Predicting Lessee Switch Behavior

An Empirical Comparison of Neural Networks and Logit Models.

Jan-Willem Feilzer - 2596978

Master Thesis Business Analytics

August 2018

Dr. E.R. Dugundji - VU supervisor
MSc. D. Stroosnier - PwC supervisor
Prof. dr. R. van der Mei - VU co-reader

Vrije Universiteit Amsterdam
Faculty of Science
Business Analytics
De Boelelan 1081a
1081 HV Amsterdam

PricewaterhouseCoopers Advisory N.V.
Consulting
Data Analytics
Thomas R. Malthusstraat 5
1066 JR Amsterdam

Preface

This thesis is written as part of the Master's degree in Business Analytics at the Vrije Universiteit Amsterdam. This two-year program aims at optimizing business results and achieving strategic objectives by harnessing students with data science, big data, statistics and machine learning knowledge. The Master's program is concluded with a six-month internship at a business or research institute. The main objective of this internship is to research and analyze a specific problem in the field of Business Analytics affecting the host organization. My internship took place at the Data Analytics team of the consulting department of PwC. The conducted research regards predicting vehicle choice of customers of a vehicle leasing company.

The past six months have proven incredibly instructive and were spent with great joy. I particularly want to thank both my supervisors. From PwC, Daan Stroosnier, for his guidance, enthusiasm and time spent on weekly meetings. From the university, Elenna Dugundji for pleasant supervision during my internship period and providing me with relevant and constructive feedback. Lastly, I would like to thank Rob van der Mei for functioning as co-reader of my thesis.

Jan-Willem Feilzer - Amsterdam, August 2018

Summary

This thesis aims to compare the strength of logit models and neural networks on a specific mode choice problem. More specifically, for a well-established vehicle leasing company it is researched which type of model is most accurate in predicting lessees' choice of vehicle make, given contract renewal. The provided data consist of terminated and operational/newly activated contracts. For the data to be used for modeling, these contracts need to be matched first. Since the leasing company only keeps track of customer IDs rather than driver IDs, this seemingly trivial task becomes nearly impossible to do flawlessly. Elaborating, a *customer* of the vehicle leasing company could be a company, whereas the employees of this company are considered *drivers*. The Norwegian branch of the leasing company provided contracts stating the previously driven vehicle, allowing for exact contract matching. Based on the characteristics of the Norwegian data an algorithm is created to match contracts of the remaining branches of the company. This algorithm matches contracts based on the number of days between termination of the ending contract and activation of the operational contract, the difference between driven mileage and agreed upon mileage per month. If all are equal, the algorithm compares the previous choice of make with the make of vehicle stated in the new contract. Since roughly 45% of drivers remain loyal to their previous make, the contract stating the same make as the previous contract is determined to be the matched contract. In addition to not being able to exactly match contracts, the absence of driver IDs causes individual-specific explanatory variables to be lost. Variables such as age, income etc. typically prove important when predicting mode choice. When all contracts are matched, these matches comprise the modeling data and are considered a valid representation of true lessee switch behavior.

The models addressed in this thesis originate from two families of model, *discrete choice models* and *(artificial) neural networks*. Regarding discrete choice models, three different *logit* models are used for prediction. Firstly, a multinomial logistic regression model is fit on the data. To fit this model, assumptions on the distribution of error terms are made which might limit the predictive power of this model. Nested multinomial logistic regression relaxes these assumptions by dividing alternatives of the choice set into nests, allowing for correlations across alternatives. However, correlations across alternatives sharing a nest are still not taken into account. To do so, a cross-nested model is introduced further relaxing the error term distribution assumptions, hence allowing for correlations across alternatives. Discrete choice models are the typical family of models used for analyzing and prediction of mode choice problems. The assumptions made on linearly independent input variables and error term distribution potentially limit the predictive capability of discrete choice models. The successful history of neural networks in many different fields give rise to applying these techniques to mode choice problems.

In this research, an infused feedforward neural network is fit on the vehicle leasing data. The structure of the network is solely determined by an *evolutionary optimization algorithm*. This technique tries to optimize a set of interrelated parameters, by applying variation and selection procedures inspired by Darwin's principles of evolution (1859). Creating a population of individuals all fighting for the same resources, *survival of the fittest* will cause individuals of the population to evolve and converge towards the ideal network structure. In addition, *embedding* layers are added to the network to map categorical variables onto continuous space. These layers map each state of a categorical variable to a fixed size continuous vector revealing the intrinsic properties of the concerned variable. A network containing such layers not only provides a measure of similarity for unique states of a categorical variable, it reduces training time of the network significantly since less parameters need to be estimated.

Performance of models indicate that (infused) feedforward neural networks are better able to accurately predict lessees' choice of vehicle. Not only do such models achieve a higher overall accuracy on test data, these models are also better able to predict minority classes. Regarding logit models, adding a nesting structure did prove helpful. However, allocation of makes across these nests should be reconsidered. The numerical explanatory variables proved most useful for prediction for all three logit models. No significant difference in importance of variables was observed between the three models. Comparing the embedding network with a standard (one-hot) network, the embedding network achieves highest accuracy on unseen data and significantly reduces training time. The embedding network is able to reveal the intrinsic properties of categorical variables, hence creating a notion of similarity. The learning curves of the networks illustrate that the provided data is not fit for modeling. Not only does the possibility exist that these data are not a valid representation of true switch behavior, the lack of individual-specific explanatory variables makes prediction a difficult task. Nonetheless, the predictive power of *embedding* neural networks of which the structure is determined by evolutionary optimization proves promising for future research.

Contents

Chapter 1

Introduction

Modeling individual's choices in selecting transportation modes has been a large area of research. However, predicting and analyzing individuals' evaluation of mode alternatives, and their corresponding decision of mode among a set of interrelated choices, remains complex. In the past, *discrete choice models* have been the typical family of models used to analyze and predict an individual's choice of one alternative from a set of mutually exclusive and collectively exhaustive alternatives (Koppelman and Bhat, 2006). These types of models are widely discussed in literature, and rose to fame when Daniel McFadden won the Nobel Prize in economics for his development of theory and methods for analyzing discrete choice (McFadden, 2001). Discrete choice models have had considerable influence on the growth of the mode choice modeling field, by trying to accommodate for both observed and unobserved effects on an individual's choice. In such models, it is assumed that an individual's preference for an alternative is captured by a value, called *utility*, and selects the alternative with highest utility. Concurrently, the assumption is made that the analyst does not have complete information, and therefore a factor of uncertainty must be taken into account (Ben-Akiva and Bierlaire, 2003). Discrete choice models are widely used due to the extent of literature available, and the relative ease of interpretation of such models.

Over the last decade however, the amount of research in the field of intelligent systems has significantly increased, and the field in which *(artificial) neural networks* are applied has greatly expanded. These types of networks have revolutionized the field of speech recognition, and natural language processing to name a few. Neural networks can in principle approximate any continuous function and piece-wise continuous function, and have therefore been able to produce state-of-the-art results in several different fields of research (Cybenko, 1989; Llanas et al., 2008). Neural networks are composed of many neurons and are usually represented by composing together multiple different functions. Due to this composition of functions, neural networks are able to capture complex relationships between features, and generally adapt well to unseen data. On the downside, the continuous nature of neural networks limits their capability to process categorical variables, which are often contained in discrete choice data, causing neural networks to be less frequently used in mode choice modeling. A recent article proposes a new way of overcoming the limitation of neural networks to work with categorical variables. This limitation is overcome by using an entity embedding method to automatically learn the representation of such variables in multi-dimensional spaces, which puts states with similar effect on the loss function close to each other, and thereby reveals the intrinsic continuity of the data (Guo and Berkhahn, 2016). In addition, finding optimal network hyper-parameters is a far from trivial task, hence these are generally found by trial and error. To avoid a trial-and-error procedure for finding optimal hyper-parameters, *evolutionary optimization* algorithms are introduced to assist in this need. These types of optimization techniques combine aspects of greedy search with a stochastic component inspired by Darwin's theories on evolution (1859). By randomly initializing a *population* of neural networks, this optimization technique iterates through *generations* to find the optimal hyper-parameters.

This thesis aims to compare the performance of the two aforementioned families of models by applying each model to a specific use case related to mode choice. Several types of discrete choice models have focused on mode choice analysis (Vovsha, 1997; Bhat and Sardesai, 2006; Chu, 2009). This thesis revolves around *logit* models, a branch of the discrete choice family. Logit models are well-represented in literature and most used for modeling mode choice (De Jong et al., 2003; Hess et al., 2012; Ding et al., 2014). Even though logit choice models have historically been most prominent in the field of mode choice modeling, such models preserve a linear relationship requiring independent explanatory variables, which is generally not a proper assumption. Neural networks are able to handle the non-linear behavior and inter-dependence of variables. Several articles comparing performance of discrete choice models and neural networks on mode choice problems exits. As early as 1996, Nijkamp et al. noted the potential of neural networks in modeling inter-urban transport flows in Italy, with such models slightly outperforming the more traditional discrete choice models. In the following years, several of such studies were conducted, all acknowledging the predictive potential of neural networks (Hensher and Ton, 2000; Vythoulkas and Koutsopoulos, 2003; Dia and Panwai, 2010; Pulugurta et al., 2013). Due to the success of neural networks in recent years, such comparison studies are more widely conducted, all having similar findings. Omrani found that neural networks slightly outperform multinomial logistic regression on a travel mode choice prediction problem of individuals in Luxembourg (2015). Roquel and Fillone found that neural networks are better able to predict minority classes, which is often considered to be a bottleneck of discrete choice models. When evolutionary algorithms are applied to neural networks, this typically corresponds to updating the network's weights through evolutionary optimization procedures, rather than standard gradient-based techniques (Montana and Davis, 1989; Mahajan and Kaur, 2013; David and Greental, 2014). Although less frequent, determination of the optimal structure of neural networks through evolutionary optimization has been researched and successfully applied (Lam et al., 2001; Castillo et al., 2003; Bahnsen and Gonzalez, 2011).

Research in this thesis is conducted for a leasing company with establishments throughout the world. This company is considered to be one of the leaders in the field of fleet management. The leasing company approached PwC to gain better understanding of its fleet data. The obtained insights should lead to improved alignment of separate entities within the company. For instance, buying, selling, and leasing of vehicles are all related, and should therefore be aligned to maximize profit. To understand customer behavior, and to provide tailored offers to these customers, it could prove greatly advantageous to model *switch behavior* of the company's lessees. A switch can be thought of as the choice of vehicle, given the customer has had a leasing contract with the company. That is to say, upon termination of the customer's current contract, the leasing company would like to know, what make of vehicle the customer will most likely lease next. It will be checked if the highly successful neural networks can outperform the more classical and widely used logit models on this particular case. Each constructed model should predict what make of vehicle a lessee is most likely to lease post termination of his contract. The main research question of the thesis can therefore be formulated as follows:

Which family of models prove most accurate for predicting lessees choice of vehicle, logit models or feedforward neural networks?

It is expected that the ability of neural networks to capture non-linear relationships across input variables will outweigh the long and successful history of logit models in mode choice analysis. Especially, the infused network containing embedding layers to learn continuous representation of categorical variables, with the network structure determined by an evolutionary optimization algorithm, will capture most unobserved non-linear input relations.

The remainder of this thesis is structured as follows. In chapter 2 the data provided by the leasing company is examined in detail. The main focus of this chapter revolves around addressing the difficulties encountered whilst processing the original data. Chapter 3 discusses the three types of discrete choice models used in this research. *Multinomial logistic regression* is the main variant of discrete choice model used for prediction. Two direct extensions of this model, *nested* and *cross-nested* multinomial logistic regression are addressed. Both these models relax the assumptions made by the multinomial logistic regression model, and should in principle allow for improved prediction accuracy. Chapter 4 discusses feedforward neural networks. The mathematical foundation of these networks is provided, the strengths and weaknesses are discussed, the mapping from categorical to continuous variables is addressed, and an *evolutionary* algorithm to optimize the network parameters is addressed. In chapter 5, the results of fitting all models on the provided data are discussed. In addition, the outcome of the optimization algorithm is addressed. Lastly, the thesis is concluded in chapter 6, with a discussion of the obtained results and the performance of all used methods.

Chapter 2

Data

This chapter discusses the data used to compare the performance all constructed models. As discussed in chapter 1, the data are provided by a vehicle leasing company with establishments throughout the world. By approaching PwC, the lease company would like to better understand customer behavior, and align entities within the company. Predicting customer switch behavior is part of this improved understanding, and the topic of this thesis. To predict these switches, a thorough understanding of the provided data is essential and is provided in this chapter. In addition, the difficulty of obtaining the desired switches by matching terminated and operational contracts, is addressed in this chapter as well.

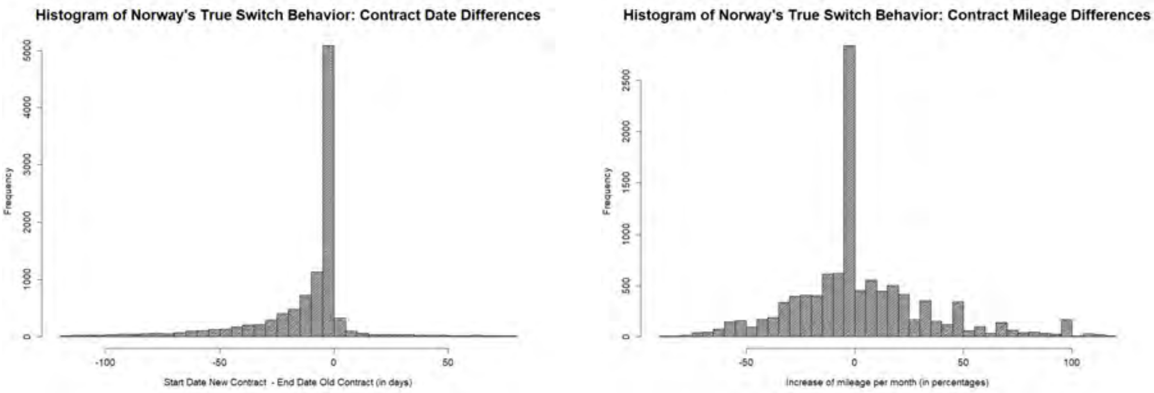
2.1 Data Construction

The provided data consist of two types of vehicle leasing contracts: *terminated* and *operational/newly activated* contracts. Each contract depicts a driver that either is or has been a client of the leasing company. Each contract contains several contract and vehicle characteristics which are used as input variables for the constructed models. To construct a data set suitable for modeling purposes, all terminated and operational contracts need to be correctly matched. Namely, for each current driver of the leasing company, the correct previous contract need be found. This presumably trivial task turned out to be one of the most challenging tasks of the conducted research. The leasing company assigns a unique driver ID to each driver, even for contract renewals. That is, if a driver renews his contract, the driver ID stated in the new contract will be different from the driver ID stated in the terminated contract. A direct and severe consequence is that matching of terminated and operational contracts becomes nearly impossible to do completely flawless. Luckily however, the leasing company does keep track of which client each driver belongs to. The distinction between a driver and a client of the company is as follows. If PwC would be a client of this company, the employees of PwC leasing a vehicle, are considered to be drivers. Therefore, for each vehicle in the company's fleet, one can only determine to which client a driver belongs. The larger the number of drivers per client, the more complex it becomes to determine the previous vehicle a driver drove. All clients of the company are divided into four different client segments: *Corporate*, *International*, *Private* and *SME*. By definition, a driver can never switch from one client segment to another. Table 2.1 displays some statistics on each client segment. The first column states the client segment, whilst the subsequent columns portray the total number of clients per client segment, the total number of drivers within that client segment, the average number of drivers per client in that particular client segment, the standard deviation of the number of drivers per client in that particular client segment, and the maximum number of drivers per client in that particular client segment. Note that the average number of drivers per client for the Private and SME segments are low, whereas this average lies much higher for the Corporate and International segments. In addition, note that the statistics on the Private segment are heavily influenced by one client having more than 14.000 drivers.

Client Seg.	**#Clients**	**#Drivers**	$\mu_{\textbf{drivers/client}}$	$\sigma_{\textbf{drivers/client}}$	$\textbf{Max}_{\textbf{drivers/client}}$
Corporate	18446	657104	35.6	221.8	15461
International	8040	319026	39.7	150.6	6699
Private	10949	40784	3.7	137.5	14258
SME	140924	319415	2.3	6.0	690

Table 2.1: Statistics on each client segment present in the raw data.

Of all branches of the company of which contracts are made available, the Norwegian branch provided data in which it is possible to match contracts with 100% accuracy. The concerned data contain an additional variable: `previous license plate`. This variable states the license plate of the vehicle the driver drove prior to his or her current vehicle. By matching the current and previous license plates, accurate switch determination can ensue. A couple interesting observations are found. One would be reasonable to assume that new contracts commence post termination of previous contracts. To investigate this claim, the number of days between commencement of the new contract, and termination of the old is calculated and shown in Figure 2.1a. The x-axis portrays the number of days between these two dates. A negative number on the x-axis means that the previous contract had not terminated upon enrollment of the new contract. Note from this Figure and Table A.1 (located in Appendix A), that the aforementioned claim is immediately refuted. Over 90% of contracts commence prior to termination of the previous contract. Additionally, roughly 30% of new contracts commence on the termination date of the old contract. Another noteworthy observation relates to the agreed upon mileage stated in each contract. When comparing both the true driven mileage of a driver whose contract had terminated, and the agreed upon mileage for the new contract, significant fluctuations occur. However, when calculating the true driven mileage per month for the terminated contracts, and the agreed upon mileage per month for the operational contract, it is noted that the majority of drivers' estimated mileage per month remains equal (approximately 21%). In addition, the remaining data follow a distribution inhabiting some normal distribution characteristics. Note from Figure 2.1b, that both tails of the distribution are roughly similar, with a slightly smaller left tail. Additionally, the data portraying the mileage differences for the true Norwegian branch data, can be found in Table A.2, located in Appendix A.



(a) Histogram showing the distribution of the number of days between operational and terminated contracts.

(b) Histogram showing the distribution of the percentage of increase or decrease between driven mileage per month of terminated contracts, and agreed upon mileage per month for operational contracts.

Figure 2.1: Histograms describing characteristics of the true Norwegian branch data.

Lastly, observe from Table 2.2 that the majority of drivers remains loyal to the make they drove. This table shows the total number and the percentage of drivers remaining loyal to their make, for all makes. The second column states the total number of drivers remaining loyal to their make, whilst the last column shows the percentage this number relates to. On average, approximately 44% of drivers remain loyal to their make. A percentage assumed to be representative for loyalty in all branches.

To match terminated and operational contracts for branches of which `previous license plate` is not provided, the observations made when examining the Norwegian branch data are used. The methodology created to perform this matching is based on the aforementioned observations and has as goal to accurately match as many contracts as possible. The methodology can be found in Algorithm 1, and will now be explained in words thoroughly.

The matching algorithm takes as input two sets of contracts T and O, denoting terminated and operational contracts respectively. In addition, integer values for both $\mathtt{lb}_{\mathtt{day}}$ and $\mathtt{ub}_{\mathtt{day}}$ need to be specified. The use of these variables will become apparent shortly. The algorithm commences by performing an *inner-join* procedure between sets T and O on the variable `clientID`, depicted on line 1. This step returns all entries of T which have matching client IDs in set O, and concatenates the matching entries of O to the corresponding entries of T. This procedure creates a set D denoting *all* possible switches within a particular client. Of course, most of these switches are infeasible and should not be taken into account. To deem a switch feasible, three feasibility measures are computed which are depicted on lines 2, 3 and 4. Firstly, the number of days between the start date of the operational contract and the end date of the terminated contract is computed, and is

Make	**#Occurrences**	**Percentage**
Audi	216	60.00
BMW	195	42.67
Citroen	22	4.47
Ford	678	65.89
Mercedes-Benz	83	20.24
Nissan	37	12.63
Opel	77	52.38
Other	342	40.17
Peugeot	229	13.58
Renault	19	5.81
Skoda	242	34.82
Volkswagen	2504	65.69
Volvo	203	58.17

Table 2.2: Table showing the *loyalty* of drivers to a make. The leftmost column shows a particular make. The middle column states the number of drivers which kept driving the same make of vehicle when agreeing on a new contract. The rightmost column shows the percentage of drivers remaining loyal to their make.

depicted by the variable $d_{\textsf{day difference}}$. Thereafter, the variable $d_{\textsf{mileage difference}}$ is calculated, depicting the increase or decrease of the to be driven mileage per month. More specifically, this value depicts the percentage of mileage per month the new agreed upon mileage per month differs from the true driven mileage per month stated in the terminated contract. The last feasibility measure acts as a Boolean value stating if the new vehicle is of the same make as the previously driven vehicle. The variable $d_{\textsf{same make}}$ is equal to 1 if the new vehicle is of the same make, and equal to 2 otherwise. Note that this is true since a union of two sets removes duplicates by definition. Next, all matched contracts ($d \in \mathsf{D}$) of which the variable $d_{\textsf{day difference}}$ does not lie within the interval $[\texttt{lb}_{\textsf{day}}, \texttt{ub}_{\textsf{day}}]$, are deleted. These bounds, provided as input variables, therefore depict the boundaries of which matches are potentially deemed feasible. For instance, if $\mathsf{lb}_{\textsf{day}}$ and $\mathsf{ub}_{\textsf{day}}$ are set to -2 and 3 respectively, matches of which the start date of the operational contract lies 5 days after the end date of the terminated contract are considered infeasible. Subsequently, all driver IDs stated in the terminated contracts ($\texttt{driverID}_{\texttt{terminated}}$) are extracted and stored in set $\mathsf{Drivers}_{\textsf{terminated}}$. Again, note that this set only contains unique values by definition. Additionally, observe that each terminated contract can potentially be matched to multiple operational contract due to the join procedure performed on line 1. Lastly, prior to the core procedure of the matching algorithm, the output is initialized as an empty set, $\mathsf{S} = \emptyset$. Matched contracts deemed feasible are iteratively added to this set.

The core procedure of the algorithm has as goal to find the best possible match for all terminated contracts, and commences on line 8. The procedure runs until set D is exhausted. That is, all matched contracts portrayed in this set, are either deemed feasible or infeasible.

At first, a set temp, depicting all matched contracts of which $\texttt{driverID}_{\texttt{terminated}}$ is present in $\mathsf{Drivers}_{\mathsf{terminated}}$, is created. It will become evident shortly why it is not possible to simply use set D. Then, for each $\texttt{driverID}_{\texttt{terminated}}$, a subset containing all these IDs present in temp is created. This subset subset is then sorted on the absolute value of the three abovementioned feasibility measures, in ascending order. Consequently, for all three feasibility measures, a low absolute value of the measure is considered to be important. Additionally, $d_{\text{day difference}} = |-1|$ is considered to be *less* than $d_{\text{day difference}} = |1|$. This decision was made due to the distribution portrayed in Figure 2.1a. In similar fashion, albeit reversed, $d_{\text{mileage difference}} = |10|$ is considered to be *less* than $d_{\text{mileage difference}} = |-10|$. Again, this decision was made due to the distribution displayed in Figure 2.1b. For illustrating purposes, the list $\boldsymbol{d}_{\text{day difference}}$ = [0, 1, -1, 2, 0 ,-1 , -2] will be returned as [0, 0, -1, -1, 1, -2, 2] by the sorting procedure. Post sorting, the first element of subset is considered to be the *most likely* switch for that particular $\texttt{driverID}_{\texttt{terminated}}$, and is added to set S.

Now, set D has been reduced to only contain unique values of $\texttt{driverID}_{\texttt{terminated}}$, and all terminated contracts have been matched to an operational contract. Note however, that this matching has not restricted set D to only contain unique values of $\texttt{driverID}_{\texttt{operational}}$. Consequently, operational contracts are potentially matched to multiple terminated contracts. To ensure unique matching of contracts, matches of which $\texttt{driverID}_{\texttt{operational}}$ is not unique, need to be reexamined. To do so, the procedures described above (lines 10 - 14), are repeated, provided some slight alterations.

Whereas the previously discussed procedures were all based on $\texttt{driverID}_{\texttt{terminated}}$, these procedures are based on $\texttt{driverID}_{\texttt{operational}}$ to ensure only unique values for $\texttt{driverID}_{\texttt{operational}}$. Note that lines 15 and 16 are just a compressed version of lines 10 - 14. In short, for each $\texttt{driverID}_{\texttt{operational}}$ present in set S, the most likely $\texttt{driverID}_{\texttt{terminated}}$ is found based on the aforementioned feasibility measures. Line 16, removes all matches of which $\texttt{driverID}_{\texttt{operational}}$ is not unique, and which are not considered to be the most likely match, from S. Next, $\mathsf{Drivers}_{\mathsf{terminated}}$ is updated to contain all $\texttt{driverID}_{\texttt{terminated}}$ which have not been matched with an operational contract yet. Lastly, the best possible match for each $\texttt{driverID}_{\texttt{terminated}}$ is deleted from set D. This is done, since the best possible match is either stored in set S, or it is deemed infeasible since it has been matched to another $\texttt{driverID}_{\texttt{terminated}}$. The matching procedure is concluded when set D is exhausted. Note that it is also possible for terminated contracts to not be matched with any operational contract. Of course, a driver of the company is free to decide not to renew his contract.

To determine the optimal value for the parameters $\texttt{lb}_{\texttt{day}}$ and $\texttt{ub}_{\texttt{day}}$, the accuracy of the matching procedure is measured on the true switching data of the Norwegian branch. Accuracy is chosen over recall as a performance measure, since one would like the matched data to accurately represent the true switch behavior, rather than having more data of which one cannot be certain of the accuracy of matching. Nonetheless, the trade-off between accuracy and enough data for prediction purposes need be balanced. In Figure 2.2, the trade-off between the accuracy of prediction of the matching algorithm, the number of matched contracts, and the length of the range [$\texttt{lb}_{\texttt{day}}$, $\texttt{ub}_{\texttt{day}}$] is portrayed for the true Norwegian data. The x-axis displays the length of [$\texttt{lb}_{\texttt{day}}$, $\texttt{ub}_{\texttt{day}}$]. Note that this length can

Algorithm 1: Heuristic Contract Matching

input : $\mathsf{T} = \big(t_1, ..., t_m\big) = \big(\{\tau_1, ..., \tau_n\}_1, ..., \{\tau_1, ..., \tau_n\}_m\big) \in \mathbb{R}^{m \times n}$
$\mathsf{O} = \big(o_1, ..., o_k\big) = \big(\{\sigma_1, ..., \sigma_n\}_1, ..., \{\sigma_1, ..., \sigma_n\}_k\big) \in \mathbb{R}^{k \times n}$
$\mathrm{lb}_{\mathrm{day}}$
$\mathrm{ub}_{\mathrm{day}}$

output: $\mathsf{S} = \big(s_1, ..., s_l\big) = \big(\{\chi_1, ..., \chi_{2n+2}\}_1, ..., \{\chi_1, ..., \chi_{2n+2}\}_l\big) \in \mathbb{R}^{l \times (2n+2)}$

$\mathsf{D} = \mathsf{T} \bowtie_{t_{\text{Client ID}} = o_{\text{Client ID}}} \mathsf{O}$
$\forall d \in \mathsf{D},\ d_{\text{day difference}} = d_{\text{start date operational}} - d_{\text{end date terminated}}$
$\forall d \in \mathsf{D},\ d_{\text{mileage difference}} = \frac{d_{\text{mileage/month operational}} - d_{\text{true mileage/month terminated}}}{d_{\text{true mileage/month terminated}}}$
$\forall d \in \mathsf{D},\ d_{\text{same make}} = |d_{\text{make terminated}} \bigcup d_{\text{make operational}}|$
$\mathsf{D} = \{d \in \mathsf{D} \mid d_{\text{day difference}} \in [\mathrm{lb}_{\mathrm{day}}, \mathrm{ub}_{\mathrm{day}}]\}$
$\mathsf{Drivers}_{\mathsf{terminated}} = \{d_{\text{driver ID terminated}}\ \forall d \in \mathsf{D}\}$
$\mathsf{S} = \emptyset$
while $\mathsf{D} \neq \emptyset$ **do**
  $\mathsf{temp} = \{d \in \mathsf{D} \mid d_{\text{driver ID terminated}} \in \mathsf{Drivers}_{\mathsf{terminated}}\}$
  for $\mathrm{driverID}_{\text{terminated}} \in \mathsf{Drivers}_{\mathsf{terminated}}$ **do**
    $\mathsf{subset} = \{t \in \mathsf{temp} \mid t_{\text{driver ID terminated}} = \mathrm{driverID}_{\text{terminated}}\}$
    $\mathtt{sort}(\mathsf{subset},\ |t_{\text{day difference}}|,\ |t_{\text{mileage difference}}|,\ |t_{\text{same make}}|,\ \mathtt{ascending})$
    $\mathtt{match} = \mathsf{subset}(1)$
    $\mathsf{S} = \mathsf{S} \bigcup \mathtt{match}$
  $\mathtt{sort}(\mathsf{S},\ s_{\text{driver ID operational}},\ |s_{\text{day difference}}|,\ |s_{\text{mileage difference}}|,\ |s_{\text{same make}}|,\ \mathtt{ascending})$
  $\mathtt{remove\_duplicates}(\mathsf{S},\ s_{\text{driver ID operational}})$
  $\mathsf{Drivers}_{\mathsf{terminated}} = \{d_{\text{driver ID terminated}}\ \forall d \in \mathsf{D}\} - \{s_{\text{driver ID terminated}}\ \forall s \in \mathsf{S}\}$
  $\mathtt{keep\_duplicates}(\mathsf{D},\ d_{\text{driver ID terminated}})$
return S

be established by multiple combinations of $\mathtt{lb}_{\mathtt{day}}$ and $\mathtt{ub}_{\mathtt{day}}$. Therefore, the combination of values achieving the highest accuracy was chosen to be portrayed. The y-axis portrays the accuracy of matching achieved by the algorithm. The labels correspond to the percentage of matched contracts that were found with respect to the actual number of switches. As expected, the accuracy of the matching algorithm decreases as the length of the interval $[\mathtt{lb}_{\mathtt{day}}, \mathtt{ub}_{\mathtt{day}}]$ increases.

Due to the abundance of contracts available from branches that cannot flawlessly be matched, it is chosen to set both $\mathtt{lb}_{\mathtt{day}}$ and $\mathtt{ub}_{\mathtt{day}}$ equal to 0. First of all, the highest possible accuracy of matching is achieved. Secondly, one does not have to deal with the uncertainty of new contracts commencing prior to termination of previous contracts. All data from the remaining branches of the company are modeled using the procedure described in Algorithm 1. The following section describes statistics on the resulting data from the matching procedure. For the remaining of this thesis, these data are assumed to correctly reflect true lessee switch behavior.

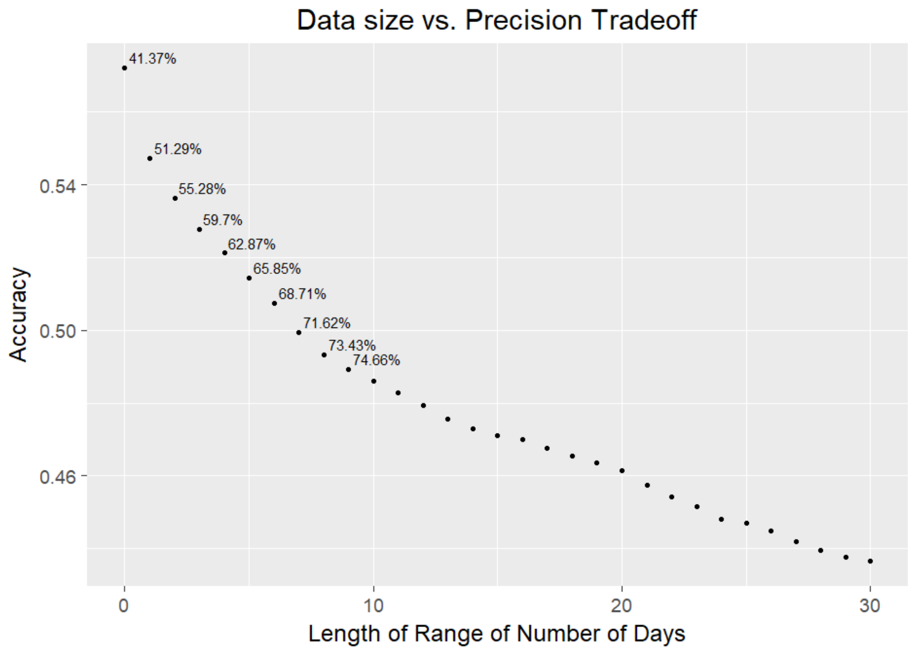


Figure 2.2: Scatter plot illustrating the trade-off between accuracy and the percentage of matched contracts found. The x-axis shows the length of the range [$\texttt{lb}_{\texttt{day}}$, $\texttt{ub}_{\texttt{day}}$]. The y-axis shows the accuracy achieved by the matching algorithm. The percentages shown in the plot, depict the percentage of matched contracts found by the matching algorithm. Note that a higher percentage does not imply a higher accuracy. It solely states the number of matched contracts found.

2.2 Data Processing

After completion of the previously discussed matching algorithm, the resulting data consist of 68,952 matched contracts or switches. The provided contracts originate from ten different branches of the company, with each branch depicting a different country. The company considers each client to belong to a particular `client segment`: *Corporate*, *International*, *Private* or *SME*. By definition of the matching procedure described in the previous section, it is not possible to switch client segments. In addition, each vehicle of the company's fleet is placed within different segments. To be more precise, all vehicles are associated with the following segments: `brand classification`, `vehicle segment`, `OEM group`, `make` and `model`. For instance, an Italian driver working for a corporate association, driving a Volkswagen Passat is classified as is portrayed in Table 2.3. All these labels are predefined by the leasing company and are stated within each provided contract. Note that most segments depend on the higher-level segment. That is, if the variable `model` is known, one knows the variables `make`, `OEM group`, `vehicle segment` and `brand classification` by definition. Recall that it is possible that a vehicle is present in multiple client segments and countries. Additionally, if only the variable `make` is known, one does know the variable `OEM group` by definition. Knowing the make does not necessarily imply the vehicle segment to be known. Since vehicles are classified to be part of different vehicle segments based on the model of vehicle, it is possible for makes to belong to multiple vehicle segments. This property proves extremely convenient when subdividing makes into nests, described in section 3.3, and will be touched upon in this section.

Country	**Client Seg.**	**Brand Class.**	**Vehicle Seg.**	**OEM**	**Make**	**Model**
Italy	Corporate	Mainstream	D	VAG	Volkswagen	Passat

Table 2.3: Example of segmentation of a client of the leasing company.

Aside from variables segmenting the vehicles of the fleet, each contract provides the following information: `customer ID`, `vehicle ID`, `fuel type`, `vehicle type`, `body style`, `lease type`, `catalogue price`, `commercial discount amount`, `standard discount percentage`, `total accessories amount`, `total options amount`, `ufwt amount`, `start mileage`, `end mileage`, `contract mileage`, `intro date model`, `end date model`, `sale date`, `sale amount`, `termination info`, `start date contract`, `end date contract` and `contract duration`. The variables `mileage per month` and `switch quarter` are extracted during and after completion of the matching algorithm. Not all variables are used for prediction purposes. Some variables are either too highly correlated, or variables are omitted due to a lack of descriptive quality.

To use variables provided alongside the matched contracts, these need be processed prior modeling. Processing of data can be separated in two parts: processing of numerical variables, and processing of categorical variables. First, processing of numerical variables is discussed. All missing values of numerical variables contained in contracts are replaced with the mean value of the concerned variable after grouping by the variables `country`, `client segment` and `make`. To illustrate the matter, if the variable `catalogue price` is missing for a Volkswagen Golf of a driver stemming from the SME segment of the Spanish branch, it is replaced with the mean value of the catalogue price for that particular vehicle in those segments. In addition, outliers are set to either the determined lower or upper boundary of the concerned variable. Some statistics on the numerical variables used for prediction can be found in Table 2.4.

To use numerical variables for prediction, these need be scaled. The need for scaling of numerical parameters becomes apparent when discussing gradient based optimization methods, provided in section 4.4. Prior to this procedure, all variables depicting a monetary value are transformed using the natural logarithmic function. The variables `catalogue price`, `commercial discount amount`, `total accessories amount`, `total options amount`, and `ufwt amount` are transformed by taking the natural log of the original value. In addition, all zero values for which the natural log is not defined, are replaced with the minimum value

	μ	σ	**Median**	**Min**	**Max**
`catalogue price`	45051.68	83363.76	23829.43	5575	1440000
`commercial discount amount`	6765.62	7835.72	4956.86	0	125760
`standard discount percentage`	10.68	7.96	10	0	35
`total accessories amount`	270.93	712.85	0	0	9877
`total options amount`	3299.08	6525.14	1000.86	0	49984
`ufwt amount`	613.26	1115.05	250	0	10000
`mileage per month`	2639.57	1248.30	2500	500	10000
`contract duration`	42.10	10.70	42	6	96

Table 2.4: Summary of all numerical variables used for modeling purposes. The rows indicate the numerical variable. The columns portray the mean, standard deviation, median, minimum value, and the maximum value of the concerned variable respectively.

for which the natural log is defined. The idea of using the natural logarithmic function for variable transformation, is to push the variable towards being normally distributed. After the logarithmic transform, the monetary variables are treated as any other numerical variable. Next, all numerical variables are standardized to have zero mean, and standard deviation of 1. This transformation refrains functions present in both discrete choice models and neural network from saturating. Again, the need for this pre-processing becomes evident when reading section 4.3 and 4.4.

Processing of categorical variables occurs in slight different fashion. No missing values occur in the data. The provided data does contain values such as *unknown*, or *country did not supply a value*. These values rarely occur and therefore do not form a significant problem. States of categorical variables that rarely occur, are either set to the state *Other*, or are merged with an already-existing state. For instance, the variable `client segment` is reduced to contain three states, since *Private* is merged with *SME*. A brief summary of the categorical variables can be found in Table 2.5.

Lastly, the provided vehicle leasing data is split into training and test data. These data are identical for all models used in this thesis. The *most recent* 10% of data are considered to be test data. Data are classified as most recent based on the date a switch occurred. Providing test data allows for models to predict on data that are seen as most presentable of the current situation. Prior to splitting data into train and test data, all data are shuffled to avoid dis-balanced data. When predicting switches using feedforward neural networks, the topic of chapter 4, training data are further processed. To train a network, 10-fold cross-validation is used. This means that the network is trained on 9 folds, whilst the remaining fold is used to validate the network's performance. In the training process, each fold functions as validation data once. So, the network is effectively trained 10 times, in which the validation data is different on each iteration.

	#States	**Max State**	**Min State**	**Min State (orig)**
`country`	8 (10)	France	Other	Austria
`client segment`	3 (4)	Corporate	SME	Private
`fuel type`	2 (6)	Diesel	Petrol	Unknown
`vehicle type`	2 (2)	Vehicle	Van	Van
`vehicle segment`	8 (12)	D	Other	F
`body style`	8 (17)	Stationwagon	Vehicle/Van	Unknown
`lease_type`	2 (2)	Operational lease	Financial lease	Financial lease
`switch quarter`	4 (4)	1	2	2
`make`	13 (41)	Volkswagen	Nissan	Chrysler

Table 2.5: Summary of all categorical variables used for modeling purposes. The rows indicate the categorical variable. The first column portrays the number of unique states per variable, with the number in parenthesis stating the number of unique variables prior processing. The remaining columns depict the most occurring state per variable, the least occurring state per variable, and the least occurring state per variable prior to processing.

Chapter 3

Discrete Choice Models

This chapter discusses the first family of models used to model the vehicle leasing data, *discrete choice models*. More precisely, this chapter discusses *logit* models, a branch of the discrete choice model family. These types of models are widely used to model mode choice. Section 3.1 describes the properties common to all discrete choice models. The general framework of such models is introduced and elaborated on. The subsequent sections dive into the three types of *logit* models used in this thesis. The models addressed in sections 3.3 and 3.4, relax the assumptions made for standard multinomial logistic regression models (section 3.2). The chapter is concluded with a section describing the applicability of these models on the provided vehicle leasing data. Note that almost all mathematical derivations and formulae are derived from Train (2009). Text, derivations and formulae not originating from this source are cited accordingly.

3.1 General Properties

Discrete choice models analyze and predict individual's choices among a set of alternatives. This set of alternatives, the *choice set*, needs to contain three characteristics. The alternatives of the choice set need be *mutually exclusive*, and the choice set must be *exhaustive* and *finite*. The first two properties are not restrictive, since data can be modeled in such ways that these properties are met. However, finity of the choice set is a restrictive property and the defining characteristic of discrete choice models. Typically, these models are derived under the assumption of *utility maximization*. Models derived under this assumption are referred to as *random utility models*. In random utility models, it is assumed that the decision maker assigns a preference value, called *utility*, to each alternative in the choice set. The decision maker is assumed to have perfect discrimination capability and therefore chooses the alternative possessing the highest utility value. In addition however, the analyst is assumed to have incomplete information, hence a factor of uncertainty needs to be taken into account. Such models can be defined as follows. If decision maker n chooses an alternative from a choice set C of size J elements, each element is assigned a utility value U. Then alternative i is chosen if and only if $U_{ni} > U_{nj}\ \forall j \neq i$.

Each utility value is composed of a deterministic and random part: $U_{ni} = V_{ni} + \epsilon_{ni}$. The deterministic part V_{ni}, or the *representative utility*, portrays the attributes of both the decision maker and the alternative, and is often specified to be linear in parameters. The random part, or error term of utility, ϵ_{ni}, captures the factors that affect utility but are not included in V_{ni}. The value of ϵ_{ni} $\forall i$ is not known a priori, hence these terms are treated as random. Assuming a joint density function $f(\epsilon_n)$ of the vector containing all random terms, the probability of individual n choosing alternative i is defined as

$$\begin{aligned}
&\mathbb{P}_{ni} = \mathbb{P}(U_{ni} > U_{nj}\ \forall j \neq i) = \\
&\mathbb{P}(V_{ni} + \epsilon_{ni} > V_{nj} + \epsilon_{nj}\ \forall j \neq i) = \\
&\mathbb{P}(\epsilon_{nj} - \epsilon_{ni} < V_{ni} - V_{nj}\ \forall j \neq i) = \\
&F(\epsilon_{nj} - \epsilon_{ni} < V_{ni} - V_{nj}\ \forall j \neq i) = \\
&\int_{\epsilon} I(\epsilon_{nj} - \epsilon_{ni} < V_{ni} - V_{nj}\ \forall j \neq i) f(\epsilon_n) d\epsilon_n,
\end{aligned} \tag{3.1}$$

in which $I(\cdot)$ denotes the indicator function, equaling one when true, and zero otherwise. Note from equation 3.1 that different types of discrete choice models arise from different distributions of $f(\epsilon_n)$. The models used in this thesis cause the integral of equation 3.1 to be of closed form due to the specification of $f(\epsilon_n)$; hence these models do not need to be evaluated numerically. The choice of distribution of the random terms, and the motivation for these different assumptions will be discussed in the following sections.

In addition, from equation 3.1 note that only the signs of the differences of utilities matter in choosing an alternative, rather than their absolute values. Consequently, this means that the only parameters able to be estimated are those capturing differences across alternatives. Due to this fact, the deterministic part of utility is often specified to be linear in parameters with a constant added. This constant captures the average effect of all factors not included in the model, on utility, and is referred to as the *alternative specific constant*. Including these constants produces the convenient property that the mean of the error terms can be assumed to equal any constant, typically zero. The deterministic part of utility is then defined as $V_{ni} = x_{ni}\beta + k_i$, where vector x_{ni} depicts the attributes of alternative i and individual n, β is the vector of coefficients of these variables to be estimated, and k_i denotes the alternative specific constant. Including alternative specific constants results in the random part of utility having zero mean by construction. If ϵ_{ni} has a nonzero mean, adding the alternative specific constants result in the remaining error term having zero mean. Therefore, without loss of generality, it can be assumed that the mean of the error terms is equal to zero by including alternative specific constants in the deterministic part of utility.

Another consequence of the fact that only the signs of differences between utilities matter, is that this property also holds for the alternative specific constants. A direct result is that it is impossible to estimate all alternative specific constants, since there are infinitely many possibilities for a and b when $a - b$ is equal to some constant. Hence one of the constants is typically normalized to zero. It does not matter which alternative specific constants is normalized, since all other constants are interpreted as being relative to whichever constant is normalized.

In addition to normalizing one of the alternative specific constants, the scale of utility must be normalized too. The necessity of this normalization can be observed from the fact that the alternative with highest utility does not change regardless of the scale of utility: the models $U_{ni} = V_{ni} + \epsilon_{ni}\ \forall i$, and $U_{ni} = \lambda V_{ni} + \lambda\epsilon_{ni}\ \forall i$ are equivalent. Generally, normalizing the scale of utility corresponds to normalizing the variance of the error terms. Observe that the scale of utility and the variance of the error terms are related by definition, since $\mathrm{Var}(aX) = a^2\mathrm{Var}(X)$. Therefore multiplying utility by λ corresponds to the variance of each ϵ_{ni} changing by a factor λ^2. The models explained in the subsequent sections assume that the error terms are independently, identically distributed (i.i.d). When the i.i.d assumption is imposed, normalization is quite simple; the error variance is normalized to some convenient value. Since the i.i.d assumption causes all error terms to have equal variance, normalizing the variance of any of the error terms sets the variance for all error terms. In addition, note that the i.i.d. assumption causes the integral of equation 3.1 to be of closed form.

3.2 Multinomial Logistic Regression

Multinomial logistic regression is the most straightforward and widely used discrete choice model since modeling is quite straightforward, and results of the model are easily interpretable. The derivation of the model is based on the framework specified in section 3.1, and the choice of distribution of unobserved utility $f(\epsilon_n)$. The multinomial logistic regression model assumes that the random parts of utility are independently, identically *extreme value* distributed. This distribution is generally referred to as *Gumbel*. The density function and cumulative distribution of the Gumbel distribution are stated in equations 3.2 and 3.3 respectively.

$$f(\epsilon_{ni}) = e^{-\epsilon_{ni}} e^{-e^{-\epsilon_{ni}}} \tag{3.2}$$

$$F(\epsilon_{ni}) = e^{-e^{-\epsilon_{ni}}} \tag{3.3}$$

The variance of this distribution is equal to $\frac{\pi^2}{6}$. Recall from section 3.1 that assuming a variance value implies normalizing the scale of utility. The mean of the Gumbel distribution is not equal to zero. However, since only the differences in utility values matter, this is irrelevant. Note that the mean of the difference of two random terms with equal mean is equal to zero by definition, hence all prerequisites are met.

To derive the choice probabilities of the multinomial logistic regression model, the assumption of independent error terms becomes significant. Using the property that the cumulative distribution becomes the product of individual cumulative distributions for independent error terms, equation 3.1 can be written as:

$$\mathbb{P}_{ni} = \int \Big(\prod_{j \neq i} e^{-e^{-(\epsilon_{ni}+V_{ni}-V_{nj})}}\Big) e^{-\epsilon_{ni}} e^{-e^{-\epsilon_{ni}}} d\epsilon_{ni} \tag{3.4}$$

Some algebraic alterations of equation 3.4, result in the final specification of the choice probabilities of the multinomial logistic regression model, stated in equation 3.5. For the derivation of this equation, the reader is referred to Train (2009).

$$\mathbb{P}_{ni} = \frac{e^{V_{ni}}}{\sum_j e^{V_{nj}}} \tag{3.5}$$

Two important properties arise from these choice probabilities. Firstly, McFadden et al. (1973) showed that the log-likelihood function of these probabilities has a global maximum, guaranteeing convergence of the maximization procedures. This property becomes extremely convenient when estimating the models discussed in sections 3.3 and 3.4, and will be touched upon in these sections. Secondly, the assumption of independence of error terms creates the notion of *independence* from *irrelevant alternatives* (IIA). This property states that for any two alternatives i and k, the ratio of probabilities remains equal when alternatives are added to the choice set.

$$\frac{\mathbb{P}_{ni}}{\mathbb{P}_{nk}} = \frac{\frac{e^{V_{ni}}}{\sum_j e^{V_{nj}}}}{\frac{e^{V_{nk}}}{\sum_j e^{V_{nj}}}} = \frac{e^{V_{ni}}}{e^{V_{nk}}} = e^{V_{ni}-V_{nk}}. \tag{3.6}$$

Note from equation 3.6 that the ratio of probabilities only depends on the alternatives i and k, and is therefore independent of any other alternatives present in the choice set. This property can impose severe limitations to the multinomial logistic regression model. These limitations are best illustrated with the famous red bus, blue bus paradox. Imagine that the decision maker has two alternatives to choose from when commuting to work: go by car, or take a blue bus. Assuming $\mathbb{P}_{car} = \mathbb{P}_{blue\ bus} = \frac{1}{2}$, equation 3.6 becomes equal to one. Adding a red bus to the choice set should intuitively not matter to the decision maker. That is, taking a red bus or blue bus is most likely irrelevant, and therefore the assumption that $\mathbb{P}_{blue\ bus} = \mathbb{P}_{red\ bus} = \frac{1}{4}$, and $\mathbb{P}_{car} = \frac{1}{2}$ is reasonable. However the IIA property states that, the ratio of probabilities of the alternatives car and blue bus remains the same. This implies that $\mathbb{P}_{blue\ bus} = \mathbb{P}_{red\ bus} = \mathbb{P}_{car} = \frac{1}{3}$. In other words, in such a model correlation across alternatives is not possible. The data used in this thesis could potentially inhabit such correlations. To overcome the limitation of standard multinomial logistic regression, *nested* multinomial logistic regression models are introduced next.

3.3 Nested Multinomial Logistic Regression

As discussed in section 3.2, using the multinomial logistic regression model implies that correlations across alternatives cannot be modeled. This section introduces a methodology that partly overcomes this limitation: nested multinomial logistic regression. These types of models can be placed in a more general framework of models: *generalized extreme value* models (GEV). The main property defining these models is that the error terms of utility for all alternatives are jointly distributed as generalized extreme value. This property allows for correlation across alternatives. It will be shown that the multinomial logistic regression model provided in section 3.2 is an instance of this family of models as well. When all correlations across alternatives are equal to zero, the GEV distribution becomes the product of independent extreme value distributions, as is the case for multinomial logistic regression.

Nested multinomial logistic regression is deemed an appropriate modeling structure when the choice set can be divided into subsets of alternatives: *nests*. Two properties hold when dividing the choice set into nests. Firstly, the IIA assumption holds for alternatives within the same nest. That is, for two alternatives in the same nest, the ratio of probabilities is independent of the remaining alternatives within that nest. Secondly, the IIA assumption does not hold for alternatives across nests, allowing for correlations across alternatives of different nests. In general, nests are visualized using a tree structure, in which each branch denotes a nest of alternatives. Within those nests, the IIA assumption holds. The leafs of the tree depict the alternatives of the choice set.

Concerning the derivation of the probabilities for the nested logistic regression model, McFadden (1978) showed that the model is consistent with the utility maximization theory provided in section 3.1. To derive these probabilities, suppose the choice set C consisting of J alternatives is to be divided into K nests B_k such that $B = \bigcup_{k=1}^{K} B_k$ and $B_k \cap B_{k'} = \varnothing \ \forall k \neq k'$. Then, the nested model is obtained by assuming that error vector ϵ_n has a type of GEV cumulative distribution stated below.

$$F(\epsilon_n) = \exp\Big(- \sum_{k=1}^{K} \big(\sum_{i \in B_k} e^{-\epsilon_{ni}/\lambda_k} \big)^{\lambda_k} \Big) \tag{3.7}$$

It can immediately be observed that equation 3.7 collapses to the product of independent extreme value distributions provided in equation 3.3 when $\lambda_k = 1 \ \forall k$; hence the model is reduced to the standard multinomial logistic regression model. This parameter measures the degree of independence of the error terms in utility among the alternatives in nest k. The higher the value of λ_k the less the correlation across alternatives in nest k. In equation 3.7, the marginal distribution of each ϵ_{nj} is still univariate extreme value, however the ϵ_{nj}'s are correlated within nests. For any two variables in different nests, the error terms are still uncorrelated.

McFadden (1978) showed that the probability of individual n choosing alternative $i \in B_k$ is defined as

$$\mathbb{P}_{ni} = \frac{e^{V_{ni}/\lambda_k} \big(\sum_{j \in B_k} e^{V_{nj}/\lambda_k} \big)^{\lambda_k - 1}}{\sum_{l=1}^{K} \big(\sum_{j \in B_l} e^{V_{nj}/\lambda_l} \big)^{\lambda_l}} \tag{3.8}$$

From equation 3.8 it can be shown that the IIA assumption still holds for alternatives sharing a nest. The fraction of probabilities of two alternatives $i \in B_k$ and $m \in B_l$ is solely defined by the numerator of the equation, since the denominator remains equal for all alternatives:

$$\frac{\mathbb{P}_{ni}}{\mathbb{P}_{nm}} = \frac{e^{V_{ni}/\lambda_k}\left(\sum_{j\in B_k} e^{V_{nj}/\lambda_k}\right)^{\lambda_k-1}}{e^{V_{nm}/\lambda_l}\left(\sum_{j\in B_l} e^{V_{nj}/\lambda_l}\right)^{\lambda_l-1}} \tag{3.9}$$

Observe that the terms in parentheses in equation 3.9 cancel out when $k = l$, resulting in the fraction of probabilities only depending on the attributes of i and m. Hence, the IIA assumption holds for alternatives sharing the same nest. Interestingly, note that some form of IIA still holds if $k \neq l$. In this case, the probability ratio only depends on all alternatives of nests k and l: *independence* from *irrelevant nests*.

To better grasp the notion of nested logistic regression models, it is possible to decompose equation 3.8 into two separate logistic regression models. To be precise, the probability of individual n choosing alternative i in nest B_k can be expressed as the product of two probabilities. Specifically, the probability of alternative $i \in B_k$ being chosen times the probability of nest B_k being chosen: $\mathbb{P}_{ni} = \mathbb{P}_{ni|B_k} \cdot \mathbb{P}_{nB_k}$. This notation allows for splitting utility into a part depending on attributes of the nest, and a part depending on attributes describing the alternative. The attributes describing the nest only vary over nests; they do not vary over alternatives within the nests. The attributes describing the alternatives vary over the alternatives within a nest. Setting $U_{ni} = W_{nk} + Y_{ni} + \epsilon_{ni}$, in which W_{nk} only depends on attributes describing nest k, and Y_{ni} depends on attributes describing alternative j, allows to write the conditional and marginal probabilities to be expressed as 3.10 and 3.11 respectively.

$$\mathbb{P}_{ni|B_k} = \frac{e^{Y_{ni}/\lambda_k}}{\sum_{j\in B_k} e^{Y_{nj}/\lambda_k}}, \tag{3.10}$$

$$\mathbb{P}_{nB_k} = \frac{e^{W_{nk}+\lambda_k I_{nk}}}{\sum_{l=1}^{K} e^{W_{nl}+\lambda_l I_{nl}}} \tag{3.11}$$

in which

$$I_{nk} = \ln \sum_{j\in B_k} e^{Y_{nj}/\lambda_k}.$$

For the derivation of equations 3.10 and 3.11 the interested reader is referred to Train (2009). Observe from 3.11 that the attributes varying over nests but not over alternatives within each nest are included. The quantity λI_{nk} is considered to be the expected utility individual n receives from the choice among alternatives within nest B_k. Quantity I_{nk} is referred to as the *inclusive utility* of nest B_k and links the *upper* and *lower* model. The upper model refers to the choice of nest; the lower model to the choice of alternative within the nest.

Nested logistic regression models maintain the property that the interval of equation 3.1 is of closed form. This convenient property allows for estimation of the model's parameters by standard maximum likelihood techniques. However, maximization could still be a demanding task due to the rugged landscape of the log-likelihood function; convergence to a global optimum is not guaranteed. To point estimation in the right direction, the analyst could commence with estimating a standard multinomial logistic regression model. The obtained parameters could then be used as starting values for the estimation of nested models. This procedure ensures appropriate starting values of the nested model and potentially eases the convergence to a global minimum. Note however, that convergences cannot be guaranteed; optimization techniques are only pointed to a possible appropriate direction.

Nested models still impose some restrictions. Note that the assumption $B = \bigcup_{k=1}^{K} B_k$ and $B_k \cap B_{k'} = \varnothing \; \forall k \neq k'$ constrain each alternative to only be part of one nest, which could potentially be an inappropriate modeling assumption. Alternatives sharing a nest are put together since it is assumed that they have similar unobserved characteristics. Of course, the possibility of an alternative sharing these characteristics with multiple nests exist; therefore it would be convenient if alternatives could belong to multiple nests. Cross-nested logistic regression models allow for this relaxation and are discussed in the subsequent section.

3.4 Cross-Nested Multinomial Logistic Regression

Cross-nested models are roughly similar to nested models with one important difference. The assumption of alternatives belonging to one nest is relaxed; alternatives can belong to multiple nests. To allow for this property, *allocation* parameters $\alpha_{ik} \geq 0$ are added to the model, indicating the degree to which alternative i belongs to nest k. Intuitively, $\alpha_{ik} = 0$ states that alternative i does not belong to nest k. For interpretability reasons the allocation parameters are usually scaled to $\sum_k \alpha_{ik} = 1 \; \forall i$. This is not a restrictive restrictive assumption however. In the remainder of this section it is assumed that this normalization has occurred. The parameter λ_k still serves the same function as in nested models and portrays the degree of independence across alternatives within nest k. Then, the probability that individual n chooses alternative i in a cross-nested structure is then defined as

$$\mathbb{P}_{ni} = \frac{\sum_k (\alpha_{ik} e^{V_{ni}})^{1/\lambda_k} \left(\sum_{j \in B_k} (\alpha_{jk} e^{V_{nj}})^{1/\lambda_k} \right)^{\lambda_k - 1}}{\sum_{l=1}^{K} \left(\sum_{j \in B_l} (\alpha_{jl} e^{V_{nj}})^{1/\lambda_l} \right)^{\lambda_l}} \tag{3.12}$$

Observe that the probability specification of a cross-nested model shares many characteristics with the probability specification of a nested structure given in equation 3.8. The difference lies in the numerator of equation 3.12 including a summation over all nests containing alternative i. The attentive reader could have observed that a cross-nested model collapses to a nested model if all alternatives are only present in one nest: $\alpha_{ik} = 1$ for $i \in B_k$.

Just as for nested models, the probability of individual n choosing alternative i can be decomposed to a marginal probability depicting the probability of choosing nest k (equation 3.13), and a conditional probability depicting the probability of choosing alternative i given nest k (equation 3.14).

$$\mathbb{P}_{nk} = \frac{\left(\sum_{j\in B_k}(\alpha_{jk}e^{V_{nj}})^{1/\lambda_k}\right)^{\lambda_k}}{\sum_{l=1}^{K}\left(\sum_{j\in B_l}(\alpha_{jl}e^{V_{nj}})^{1/\lambda_l}\right)^{\lambda_l}} \tag{3.13}$$

$$\mathbb{P}_{ni|B_k} = \frac{(\alpha_{ik}e^{V_{ni}})^{1/\lambda_k}}{\sum_{j\in B_k}(\alpha_{jk}e^{V_{nj}})^{1/\lambda_k}} \tag{3.14}$$

Note that the inclusive utility has dropped out of the equations due to the relaxation that alternatives can belong to multiple nests. For the mathematical derivation of these equations, the reader is referred to Train (2009).

Concerning optimization of the model's parameter, convergence cannot be guaranteed. Just as for the nested model, the rugged landscape of the log-likelihood function could produce several local optima. Again, parameters of the cross-nested model could be initialized by first estimating either a standard or nested model and using these estimates as initial parameter values.

3.5 Application to Vehicle Leasing Data

This section addresses how the models discussed in this chapter can be applied to the vehicle leasing data. The section is mainly concerned with defining appropriate nests for the nested and cross-nested models. No nests are required for a standard multinomial logistic regression model; fitting such a model on the vehicle leasing data is straightforward. However, as discussed in previous sections, the outcome of the standard model is very useful. The estimates of the parameters are used as starting values for both the nested and cross-nested models. Besides pointing the maximization procedure in the right direction, this procedure significantly reduces computation time.

All three models use the same explanatory variables, which can be classified as numerical and categorical variables. Recall from Chapter 2 that the numerical variables consist of the variables `catalogue price`, `commercial discount amount`, `standard discount percentage`, `total accessories amount`, `ufwt amount`, `mileage per month` and `contract duration`. The remaining variables consist of the categorical variables `country`, `client segment`, `fuel type`, `vehicle type`, `vehicle segment`, `body style`, `lease type`, `switch quarter` and `make`. Regarding categorical variables, rather than estimating one β per state of the variable, one β per state of the variable *per alternative* is estimated. Namely, for state *Diesel* of the variable `fuel type`, one β per alternative is estimated ($\beta_{\text{Diesel-Audi}}$, $\beta_{\text{Diesel-BMW}}$, etc.). Of course, as discussed in section 3.1, one of the βs per categorical variable is held fixed; all other βs are estimated with respect to the fixed β. In addition, a categorical variable with n unique states, produces $n-1$ (times the number of unique alternatives) different βs to be estimated, since the n^{th} state is a perfect linear combination of the previous $n-1$ states. Additionally, one β per alternative for each numerical variable is estimated ($\beta_{\texttt{catalogue price}\text{-Audi}}$, $\beta_{\texttt{catalogue price}\text{ -BMW}}$, etc.).

The multinomial logistic regression uses all these explanatory variables to analyze and predict. As discussed in section 3.2, the IIA assumption does not allow for correlation across alternatives. One can imagine however, that for instance adding a *Fiat 500* to the choice set should not change the decision maker's choice when he is looking for an *SUV* type of vehicle. To allow for correlations across alternatives, the alternatives are divided into nests. This division proves relatively straightforward due to the vehicle segmentation provided by the leasing company. Section 2.2 states that the leasing company assigns each model of vehicle to one particular vehicle segment. Note however, that the make of vehicle can belong to multiple vehicle segments, since a make of vehicle consists of multiple models. The exact distribution of the target variable `new make` over the variable `new vehicle segment` is shown in Table 3.1. Note that the variable `new vehicle segment` is not used for prediction. It is solely used to divide `new make` into nests.

	A	**B**	**C**	**D**	**E**	**F**	**LCV**	**MPV**	**Pickup**	**S**	**SUV**
Audi	0	167	1413	**2923**	1196	3	0	0	0	42	1308
BMW	0	0	789	**2202**	1194	10	0	525	0	2	1835
Citroen	40	316	520	73	0	0	**1333**	916	0	0	10
Ford	3	747	**1646**	926	0	0	1290	886	66	6	390
Merc-B	0	0	574	**2706**	1046	3	427	204	0	5	1158
Nissan	0	71	108	0	0	0	56	12	22	0	**1173**
Opel	2	359	**1714**	1121	0	0	322	251	0	0	224
Other	192	1139	1564	595	88	54	422	420	111	4	**2490**
Peugeot	37	616	**1839**	693	0	0	1190	499	0	1	1142
Renault	7	2287	1166	425	0	0	**2298**	864	0	0	448
Skoda	1	72	**1294**	752	0	0	0	1	0	0	134
Volksw.	28	312	2345	**3360**	24	0	1221	1476	35	0	942
Volvo	0	0	287	496	340	0	0	0	0	0	**906**
Total	310	6086	15259	16272	3888	70	8559	6054	234	60	12160

Table 3.1: Overview of the distribution of the target variable `new make` over the variable `new vehicle segment`. This segmentation is provided by the leasing company. The vehicle segment to which an alternative is assigned to most often, is stated in bold. The rows indicate the alternatives, whilst the columns indicate the states of the variable `vehicle segment`.

From Table 3.1, it becomes clear that all makes belong to multiple vehicle segments. For the nested multinomial logistic regression model, alternatives are restricted to be part on only one nest. To determine the nesting structure of this model, and to determine which nest each alternative belongs to, each alternative is assigned to the vehicle segment in which the alternative occurs most. This assignment is shown in Table 3.2. Observe from this table that only four unique vehicle segments are assigned to be a nest: *C*, *D*, *LCV* and *SUV*. Each nest has at least two alternatives that belong to it. A schematic representation of the nesting structure of the model is visualized in Figure 3.1.

Alternative	Nest
Audi	D
BMW	D
Citroen	LCV
Ford	C
Mercedes-Benz	D
Nissan	SUV
Opel	C
Other	SUV
Peugeot	C
Renault	LCV
Skoda	C
Volkswagen	D
Volvo	SUV

Table 3.2: Assignment of nests to each alternative of the choice set. Each nest corresponds to the nest in which the alternative occurs most.

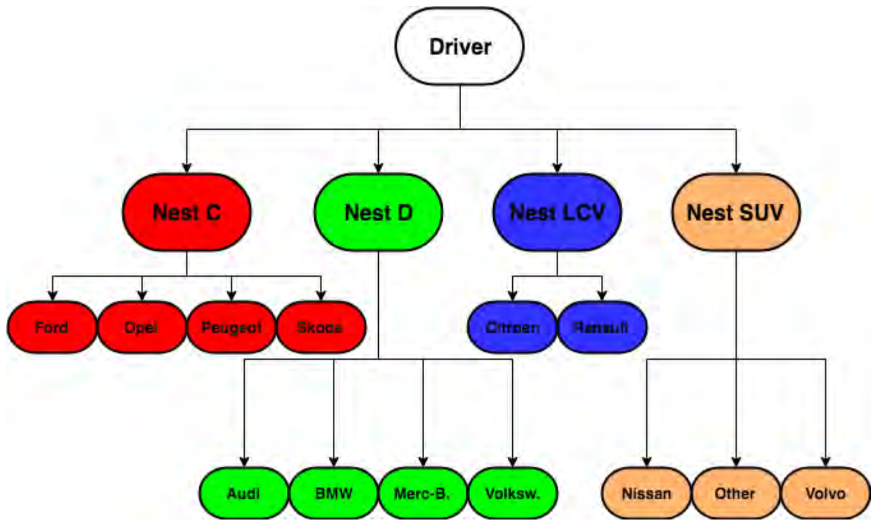


Figure 3.1: Schematic overview of the nesting structure used for the nested multinomial logistic regression model. Each nest has its own color. In addition, each alternative is only part of one nest.

Regarding the cross-nested multinomial logistic regression, the restriction of each alternative belonging to one nest is dropped. Each alternative is allowed to be contained in multiple nests. The model estimates the allocation parameters α, indicating the degree to which an alternative belongs to a particular nest. To determine the nests each alternative belongs to, and the allocation parameters α corresponding to these nests, Table 3.1 is used. First, each number of this table is divided by the total occurrences of the alternative in the data. This procedure results in a table denoting the fraction of the number of times an alternative is assigned to a particular vehicle segment over the total number of occurrences of the alternative in the data. The idea is that these fractions function as starting values for the allocation parameters α in the cross-nested multinomial logistic regression model. All fractions less than 0.10 are considered to equal zero. For interpretability reasons, the allocation parameters are usually scaled to $\sum_k \alpha_{ik} = 1 \ \forall i$, with i denoting the alternative and k the nest. Therefore, all fractions less than 0.10, but greater than zero are added to different α. Since these α solely function as starting values for the cross-nested model, this should not impose a problem. Table 3.3 depicts the starting values of the allocation parameters α stemming from the above-mentioned procedures. Figure 3.2 depicts a schematic overview of the cross-nested structure. Note that for interpretability reasons, only three nests are depicted.

	B	**C**	**D**	**E**	**LCV**	**MPV**	**SUV**
Audi		0.23	0.41	0.17			0.19
BMW		0.20	0.34	0.18			0.28
Citroen		0.29			0.42	0.29	
Ford	0.19	0.28	0.16		0.22	0.15	
Mercedes-Benz			0.64	0.17			0.19
Nissan							1.00
Opel		0.72	0.28				
Other	0.41	0.22					0.37
Peugeot		0.49	0.12		0.20		0.19
Renault	0.41	0.16			0.31	0.12	
Skoda		0.67	0.33				
Volkswagen		0.38	0.34		0.13	0.15	
Volvo		0.14	0.24	0.17			0.45

Table 3.3: Distribution of the alternatives over the different nests. Each value depicts the starting value of the allocation parameter α associated with the alternative and the nest. Note that the sum of each row is equal to 1, satisfying the normalization $\sum_k \alpha_{ik} = 1 \ \forall i$. The rows indicate the alternative, whilst the columns indicate the nests.

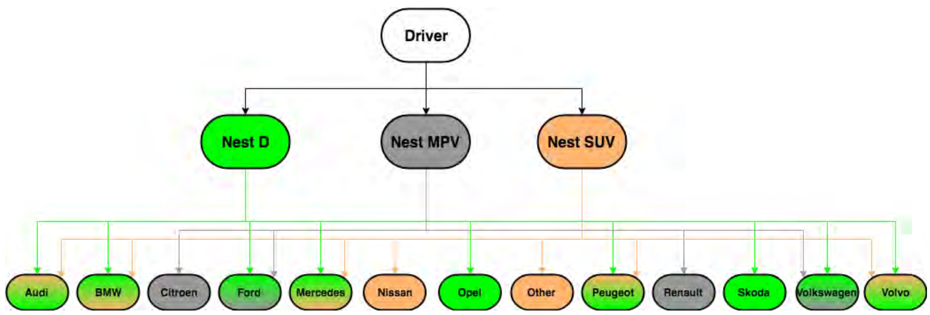


Figure 3.2: Schematic overview of the nesting structure used for the cross-nested multinomial logistic regression model. Each nest is associated with its own color. Contrary to the nested model, alternatives are allowed to belong to multiple nests. The color(s) of the alternatives indicate the nest(s) they belong to. Note that for interpretability reasons, only three nests are shown.

Chapter 4

Feedforward Neural Networks

This chapter discusses the second family of models used to model the vehicle leasing data, *feedforward neural networks*. These models are not frequently used to model mode choice. However, due to their successful history in many fields, it is researched whether such models can outperform discrete choice models on this particular case. Section 4.1 describes the properties common to all feedforward neural networks. The general framework of such models is introduced and elaborated on. The subsequent sections dive into the relevant parts of the network. Section 4.2 discusses the output produced by such networks. In addition, the loss function used to produce these outputs is touched upon. Section 4.3 describes the activation functions considered for this thesis, and dives into the strengths and weaknesses of each of them. Section 4.4 addresses the methods used by which neural networks *learn* the optimal parameters. Note that a significant part of this section is readily applicable to the methods discussed in chapter 3. The next two sections describe characteristics of the network produced for this thesis. Section 4.5 describes a methodology used to overcome some limitations of a neural network. Section 4.6 addresses *Evolutionary Optimization*. This technique is used to find the optimal hyper-parameters of the network. The chapter is concluded with a section discussing the applicability of feedforward neural networks on the vehicle leasing data. Note that almost all mathematical derivations and formulae are derived from Goodfellow et al. (2016). Text, derivations and formulae not originating from this source are cited accordingly.

4.1 General Properties

Feedforward neural networks are the best known and most widely used deep learning models. These models form the basis for many different kinds of neural networks, such as convolutional and recurrent neural networks, which are widely used in the fields of image recognition and natural language processing. Feedforward neural networks approximate some function $y = f^*(x)$, mapping input x to output y. These networks define a mapping $y = f(x; \theta)$ and learn the parameters θ that best approximate $f^*(x)$. They are called feedforward since information only flows forward within the network; no feedback connections occur. Intermediate output of the network is only used as input of a subsequent layer, never as input to the same or a preceding layer of the network. Neural networks containing feedback connections are considered recurrent neural networks, and are better able to model time dependencies. The process of mapping input x to output y typically consists of many transformations of x performed by multiple functions contained within the network. All functions used to map x to y define the network and perform intermediate computations to define f. In other words, feedforward neural networks can be defined as *directed acyclic graphs*, describing how all functions approximate the behavior of $f^*(x)$. If a network is composed of three functions $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$, the network can be described as $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$, in which each function depicts a *layer* of the network. The *depth* of a network is defined as the total number of layers of which the network is composed. By definition, the first and last layer are defined as the input and output layer of the network respectively. The behavior of the intermediate layers is defined by the data used to train the network. Since many dependencies between layers exist, the network should figure out how to use each layer to best approximate $f^*(x)$. Since output of the intermediate layers is typically only used as input to subsequent layers, and not shown to the analyst, these layers are referred to as *hidden* layers.

The link with biology stems from the construction of the layers of a network. Generally, each layer of the network is vector valued, and comprises of *neurons*. The neurons of a layer mimic the behavior of the neurons of a human brain. They receive input from many other neurons and only activate if a certain threshold is exceeded. Due to the individual behavior of each neuron, the layers in a network can be thought of as functions representing many individual units acting in parallel. All neurons take input x, and multiply the input by a set of weights corresponding to the neuron. A *bias* or threshold value is added to this multiplication, depicting the activation threshold of the neuron. The difference with for instance logistic regression models discussed in chapter 3, lies in the fact that this linear in parameter value $w^\top x + b$ is then transformed using a non-linear *activation* function ϕ. The output or activation a of a neuron can then be defined as $a = \phi(w^\top x + b)$. Using this information, the first layer of a network is then defined as

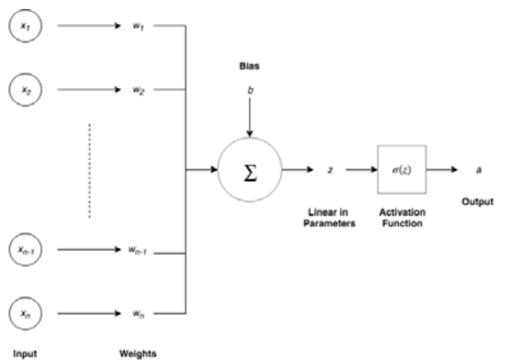
$$h^{(1)} = \phi^{(1)}\Big(W^{(1)\top}x + b^{(1)}\Big), \tag{4.1}$$

in which $W^{(1)\top}$ depicts the transpose of the weight matrix. Each column $w_i \in W^{(j)}$ corresponds to the weight vector of neuron i of layer j. The successive layers are defined in similar fashion:

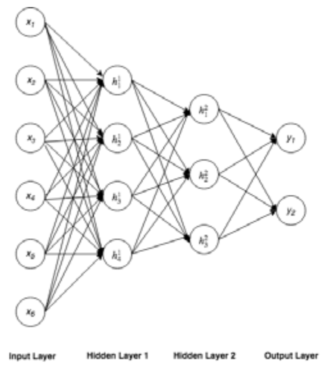
$$h^{(j)} = \phi^{(j)}\Big(W^{(j)\top}h^{(j-1)} + b^{(j)}\Big)\ \forall j \geq 2. \tag{4.2}$$

Note that the only difference between equations 4.1 and 4.2 consists of the input the layers receive. All layers $j \geq 2$, receive the transformed output of a preceding layer as input.

Figure 4.1 portrays the structure of one neuron and a schematic overview of an entire feedforward neural network. Figure 4.1a depicts the schematic overview of a neuron, mapping input x to output y, by representing the input as linear in parameter transformed by a non-linear activation function. Figure 4.1b depicts the general structure of a network. The size of the input layer corresponds to the number of input features the network receives. The number of hidden layers and the number of neurons per hidden layer are hyper-parameters that should be set by the analyst. The size of the output layer corresponds to the number of desired output values. Note that the output value of a neuron in layer l is used as the input value to all neurons in layer $l+1$.



(a) Schematic overview of a neuron in a neural network. All x denote the input variables, w denote the weights corresponding to each input parameter, b denotes the bias value, z denotes the linear in parameter, $\sigma(z)$ portrays the activation function associated with the neuron, and a the final output of the neuron.



(b) Schematic overview of a neural network architecture. The network consists of an input layer with six neurons, two hidden layers containing four and three neurons respectively, and an output layer of two neurons.

Figure 4.1: Schematic overview of a neuron (left) and of an entire network (right).

4.2 The Output Layer and Loss Functions

One of the main limitations of linear models is their restriction to being modeled by linear functions, resulting in models not able to understand the possible interaction between two input variables. The addition of activation functions assists in tackling this problem. Transforming the input allows for interaction between input variables, however it imposes a different problem: optimization of model parameters becomes quite challenging. Prior to diving into the techniques used to optimize performance of neural networks, a better understanding of the loss and activation functions present in these networks is paramount. The choice of loss and activation functions are heavily intertwined and determine the strength of the network. Most loss functions used to estimate linear models can be used when optimizing a neural network. In most cases, as is the case for the vehicle leasing data set, the network defines a distribution $\mathbb{P}(y|x;\theta)$, and maximum likelihood is used to optimize the network's parameters θ. The negative log-likelihood, or equivalently, the *cross-entropy* between the training data and the model's distribution is used as the loss function. The loss function is then given by

$$\mathcal{L}(\theta) = -\mathbb{E}_{x,y\sim\hat{\mathbb{P}}_{\text{data}}} \ln(\mathbb{P}_{\text{model}}(y|x)), \tag{4.3}$$

and the estimator is the value of θ that maximizes this function. Observe that the negative log-likelihood loss function is used to optimize all models discussed in this thesis. Recall however that McFadden et al. (1973) showed that equation 4.3 is only globally concave for the multinomial logistic regression model. The loss functions of the (cross-)nested models discussed in sections 3.3 and 3.4 could potentially possess more than one local maximum, and are therefore estimated using gradient-based optimization techniques. It should be intuitive that the non-linearity of a neural network causes its loss function to have a rugged landscape. Gradient-based optimization and its applicability to neural networks are discussed in section 4.4.

The non-linear nature of neural networks creates several advantages. However, these advantages go hand in hand with some difficulties. One of the main difficulties of constructing a neural network is the problem of *vanishing gradients*. It is imperative that the gradient of the loss function is large and predictable enough to pose as a good measure for the network. Functions that become very flat, or *saturate*, cause the gradient to become vanishingly small, preventing the weights of the network from updating. In the worst case scenario, the loss value becomes unidentifiable and the network is prevented from further training. Most of the time, a vanishing gradient occurs when activation functions of neurons saturate. Since most activation functions of neurons possess an exponential function that has potential to saturate, negative log-likelihood is a good way of avoiding this problem. The natural log term of equation 4.3, undoes the exponential term of the activation function, preventing neurons from saturating. For this exact reason, loss functions such as mean absolute error and mean squared error, are typically less applicable to neural networks; they do not contain a logarithmic term.

It should be apparent to the reader that the choice of loss function and the choice of activation functions of each layer of the network are strongly related. A discussion of the link between the loss function and the output neurons of a network will further ratify this observation. Note that each output neuron of a network could be used as a hidden neuron as well. However the choice of structure of the output neurons determines the form of the loss function, and therefore a thorough discussion is appropriate. This discussion will focus on output neurons used in the vehicle leasing data case: *multinomial softmax neurons*. For a broader overview of all types of output neurons, the reader is referred to Goodfellow et al. (2016).

The output layer of the network has the task of transforming the provided hidden features $h = f(x;\theta)$ to the desired output format. When dealing with the vehicle leasing data, the network should specify a probability distribution over all of the alternatives of choice set C of size J. For all alternatives $i = 1, ..., J$, the network should predict the probability that individual n chooses alternative i: $\mathbb{P}_{ni} = \mathbb{P}(y = i \mid x)$. Concatenating all these probabilities, the output layer should predict a vector $\hat{y}$ of size J, with $\hat{y}_i = \mathbb{P}_{ni} = \mathbb{P}(y = i \mid x)$. To achieve this task, the output layer, as for any layer of the network, first transforms the input by a linear in parameter multiplication. Suppose that $z = W^\top h + b$ represents the unnormalized log probabilities. To produce the desired, valid probability output vector, these log probabilities should be normalized. Recall from section 3.2, that equation 3.5, the *softmax* function, exactly achieves this requirement by exponentiation and normalization. Rewriting this equation to comply with the used notation, the output vector is then calculated as follows:

$$\hat{y} = \frac{e^{z_i}}{\sum_j e^{z_j}}, \tag{4.4}$$

where $z_i = w^\top h + b_i$. Again, the exponential terms of the softmax are useful to avoid saturation of the loss function. Taking the natural log of equation 4.4 one obtains,

$$\ln\hat{y} = z_i - \ln \sum_j e^{z_j} \tag{4.5}$$

From equation 4.5, observe the importance of the term z_i. It implies both direct contribution of the input to the loss function, and the avoidance of saturation of the loss function. The contribution of the term $\ln \sum_j e^{z_j}$ can roughly be interpreted as follows. This term can more or less be approximated by the maximum value of z. Elaborating on this approximation, the assumption is made that any e^{z_j} is deemed insignificant for any z_j significantly lower than the maximum of z. Then, one can recognize that the negative log-likelihood loss function heavily punishes the most incorrect prediction. If the correct alternative already has the largest input to equation 4.4, the terms of equation 4.5 will roughly cancel, and the training example will barely contribute to the total loss value.

All properties of the output layer and the loss function described so far are common to most machine learning algorithms using gradient based optimization. The output layer of the multi-class classification network just described is closely related to the multinomial logistic regression model discussed in section 3.2. The main difference consists of the input received by the multinomial logistic regression model, and the output layer of the neural network. The logistic regression model receives the input as specified by the user, whereas the output layer of the network receives input consisting of hidden features generated by the preceding layers. Although modest at first glance, the difference in received input could potentially be substantial. The next section gives a thorough explanation of the input received by the output layer of the network. The hidden neurons of a network, a unique characteristic of neural networks, are discussed in detail. These neurons are considered to be the most significant difference between models discussed in chapter 3 and neural networks; they are able to capture unobserved and complicated relationships between input variables affecting the loss of a model.

4.3 Hidden Neurons and Activation Functions

This section concerns the defining property of neural networks, activation functions of hidden neurons. It is assumed, as is the case for the vehicle leasing network, that each neuron accepts a vector of input values x, transforms these inputs by the linear in-parameter transformation $z = W^\top x + b$, and applies an element-wise non-linear activation function $\sigma(z)$ to produce an output vector associated with that neuron. One can imagine that the choice of activation function σ greatly affects the output of each neuron. Some relevant and widely used activation functions are discussed in this section. The properties and (dis)advantages of each of these functions are addressed. The discussion of activation functions is restricted to the functions used in this thesis. The choice of activation functions is based on an empirical study and the availability of functions in the packages used for modeling (Chollet et al., 2015). The functions discussed in this section, in respective order, are the *sigmoid*, *hyperbolic tangent*, *ReLU* and *ELU* activation functions.

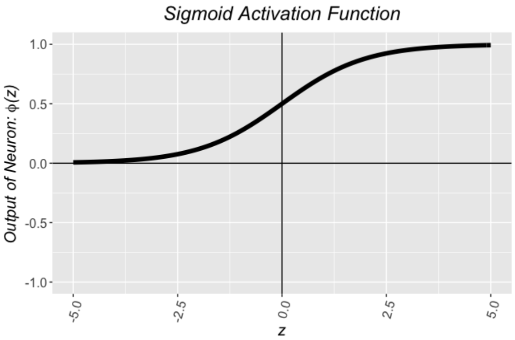
Prior to the introduction of the ReLU activation function, most networks used the (logistic) sigmoid activation function. The attentive reader should note that this function has already been discussed in detail. This function was first introduced in section 3.2 as the probability distribution of the multinomial logistic regression model, and is given in equation 3.5. In addition, this activation function is used within the network to define a probability distribution over all output alternatives. The sigmoid activation function is one of the main reasons why all numerical input variables are scaled to have mean zero, and a standard deviation of one, as discussed in section 2.2. Sigmoid neurons saturate across most of their domain. Observe from Figure 4.2a, that the sigmoid activation only does not saturate when z is relatively close to zero. Input values significantly deviating from zero may cause the neuron to saturate, restricting the network from converging. Scaling of the input values, and the choice of an appropriate loss function are ways of preventing saturation of neurons. However, other activation functions might be more applicable.

Another popular activation function, widely used before the rise of the ReLU function, is the hyperbolic tangent (*tanh*) function. This activation function is simply a rescaled version of the sigmoid activation function. Whereas, the sigmoid function produces output values in the range $(0, 1)$, the hyperbolic tangent allows for negative output values in the range $(-1, 1)$. Since scaling of the numerical values evolves around zero, with standard deviation of one, one can imagine that the hyperbolic tangent is generally a more applicable activation function. The hyperbolic tangent is defined as

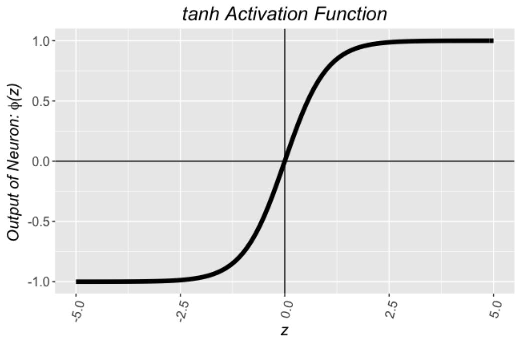
$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \tag{4.6}$$

Observe that equation 4.6 is equal to $2\sigma(2z) - 1$, in which σ denotes the sigmoid activation function. The slope of the hyperbolic tangent is given in Figure 4.2b. Observe that the output of tanh-neurons indeed ranges from -1 to 1. In addition, note that the range of values of z for which tanh saturates is even smaller than for the sigmoid function. However, since input values are almost always scaled, this should not impose greater problems than for the sigmoid function.

The next activation function, the Rectified Linear Unit (ReLU) activation function, was first proposed for restricted Boltzmann machines (Nair and Hinton, 2010), after which this activation function was successfully applied to neural networks (Glorot et al., 2011). ReLU



(a) Sigmoid activation function producing output values in the range $(0, 1)$.



(b) tanh activation function producing output values in the range $(-1, 1)$

Figure 4.2: Sigmoid (left) and tanh (right) activation functions.

is one of the most frequently used activation functions, due to the fact that it remains very close to being linear. The function is considered as a piece-wise linear function with two linear pieces and is given by

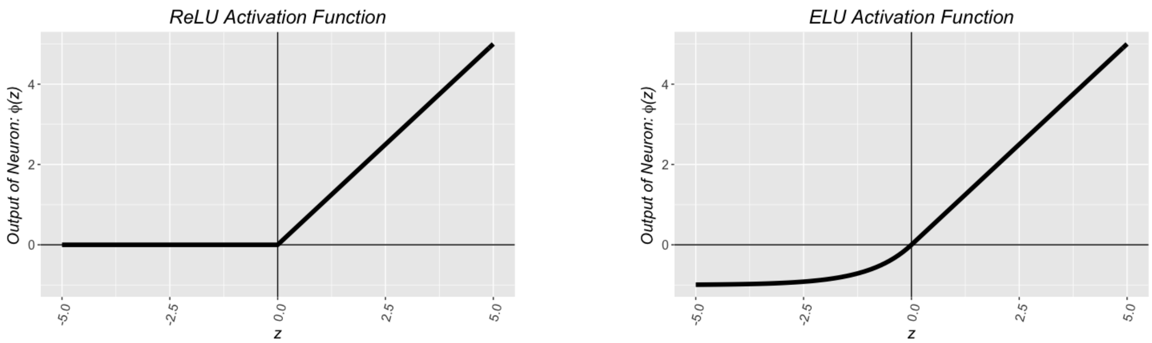
$$\sigma(z) = \max(0, z).$$

The near-linear nature of ReLU preserves properties that make linear models straightforward to optimize by gradient-based methods. Observe from Figure 4.3a that the only difference with a linear neuron is that ReLU outputs zero across half its domain. A convenient property is that the more active a ReLU neuron is, the larger the derivatives through this neuron become. This property causes the gradients to be large and consistent. However, note that a ReLU neuron is not differentiable across half its domain. Intuitively, one would assume that gradient-based optimization methods are not applicable. However, since neural networks almost never arrive at a local minimum of the loss function, it is not expected that training reaches a point where the gradient vector is equal to zero. Therefore, it is generally not a problem that minima of the loss function correspond to points with an undefined gradient vector. In addition, neurons that are not differentiable at all points, are typically differentiable across most of its domain. One possible method of avoiding ReLU neurons to saturate, is to initialize the weights of these neurons to be small positive numbers greater than zero. Doing so increases the chance of these neurons to be active at initialization. In addition to the potential non-differentiablity of ReLUs, these neurons are always non-negative and therefore have a mean activation greater than zero. Neurons with mean activation greater than zero could potentially act as a bias for the next layer. If such neurons do not cancel each other out, learning could cause a bias shift for neurons in the next layer. The more correlated neurons, the higher their bias shift. Bias shifts can greatly complicate gradient-based optimization, hence one typically aims at activation function with activation means close to zero (Clevert et al., 2015). Even though the almost-linear properties of ReLU cause this activation function to be widely used, generalizations dealing with the potential non-differentiablity of ReLUs and non-zero mean activation values do exist.

The Exponential Linear Unit (ELU) activation function deals with both these limitations. This function is defined as

$$\sigma(z) = \begin{cases} z & \text{if } \ z > 0 \\ \alpha e^{z} - \alpha & \text{if } \ z \leq 0, \end{cases} \tag{4.7}$$

and was first introduced by Clevert et al. (2015). The parameter α in equation 4.7 controls the value to which an ELU saturates for negative input values. Contrary to ReLUs, ELUs allow for mean activation values close to zero. Not only does this simplify the learning procedure, this property also ensures faster learning (Clevert et al., 2015). In addition, these neurons saturate to a smaller value with smaller input values, allowing the gradient vector to exist. A visual representation of the ELU activation function is depicted in Figure 4.3b.



(a) ReLU activation function producing output values in the range $[0, z)$.

(b) ELU activation function producing output values in the range $(-1, z)$. The saturation parameter α is set to one.

Figure 4.3: ReLU (left) and ELU (right) activation functions.

4.4 Learning in Neural Networks

The non-linear nature of neural networks cause optimization of network parameters to be a challenging task. The optimal set of weights is usually found using iterative gradient-based optimizers. Due to the maximum likelihood principle, optimization of this kind is also applicable to the methodologies discussed in chapter 3. This section will mainly focus on the methods used to train a neural network, yet the discussed gradient-based techniques are readily applicable to the logistic regression models.

In each feedforward neural network, input x is distributed forward through the network producing output vector $\hat{y}$. This procedure is referred to as *forward-propagation* and produces loss value $J(\theta)$ based on the loss function $\mathcal{L}(\theta)$. To achieve an optimal loss value, this information is then distributed backwards through the network: *back-propagation.* This procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the output vector of the network $\hat{y}$, and the desired output vector y (Rumelhart et al., 1986). A fundamental part of the back-propagation algorithm consists of *gradient descent*, first discovered by Cauchy (1847).

Ideally, the analyst will find the lowest possible value of $J(\theta)$, the global minimum of the loss function, thus finding the optimal weights of the network. However, the rugged landscape of the loss function makes this nearly impossible. Besides the possibility of the loss function containing many local minima, the loss function could also hold many *saddle points*, critical points that are neither local minima nor maxima. These aspects of the network cause optimization algorithms to seek for a significantly low value of $J(\theta)$, rather than finding the global minimum of the loss function. Consider the minimization of loss function $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$. Note that this example is applicable to the vehicle leasing case, since during training the input features are mapped to one loss value. Recall from simple calculus that at critical points of a function, all derivatives are equal to zero. The gradient of $\mathcal{L}(\theta)$, $\nabla_\theta \mathcal{L}(\theta)$, denotes the vector containing all partial derivatives with respect to θ. The goal of gradient descent is therefore to arrive at a point where $\nabla_\theta \mathcal{L}(\theta) = 0$, and the loss value $J(\theta)$ is significantly low. To achieve this objective, the gradient vector of $\mathcal{L}$ is iteratively calculated and the weights θ are updated on each iteration. To minimize $\mathcal{L}$ it is imperative to know in which direction $\mathcal{L}$ decreases the fastest. Suppose all weights $\theta_i \in \theta$ are moved $\Delta\theta_i$ in the θ_i direction. The change in $\mathcal{L}$ can be denoted by

$$\Delta\mathcal{L} \approx \frac{\partial \mathcal{L}}{\partial \theta_1}\Delta\theta_1 + ... + \frac{\partial \mathcal{L}}{\partial \theta_n}\Delta\theta_n = \nabla_\theta \mathcal{L}(\theta) \cdot \Delta\theta.$$

Observe that decreasing $\mathcal{L}$ corresponds to $\Delta\mathcal{L}$ being negative. Negativity of $\Delta\mathcal{L}$ can be enforced by restricting $\Delta\theta = -\epsilon\nabla_\theta \mathcal{L}(\theta)$, in which ϵ denotes a small positive value, the step size or *learning rate.* Doing so, the changes in $\mathcal{L}$ can be written as:

$$\Delta\mathcal{L} \approx -\epsilon||\nabla_\theta \mathcal{L}(\theta)||^2. \tag{4.8}$$

Forcing the learning rate to be positive ensures that the value of $\mathcal{L}$ will always be decreased. So, by iteratively moving in the opposite direction of the gradient, $\mathcal{L}$ is slowly pushed towards a minimum value. The weights are then updated as follows:

$$\theta \to \theta' = \theta - \epsilon\nabla_\theta \mathcal{L}(\theta) \tag{4.9}$$

When all weights are updated and the stopping criterion is not met, the same procedure is repeated. That is, the network produces an output value $J(\theta)$ based on the updated set of weights, after which the gradient descent algorithm is performed again. This procedure is typically repeated until either a specific number of *epochs* (iterations) have been completed, or $J(\theta)$ does not improve anymore.

In most networks, an extension of gradient descent, *stochastic* gradient descent is used as optimization algorithm. As the size of the provided data grows, the time it takes to compute the sum of all gradients can become excessively large. The underlying thought of stochastic gradient descent is that the gradient is an expectation, and could therefore be approximately estimated by using a small set of samples. Stochastic gradient descent randomly draws a *mini-batch* of examples from the entire data set. This mini batch is then used to calculate an estimate of the true gradient:

$$g = \frac{1}{m} \nabla_\theta \sum_{i=1}^{m} \mathcal{L}(x^{(i)}, y^{(i)}; \theta),$$

in which m denotes the size of the mini-batch. The estimate of the gradient g is then used to update the weights θ, in similar fashion as in equation 4.9.

One can imagine that computing the partial derivatives with respect to all weights and biases in the network can become computationally expensive. The back-propagation algorithm is a way of circumventing this computational burden. Back-propagation itself is not used to update the parameters of the network. It is simply an effective and clever way of computing the gradient of the loss function with respect to any weight or bias in the network. Since most neural networks are composed of multiple layers, or equivalently multiple functions, many gradient calculations need be repeated several times. Back-propagation avoids this problem by computing the chain rule of calculus with a specific order of operations that is highly efficient. To illustrate the matter, suppose that $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, $g : \mathbb{R}^m \to \mathbb{R}^n$, and $f : \mathbb{R}^n \to \mathbb{R}$. If $y = g(x)$, and $z = f(y)$, then the chain rule states that

$$\nabla_x z = \left(\frac{\partial y}{\partial x}\right)^\top \nabla_y z, \tag{4.10}$$

in which $\frac{\partial y}{\partial x}$ denotes the *Jacobian* matrix of g. The Jacobian matrix can be defined as the matrix of all first-order partial derivatives of a vector-valued function. From equation 4.10 it becomes clear that the gradient vector of layer $l-1$ of the network can be computed by multiplying the gradient vector of layer l by the Jacobian matrix of layer l with respect to layer $l-1$. Back-propagation consist of performing equation 4.10 for each layer of the network, and is therefore able to quickly calculate the gradients for all weights and biases of the network.

Summarizing, at initialization of the network all weights and biases of the network are set to small random values. Next, for each training input x, the following steps are performed. First, x is propagated forward through the network, and the output error is captured in the loss value $J(\theta)$. Following, for each layer contained in the network the gradient is calculated and equation 4.10 is performed. Lastly, all weights and biases are updated using gradient descent such that the loss function is moved to its minimum value by moving in the opposite direction of the gradient. This process is repeated until a stopping criterion is met.

The degree to which training of a network is considered successful is dependent on the provided training data. The non-linear hidden layers of a network allow for learning complicated relations between inputs and outputs. When training data is sparse however, many of these relationships are the results of sampling noise. These relationships will only exist in training data, and the network will therefore not generalize well on unseen data. This phenomenon is better known as overfitting, and many techniques to avoid this exist. Introducing weight penalties and stopping as soon as performance on validation data worsens are two of them. The best way of avoiding overfitting of a network of fixed structure is to average the predictions of all possible settings of the network parameters. Of course, this is infeasible, due to the computational power required. The technique used in this research to prevent overfitting tries to accomplish this and is called *dropout* (Srivastava et al., 2014). Dropout prevents from overfitting and offers a way of approximately combining exponentially many different network structures effectively. Dropout simply refers to temporarily removing both hidden and visible neurons from the network, along with all its incoming and outgoing connections. If dropout is added to a layer, a fixed amount of neurons is dropped from the layer during training. The choice of which neurons is random. During training of the network these dropped neurons are simply ignored, and training continuous with a thinned network. Therefore, a neural network containing n neurons can be viewed as a collection of 2^n possible thinned networks. All these networks still share weights. For each training case, a new thinned network is sampled and trained. So, training a network with dropout can be seen as training a collection of 2^n thinned networks with extensive weight sharing, where each thinned network gets trained very rarely, if at all (Srivastava et al., 2014).

4.5 Entity Embeddings of Categorical Variables

As stated in earlier sections, neural networks can in principle approximate any continuous function and piece-wise continuous function (Cybenko, 1989; Llanas et al., 2008). However, neural networks typically do not excel when approximating arbitrary non-continuous functions; the continuous nature of neural networks limits their applicability to categorical variables (Guo and Berkhahn, 2016). There are several possibilities of transforming categorical variables such that these can be used as input to the network. However, all these methods possess severe weaknesses. Using integer representation for categorical variables does not work well. For example, when mapping the categorical variable *make* with three categories, Audi, BMW and Volkswagen to their integer representation 1, 2 and 3 respectively, the network assumes that Audi and BMW are more similar to each other than Audi and Volkswagen, since 1 is closer to 2, than 1 is to 3. To avoid these possible non-existing relationships, categorical variables are typically transformed to their *one-hot encoding*; each state of the categorical variable is mapped to a dummy variable and is used as a separate input variable of the network. This method solves the problem of non-existing relationships between states. However, it imposes two new restrictions. Firstly, it treats different states of categorical variables completely independent of each other and ignores the informative relations between them (Guo and Berkhahn, 2016). Secondly, this procedure can become computationally expensive. A categorical variable with n states, is mapped to n - 1 input features. The larger n becomes, the more parameters the network should learn, and the more time it takes for the network to converge.

To obtain a measure of similarity for categorical variables, and to avoid the problem of significant training time, *embedding layers* are introduced. These types of layers turn positive integers into dense vectors of fixed size (Chollet et al., 2015). Using these layers, allows for the network to learn the representation of categorical features in multi-dimensional spaces which puts values with similar effect on the loss function of the network close to each other, and thereby reveals the intrinsic continuity of the data (Guo and Berkhahn, 2016). Embedding layers are not commonly used to map categorical features to continuous space. Nonetheless, these layers have revolutionized the field of natural language processing (NLP). Mikolov et al. proposed a new model architecture in which words and phrases are mapped into a continuous distributed vector in a semantic space (2013). Doing so, significantly reduced training time of the network, whilst concurrently outperforming state-of-the-art NLP-methods. Interestingly enough, they showed that not only the distance between word representations matter, the direction of the difference is also meaningful.

Next, the derivation of embeddings of categorical variables is explained. This methodology slightly differs from the approach of Mikolov et al. Their approach uses a sequence of words to create embedding representations, whilst the approach used in this thesis only uses the labeled representation of the categorical variable as input to the embedding layer. For a thorough explanation of the *word embedding* methodology, the reader is referred to Mikolov et al. (2013).

Intuitively, the goal of the network remains to approximate some arbitrary function $y = f^*(x)$. The only difference arises from the way categorical variables are processed by the network. To start, suppose the goal is to map each state i of a categorical variable x with m different states, to its embedding representation of size n:

$$e_i : x_i \mapsto \mathrm{x}_i,$$

where $i \in [1, m]$, $|\mathrm{x}_i| = n$, $|x_i| = m$, and $m > n$. Note that the value of n is a hyper-parameter to be set by the analyst. To achieve this mapping, each state of the categorical variable x is first mapped to its corresponding one-hot encoding:

$$u_i : x_i \mapsto \delta_{x_i\alpha}. \tag{4.11}$$

In equation 4.11, $\delta_{x_i\alpha}$ depicts the *Kronecker-delta* function, defined as,

$$\delta_{x_i\alpha} = \begin{cases} 1 & \text{if } x_i = \alpha \\ 0 & \text{otherwise,} \end{cases}$$

and α spans the same range as i: $\alpha \in [1, m]$. Then $|\delta_{x_i\alpha}| = m$, and will only be non-zero when $\alpha = x_i$. To obtain the desired embedding representations of all states, the one-hot representations are used as input to the embedding layer. The embedding layer solely consists of linear neurons and is therefore defined as

$$\mathrm{x}_i = \sum_{\alpha} w_\alpha \delta_{x_i\alpha} \tag{4.12}$$

Note that the Kronecker-delta function has the useful property that $\sum_i a_i \delta_{ij} = a_j$. Using this property equation 4.11 can be rewritten as

$$\mathrm{x}_i = \sum_{\alpha} w_\alpha \delta_{x_i \alpha} = w_{x_i}.$$

From this property, one can observe that the learned embedding representations are simply the weights of the embedding layer, and can be learned by the network using standard learning techniques.

Making this methodology more concrete with an example, suppose x is a categorical variable with $m = 4$ different states, and the desired size of the embeddings is $n = 2$. The procedure described above is visualized in Figure 4.4 for x_1 and x_3 respectively. Note that, as for any weight of the network, the weights of the embedding layer are initialized randomly. They are updated in exact same fashion as described in section 4.4.

Figure 4.4: The embedding procedure visualized for states 1 and 3 of a categorical variable x with $m = 4$ different states. All states are mapped to an embedding vector of size $n = 2$.

To learn these embedding vectors for each state of each categorical variable, whilst simultaneously training the network, the network structure is changed slightly. Prior to training of the network, all numerical and categorical variables are split. The numerical variables are treated as they usually are. For each categorical variable, an embedding layer is added to the network. All embedding layers, and the hidden layer processing all numerical variables are merged horizontally to produce a *new* input layer. This network structure allows for simultaneously learning the embedding vectors and training the network. A visual representation of the structure of the network is shown in Figure 4.5. Observe that the network acts as any feedforward neural network after concatenation of the embedding layers and the first hidden layer processing the numerical variables.

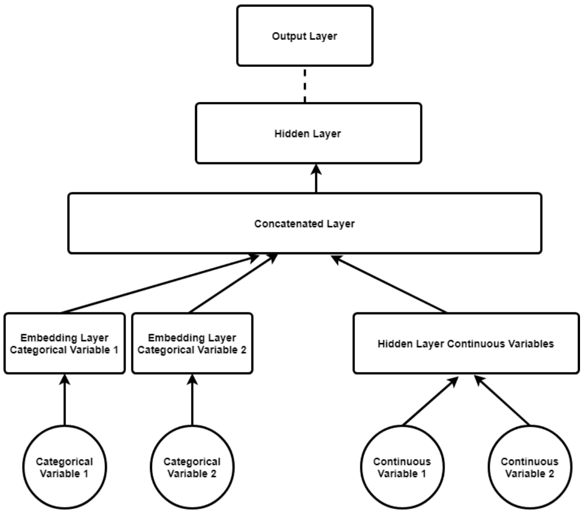


Figure 4.5: Structure of an *embedding network* with two categorical and two numerical input variables. For each categorical variable, an embedding layer is added to the network. These layers act in parallel with the first hidden layer processing the numerical variables. Note that the intermediate one-hot transformation of the categorical variables is omitted.

4.6 Evolutionary Optimization of Network Hyper-Parameters

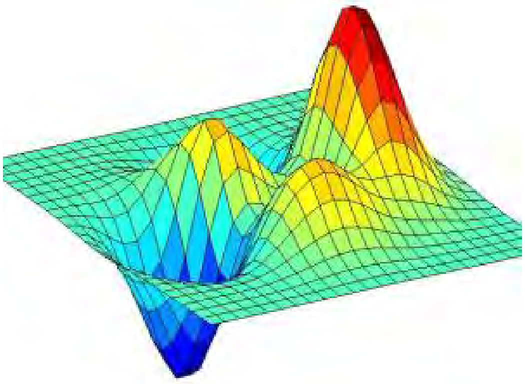
From previous sections it has become evident that many different network topologies can arise from different sets of hyper-parameters. Finding hyper-parameters that optimize performance of the network usually occurs by means of trial and error. However, trial-and-error procedures can almost never ensure optimal network hyper-parameters. To avoid such a methodology, and to avoid a greedy search procedure of all possible hyper-parameters, this section introduces *evolutionary optimization* algorithms. The goal of such optimization techniques is to optimize some function f by applying principles based on Darwin's *The Origin of Species* (1859). Although known by most people, a brief recap of Darwin's theory of evolution will ease the introduction of algorithms inspired by his theory. Note that most information stated in this section stems from Eiben et al. (2003). Information originating from other sources is cited accordingly.

Darwin's theory of evolution explains the origins of biological diversity. The principle of natural selection plays a significant role in his theory. It is assumed that the environment can host a limited number of *individuals*. Since these individuals possess the instinct to reproduce, and the *population* size is not to grow exponentially, *selection* becomes inevitable. The principle of natural selection favors the individuals that compete for the given resources most effectively. This principle is better known as *survival of the fittest*. The above-introduced competition-based selection is seen as the first building block of evolutionary progress. The second building block identified by Darwin arises from *phenotypic* variations among individuals. Phenotypic variations are defined as physical and behavioral traits of an individual affecting its response to the environment and other individuals. This second primary force is seen as an individual's *fitness*. Each individual represents a unique mixture of these phenotypic traits that is evaluated by the environment. Or in other words, if an individual's fitness is evaluated positively, the individual has a greater chance to reproduce. If individuals with high fitness reproduce, favorable traits can be inherited by their offspring. In addition, Darwin's understanding was that *mutations*, or small, random changes, in phenotypic traits occur during reproduction. These mutations cause new mixtures of phenotypic traits, or fitnesses, to exit and to be evaluated. Summarizing, a population consists of a group of individuals all competing for the same resources in some given environment. The better these individuals adapt to the environment, the more chance they have of producing offspring. As the population evolves, and the more successful individuals reproduce, new phenotypic traits arise from cross-over and mutation, and their performance is to be evaluated. Hence, as time passes, the structure of the population changes. The population is considered to be the unit of evolution (Eiben et al., 2003).

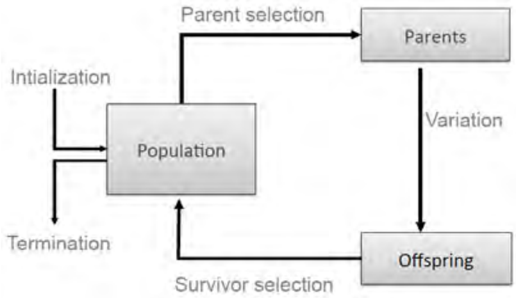
Translating this analogy to a more visual representation, observe the *adaptive landscape* (Wright, 1932) shown in Figure 4.6a. The z-dimension, or height, of this landscape portrays the fitness of individuals of the population. The x and y-dimensions correspond to the *genes* of individuals, determining their phenotypic traits. In this three-dimensional example the xy-plane depicts all possible gene combinations. Hence, an individual is defined as a point on the landscape, a population as a group of points located on the landscape, and each peak of the landscape represents a combination of successful genes. Evolution is then perceived as the process of gradual movement of the population towards high-fitness areas. This process is empowered by *variation* (cross-over and mutation) and selection. Of course, it is not guaranteed that the population moves towards a global optimum. The finite size of the population and the stochastic nature of variation and selection procedures can trigger a phenomenon called *genetic drift*, through which highly fit individuals might be lost from the population. In addition, the population may become less diverse from a loss of some traits. A result of genetic drift might be that populations get stuck in low-fitness valleys. Escaping such regions is possible. According to Wright's *shifting balance theorem* the maximum of a fixed landscape can be reached (1932).

Using Darwin's theory of evolution for automated problem solving dates back to the late 40s of the previous century. In one of his articles, Turing proposed the notion of "genetical or evolutionary search by which a combination of genes is looked for, the criterion being the survival value (Turing, 1948)." In the early sixties, Bremermann was the first to complete computer experiments based on Darwin's principles (1962). Following, from the sixties through the nineties of the previous century, *evolutionary programming*, *evolution strategies*, *genetic algorithms* and *genetic programming* were introduced (Fogel et al., 1966; Schwefel et al., 1995; Holland, 1973; Koza, 1992). All these families of algorithms inspired by Darwin's legacy now belong to one family of models called *evolutionary algorithms* (Back, 1996).

The general scheme of an evolutionary algorithm is closely related to the theory described above. In evolutionary algorithms a candidate solution to the problem one is solving is con-



(a) Illustration of an adaptive landscape as perceived by Wright (1932).

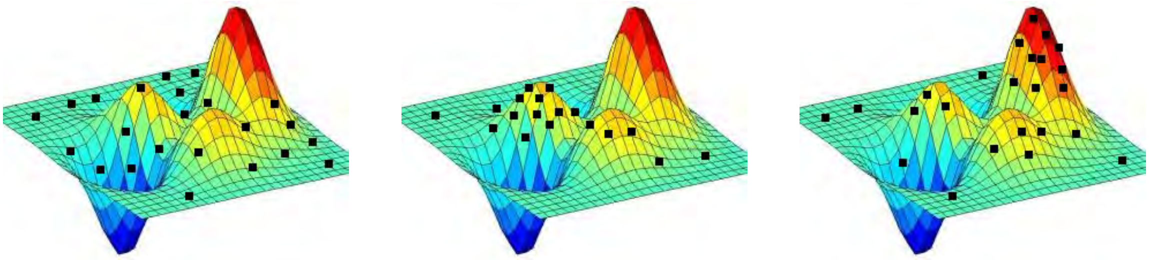


(b) Schematic overview of an evolutionary algorithm, taken from Eiben et al. (2003).

Figure 4.6

sidered to be an individual. In the case of finding optimal network hyper-parameters, an individual can for instance be represented as a two-dimensional vector containing the following genes: `number of layers` and `number of epochs`, each representing a component defining the structure of the network. The goal of the evolutionary algorithm is then to find genes corresponding to optimal fitness. Fitness is a measure to be defined by the analyst and could for instance equal the loss value achieved by the network, or the accuracy of performance of the network on test data. A general scheme of an evolutionary algorithm is given in Figure 4.6b. Initialization of the algorithm corresponds to creating a population of individuals of size n. Returning to the previous example, n two-dimensional vectors are created by randomly initializing each of the dimensions. Each dimension is related to a set of values. Randomly initializing each dimension corresponds to drawing one of these values at random per dimension. This set of values can be either finite or infinite, depending on the corresponding hyper-parameter. Of course `number of layers` $= -2$ should not be possible. The set of possible values regarding this gene should therefore be bounded. Post initialization, a fitness value is obtained for each member of the population. Next, based on these fitness values, a group of individuals is selected to produce offspring. Each gene of each offspring could potentially be mutated. Mutation of genes is associated with some mutation probability. Again, returning to the aforementioned example, the value for `number of epochs` could be subjected to mutation: `number of epochs` $= 15 \rightarrow$ `number of epochs` $= 14$. Mutation of a gene typically corresponds to slight alteration of the value associated with that gene. Post variation procedures (cross-over and mutation), survivors are selected. All offspring are added to the already existing population. Of this population, a group of n individuals is selected to represent the next generation. These procedures proceed until a stopping criterion is met. The stopping criterion could for instance correspond to the mean fitness of the population stabilizing, the best fitness value being equal for several generations, or a fixed number of iterations being completed.

Random initialization of individuals causes the population to be spread out over the entire fitness landscape, as shown in Figure 4.7a. Of course, for most optimization problems, one does not know the shape of the landscape to be examined. Therefore, random initialization and a large enough population size are crucial. Both these aspects ensure exploration of the landscape. As time passes, the goal of the algorithm is to push the population towards a global, or satisfactory local, optimum of the fitness landscape. The selection and variation procedures assist in this task and are considered the two main forces of evolutionary algorithms. Selection acts as a force increasing the mean *quality* of solutions in the population, whilst recombination and mutation create the necessary diversity within the population, and thereby facilitate *novelty* (Eiben et al., 2003). The balance between quality and novelty is extremely important. An algorithm focused too much on quality can cause the population to get stuck in a non-optimal region, as shown in Figure 4.7b. Selecting individuals only based on fitness values can cause fast convergence towards undesirable regions, whilst not having explored the entire search space.



(a) Random initialization of a population of individuals.

(b) Algorithm focused on quality. The population is stuck on a local optimum.

(c) Accurate balance of quality and novelty.

Figure 4.7: Populations spread out over the fitness landscape.

To avoid populations to get stuck in non-optimal regions, variation operators are indispensable. It could occur that, at initialization, many individuals are located near a local optimum. If only selection takes place, the population is deemed to converge towards this local optimum. By introducing cross-over and mutation procedures new individuals receive the possibility of exploring the search space. If one or more of these individuals discover a new promising region, the population can be guided away from the local optimum and guided towards the true optimum. An accurate balance between quality and novelty is portrayed in Figure 4.7c. The novelty operators ensure exploration of the search space. The quality operators push the population towards the desired region.

4.7 Application to Vehicle Leasing Data

This section addresses how the methods discussed in this chapter can be applied to the vehicle leasing data. The section is mainly concerned with determining the structure of the network. The structure of the network and all the hyper-parameters involved are solely determined by the constructed evolutionary algorithm. Hence, explanation of this algorithm will suffice to determine the network structure. For model comparison reasons, the set of variables used as input to the network is the same as for the models discussed in chapter 3. The ways of processing these variables remain equal too. A general outline of the evolutionary optimization algorithm is given in Algorithm 2. Each line of the algorithm denotes a function assisting in finding the optimal network hyper-parameters. The algorithm will now be addressed in detail.

Algorithm 2: Evolutionary Optimization Algorithm

```
   input : max_iteration
1  initialize population
2  evaluate population
3  while iteration ≠ max_iteration do
4      select parents
5      create offspring
6      mutate offspring
7      evaluate offspring
8      select next generation
9      update best individual
10 return best individual
```

The start of the algorithm consists of randomly initializing a population of individuals, denoted by the function `initialize population` on line 1. An individual can be thought of as a dictionary mapping *keys* to *values*. Each key is associated with one or more values. Keys correspond to the hyper-parameters to be tuned, values are the values associated with these parameters. All keys composing an individual are shown in Table 4.1. In addition, the range of possible values for each key is specified. Most keys and values are self-explanatory. Nonetheless, all are elaborated on. Upon initialization of an individual, the first key to be determined is `hidden layers`. Many of the other keys are dependent on this key, and therefore the value associated with this key should be determined first. To determine the value associated with the key, an integer is randomly drawn from the specified range. To clarify, if the value 2 is drawn, the network is defined to consist of $2 + 2 = 4$ layers. By definition, a network consists of an input and output layer, hence the two additional layers. The keys `neurons per layer`, `activation functions` and `dropout ratio` are all dependent on the value associated with `hidden layers`. Namely, for each hidden layer it need be determined the number of neurons associated with the layer, the activation function related to the neurons of each layer, and if dropout needs to be applied prior to the hidden layer. The value associated with `hidden layers` determines the number of values drawn for the aforementioned keys. Again, if `hidden layers` = 2, two integers are randomly drawn from the specified range to determine `neurons per layer` for *each* layer. The same principle applies to `activation functions`, for which two functions are drawn at random from the specified possible values. Lastly, two real numbers are drawn at random to be associated with `dropout ratio`. In addition, two Booleans are randomly drawn to determine the presence of dropout-layers. If for instance the values *False* and *True* are drawn, the first drawn value associated with `dropout ratio` is omitted. The second value is used to determine the ratio of dropout applied prior to the second hidden layer.

Hyper-Parameter	**Range**
`hidden layers`	$[1, ..., 4] \in \mathbb{Z}$
`neurons per layer`	$[50, ..., 500] \in \mathbb{Z}$
`activation functions`	Sigmoid, tanh, ReLU and ELU
`epochs`	$[5, ..., 50] \in \mathbb{Z}$
`batch size`	$[100, ..., 10000] \in \mathbb{Z}$
`optimizers`	Adam, RMSProp and SGD
`learning rate`	$(0, ..., 0.2] \in \mathbb{R}$
`dropout ratio`	$(0, ..., 0.3] \in \mathbb{R}$
`embedding size`	$[1, 2, 3] \in \mathbb{Z}$

Table 4.1: All hyper-parameters to be tuned by the evolutionary optimization algorithm. The right column specifies the ranges of possible values associated with each parameter.

The integer values related to the keys `epochs` and `batch size` are drawn in similar fashion as for the key `hidden layers`. These values correspond to the number of training epochs and the size of the mini-batches used for training respectively. The optimization function used to find the optimal weights of the network is determined by randomly drawing a function from the list associated with `optimizers`. Both RMSProp and Adam are optimization methodologies based on the gradient techniques discussed in section 4.4. These methods are slight alterations of Stochastic Gradient Descent (SGD). For a thorough explanation of both Adam and RMSProp, the reader is referred to Kingma and Ba (2014) and Tieleman and Hinton (2012) respectively. The value for `learning rate` is determined by drawing a random real number from the corresponding range of possible values. Lastly, to determine the *size* of each embedding layer, n values for `embedding size` are drawn from the specified range, with n corresponding to the number of categorical variables. For each categorical variable a value is drawn, e.g. $\texttt{embedding size}_{\texttt{make}} = 2$, and $\texttt{embedding size}_{\texttt{vehicle segment}} = 3$. Note that the specified range only exists of three integer values. These values were chosen since no significant improvement of performance is observed when increasing the upper bound of this key. Categorical variables only containing two states are mapped to an embedding of one dimension. The lower bound of `embedding size` is defined to equal two for categorical variables containing more than two states. These values were chosen since embedding layers of one, two or three dimensions are readily interpretable and easy to visualize.

Each individual of the first generation is initialized as described above. The number of individuals initialized corresponds to the value associated with the parameter `population size` specified in Table 4.2. This table shows all parameters used for the evolutionary algorithm. The choice of values for $\texttt{max}_{\texttt{iteration}}$ and `population size` is related to the provided computational power. Since training a network typically takes a significant amount of time, and one individual corresponds to training an entire network, the analyst is limited to the provided computational power. The choice of remaining parameter values originates from empirical observations. These values were chosen to obtain an ideal relation between *quality* and *novelty*, resulting in convergence towards a global optimum, as discussed in section 4.6.

Parameter	**Value**
$\texttt{max}_{\texttt{iteration}}$	10
`population size`	32
$\texttt{parent ratio}_{\texttt{random}}$	$\frac{4}{\texttt{population size}}$
$\texttt{parent ratio}_{\texttt{quality}}$	$\frac{3}{4}$ - $\texttt{parent ratio}_{\texttt{random}}$
`mutation probability`	0.2
`mutation rate`	0.2
$\texttt{survivor ratio}_{\texttt{random}}$	0.2
$\texttt{survivor ratio}_{\texttt{quality}}$	0.8

Table 4.2: All parameters used for the evolutionary algorithm (left) with their corresponding values (right).

Post initialization of the population, each individual is evaluated as portrayed on line 2 of Algorithm 2. Evaluation of an individual is as straightforward as measuring *accuracy* of prediction on test data. Recall from section 2.2 that 10% of the provided data are used for testing purposes. In addition, these data correspond to the most recent 10% of switches. Observe that the loss objective of the network differs from the fitness measured by the evolutionary algorithm. Therefore, it is possible that a network performs well on training data, whereas the network does not generalize well and does not achieve great accuracy on test data. Accuracy is chosen as fitness measure since achieving great performance on unseen data is seen as more valuable than performance on training data.

Next, the procedures described on lines 4 - 9 are repeated until the maximum number of iterations is reached. First, part of the population is selected to function as parents. To stimulate both quality and novelty, part of the parents are selected based on fitness values, whereas the remaining parents are selected at random. Parameter $\texttt{parent ratio}_{\texttt{quality}}$ determines the number of parents selected based on fitness measure. The best performing $\texttt{parent ratio}_{\texttt{quality}} \cdot \texttt{population size}$ are selected to be parents. Of the remaining population, $\texttt{parent ratio}_{\texttt{random}} \cdot \texttt{population size}$ individuals are randomly selected to create offspring.

Following, the procedure `create offspring` randomly matches individuals to create pairs of parents. Per pair of parents, a pair of children is created. Each child inherits genes from both its parents. For both children, `hidden layers` is determined first. The first child inherits `hidden layers` from its mother, whereas the second child inherits from its father. Next, per layer `neurons per layer` is determined. For the overlapping layers, each child is assigned a value per layer at random. One child inherits this value from its mother, the other child from its father. If both children (and therefore both parents) have the same value associated with `hidden layers`, this procedure stops here. If $\texttt{hidden layers}_{\text{child}_1} \neq \texttt{hidden layers}_{\text{child}_2}$, the child of which the value for `hidden layers` is greatest, inherits the remaining values of `neurons per layer`. Note that the remaining values must come from one of the parents only. Genes `activation functions` and `dropout ratio` are assigned in exact similar fashion as described above. The remaining genes of the children are determined as follows. One of the children inherits from its mother, whilst the other child inherits this gene from its father. To clarify the `create offspring` procedure, an example is illustrated in Table 4.3.

Hyper-Parameter	**Parent$_1$**	**Parent$_2$**	**Child$_1$**	**Child$_2$**
`hidden layers`	2	3	2	3
`neurons/layer`	[110, 70]	[354, 130, 300]	[354, 130]	[110, 70, 300]
`activations`	[ReLU, ReLU]	[ELU, ELU, tanh]	[ReLU, ELU]	[ELU, ReLU, tanh]
`epochs`	24	28	28	24
`batch size`	1500	1234	1500	1234
`optimizers`	SGD	SGD	SGD	SGD
`learning rate`	0.12	0.06	0.06	0.12
`dropout ratio`	0.04	0.23	0.04	0.23
$\texttt{embedding}_{\texttt{make}}$	2	3	2	3

Table 4.3: Example of the procedure `create offspring`. Note that `embedding size` is only shown for the categorical variable `make`. All other embedding sizes of categorical variables are determined in similar fashion and are therefore omitted.

Once `create offspring` terminates, each produced offspring serves as input to the function `mutate offspring`. This function takes the parameters `mutation probability` and `mutation rate` as input. The former denotes the probability of each gene being mutated, whilst the latter denotes the rate by which genes mutate. Only the genes `neurons per layer`, `epochs` `batch size`, `learning rate` and `dropout ratio` are affected by `mutation probability` and are mutated as follows. First, per gene, a random number $r \in \mathbb{R}$ between 0 and 1 is drawn. If $r <$ `mutation probability`, the gene is mutated. Mutation corresponds to randomly drawing a value in the range $[(1 - \texttt{mutation rate}) \cdot v, ..., (1 + \texttt{mutation rate}) \cdot v])$, in which v corresponds to the parameter value associated with the gene in question. This range consists of either real numbers or integers, depending on the gene to be mutated. To clarify, assume that the gene to be mutated is `batch size`, $r < \texttt{mutation probability}$, $v = 1234$ and $\texttt{mutation probability} = 0.1$. Then, the range of possible values for the mutated gene is $[0.9 \cdot 1234, ..., 1.1 \cdot 1234] = [1111, ..., 1357] \in \mathbb{Z}$. Mutation of the genes `activation functions` and `optimizers` happens as follows. If $r < \texttt{mutation probability}$, a new value for the corresponding gene is drawn from the range of possible values as specified in Table 4.1. Note that per hidden layer a value for r is drawn. Mutation of the remaining genes `hidden layers` and `embedding size` is a mixture of both aforementioned mutation methods. If $r <$ `mutation probability` a value of one is either added or subtracted from the value corresponding to the gene. If, post mutation, the value of the gene lies outside of the boundaries corresponding to the gene, the value is reset to either the lower or upper boundary as specified in Table 4.1. Again, clarifying with an example, suppose that the gene to be mutated is `hidden layers`, $r <$ `mutation probability`, $v = 4$ and the `mutate offspring` procedure determines mutation to correspond to adding one layer. Then, $v = 4 \rightarrow v = 5$. Since $v = 5 > \texttt{hidden layers}_{\texttt{upper boundary}}$, v is reset to equal 4.

Once mutation of all offspring has completed, each offspring is evaluated. As for `mutate population`, `mutate offspring` corresponds to measuring accuracy of prediction of a constructed network. Per offspring, a network is created based on individuals' genes. Post training of the network, a fitness value is obtained by measuring accuracy of prediction on test data. Following, all offspring are added to the population. The procedure `select next generation` selects `population size` survivors to form the next generation. Selection of survivors is determined by the parameters $\texttt{survivor ratio}_{\texttt{quality}}$ and $\texttt{survivor ratio}_{\texttt{random}}$. The best $\texttt{survivor ratio}_{\texttt{quality}} \cdot \texttt{population size}$ performing individuals are selected to be part of the next generation. In addition, of the remaining population $\texttt{survivor ratio}_{\texttt{quality}} \cdot \texttt{population size}$ individuals are selected at random to be added to the next generation. Once the next generation of individuals is determined, the individual achieving highest accuracy on the test data is selected and memorized by the algorithm. This completes one iteration. Once $\texttt{max}_{\texttt{iteration}}$ number of iterations are completed, the algorithm terminates and returns the best performing individual. This individual is considered the final network and used for modeling the car leasing data.

This final network contains several embedding layers, all having embedding sizes determined by the network. To test the strength of networks in which categorical variables are mapped to continuous space, a network in which these embedding layers are not present is created. The structure of the network is nearly identical to the network the evolutionary algorithm returned. In this network, categorical variables are transformed to their one-hot encoding. These encodings are used as input in identical fashion as the continuous variables are used for input. So, only one input layer exists in this network, and therefore no parallel input layer exists. Of course, the number of neurons of the input layer increases. The number of neurons in the remaining layers, and all other hyper-parameters, are identical to those in the embedding-network.

Chapter 5

Results

This chapter describes the results obtained by the models discussed in chapters 3 and 4. Recall that all models use the exact same data as input to analyze and predict. The chapter is structured as follows. Section 5.1 describes the results obtained by all variants of discrete choice models discussed in chapter 3, whilst section 5.2 discusses all results obtained by neural networks addressed in chapter 4. In addition, this section describes the results obtained by the evolutionary algorithm, meaning the final network structure.

5.1 Discrete Choice Models

Observe from Table 5.1 the results regarding all discrete choice models. The leftmost column states performance measures, whereas the remaining columns indicate the values associated with these measures for the standard, nested, and cross-nested multinomial logistic regression models respectively. The statistic $\mathcal{L}(0)$ corresponds to the null log likelihood. The null log likelihood is the log likelihood of the sample for a logistic regression model such that the deterministic part of the utility function is zero for all alternatives, that is

$$\mathcal{L}(0) = \sum_{n \in \text{sample}} w_n \ln \frac{1}{\#C_n},$$

where $\#C_n$ is the number of alternatives available to individual n and w_n is the associated weight (Bierlaire, 2015). In addition, $\mathcal{L}(c)$ is defined as the log likelihood of the sample where the the deterministic part of utility of each alternative contains only the alternative specific constants. Since all alternatives are always available, this corresponds to

$$\mathcal{L}(c) = \sum_{j \in C} n_j \ln n_j - n \ln n,$$

in which n_j is the number of times alternative j has been chosen, and $n = \sum_{j \in C} n_j$ is the number of observations in the sample (Bierlaire, 2015). The statistic $\mathcal{L}(\hat{\beta})$ denotes the final log likelihood of the estimated model, and $-2[\mathcal{L}(0) - \mathcal{L}(\hat{\beta})]$ denotes the likelihood ratio test. The likelihood ratio test compares the goodness of fit of two models, the null and final model. Lastly, ρ^2 and $\hat{\rho}^2$ are defined as the likelihood ratio index and the adjusted likelihood ratio index respectively. The former is defined as

$$\rho^2 = 1 - \frac{\mathcal{L}(\hat{\beta})}{\mathcal{L}(0)},$$

whilst the latter is a slight adjustment of this definition by taking into account the number of estimated parameters K:

$$\rho^2 = 1 - \frac{\mathcal{L}(\hat{\beta}) - K}{\mathcal{L}(0)}.$$

Summary Statistics	**MNL**	**nested MNL**	**cross-nested MNL**
$\mathcal{L}(0)$	$-159,170.497$		
$\mathcal{L}(c)$	$-152,117.028$		
$\mathcal{L}(\hat{\beta})$	$-106,158.670$	$-106,121.880$	$\mathbf{-105,995.517}$
$-2[\mathcal{L}(0) - \mathcal{L}(\hat{\beta})]$	$106,023.655$	$106,097.235$	$\mathbf{106,349.960}$
ρ^2	0.333	0.333	**0.334**
$\bar{\rho}^2$	0.329	0.329	**0.330**

Table 5.1: Performance of all Discrete Choice Models on training data. Each model was estimated using 62,056 observations. Values equal for all three models are only stated in the second column and are left blank in the subsequent columns.

Observe from Table 5.1 that the cross-nested model achieves best results on all performance measures. Of course, it is to be expected that both the nested and cross-nested models outperform the standard multinomial logistic regression model, since estimates of this model are taken as starting values of the nested and cross-nested models. The likelihood ratio test indicates that all three models are significant improvements relative to the null-model. Comparing this statistic for all three models, the more restrictive assumptions are relaxed, the better fit the model is on training data. Only for the cross-nested model the improved final log likelihood with respect to the other two models, results in a better (adjusted) likelihood ratio index. Observe that the improvement of final log likelihood value of the nested model is not significant enough to obtain a better likelihood ratio index.

Since performance of all three models is roughly similar, it need be checked if dropping the IIA assumption is relevant. To do so, note the estimates of the nest parameters stated in Table 5.2. Recall from section 3.3 that the nested model collapses to a standard multinomial logistic regression model if all nest parameters are equal to one, $\lambda_k = 1 \; \forall k$. Note that both nest `D` and `SUV` are significant irrelevant of the maintained significance level. Nest `C` is deemed appropriate depending on the maintained significance level. Albeit slight, all estimated nest parameters are greater than one, hence validating relaxing of IIA.

Nest	Estimate	Std. Error	t-stat	p-value
C	1.24	0.124	1.92	0.05
D	1.66	0.126	5.18	0.00
LCV	1.00	FIXED		
SUV	1.73	0.249	2.93	0.00

Table 5.2: Relevance of nest parameters of the nested multinomial logistic regression model.

Further relaxation of assumptions leads to the cross-nested model, of which the nest and allocation parameters are displayed in Table 5.3 and 5.4 respectively. Note that some allocation parameters are fixed. These parameters were estimated with infinite standard error at first. The cross-nested model was estimated again fixing these allocation parameters at the estimated value produced on the first run. Interestingly, all nest parameters are significant and all estimates of these parameters differ significantly from one, hence relaxing assumptions is again validated. Observe that nest MPV has the highest estimated parameter value. The higher the nest parameter estimate, the more correlated alternatives of this nest are within the nest, rather than outside the nest.

Nest	Estimate	Std. Error	t-stat	p-value
B	1.75	0.319	5.490	0.00
C	1.54	0.034	45.28	0.00
D	2.08	0.066	31.46	0.00
E	1.24	0.148	8.420	0.00
LCV	1.10	FIXED		
MPV	2.89	0.160	18.02	0.00
SUV	1.94	0.177	10.98	0.00

Table 5.3: Relevance of nest parameters of the cross-nested multinomial logistic regression model. Note that no statistics are displayed for nest LCV. This parameter was estimated with an infinite standard error at first. On the second run this parameter was therefore held fixed at the estimated value of the first run.

Even though all nest parameters of the cross-nested model are deemed significant and relevant, Table 5.4 states some allocation parameters indicating that inclusion of the corresponding alternative in the concerned nest is not strongly supported by the given data. In other words, some allocation parameters are deemed insignificant. The third column of this table states the starting value of the respective allocation parameters. Recall that these estimates are solely based on the nesting structure provided by the vehicle leasing company. Note that most estimated values do not differ much from the provided starting value of the parameter. Interestingly enough, estimates of allocation parameters that differ much from the corresponding starting value are typically significant. Lastly, note that for each of the alternatives at least one of the allocation parameters is significant, indicating inclusion in one of the nests is indicated by the data.

Allocation Parameter	**Estimate**	**Start. Value**	**Std. Error**	***t*-stat**	***p*-value**
$\alpha_{Audi-\texttt{C}}$	0.367	0.23	0.222	1.65	0.10
$\alpha_{Audi-\texttt{D}}$	0.378	0.41	0.243	1.56	0.12
$\alpha_{Audi-\texttt{E}}$	0.111	0.17	FIXED		
$\alpha_{Audi-\texttt{SUV}}$	0.145	0.19	FIXED		
$\alpha_{BMW-\texttt{C}}$	0.103	0.20	0.105	0.98	0.33
$\alpha_{BMW-\texttt{D}}$	0.665	0.34	0.0546	12.18	0.00
$\alpha_{BMW-\texttt{E}}$	0.0908	0.18	0.0950	0.96	0.34
$\alpha_{BMW-\texttt{SUV}}$	0.141	0.28	0.0421	3.35	0.00
$\alpha_{Citroen-\texttt{C}}$	0.147	0.29	0.109	1.35	0.18
$\alpha_{Citroen-\texttt{LCV}}$	0.212	0.42	FIXED		
$\alpha_{Citroen-\texttt{MPV}}$	0.641	0.29	0.165	3.88	0.00
$\alpha_{Ford-\texttt{B}}$	0.0959	0.19	0.0714	1.34	0.18
$\alpha_{Ford-\texttt{C}}$	0.153	0.28	0.0996	1.54	0.12
$\alpha_{Ford-\texttt{D}}$	0.485	0.16	0.122	3.97	0.00
$\alpha_{Ford-\texttt{LCV}}$	0.190	0.22	0.0608	3.13	0.00
$\alpha_{Ford-\texttt{MPV}}$	0.0757	0.15	0.0186	4.08	0.00
$\alpha_{Mercedes\text{-}Benz-\texttt{D}}$	0.685	0.64	0.0514	13.33	0.00
$\alpha_{Mercedes\text{-}Benz-\texttt{E}}$	0.102	0.17	0.0680	1.50	0.13
$\alpha_{Mercedes\text{-}Benz-\texttt{SUV}}$	0.213	0.19	0.0425	5.01	0.00
$\alpha_{Nissan-\texttt{SUV}}$	1.0	1.0	FIXED		
$\alpha_{Opel-\texttt{C}}$	0.365	0.72	FIXED		
$\alpha_{Opel-\texttt{D}}$	0.635	0.28	FIXED		
$\alpha_{Other-\texttt{B}}$	0.231	0.41	0.104	2.23	0.03
$\alpha_{Other-\texttt{C}}$	0.111	0.22	0.0384	2.89	0.00
$\alpha_{Other-\texttt{SUV}}$	0.658	0.37	FIXED		
$\alpha_{Peugeot-\texttt{C}}$	0.379	0.49	FIXED		
$\alpha_{Peugeot-\texttt{D}}$	0.351	0.12	0.0741	4.74	0.00
$\alpha_{Peugeot-\texttt{LCV}}$	0.101	0.20	FIXED		
$\alpha_{Peugeot-\texttt{SUV}}$	0.168	0.19	0.0299	5.62	0.00
$\alpha_{Renault-\texttt{B}}$	0.336	0.41	0.113	2.98	0.00
$\alpha_{Renault-\texttt{C}}$	0.184	0.16	0.0966	1.91	0.06
$\alpha_{Renault-\texttt{LCV}}$	0.378	0.31	0.196	1.92	0.05
$\alpha_{Renault-\texttt{MPV}}$	0.102	0.12	0.0230	4.43	0.00
$\alpha_{Skoda-\texttt{C}}$	0.338	0.67	0.0784	4.32	0.00
$\alpha_{Skoda-\texttt{D}}$	0.662	0.33	0.0784	8.44	0.00
$\alpha_{Volkswagen-\texttt{C}}$	0.225	0.38	0.148	1.52	0.13
$\alpha_{Volkswagen-\texttt{D}}$	0.211	0.34	0.0680	3.11	0.00
$\alpha_{Volkswagen-\texttt{LCV}}$	0.206	0.13	0.0604	3.41	0.00
$\alpha_{Volkswagen-\texttt{MPV}}$	0.357	0.15	FIXED		
$\alpha_{Volvo-\texttt{C}}$	0.0706	0.14	FIXED		

$\alpha_{Volvo-\mathtt{D}}$	0.131	0.24	0.0773	1.70	0.09
$\alpha_{Volvo-\mathtt{E}}$	0.0858	0.17	0.0878	0.98	0.33
$\alpha_{Volvo-\mathtt{SUV}}$	0.712	0.45	0.0932	7.64	0.00

Table 5.4: Statistics on the allocation parameters α. In $\alpha_{i\mathtt{k}}$, i denotes the alternative and $\mathtt{k}$ the corresponding nest. All values for which statistics are not displayed were fixed at run-time. These parameters were estimated with infinite standard error at first. On the second run they were fixed at the produced output value of the first run.

Since this research aims to construct *predictive* models rather than *descriptive* models, performance on test data is important. Table 5.5 states the achieved accuracy of each model on test data. Recall that the test data consist of the most recent 10% of switches. In addition to the achieved accuracy of the aforementioned models, accuracy of a benchmark model is stated as well. This benchmark model assumes loyalty of drivers. It assumes that a driver chooses the same make as he or she was driving prior to his or her current vehicle, hence this model solely copies the variable `make` to the choice variable `new make`. Observe that, in the test data, almost 45% of drivers remain loyal to their previously driven make. Interestingly enough, the nested model generalizes worst on unseen data, whilst best performance of the cross-nested model on training data directly relates to best performance on unseen data. Note that an accuracy value of 1% translates to roughly 69 correctly predicted switches. Additionally, Table B.4 located in Appendix B portrays the breakdown of achieved accuracy score per model over all alternatives of the choice set. Observe that all logit models achieve highest accuracy when predicting alternatives occurring relatively often in the test data. The logit models achieve relatively low accuracy when predicting fewer occurring alternatives.

Model	**Accuracy**
MNL	48.26%
nested MNL	47.96%
cross-nested MNL	**48.39%**
benchmark	44.90%

Table 5.5: Performance of all Discrete Choice Models on test data. Each model predicts the most likely next make of car given previous contract attributes. The test data contain 6,896 matched contracts.

Next, the relevance and influence of the explanatory variables is discussed. All estimated parameters included in the standard, nested and cross-nested models are depicted in Tables B.1, B.2 and B.3 respectively, all located in Appendix B. Note that insignificant parameters are also displayed. These parameters serve two purposes. Firstly, insignificance of parameters could serve an explanatory purpose regarding inclusion in models. Secondly, exclusion of these parameters cause both the final log-likelihood of models and the achieved accuracy on test data to worsen. Therefore, all insignificant estimates were maintained and used for prediction. Since nearly 45% of drivers stays loyal to their make, it is expected that parameters regarding `make` are of great importance. Table 5.6 portrays the estimates for these parameters. Per state of `make` only the two parameters with the highest estimated value are shown. Note that for nearly all these βs, the one indicating make loyalty has the highest value. Only the parameters

Parameter	**Estimate**	**Std. Error**	***t*-stat**	***p*-value**
$\beta_{\texttt{make Audi } \textit{Peugeot}}$	-0.854	0.248	-3.44	0.00
$\beta_{\texttt{make BMW } \textit{Opel}}$	1.02	0.226	4.49	0.00
$\beta_{\texttt{make BMW } \textit{BMW}}$	0.917	0.104	8.81	0.00
$\beta_{\texttt{make Citroen } \textit{Citroen}}$	3.32	0.434	7.65	0.00
$\beta_{\texttt{make Citroen } \textit{Peugeot}}$	2.30	0.261	8.80	0.00
$\beta_{\texttt{make Ford } \textit{Ford}}$	2.70	0.165	16.39	0.00
$\beta_{\texttt{make Ford } \textit{Citroen}}$	2.12	0.430	4.93	0.00
$\beta_{\texttt{make Mercedes-Benz } \textit{Mercedes-Benz}}$	1.54	0.122	12.58	0.00
$\beta_{\texttt{make Mercedes-Benz } \textit{Opel}}$	0.560	0.252	2.22	0.03
$\beta_{\texttt{make Nissan } \textit{Nissan}}$	3.35	0.363	9.23	0.00
$\beta_{\texttt{make Nissan } \textit{Opel}}$	1.50	0.328	4.56	0.00
$\beta_{\texttt{make Opel } \textit{Opel}}$	2.88	0.225	12.79	0.00
$\beta_{\texttt{make Opel } \textit{Peugeot}}$	1.52	0.249	6.12	0.00
$\beta_{\texttt{make Other } \textit{Other}}$	1.85	0.134	13.82	0.00
$\beta_{\texttt{make Other } \textit{Citroen}}$	1.51	0.430	3.52	0.00
$\beta_{\texttt{make Peugeot } \textit{Peugeot}}$	2.53	0.247	10.26	0.00
$\beta_{\texttt{make Peugeot } \textit{Citroen}}$	1.76	0.429	4.11	0.00
$\beta_{\texttt{make Renault } \textit{Citroen}}$	2.54	0.428	5.92	0.00
$\beta_{\texttt{make Renault } \textit{Renault}}$	2.07	0.186	11.11	0.00
$\beta_{\texttt{make Skoda } \textit{Skoda}}$	2.63	0.190	13.83	0.00
$\beta_{\texttt{make Skoda } \textit{Nissan}}$	1.57	0.385	4.08	0.00
$\beta_{\texttt{make Volkswagen } \textit{Volkswagen}}$	1.41	0.107	13.16	0.00
$\beta_{\texttt{make Volkswagen } \textit{Peugeot}}$	0.573	0.240	2.39	0.02

Table 5.6: Importance of previously driven make of car for the multinomial logistic regression model. The two βs with the highest value are shown per previous make. The parameter $\beta_{\texttt{make Audi } \textit{Audi}}$ is omitted, since this parameter is fixed at zero. Note that insignificant βs are not taken into account. All alternatives are stated in italics. For a complete overview of parameters the reader is referred to Table B.1.

$\beta_{\texttt{make BMW } \textit{BMW}}$ and $\beta_{\texttt{make Renault } \textit{Renault}}$ are not the estimate with the highest value for that particular state of the variable. In addition, note that all these estimates are significant. This table portrays the estimates of the multinomial logistic regression model. Similar phenomena are observed for both the nested and cross-nested models.

Regarding the remaining explanatory variables, observe that estimates of the parameters included in all three models portray similar characteristics. Most parameters considered important in one model, prove important for the other two models as well. Recall that the leasing company segments drivers by `country` and `client segment`. The estimates regarding these parameters indicate that segmentation by `country` is often more relevant than by `client segment`. Noteworthy, the estimates regarding the state `Norway` of parameter `country` are all significant for prediction. This is the only branch of the company for which true data is provided. The categorical variable `fuel type` seems to play an important predictive role, even though that the distribution of the states of this parameter is dis-balanced. In addition, the parameter `vehicle segment` proves its relevance depending on the state of the variable. It seems that the states chosen to serve as nests prove slightly more predictive power than the excluded states. Observe that nearly all states of the variable `body style` do not add much explanatory power to the model. Only the states `Car Van` and `Delivery Van` add some strength to the model. Regarding the numerical variables included, the variable `catalogue price` proves its explanatory power for all alternatives except *Volvo*. In addition, the parameters `mileage month` and `standard discount percentage` influence most drivers when choosing a new make of vehicle.

5.2 Feedforward Neural Networks

Prior to discussing performance of the final feedforward neural network, the structure of this network is addressed. Recall that this structure is solely determined by the evolutionary algorithm. The final network structure is given in Table 5.7. The final network consists of one input, two hidden and one output layer. The hidden layers contain 347 and 154 neurons respectively. Both hidden layers use the Exponential Linear Unit (ELU) activation function to transform their inputs. For both layers' activation function the parameter α is fixed at one by default. In addition to the final network choosing ELU, most successful networks neglected the more traditional sigmoid and tanh activation functions, and used either ReLU or ELU to transform their inputs. Note the surprisingly few number of epochs used for training the network. In addition, the network is trained on mini-batches of size 2105 using the optimization function RMSProp. To avoid possible overfitting, the dropout technique can potentially be added to the network. The final network only uses this technique prior to the second layer with a relatively high ratio of 0.279. Lastly, note the determined embedding sizes of each categorical variable.

Figure 5.1 portrays some relevant statistics of the evolutionary algorithm over time. The maximum, minimum, mean and standard deviation of fitness are shown per iteration. Recall that the fitness of an individual is measured by obtaining the accuracy of prediction on test data. Observe that the maximum fitness value barely increased. One individual of the first generation achieved an accuracy of 49.97% on test data. Only two individuals performed better than this individual of the first generation, one of which being an individual part of the third generation achieving an accuracy of 50.20% on test data. The highest accuracy value was obtained in the fourth generation. Observe that the following generation tried to escape this potential local optimum. New individuals tried to explore new areas of

Hyper-Parameter	Value(s)	Hyper-Parameter	Value(s)
`hidden layers`	2	`embedding size`$_{\texttt{body style}}$	2
`neurons per layer`	347, 154	`embedding size`$_{\texttt{client segment}}$	2
`activation functions`	ELU, ELU	`embedding size`$_{\texttt{country}}$	3
`epochs`	12	`embedding size`$_{\texttt{fuel type}}$	1
`batch size`	2105	`embedding size`$_{\texttt{lease type}}$	1
`optimizers`	RMSProp	`embedding size`$_{\texttt{make}}$	3
`learning rate`	0.002178	`embedding size`$_{\texttt{switch quarter}}$	2
`dropout ratio`	`NA`, 0.279	`embedding size`$_{\texttt{vehicle segment}}$	2
		`embedding size`$_{\texttt{vehicle type}}$	1

Table 5.7: Hyper-parameters of best performing neural network. All hyper-parameters are determined by the evolutionary optimization algorithm.

the fitness-landscape. Doing so increased diversity among the population. Despite these efforts, no individual found an area of the landscape resulting in higher accuracy. Note that towards the last generation of individuals, the population converged in the direction of the previously obtained highest fitness value. In the last generation of individuals, almost no variation among individuals existed. The population converged either to a local or global optimum.

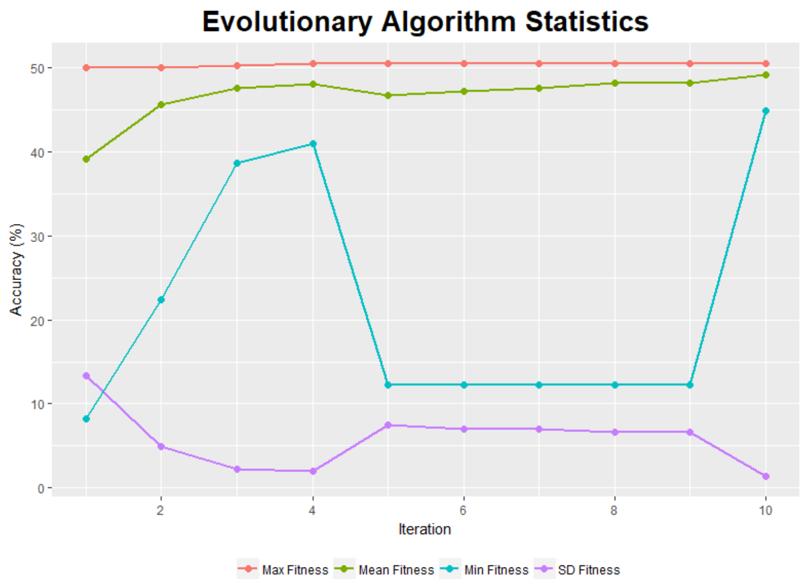


Figure 5.1: Evolutionary Algorithm Statistics over time.

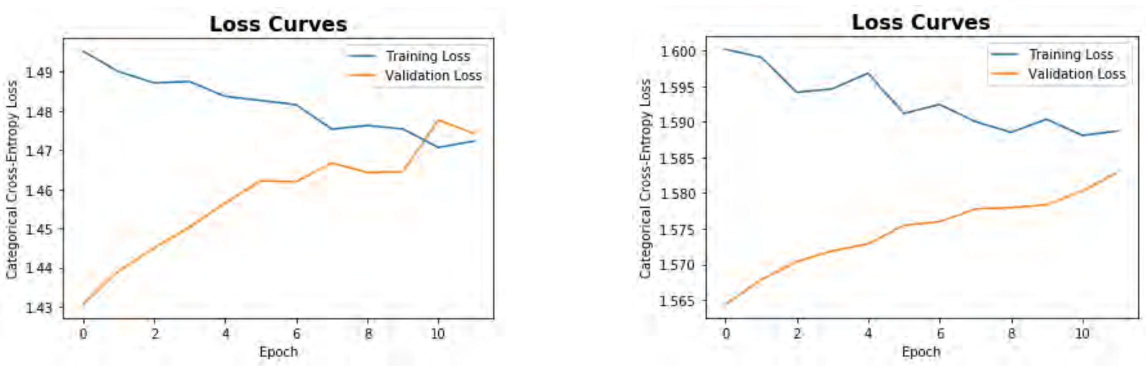
Post termination of the evolutionary algorithm, the final network structure is determined and a *one-hot network* is constructed based on the hyper-parameters of the final network. As discussed in section 4.7, this one-hot network is identical to the embedding network except for the way of handling categorical variables. Table 5.8 states the achieved accuracy of the final network and the benchmark model addressed in the previous section. In addition, the last column states the time both networks needed to learn the optimal weights. Due to the stochastic nature of neural networks and the fact that many different structures for the embedding network were tried, the one-hot network was randomly initialized ten different times. The accuracy stated in Table 5.8 refers to the highest achieved accuracy of one of the networks on test data. The training time stated in this table is the average of all ten networks' training time. Observe that the embedding network outperforms all ten instances of the one-hot network. Additionally, observe that training the embedding network takes nearly 2.5 times less than a one-hot network. As stated in the previous section, Table B.4 denotes a breakdown of achieved accuracy per model over all alternatives of the choice set. Both networks achieve similar results. However, the embedding network manages to significantly perform better when predicting the alternative `Other`. Additionally, note that both types of networks are better able to predict minority alternatives with respect to the logit models.

Model	**Accuracy**	**Training Time**
Embedding Neural Net	50.51%	136.60
One-Hot Neural Net	49.68%	333.49
benchmark	44.90%	

Table 5.8: Performance and training time (in seconds) of both types of neural networks. In addition, performance of the benchmark model is stated. Each model predicts the most likely next make of car given previous contract attributes. The test data contain 6,896 matched contracts.

The achieved loss values of the final embedding network and the best achieving one-hot network are displayed in Figures 5.2a and 5.2b respectively. Ideally, these curves are smooth and L-shaped. A troublesome observation is that the validation loss worsens over time. Recall from section 4.4 that typically, to avoid overfitting, training of a network is terminated once performance of the network on validation data worsens. Since dropout techniques are imposed to avoid overfitting, worsening of performance on validation data is not considered as stopping criterion. Nonetheless, these curves indicate that the found relationships by the network are the result of sampling noise.

Lastly, the learned embeddings are discussed. Figures 5.3a and 5.3b portray the learned embeddings in 3-dimensional space for the categorical variables `make` and `country` respectively. In addition, for each state of the categorical variables `make` and `country`, Tables 5.9 and 5.10 present the *nearest* state to and *furthest* state from the concerned state. The nearest state is defined as the state of the categorical variable that is most similar based on cosine similarity, whilst the furthest state is defined as the least similar state based on the same measure. Cosine similarity measures the cosine of the angle between two vectors. This measure is calculated as stated below,

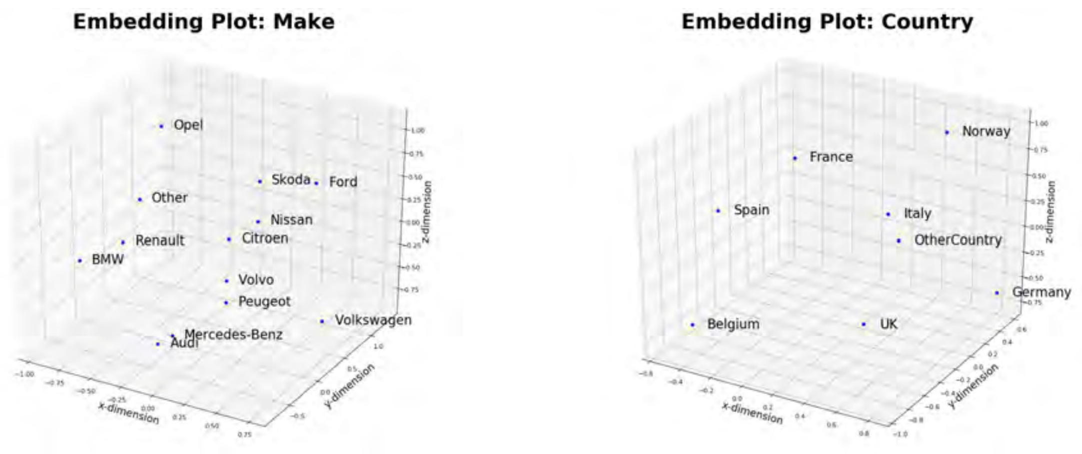


(a) Embedding-network (b) One-hot-network

Figure 5.2: Loss values of training and validation data over time for both types of networks.

$$\cos\,\theta = \frac{x \cdot y}{||x|| * ||y||},$$

in which x and y denote vectors, $||\cdot||$ denotes the norm of $\cdot$ and θ defines the angle between x and y. Cosine similarity outputs values in the range $[-1, 1]$, with a value of 1 depicting equal vectors and a similarity score of -1 meaning vectors pointing in opposite directions. Hence, if states of a categorical have similar effect on the loss function of the network it can be expected that these states contain a high cosine similarity score.



(a) `make` (b) `country`

Figure 5.3: The learned embeddings by the network, mapped to 3-dimensional space for the categorical variables `make` (left) and `country` (right).

Observe that the most similar states of `make` are `Nissan` and `Skoda`, having almost identical contribution to the loss function. The state `Volkswagen` can be considered a separate entity containing the most unique characteristics. Looking at the variable `country`, observe that `France` shares almost no characteristics with other states whatsoever. In general, the states of `country` share less unobserved characteristics than `make`. Interestingly enough, the state `Italy` shares the most characteristics with `Other`. These are the two states containing the least number of observations.

`make`	**Nearest**	**cos θ**	**Furthest**	**cos θ**
`Audi`	`Peugeot`	0.749	`Nissan`	-0.878
`BMW`	`Renault`	0.631	`Volkswagen`	-0.759
`Citroen`	`Renault`	0.658	`Volvo`	-0.954
`Ford`	`Citroen`	0.639	`Mercedes-Benz`	-0.914
`Mercedes-Benz`	`Audi`	0.707	`Ford`	-0.914
`Nissan`	`Skoda`	0.953	`France`	-0.878
`Opel`	`Other`	0.498	`Peugeot`	-0.824
`Other`	`Skoda`	0.759	`Peugeot`	-0.886
`Peugeot`	`Audi`	0.749	`Other`	-0.886
`Renault`	`Citroen`	0.658	`Volvo`	-0.854
`Skoda`	`Nissan`	0.953	`Audi`	-0.816
`Volkswagen`	`Volvo`	0.485	`Renault`	-0.787
`Volvo`	`Skoda`	0.668	`Citroen`	-0.954

Table 5.9: Nearest and furthest neighbors based on learned embedding weights for variable `make`.

`country`	**Nearest**	**cos θ**	**Furthest**	**cos θ**
`Belgium`	`UK`	0.400	`Norway`	-0.939
`France`	`Norway`	0.097	`UK`	-0.958
`Germany`	`UK`	0.420	`Spain`	-0.798
`Italy`	`Other`	0.972	`Belgium`	-0.543
`Norway`	`Italy`	0.771	`Belgium`	-0.939
`Other`	`Italy`	0.972	`France`	-0.690
`Spain`	`Other`	0.595	`Germany`	-0.798
`UK`	`Other`	0.468	`France`	-0.958

Table 5.10: Nearest and furthest neighbors based on learned embedding weights for variable `country`.

Chapter 6

Discussion

Logit models have historically proven successful for analyzing and predicting mode choice. Previous research revealed promising performance of feedforward neural networks regarding this field. This thesis presented a mode choice case study by which performance of the more classical logit models was compared to that of infused feedforward neural networks containing embedding layers, of which the network structure was determined using an evolutionary optimization algorithm. It was expected that these networks would outperform the logit models. The presented results did indeed indicate better performance of such models. More precisely, these models generalize better on unseen data, obtaining higher accuracy on test data.

Regarding logit models, a standard multinomial logistic regression model was estimated first. Due to a guarantee of convergence, such models prove extremely convenient when relaxing assumptions on the distribution of error terms. When defining a nesting structure the estimated parameters of the standard model were taken as starting values for the parameters of the nested models. The presented results indicate that the leasing company defined nesting structure proves accurate, with almost all estimated nest parameters being significant. Whereas performance of the nested logistic regression model was slightly better than that of the standard model on training data, it generalized worse on unseen data. Performance of the cross-nested model was better on both training and test data. Considering that the increase in log-likelihood of the nested model relative to the standard model was negligible, both an increase in log-likelihood and accuracy on test data was observed for the cross-nested model. Even though the nesting structure used for the cross-nested model was considered accurate, the training data did not support the entire allocation of makes across nests. Relaxing assumptions on error term distribution does indeed improve performance of models, albeit slight. From estimation results it can be concluded that correlations across alternatives indeed exist, hence assigning each make one or multiple nests is justified.

The achieved accuracy on test data by both types of neural networks justify relaxation of assumptions. These networks do not assume linear uncorrelated inputs and the absence of correlation across alternatives, resulting in better performance than logit models. Not only did both types of networks outperform all logit models, both networks were also able to better predict minority alternatives. This property could prove convenient when all vehicles of the leasing company's fleet are included in the choice set. It is expected that neural networks are able to predict relatively low-occurring alternatives better than logit models when the entire fleet is included. The addition of embedding layers indeed improves performance and decreases training time. The embedding layer mapping the variable `make` onto continuous space ratifies the decision of dividing alternatives into nests. It allows for grouping together different makes of vehicles. Comparing the distribution of makes mapped onto continuous space with the allocation of makes in the cross-nested model, no overlap is observed. Nonetheless, when dividing makes into nests for a (cross-)nested model, embedding layers can prove useful. The learned embeddings can be used to redefine the nesting structure of makes. In addition, these embeddings can be used by the leasing company to segment their fleet of vehicles. Embedding layers did prove their usefulness. However, such layers will most likely prove more important when dealing with categorical variables containing more unique states. These variables will cause sparser training data. Additionally, training time improved significantly, but it is to be expected that embedding layers exponentially reduce training time for networks containing categorical variables with many unique states. Of course, one cannot conclude that embedding networks are better than one-hot networks. The determined optimal structure of the embedding network might not be the optimal structure of the one-hot network. If the optimal network structure was determined for a one-hot network, using this structure for an embedding network might result in better performance of the one-hot network. Nonetheless, it can be concluded that embedding networks significantly reduce training time, simply by the fact that less network weights need be estimated. Determining the structure of the final network through evolutionary optimization did prove its practicality. Determining the structure by trial and error was considered too prone for error and impractical, whereas an entire grid-search of hyper-parameters was too time consuming. Evolutionary algorithms have proven their strength in discovering high-fitness areas of the fitness landscape, and conveniently combine the best aspects of trial and error and grid-search procedures. The constructed evolutionary algorithm showed both *quality* and *novelty* aspects through generations. It showed early convergence towards a promising fitness-region, followed by an attempt to escape this potential local optimum. Towards the last generation, almost no variation among individuals existed, hence convergence towards an optimum took place.

The learning curves of both types of networks shown in Figure 5.2 illustrate the most troublesome part of this thesis. These figures indicate that the data is not fit for modeling and prediction. The fact that the training loss curve was not smoothly L-shaped and the minimum validation loss value was observed at the first epoch, strengthen this belief. Possible explanations for the incapability of modeling are the following. Firstly, the absence of driver IDs cause the data to be an approximation of true switch behavior. Recall from chapter 2 that this absence makes flawlessly matching contracts impossible. The data resulting from the matching procedure might exhibit non-existing relationships influencing the predictive power of the models. Another result from the absence of driver IDs is that no individual-specific information is provided. All explanatory variables either regard vehicle or contract attributes. The belief is that individual-specific variables (age, income etc.) will add great predictive power and cause the data to be less noisy.

In future research, it can be examined if the two families of models can supplement each other. An ensemble of a (cross-nested) logit model and a neural network could potentially outperform any individual model. In addition, the learned embeddings can be used as input for logit models as well. This will significantly reduce estimation time, resulting in more nesting structures able to be tested. On the downside, the ease of interpretation of logit models will become less since categorical variables will be split in dimensions rather than states of the variable. For the evolutionary algorithm a bigger population size, more generations and a higher degree of variation should be tested. A bigger population size and more generations might improve the final network's achieved accuracy. However, a more powerful computer or cloud-computing need be used. Evolutionary algorithms for neural network structure determination are computationally expensive, hence the choice of algorithm parameters.

In conclusion, this thesis aimed to empirically compare infused feedforward neural networks with variants of logit models on a mode choice problem. As expected, neural networks achieved higher accuracy of prediction on unseen data. Both the addition of embedding layers, and optimization of the network structure through proven their usefulness. Regarding logit models, relaxing assumptions on error term distribution allows for better capturing of correlations across alternatives. For the leasing company to use the models discussed in this thesis, the company should keep track of the drivers of their vehicles. Doing so will allow for accurate matching of contracts, whilst concurrently enhancing predictive power by addition of explanatory variables.

Bibliography

Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

Bahnsen, A. C. and Gonzalez, A. M. (2011). Evolutionary algorithms for selecting the architecture of a mlp neural network: A credit scoring case. In *2011 11th IEEE International Conference on Data Mining Workshops*, pages 725–732. IEEE.

Ben-Akiva, M. and Bierlaire, M. (2003). Discrete choice models with applications to departure time and route choice. In *Handbook of transportation science*, pages 7–37. Springer.

Bhat, C. R. and Sardesai, R. (2006). The impact of stop-making and travel time reliability on commute mode choice. *Transportation Research Part B: Methodological*, 40(9):709–730.

Bierlaire, M. (2015). Bisonbiogeme: estimating a first model. Technical report.

Bremermann, H. J. (1962). Optimization through evolution and recombination. *Self-organizing systems*, 93:106.

Castillo, P., Arenas, M., Castillo-Valdivieso, J., Merelo, J., Prieto, A., and Romero, G. (2003). Artificial neural networks design using evolutionary algorithms. In *Advances in Soft Computing*, pages 43–52. Springer.

Cauchy, A. (1847). Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Chu, Y.-L. (2009). Work departure time analysis using dogit ordered generalized extreme value model. *Transportation Research Record: Journal of the Transportation Research Board*, (2132):42–49.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Darwin, C. (1859). On the origin of species by means of natural selection. *Murray, London.*

David, O. E. and Greental, I. (2014). Genetic algorithms for evolving deep neural networks. In *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 1451–1452. ACM.

De Jong, G., Daly, A., Pieters, M., Vellay, C., Bradley, M., and Hofman, F. (2003). A model for time of day and mode choice using error components logit. *Transportation Research Part E: Logistics and Transportation Review*, 39(3):245–268.

Dia, H. and Panwai, S. (2010). Evaluation of discrete choice and neural network approaches for modelling driver compliance with traffic information. *Transportmetrica*, 6(4):249–270.

Ding, C., Mishra, S., Lin, Y., and Xie, B. (2014). Cross-nested joint model of travel mode and departure time choice for urban commuting trips: Case study in maryland–washington, dc region. *Journal of Urban Planning and Development*, 141(4):04014036.

Eiben, A. E., Smith, J. E., et al. (2003). *Introduction to evolutionary computing*, volume 53. Springer.

Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). Intelligent decision making through a simulation of evolution. *Behavioral science*, 11(4):253–272.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Guo, C. and Berkhahn, F. (2016). Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*.

Hensher, D. A. and Ton, T. T. (2000). A comparison of the predictive potential of artificial neural networks and nested logit models for commuter mode choice. *Transportation Research Part E: Logistics and Transportation Review*, 36(3):155–172.

Hess, S., Fowler, M., Adler, T., and Bahreinian, A. (2012). A joint model for vehicle type and fuel type choice: evidence from a cross-nested logit study. *Transportation*, 39(3):593–625.

Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing*, 2(2):88–105.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koppelman, F. S. and Bhat, C. (2006). A self instructing course in mode choice modeling: multinomial and nested logit models.

Koza, J. R. (1992). *Genetic programming II, automatic discovery of reusable subprograms*. MIT Press, Cambridge, MA.

Lam, H., Ling, S., Leung, F. H., and Tam, P. K.-S. (2001). Tuning of the structure and parameters of neural network using an improved genetic algorithm. In *Industrial Electronics Society, 2001. IECON'01. The 27th Annual Conference of the IEEE*, volume 1, pages 25–30. IEEE.

Llanas, B., Lantarón, S., and Sáinz, F. J. (2008). Constructive approximation of discontinuous functions by neural networks. *Neural Processing Letters*, 27(3):209–226.

Mahajan, R. and Kaur, G. (2013). Neural networks using genetic algorithms. *International Journal of Computer Applications*, 77(14).

McFadden, D. (1978). Modeling the choice of residential location. *Transportation Research Record*, (673).

McFadden, D. (2001). Economic choices. *American economic review*, 91(3):351–378.

McFadden, D. et al. (1973). Conditional logit analysis of qualitative choice behavior.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Montana, D. J. and Davis, L. (1989). Training feedforward neural networks using genetic algorithms. In *IJCAI*, volume 89, pages 762–767.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Nijkamp, P., Reggiani, A., and Tritapepe, T. (1996). Modelling inter-urban transport flows in italy: A comparison between neural network analysis and logit analysis.

Omrani, H. (2015). Predicting travel mode of individuals by machine learning. *Transportation Research Procedia*, 10:840–849.

Pulugurta, S., Arun, A., and Errampalli, M. (2013). Use of artificial intelligence for mode choice analysis and comparison with traditional multinomial logit model. *Procedia-Social and Behavioral Sciences*, 104:583–592.

Roquel, K. I. D. Z. and Fillone, A. M. (2016). Comparison of logit and neural network models in inter-island discrete choice analysis.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533.

Schwefel, H.-P., Schwefel, H.-P., Schwefel, H.-P., and Schwefel, H.-P. (1995). *Evolution and optimum seeking*, volume 1515. Wiley New York.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.

Train, K. E. (2009). *Discrete choice methods with simulation.* Cambridge university press.

Turing, A. (1948). Intelligent machinery. report for national physical laboratory. reprinted in ince, dc (editor). 1992. mechanical intelligence: Collected works of am turing.

Vovsha, P. (1997). Application of cross-nested logit model to mode choice in tel aviv, israel, metropolitan area. *Transportation Research Record: Journal of the Transportation Research Board*, (1607):6–15.

Vythoulkas, P. C. and Koutsopoulos, H. N. (2003). Modeling discrete choice behavior using concepts from fuzzy set theory, approximate reasoning and neural networks. *Transportation Research Part C: Emerging Technologies*, 11(1):51–73.

Wright, S. (1932). *The roles of mutation, inbreeding, crossbreeding, and selection in evolution*, volume 1. na.

Appendix A

Data Statistics

#Days	Freq	Perc
0	3290	29.73
-1	769	6.95
-3	383	3.46
-2	331	2.99
-4	311	2.81
-7	287	2.59
-5	264	2.39
-6	263	2.38
-8	191	1.73
-14	166	1.50
-13	160	1.45
1	159	1.44
-12	149	1.35
-11	133	1.20
-15	129	1.17

#Days	Freq	Perc
-9	123	1.11
-10	115	1.04
-21	110	0.99
-16	103	0.93
-20	90	0.81
-18	84	0.76
-17	83	0.75
-22	80	0.72
-19	75	0.68
-35	70	0.63
-27	69	0.62
-24	66	0.60
-28	64	0.58
3	61	0.55
-23	57	0.52

Table A.1: Table showing the data used to create Figure 2.1a. The first column shows the number of days between the start of a new contract and the termination date of the old contract. Hence, a negative number means that the new contract commenced prior to termination of the old contract. The second column illustrates the frequency of these differences occurring. Lastly, the third column portrays the percentage of these frequencies relative to the total number of matched contracts. Note that only the 30 most occurring entries are shown.

Perc. Mileage Difference	Frequency	Percentage
0.00	2293	20.72
50.00	279	2.52
25.00	278	2.51
33.33	271	2.45
-33.33	202	1.83
-25.00	193	1.74
-20.00	193	1.74
100.00	164	1.48
-16.67	160	1.45
20.00	160	1.45
66.67	118	1.07
-50.00	104	0.94
-14.29	104	0.94
11.11	94	0.85
-12.50	93	0.84
-57.66	88	0.80
16.67	80	0.72
14.29	76	0.69
-10.00	73	0.66
-1.92	73	0.66
12.50	59	0.53
-11.11	56	0.51
-40.00	54	0.49
42.86	51	0.46
60.00	50	0.45
-28.57	47	0.42
9.09	45	0.41
150.00	43	0.39
-9.09	42	0.38
15.38	41	0.37

Table A.2: Table showing the data used to create Figure 2.1b. The first column shows the percentage increase or decrease between the driven mileage per month of the terminated contract, and the agreed upon mileage per month for the new contract. Hence, a negative percentage indicated the mileage per month stated in the new contract is lower than the driven mileage per month of the terminated contract. The second column illustrates the frequency of these percentages occurring. Lastly, the third column portrays the percentage of these frequencies relative to the total number of matched contracts. Note that only the 30 most occurring entries are shown.

Appendix B

Estimates

Parameter	Estimate	Std. Error	t-stat	p-value
ASC_{BMW}	0.781	0.402	1.94	0.05
$\text{ASC}_{Citroen}$	-2.21	0.556	-3.98	0.00
ASC_{Ford}	-0.210	0.376	-0.56	0.58
$\text{ASC}_{Mercedes\text{-}Benz}$	0.501	0.367	1.36	0.17
ASC_{Nissan}	-0.00868	0.525	-0.02	0.99
ASC_{Opel}	-1.60	0.441	-3.63	0.00
ASC_{Other}	2.22	0.332	6.69	0.00
$\text{ASC}_{Peugeot}$	-2.61	0.414	-6.30	0.00
$\text{ASC}_{Renault}$	-0.881	0.374	-2.35	0.02
ASC_{Skoda}	-1.89	0.552	-3.43	0.00
$\text{ASC}_{Volkswagen}$	0.570	0.330	1.73	0.08
ASC_{Volvo}	1.63	0.493	3.32	0.00
$\beta_{\texttt{body style APV MPV Monovolume}\ BMW}$	-0.170	0.154	-1.10	0.27
$\beta_{\texttt{body style APV MPV Monovolume}\ Citroen}$	0.424	0.239	1.77	0.08
$\beta_{\texttt{body style APV MPV Monovolume}\ Ford}$	0.495	0.184	2.69	0.01
$\beta_{\texttt{body style APV MPV Monovolume}\ Mercedes\text{-}Benz}$	-0.510	0.173	-2.95	0.00
$\beta_{\texttt{body style APV MPV Monovolume}\ Nissan}$	-0.0995	0.237	-0.42	0.67
$\beta_{\texttt{body style APV MPV Monovolume}\ Opel}$	-0.263	0.251	-1.05	0.30
$\beta_{\texttt{body style APV MPV Monovolume}\ Other}$	-0.565	0.165	-3.43	0.00
$\beta_{\texttt{body style APV MPV Monovolume}\ Peugeot}$	0.400	0.172	2.33	0.02
$\beta_{\texttt{body style APV MPV Monovolume}\ Renault}$	-0.0803	0.200	-0.40	0.69
$\beta_{\texttt{body style APV MPV Monovolume}\ Skoda}$	-0.269	0.240	-1.12	0.26
$\beta_{\texttt{body style APV MPV Monovolume}\ Volkswagen}$	0.250	0.150	1.67	0.10
$\beta_{\texttt{body style APV MPV Monovolume}\ Volvo}$	-0.198	0.210	-0.94	0.35
$\beta_{\texttt{body style Car Van}\ BMW}$	-0.544	0.405	-1.34	0.18
$\beta_{\texttt{body style Car Van}\ Citroen}$	-1.30	0.412	-3.16	0.00
$\beta_{\texttt{body style Car Van}\ Ford}$	-1.92	0.387	-4.95	0.00
$\beta_{\texttt{body style Car Van}\ Mercedes\text{-}Benz}$	-0.910	0.426	-2.14	0.03

$\beta_{\texttt{body style Car Van } \textit{Nissan}}$	-1.30	0.444	-2.92	0.00
$\beta_{\texttt{body style Car Van } \textit{Opel}}$	-1.58	0.427	-3.70	0.00
$\beta_{\texttt{body style Car Van } \textit{Other}}$	-0.773	0.339	-2.28	0.02
$\beta_{\texttt{body style Car Van } \textit{Peugeot}}$	-0.689	0.364	-1.90	0.06
$\beta_{\texttt{body style Car Van } \textit{Renault}}$	-0.716	0.366	-1.96	0.05
$\beta_{\texttt{body style Car Van } \textit{Skoda}}$	-1.30	0.473	-2.75	0.01
$\beta_{\texttt{body style Car Van } \textit{Volkswagen}}$	-0.703	0.333	-2.11	0.03
$\beta_{\texttt{body style Car Van } \textit{Volvo}}$	-0.941	0.535	-1.76	0.08
$\beta_{\texttt{body style Delivery Van } \textit{BMW}}$	-0.419	0.374	-1.12	0.26
$\beta_{\texttt{body style Delivery Van } \textit{Citroen}}$	-1.88	0.392	-4.78	0.00
$\beta_{\texttt{body style Delivery Van } \textit{Ford}}$	-0.725	0.332	-2.18	0.03
$\beta_{\texttt{body style Delivery Van } \textit{Mercedes-Benz}}$	0.432	0.345	1.25	0.21
$\beta_{\texttt{body style Delivery Van } \textit{Nissan}}$	-1.05	0.397	-2.65	0.01
$\beta_{\texttt{body style Delivery Van } \textit{Opel}}$	-1.10	0.385	-2.86	0.00
$\beta_{\texttt{body style Delivery Van } \textit{Other}}$	-1.13	0.305	-3.71	0.00
$\beta_{\texttt{body style Delivery Van } \textit{Peugeot}}$	-1.16	0.334	-3.48	0.00
$\beta_{\texttt{body style Delivery Van } \textit{Renault}}$	-0.835	0.332	-2.52	0.01
$\beta_{\texttt{body style Delivery Van } \textit{Skoda}}$	-0.816	0.439	-1.86	0.06
$\beta_{\texttt{body style Delivery Van } \textit{Volkswagen}}$	-0.543	0.295	-1.84	0.07
$\beta_{\texttt{body style Delivery Van } \textit{Volvo}}$	0.274	0.375	0.73	0.46
$\beta_{\texttt{body style Hatchback } \textit{BMW}}$	-0.0761	0.121	-0.63	0.53
$\beta_{\texttt{body style Hatchback } \textit{Citroen}}$	-0.405	0.253	-1.60	0.11
$\beta_{\texttt{body style Hatchback } \textit{Ford}}$	-0.450	0.174	-2.58	0.01
$\beta_{\texttt{body style Hatchback } \textit{Mercedes-Benz}}$	-0.299	0.136	-2.20	0.03
$\beta_{\texttt{body style Hatchback } \textit{Nissan}}$	-0.415	0.198	-2.09	0.04
$\beta_{\texttt{body style Hatchback } \textit{Opel}}$	-1.20	0.224	-5.37	0.00
$\beta_{\texttt{body style Hatchback } \textit{Other}}$	-0.389	0.131	-2.97	0.00
$\beta_{\texttt{body style Hatchback } \textit{Peugeot}}$	-0.186	0.186	-1.00	0.32
$\beta_{\texttt{body style Hatchback } \textit{Renault}}$	-0.194	0.187	-1.04	0.30
$\beta_{\texttt{body style Hatchback } \textit{Skoda}}$	-0.868	0.218	-3.99	0.00
$\beta_{\texttt{body style Hatchback } \textit{Volkswagen}}$	-0.232	0.133	-1.74	0.08
$\beta_{\texttt{body style Hatchback } \textit{Volvo}}$	0.0202	0.176	0.11	0.91
$\beta_{\texttt{body style Other } \textit{BMW}}$	-0.0812	0.145	-0.56	0.57
$\beta_{\texttt{body style Other } \textit{Citroen}}$	-0.518	0.322	-1.61	0.11
$\beta_{\texttt{body style Other } \textit{Ford}}$	-0.355	0.220	-1.61	0.11
$\beta_{\texttt{body style Other } \textit{Mercedes-Benz}}$	-0.168	0.160	-1.05	0.29
$\beta_{\texttt{body style Other } \textit{Nissan}}$	-0.181	0.291	-0.62	0.53
$\beta_{\texttt{body style Other } \textit{Opel}}$	-0.411	0.260	-1.58	0.11
$\beta_{\texttt{body style Other } \textit{Other}}$	-0.259	0.172	-1.51	0.13
$\beta_{\texttt{body style Other } \textit{Peugeot}}$	-0.265	0.244	-1.08	0.28
$\beta_{\texttt{body style Other } \textit{Renault}}$	-0.213	0.235	-0.91	0.36
$\beta_{\texttt{body style Other } \textit{Skoda}}$	-0.161	0.242	-0.67	0.51

$\beta_{\texttt{body style Other } \textit{Volkswagen}}$	-0.0901	0.159	-0.57	0.57
$\beta_{\texttt{body style Other } \textit{Volvo}}$	0.0455	0.204	0.22	0.82
$\beta_{\texttt{body style Sedan } \textit{BMW}}$	-0.0411	0.125	-0.33	0.74
$\beta_{\texttt{body style Sedan } \textit{Citroen}}$	0.345	0.280	1.23	0.22
$\beta_{\texttt{body style Sedan } \textit{Ford}}$	0.0588	0.188	0.31	0.75
$\beta_{\texttt{body style Sedan } \textit{Mercedes-Benz}}$	-0.244	0.138	-1.77	0.08
$\beta_{\texttt{body style Sedan } \textit{Nissan}}$	0.0284	0.251	0.11	0.91
$\beta_{\texttt{body style Sedan } \textit{Opel}}$	-0.738	0.236	-3.13	0.00
$\beta_{\texttt{body style Sedan } \textit{Other}}$	0.0698	0.146	0.48	0.63
$\beta_{\texttt{body style Sedan } \textit{Peugeot}}$	0.136	0.210	0.65	0.52
$\beta_{\texttt{body style Sedan } \textit{Renault}}$	-0.0757	0.210	-0.36	0.72
$\beta_{\texttt{body style Sedan } \textit{Skoda}}$	-0.203	0.229	-0.88	0.38
$\beta_{\texttt{body style Sedan } \textit{Volkswagen}}$	0.0784	0.140	0.56	0.58
$\beta_{\texttt{body style Sedan } \textit{Volvo}}$	-0.0274	0.187	-0.15	0.88
$\beta_{\texttt{body style Stationwagon } \textit{BMW}}$	-0.167	0.113	-1.47	0.14
$\beta_{\texttt{body style Stationwagon } \textit{Citroen}}$	-0.532	0.252	-2.11	0.03
$\beta_{\texttt{body style Stationwagon } \textit{Ford}}$	0.0355	0.168	0.21	0.83
$\beta_{\texttt{body style Stationwagon } \textit{Mercedes-Benz}}$	-0.278	0.125	-2.23	0.03
$\beta_{\texttt{body style Stationwagon } \textit{Nissan}}$	-0.293	0.204	-1.44	0.15
$\beta_{\texttt{body style Stationwagon } \textit{Opel}}$	-0.733	0.222	-3.31	0.00
$\beta_{\texttt{body style Stationwagon } \textit{Other}}$	-0.0663	0.126	-0.53	0.60
$\beta_{\texttt{body style Stationwagon } \textit{Peugeot}}$	0.112	0.189	0.59	0.55
$\beta_{\texttt{body style Stationwagon } \textit{Renault}}$	-0.0887	0.186	-0.48	0.63
$\beta_{\texttt{body style Stationwagon } \textit{Skoda}}$	-0.110	0.206	-0.54	0.59
$\beta_{\texttt{body style Stationwagon } \textit{Volkswagen}}$	0.249	0.128	1.94	0.05
$\beta_{\texttt{body style Stationwagon } \textit{Volvo}}$	0.0853	0.160	0.53	0.59
$\beta_{\texttt{catalogue price } \textit{BMW}}$	0.411	0.0895	4.59	0.00
$\beta_{\texttt{catalogue price } \textit{Citroen}}$	-1.81	0.140	-12.92	0.00
$\beta_{\texttt{catalogue price } \textit{Ford}}$	-1.45	0.113	-12.84	0.00
$\beta_{\texttt{catalogue price } \textit{Mercedes-Benz}}$	0.295	0.0966	3.06	0.00
$\beta_{\texttt{catalogue price } \textit{Nissan}}$	-1.35	0.155	-8.73	0.00
$\beta_{\texttt{catalogue price } \textit{Opel}}$	-1.86	0.132	-14.10	0.00
$\beta_{\texttt{catalogue price } \textit{Other}}$	-0.318	0.0934	-3.41	0.00
$\beta_{\texttt{catalogue price } \textit{Peugeot}}$	-1.57	0.117	-13.36	0.00
$\beta_{\texttt{catalogue price } \textit{Renault}}$	-1.10	0.112	-9.79	0.00
$\beta_{\texttt{catalogue price } \textit{Skoda}}$	-1.19	0.142	-8.41	0.00
$\beta_{\texttt{catalogue price } \textit{Volkswagen}}$	-0.788	0.0894	-8.82	0.00
$\beta_{\texttt{catalogue price } \textit{Volvo}}$	0.214	0.143	1.50	0.13
$\beta_{\texttt{client segment Corporate } \textit{BMW}}$	0.0549	0.0672	0.82	0.41
$\beta_{\texttt{client segment Corporate } \textit{Citroen}}$	0.216	0.0880	2.45	0.01
$\beta_{\texttt{client segment Corporate } \textit{Ford}}$	-0.148	0.0723	-2.05	0.04
$\beta_{\texttt{client segment Corporate } \textit{Mercedes-Benz}}$	0.0265	0.0774	0.34	0.73

$\beta_{\texttt{client segment Corporate}\ \textit{Nissan}}$	-0.774	0.0902	-8.58	0.00
$\beta_{\texttt{client segment Corporate}\ \textit{Opel}}$	0.185	0.0854	2.17	0.03
$\beta_{\texttt{client segment Corporate}\ \textit{Other}}$	-0.0805	0.0619	-1.30	0.19
$\beta_{\texttt{client segment Corporate}\ \textit{Peugeot}}$	0.399	0.0689	5.78	0.00
$\beta_{\texttt{client segment Corporate}\ \textit{Renault}}$	0.544	0.0742	7.33	0.00
$\beta_{\texttt{client segment Corporate}\ \textit{Skoda}}$	-0.307	0.101	-3.03	0.00
$\beta_{\texttt{client segment Corporate}\ \textit{Volkswagen}}$	0.0515	0.0612	0.84	0.40
$\beta_{\texttt{client segment Corporate}\ \textit{Volvo}}$	-0.183	0.100	-1.83	0.07
$\beta_{\texttt{client segment International}\ \textit{BMW}}$	0.354	0.0686	5.16	0.00
$\beta_{\texttt{client segment International}\ \textit{Citroen}}$	0.628	0.0853	7.36	0.00
$\beta_{\texttt{client segment International}\ \textit{Ford}}$	-0.128	0.0778	-1.64	0.10
$\beta_{\texttt{client segment International}\ \textit{Mercedes-Benz}}$	0.407	0.0779	5.23	0.00
$\beta_{\texttt{client segment International}\ \textit{Nissan}}$	-1.22	0.103	-11.81	0.00
$\beta_{\texttt{client segment International}\ \textit{Opel}}$	0.0433	0.0936	0.46	0.64
$\beta_{\texttt{client segment International}\ \textit{Other}}$	-0.583	0.0681	-8.56	0.00
$\beta_{\texttt{client segment International}\ \textit{Peugeot}}$	0.162	0.0710	2.29	0.02
$\beta_{\texttt{client segment International}\ \textit{Renault}}$	0.904	0.0747	12.10	0.00
$\beta_{\texttt{client segment International}\ \textit{Skoda}}$	0.000226	0.101	0.00	1.00
$\beta_{\texttt{client segment International}\ \textit{Volkswagen}}$	0.0225	0.0628	0.36	0.72
$\beta_{\texttt{client segment International}\ \textit{Volvo}}$	-0.0529	0.100	-0.53	0.60
$\beta_{\texttt{commercial discount amount}\ \textit{BMW}}$	0.0103	0.0216	0.48	0.63
$\beta_{\texttt{commercial discount amount}\ \textit{Citroen}}$	-0.0139	0.0295	-0.47	0.64
$\beta_{\texttt{commercial discount amount}\ \textit{Ford}}$	0.0648	0.0259	2.50	0.01
$\beta_{\texttt{commercial discount amount}\ \textit{Mercedes-Benz}}$	-0.0485	0.0226	-2.15	0.03
$\beta_{\texttt{commercial discount amount}\ \textit{Nissan}}$	0.0933	0.0326	2.86	0.00
$\beta_{\texttt{commercial discount amount}\ \textit{Opel}}$	-0.0155	0.0246	-0.63	0.53
$\beta_{\texttt{commercial discount amount}\ \textit{Other}}$	0.0324	0.0222	1.46	0.14
$\beta_{\texttt{commercial discount amount}\ \textit{Peugeot}}$	0.0196	0.0256	0.76	0.44
$\beta_{\texttt{commercial discount amount}\ \textit{Renault}}$	0.00618	0.0283	0.22	0.83
$\beta_{\texttt{commercial discount amount}\ \textit{Skoda}}$	-0.00380	0.0282	-0.13	0.89
$\beta_{\texttt{commercial discount amount}\ \textit{Volkswagen}}$	0.0865	0.0236	3.67	0.00
$\beta_{\texttt{commercial discount amount}\ \textit{Volvo}}$	-0.109	0.0262	-4.16	0.00
$\beta_{\texttt{contract duration}\ \textit{BMW}}$	0.0225	0.0247	0.91	0.36
$\beta_{\texttt{contract duration}\ \textit{Citroen}}$	-0.190	0.0320	-5.95	0.00
$\beta_{\texttt{contract duration}\ \textit{Ford}}$	-0.111	0.0274	-4.06	0.00
$\beta_{\texttt{contract duration}\ \textit{Mercedes-Benz}}$	-0.141	0.0246	-5.74	0.00
$\beta_{\texttt{contract duration}\ \textit{Nissan}}$	-0.195	0.0427	-4.57	0.00
$\beta_{\texttt{contract duration}\ \textit{Opel}}$	-0.125	0.0299	-4.18	0.00
$\beta_{\texttt{contract duration}\ \textit{Other}}$	-0.187	0.0249	-7.50	0.00
$\beta_{\texttt{contract duration}\ \textit{Peugeot}}$	0.0183	0.0290	0.63	0.53
$\beta_{\texttt{contract duration}\ \textit{Renault}}$	-0.0555	0.0271	-2.05	0.04
$\beta_{\texttt{contract duration}\ \textit{Skoda}}$	0.148	0.0374	3.95	0.00

$\beta_{\texttt{contract duration } \textit{Volkswagen}}$	0.0645	0.0241	2.68	0.01
$\beta_{\texttt{contract duration } \textit{Volvo}}$	0.0332	0.0384	0.86	0.39
$\beta_{\texttt{country Belgium } \textit{BMW}}$	0.189	0.0957	1.98	0.05
$\beta_{\texttt{country Belgium } \textit{Citroen}}$	0.118	0.156	0.75	0.45
$\beta_{\texttt{country Belgium } \textit{Ford}}$	-0.691	0.121	-5.72	0.00
$\beta_{\texttt{country Belgium } \textit{Mercedes-Benz}}$	0.0191	0.103	0.19	0.85
$\beta_{\texttt{country Belgium } \textit{Nissan}}$	-0.716	0.198	-3.62	0.00
$\beta_{\texttt{country Belgium } \textit{Opel}}$	-1.28	0.124	-10.28	0.00
$\beta_{\texttt{country Belgium } \textit{Other}}$	-0.123	0.0995	-1.24	0.22
$\beta_{\texttt{country Belgium } \textit{Peugeot}}$	0.0826	0.148	0.56	0.58
$\beta_{\texttt{country Belgium } \textit{Renault}}$	-0.320	0.121	-2.64	0.01
$\beta_{\texttt{country Belgium } \textit{Skoda}}$	-0.549	0.160	-3.43	0.00
$\beta_{\texttt{country Belgium } \textit{Volkswagen}}$	0.0100	0.101	0.10	0.92
$\beta_{\texttt{country Belgium } \textit{Volvo}}$	0.413	0.153	2.69	0.01
$\beta_{\texttt{country France } \textit{BMW}}$	-0.737	0.113	-6.51	0.00
$\beta_{\texttt{country France } \textit{Citroen}}$	1.48	0.161	9.19	0.00
$\beta_{\texttt{country France } \textit{Ford}}$	-0.206	0.132	-1.56	0.12
$\beta_{\texttt{country France } \textit{Mercedes-Benz}}$	-0.924	0.120	-7.71	0.00
$\beta_{\texttt{country France } \textit{Nissan}}$	0.0226	0.200	0.11	0.91
$\beta_{\texttt{country France } \textit{Opel}}$	-1.21	0.144	-8.44	0.00
$\beta_{\texttt{country France } \textit{Other}}$	-0.580	0.114	-5.11	0.00
$\beta_{\texttt{country France } \textit{Peugeot}}$	2.55	0.149	17.20	0.00
$\beta_{\texttt{country France } \textit{Renault}}$	1.24	0.124	9.96	0.00
$\beta_{\texttt{country France } \textit{Skoda}}$	-1.43	0.190	-7.55	0.00
$\beta_{\texttt{country France } \textit{Volkswagen}}$	0.641	0.110	5.85	0.00
$\beta_{\texttt{country France } \textit{Volvo}}$	-1.15	0.180	-6.42	0.00
$\beta_{\texttt{country Germany } \textit{BMW}}$	0.0911	0.118	0.77	0.44
$\beta_{\texttt{country Germany } \textit{Citroen}}$	-0.148	0.208	-0.71	0.48
$\beta_{\texttt{country Germany } \textit{Ford}}$	0.846	0.135	6.25	0.00
$\beta_{\texttt{country Germany } \textit{Mercedes-Benz}}$	0.205	0.121	1.70	0.09
$\beta_{\texttt{country Germany } \textit{Nissan}}$	-2.70	0.384	-7.05	0.00
$\beta_{\texttt{country Germany } \textit{Opel}}$	-0.898	0.147	-6.12	0.00
$\beta_{\texttt{country Germany } \textit{Other}}$	-1.66	0.142	-11.70	0.00
$\beta_{\texttt{country Germany } \textit{Peugeot}}$	-1.80	0.267	-6.75	0.00
$\beta_{\texttt{country Germany } \textit{Renault}}$	-1.54	0.179	-8.61	0.00
$\beta_{\texttt{country Germany } \textit{Skoda}}$	-0.939	0.192	-4.89	0.00
$\beta_{\texttt{country Germany } \textit{Volkswagen}}$	0.307	0.117	2.63	0.01
$\beta_{\texttt{country Germany } \textit{Volvo}}$	-0.978	0.184	-5.32	0.00
$\beta_{\texttt{country Italy } \textit{BMW}}$	-0.0195	0.112	-0.17	0.86
$\beta_{\texttt{country Italy } \textit{Citroen}}$	0.910	0.171	5.32	0.00
$\beta_{\texttt{country Italy } \textit{Ford}}$	0.606	0.122	4.96	0.00
$\beta_{\texttt{country Italy } \textit{Mercedes-Benz}}$	-0.649	0.139	-4.69	0.00

$\beta_{\texttt{country Italy}\ \textit{Nissan}}$	0.522	0.176	2.97	0.00
$\beta_{\texttt{country Italy}\ \textit{Opel}}$	-0.458	0.135	-3.40	0.00
$\beta_{\texttt{country Italy}\ \textit{Other}}$	1.15	0.106	10.90	0.00
$\beta_{\texttt{country Italy}\ \textit{Peugeot}}$	1.74	0.145	11.95	0.00
$\beta_{\texttt{country Italy}\ \textit{Renault}}$	0.354	0.136	2.60	0.01
$\beta_{\texttt{country Italy}\ \textit{Skoda}}$	-0.499	0.192	-2.61	0.01
$\beta_{\texttt{country Italy}\ \textit{Volkswagen}}$	0.564	0.115	4.92	0.00
$\beta_{\texttt{country Italy}\ \textit{Volvo}}$	0.0527	0.192	0.27	0.78
$\beta_{\texttt{country Norway}\ \textit{BMW}}$	-1.30	0.359	-3.62	0.00
$\beta_{\texttt{country Norway}\ \textit{Citroen}}$	6.89	0.560	12.31	0.00
$\beta_{\texttt{country Norway}\ \textit{Ford}}$	6.27	0.439	14.28	0.00
$\beta_{\texttt{country Norway}\ \textit{Mercedes-Benz}}$	-1.29	0.388	-3.32	0.00
$\beta_{\texttt{country Norway}\ \textit{Nissan}}$	5.83	0.609	9.56	0.00
$\beta_{\texttt{country Norway}\ \textit{Opel}}$	5.82	0.522	11.14	0.00
$\beta_{\texttt{country Norway}\ \textit{Other}}$	1.61	0.368	4.38	0.00
$\beta_{\texttt{country Norway}\ \textit{Peugeot}}$	8.15	0.465	17.54	0.00
$\beta_{\texttt{country Norway}\ \textit{Renault}}$	3.44	0.442	7.78	0.00
$\beta_{\texttt{country Norway}\ \textit{Skoda}}$	5.82	0.559	10.40	0.00
$\beta_{\texttt{country Norway}\ \textit{Volkswagen}}$	4.81	0.354	13.62	0.00
$\beta_{\texttt{country Norway}\ \textit{Volvo}}$	-0.198	0.556	-0.36	0.72
$\beta_{\texttt{country Other}\ \textit{BMW}}$	-0.301	0.127	-2.37	0.02
$\beta_{\texttt{country Other}\ \textit{Citroen}}$	-2.20	0.556	-3.96	0.00
$\beta_{\texttt{country Other}\ \textit{Ford}}$	0.726	0.148	4.92	0.00
$\beta_{\texttt{country Other}\ \textit{Mercedes-Benz}}$	-0.749	0.142	-5.28	0.00
$\beta_{\texttt{country Other}\ \textit{Nissan}}$	0.0922	0.252	0.37	0.71
$\beta_{\texttt{country Other}\ \textit{Opel}}$	-0.755	0.188	-4.01	0.00
$\beta_{\texttt{country Other}\ \textit{Other}}$	0.289	0.132	2.20	0.03
$\beta_{\texttt{country Other}\ \textit{Peugeot}}$	0.891	0.224	3.98	0.00
$\beta_{\texttt{country Other}\ \textit{Renault}}$	-0.933	0.224	-4.17	0.00
$\beta_{\texttt{country Other}\ \textit{Skoda}}$	1.21	0.180	6.75	0.00
$\beta_{\texttt{country Other}\ \textit{Volkswagen}}$	1.39	0.123	11.25	0.00
$\beta_{\texttt{country Other}\ \textit{Volvo}}$	0.693	0.184	3.77	0.00
$\beta_{\texttt{country Spain}\ \textit{BMW}}$	-0.460	0.161	-2.86	0.00
$\beta_{\texttt{country Spain}\ \textit{Citroen}}$	-0.281	0.215	-1.31	0.19
$\beta_{\texttt{country Spain}\ \textit{Ford}}$	0.0621	0.162	0.38	0.70
$\beta_{\texttt{country Spain}\ \textit{Mercedes-Benz}}$	-0.871	0.173	-5.04	0.00
$\beta_{\texttt{country Spain}\ \textit{Nissan}}$	0.558	0.246	2.27	0.02
$\beta_{\texttt{country Spain}\ \textit{Opel}}$	-0.173	0.170	-1.02	0.31
$\beta_{\texttt{country Spain}\ \textit{Other}}$	1.02	0.138	7.41	0.00
$\beta_{\texttt{country Spain}\ \textit{Peugeot}}$	2.03	0.174	11.66	0.00
$\beta_{\texttt{country Spain}\ \textit{Renault}}$	1.49	0.150	9.92	0.00
$\beta_{\texttt{country Spain}\ \textit{Skoda}}$	-1.03	0.230	-4.49	0.00

$\beta_{\texttt{country Spain}\ Volkswagen}$	1.23	0.137	8.99	0.00
$\beta_{\texttt{country Spain}\ Volvo}$	-0.120	0.222	-0.54	0.59
$\beta_{\texttt{fuel type Diesel}\ BMW}$	-0.112	0.0885	-1.26	0.21
$\beta_{\texttt{fuel type Diesel}\ Citroen}$	0.342	0.149	2.29	0.02
$\beta_{\texttt{fuel type Diesel}\ Ford}$	0.268	0.103	2.61	0.01
$\beta_{\texttt{fuel type Diesel}\ Mercedes\text{-}Benz}$	-0.196	0.0950	-2.06	0.04
$\beta_{\texttt{fuel type Diesel}\ Nissan}$	-0.405	0.126	-3.22	0.00
$\beta_{\texttt{fuel type Diesel}\ Opel}$	0.388	0.105	3.70	0.00
$\beta_{\texttt{fuel type Diesel}\ Other}$	-0.412	0.0810	-5.09	0.00
$\beta_{\texttt{fuel type Diesel}\ Peugeot}$	0.909	0.138	6.60	0.00
$\beta_{\texttt{fuel type Diesel}\ Renault}$	0.852	0.142	6.01	0.00
$\beta_{\texttt{fuel type Diesel}\ Skoda}$	0.293	0.131	2.24	0.03
$\beta_{\texttt{fuel type Diesel}\ Volkswagen}$	0.160	0.0909	1.76	0.08
$\beta_{\texttt{fuel type Diesel}\ Volvo}$	-0.198	0.139	-1.42	0.15
$\beta_{\texttt{lease type Financial lease}\ BMW}$	-0.270	0.131	-2.06	0.04
$\beta_{\texttt{lease type Financial lease}\ Citroen}$	0.791	0.171	4.61	0.00
$\beta_{\texttt{lease type Financial lease}\ Ford}$	0.940	0.139	6.78	0.00
$\beta_{\texttt{lease type Financial lease}\ Mercedes\text{-}Benz}$	0.672	0.109	6.19	0.00
$\beta_{\texttt{lease type Financial lease}\ Nissan}$	-1.04	0.437	-2.38	0.02
$\beta_{\texttt{lease type Financial lease}\ Opel}$	0.632	0.175	3.60	0.00
$\beta_{\texttt{lease type Financial lease}\ Other}$	-0.619	0.163	-3.80	0.00
$\beta_{\texttt{lease type Financial lease}\ Peugeot}$	-0.273	0.212	-1.29	0.20
$\beta_{\texttt{lease type Financial lease}\ Renault}$	-0.239	0.188	-1.27	0.20
$\beta_{\texttt{lease type Financial lease}\ Skoda}$	0.342	0.214	1.60	0.11
$\beta_{\texttt{lease type Financial lease}\ Volkswagen}$	0.0552	0.137	0.40	0.69
$\beta_{\texttt{lease type Financial lease}\ Volvo}$	-0.996	0.263	-3.78	0.00
$\beta_{\texttt{make Audi}\ BMW}$	-0.929	0.104	-8.89	0.00
$\beta_{\texttt{make Audi}\ Citroen}$	-0.738	0.442	-1.67	0.10
$\beta_{\texttt{make Audi}\ Ford}$	-1.79	0.166	-10.80	0.00
$\beta_{\texttt{make Audi}\ Mercedes\text{-}Benz}$	-0.985	0.113	-8.69	0.00
$\beta_{\texttt{make Audi}\ Nissan}$	-1.18	0.356	-3.32	0.00
$\beta_{\texttt{make Audi}\ Opel}$	-0.958	0.239	-4.01	0.00
$\beta_{\texttt{make Audi}\ Other}$	-1.22	0.125	-9.76	0.00
$\beta_{\texttt{make Audi}\ Peugeot}$	-0.854	0.248	-3.44	0.00
$\beta_{\texttt{make Audi}\ Renault}$	-2.09	0.195	-10.72	0.00
$\beta_{\texttt{make Audi}\ Skoda}$	-1.02	0.165	-6.21	0.00
$\beta_{\texttt{make Audi}\ Volkswagen}$	-1.25	0.109	-11.43	0.00
$\beta_{\texttt{make Audi}\ Volvo}$	-3.16	0.117	-27.08	0.00
$\beta_{\texttt{make BMW}\ BMW}$	0.917	0.104	8.81	0.00
$\beta_{\texttt{make BMW}\ Citroen}$	0.395	0.450	0.88	0.38
$\beta_{\texttt{make BMW}\ Ford}$	-0.865	0.171	-5.06	0.00
$\beta_{\texttt{make BMW}\ Mercedes\text{-}Benz}$	-0.00883	0.116	-0.08	0.94

$\beta_{\texttt{make BMW } Nissan}$	0.453	0.339	1.33	0.18
$\beta_{\texttt{make BMW } Opel}$	1.02	0.226	4.49	0.00
$\beta_{\texttt{make BMW } Other}$	-0.321	0.130	-2.46	0.01
$\beta_{\texttt{make BMW } Peugeot}$	0.212	0.258	0.82	0.41
$\beta_{\texttt{make BMW } Renault}$	-0.779	0.198	-3.92	0.00
$\beta_{\texttt{make BMW } Skoda}$	-0.315	0.177	-1.77	0.08
$\beta_{\texttt{make BMW } Volkswagen}$	-0.533	0.117	-4.56	0.00
$\beta_{\texttt{make BMW } Volvo}$	-2.27	0.119	-19.07	0.00
$\beta_{\texttt{make Citroen } BMW}$	-0.0300	0.191	-0.16	0.88
$\beta_{\texttt{make Citroen } Citroen}$	3.32	0.434	7.65	0.00
$\beta_{\texttt{make Citroen } Ford}$	0.362	0.209	1.73	0.08
$\beta_{\texttt{make Citroen } Mercedes\text{-}Benz}$	0.236	0.199	1.19	0.24
$\beta_{\texttt{make Citroen } Nissan}$	1.40	0.368	3.82	0.00
$\beta_{\texttt{make Citroen } Opel}$	1.59	0.265	6.01	0.00
$\beta_{\texttt{make Citroen } Other}$	0.869	0.179	4.85	0.00
$\beta_{\texttt{make Citroen } Peugeot}$	2.30	0.261	8.80	0.00
$\beta_{\texttt{make Citroen } Renault}$	0.917	0.206	4.45	0.00
$\beta_{\texttt{make Citroen } Skoda}$	0.820	0.237	3.46	0.00
$\beta_{\texttt{make Citroen } Volkswagen}$	0.205	0.166	1.24	0.21
$\beta_{\texttt{make Citroen } Volvo}$	-1.39	0.227	-6.12	0.00
$\beta_{\texttt{make Ford } BMW}$	0.0206	0.145	0.14	0.89
$\beta_{\texttt{make Ford } Citroen}$	2.12	0.430	4.93	0.00
$\beta_{\texttt{make Ford } Ford}$	2.70	0.165	16.39	0.00
$\beta_{\texttt{make Ford } Mercedes\text{-}Benz}$	0.177	0.149	1.19	0.24
$\beta_{\texttt{make Ford } Nissan}$	1.66	0.341	4.88	0.00
$\beta_{\texttt{make Ford } Opel}$	1.23	0.240	5.12	0.00
$\beta_{\texttt{make Ford } Other}$	0.744	0.149	5.00	0.00
$\beta_{\texttt{make Ford } Peugeot}$	1.93	0.250	7.72	0.00
$\beta_{\texttt{make Ford } Renault}$	0.524	0.194	2.71	0.01
$\beta_{\texttt{make Ford } Skoda}$	0.525	0.191	2.75	0.01
$\beta_{\texttt{make Ford } Volkswagen}$	0.225	0.136	1.65	0.10
$\beta_{\texttt{make Ford } Volvo}$	-1.57	0.166	-9.50	0.00
$\beta_{\texttt{make Mercedes-Benz } BMW}$	0.399	0.121	3.30	0.00
$\beta_{\texttt{make Mercedes-Benz } Citroen}$	0.590	0.454	1.30	0.19
$\beta_{\texttt{make Mercedes-Benz } Ford}$	-0.587	0.188	-3.13	0.00
$\beta_{\texttt{make Mercedes-Benz } Mercedes\text{-}Benz}$	1.54	0.122	12.58	0.00
$\beta_{\texttt{make Mercedes-Benz } Nissan}$	0.726	0.376	1.93	0.05
$\beta_{\texttt{make Mercedes-Benz } Opel}$	0.560	0.252	2.22	0.03
$\beta_{\texttt{make Mercedes-Benz } Other}$	0.168	0.147	1.14	0.25
$\beta_{\texttt{make Mercedes-Benz } Peugeot}$	0.797	0.271	2.94	0.00
$\beta_{\texttt{make Mercedes-Benz } Renault}$	-0.358	0.213	-1.68	0.09
$\beta_{\texttt{make Mercedes-Benz } Skoda}$	0.0384	0.212	0.18	0.86

$\beta_{\texttt{make Mercedes-Benz}\ \textit{Volkswagen}}$	0.126	0.128	0.98	0.33
$\beta_{\texttt{make Mercedes-Benz}\ \textit{Volvo}}$	-1.74	0.149	-11.64	0.00
$\beta_{\texttt{make Nissan}\ \textit{BMW}}$	0.0161	0.240	0.07	0.95
$\beta_{\texttt{make Nissan}\ \textit{Citroen}}$	1.37	0.497	2.76	0.01
$\beta_{\texttt{make Nissan}\ \textit{Ford}}$	0.832	0.254	3.27	0.00
$\beta_{\texttt{make Nissan}\ \textit{Mercedes-Benz}}$	-0.00160	0.250	-0.01	0.99
$\beta_{\texttt{make Nissan}\ \textit{Nissan}}$	3.35	0.363	9.23	0.00
$\beta_{\texttt{make Nissan}\ \textit{Opel}}$	1.50	0.328	4.56	0.00
$\beta_{\texttt{make Nissan}\ \textit{Other}}$	1.13	0.212	5.35	0.00
$\beta_{\texttt{make Nissan}\ \textit{Peugeot}}$	1.24	0.311	4.00	0.00
$\beta_{\texttt{make Nissan}\ \textit{Renault}}$	0.378	0.270	1.40	0.16
$\beta_{\texttt{make Nissan}\ \textit{Skoda}}$	0.799	0.315	2.54	0.01
$\beta_{\texttt{make Nissan}\ \textit{Volkswagen}}$	0.634	0.207	3.07	0.00
$\beta_{\texttt{make Nissan}\ \textit{Volvo}}$	-2.22	0.319	-6.98	0.00
$\beta_{\texttt{make Opel}\ \textit{BMW}}$	-0.150	0.131	-1.14	0.25
$\beta_{\texttt{make Opel}\ \textit{Citroen}}$	1.01	0.439	2.30	0.02
$\beta_{\texttt{make Opel}\ \textit{Ford}}$	0.00935	0.167	0.06	0.96
$\beta_{\texttt{make Opel}\ \textit{Mercedes-Benz}}$	-0.460	0.146	-3.16	0.00
$\beta_{\texttt{make Opel}\ \textit{Nissan}}$	1.10	0.343	3.22	0.00
$\beta_{\texttt{make Opel}\ \textit{Opel}}$	2.88	0.225	12.79	0.00
$\beta_{\texttt{make Opel}\ \textit{Other}}$	0.103	0.141	0.73	0.46
$\beta_{\texttt{make Opel}\ \textit{Peugeot}}$	1.52	0.249	6.12	0.00
$\beta_{\texttt{make Opel}\ \textit{Renault}}$	-0.0217	0.193	-0.11	0.91
$\beta_{\texttt{make Opel}\ \textit{Skoda}}$	0.0961	0.186	0.52	0.61
$\beta_{\texttt{make Opel}\ \textit{Volkswagen}}$	-0.169	0.128	-1.32	0.19
$\beta_{\texttt{make Opel}\ \textit{Volvo}}$	-2.54	0.181	-14.06	0.00
$\beta_{\texttt{make Other}\ \textit{BMW}}$	0.267	0.129	2.06	0.04
$\beta_{\texttt{make Other}\ \textit{Citroen}}$	1.51	0.430	3.52	0.00
$\beta_{\texttt{make Other}\ \textit{Ford}}$	0.389	0.169	2.31	0.02
$\beta_{\texttt{make Other}\ \textit{Mercedes-Benz}}$	0.113	0.141	0.80	0.42
$\beta_{\texttt{make Other}\ \textit{Nissan}}$	1.37	0.332	4.12	0.00
$\beta_{\texttt{make Other}\ \textit{Opel}}$	1.45	0.232	6.23	0.00
$\beta_{\texttt{make Other}\ \textit{Other}}$	1.85	0.134	13.82	0.00
$\beta_{\texttt{make Other}\ \textit{Peugeot}}$	1.19	0.250	4.77	0.00
$\beta_{\texttt{make Other}\ \textit{Renault}}$	0.744	0.181	4.10	0.00
$\beta_{\texttt{make Other}\ \textit{Skoda}}$	0.988	0.177	5.58	0.00
$\beta_{\texttt{make Other}\ \textit{Volkswagen}}$	0.226	0.129	1.75	0.08
$\beta_{\texttt{make Other}\ \textit{Volvo}}$	-2.11	0.163	-12.98	0.00
$\beta_{\texttt{make Peugeot}\ \textit{BMW}}$	-0.144	0.152	-0.95	0.34
$\beta_{\texttt{make Peugeot}\ \textit{Citroen}}$	1.76	0.429	4.11	0.00
$\beta_{\texttt{make Peugeot}\ \textit{Ford}}$	0.636	0.174	3.66	0.00
$\beta_{\texttt{make Peugeot}\ \textit{Mercedes-Benz}}$	-0.302	0.167	-1.81	0.07

$\beta_{\texttt{make Peugeot}\ Nissan}$	1.10	0.344	3.20	0.00
$\beta_{\texttt{make Peugeot}\ Opel}$	1.11	0.244	4.55	0.00
$\beta_{\texttt{make Peugeot}\ Other}$	0.356	0.151	2.35	0.02
$\beta_{\texttt{make Peugeot}\ Peugeot}$	2.53	0.247	10.26	0.00
$\beta_{\texttt{make Peugeot}\ Renault}$	0.629	0.188	3.34	0.00
$\beta_{\texttt{make Peugeot}\ Skoda}$	0.446	0.198	2.25	0.02
$\beta_{\texttt{make Peugeot}\ Volkswagen}$	0.0441	0.137	0.32	0.75
$\beta_{\texttt{make Peugeot}\ Volvo}$	-2.47	0.216	-11.48	0.00
$\beta_{\texttt{make Renault}\ BMW}$	0.0258	0.154	0.17	0.87
$\beta_{\texttt{make Renault}\ Citroen}$	2.54	0.428	5.92	0.00
$\beta_{\texttt{make Renault}\ Ford}$	0.477	0.180	2.64	0.01
$\beta_{\texttt{make Renault}\ Mercedes\text{-}Benz}$	-0.0138	0.165	-0.08	0.93
$\beta_{\texttt{make Renault}\ Nissan}$	1.85	0.339	5.47	0.00
$\beta_{\texttt{make Renault}\ Opel}$	1.42	0.241	5.91	0.00
$\beta_{\texttt{make Renault}\ Other}$	1.03	0.147	6.96	0.00
$\beta_{\texttt{make Renault}\ Peugeot}$	1.77	0.251	7.05	0.00
$\beta_{\texttt{make Renault}\ Renault}$	2.07	0.186	11.11	0.00
$\beta_{\texttt{make Renault}\ Skoda}$	0.102	0.216	0.47	0.64
$\beta_{\texttt{make Renault}\ Volkswagen}$	0.174	0.140	1.25	0.21
$\beta_{\texttt{make Renault}\ Volvo}$	-1.45	0.167	-8.66	0.00
$\beta_{\texttt{make Skoda}\ BMW}$	0.00230	0.193	0.01	0.99
$\beta_{\texttt{make Skoda}\ Citroen}$	0.883	0.551	1.60	0.11
$\beta_{\texttt{make Skoda}\ Ford}$	0.0138	0.223	0.06	0.95
$\beta_{\texttt{make Skoda}\ Mercedes\text{-}Benz}$	0.207	0.197	1.05	0.29
$\beta_{\texttt{make Skoda}\ Nissan}$	1.57	0.385	4.08	0.00
$\beta_{\texttt{make Skoda}\ Opel}$	0.624	0.295	2.11	0.03
$\beta_{\texttt{make Skoda}\ Other}$	0.753	0.182	4.13	0.00
$\beta_{\texttt{make Skoda}\ Peugeot}$	0.692	0.321	2.15	0.03
$\beta_{\texttt{make Skoda}\ Renault}$	-0.264	0.299	-0.88	0.38
$\beta_{\texttt{make Skoda}\ Skoda}$	2.63	0.190	13.83	0.00
$\beta_{\texttt{make Skoda}\ Volkswagen}$	0.275	0.171	1.61	0.11
$\beta_{\texttt{make Skoda}\ Volvo}$	-1.83	0.230	-7.99	0.00
$\beta_{\texttt{make Volkswagen}\ BMW}$	-0.179	0.111	-1.61	0.11
$\beta_{\texttt{make Volkswagen}\ Citroen}$	0.675	0.425	1.59	0.11
$\beta_{\texttt{make Volkswagen}\ Ford}$	-0.120	0.154	-0.78	0.44
$\beta_{\texttt{make Volkswagen}\ Mercedes\text{-}Benz}$	-0.0323	0.119	-0.27	0.79
$\beta_{\texttt{make Volkswagen}\ Nissan}$	0.657	0.328	2.00	0.05
$\beta_{\texttt{make Volkswagen}\ Opel}$	0.433	0.226	1.92	0.06
$\beta_{\texttt{make Volkswagen}\ Other}$	-0.0858	0.127	-0.67	0.50
$\beta_{\texttt{make Volkswagen}\ Peugeot}$	0.573	0.240	2.39	0.02
$\beta_{\texttt{make Volkswagen}\ Renault}$	-0.480	0.178	-2.70	0.01
$\beta_{\texttt{make Volkswagen}\ Skoda}$	0.516	0.156	3.31	0.00

$\beta_{\texttt{make Volkswagen}\ \textit{Volkswagen}}$	1.41	0.107	13.16	0.00
$\beta_{\texttt{make Volkswagen}\ \textit{Volvo}}$	-2.02	0.122	-16.62	0.00
$\beta_{\texttt{mileage month}\ \textit{BMW}}$	-0.0480	0.0252	-1.90	0.06
$\beta_{\texttt{mileage month}\ \textit{Citroen}}$	0.0948	0.0320	2.97	0.00
$\beta_{\texttt{mileage month}\ \textit{Ford}}$	0.158	0.0284	5.56	0.00
$\beta_{\texttt{mileage month}\ \textit{Mercedes-Benz}}$	-0.109	0.0259	-4.22	0.00
$\beta_{\texttt{mileage month}\ \textit{Nissan}}$	0.0186	0.0413	0.45	0.65
$\beta_{\texttt{mileage month}\ \textit{Opel}}$	0.145	0.0314	4.61	0.00
$\beta_{\texttt{mileage month}\ \textit{Other}}$	-0.0796	0.0263	-3.03	0.00
$\beta_{\texttt{mileage month}\ \textit{Peugeot}}$	0.254	0.0273	9.30	0.00
$\beta_{\texttt{mileage month}\ \textit{Renault}}$	0.166	0.0275	6.04	0.00
$\beta_{\texttt{mileage month}\ \textit{Skoda}}$	0.450	0.0357	12.60	0.00
$\beta_{\texttt{mileage month}\ \textit{Volkswagen}}$	0.207	0.0234	8.81	0.00
$\beta_{\texttt{mileage month}\ \textit{Volvo}}$	0.0440	0.0386	1.14	0.25
$\beta_{\texttt{standard discount percentage}\ \textit{BMW}}$	-0.0406	0.0494	-0.82	0.41
$\beta_{\texttt{standard discount percentage}\ \textit{Citroen}}$	0.246	0.0504	4.87	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Ford}}$	0.125	0.0454	2.74	0.01
$\beta_{\texttt{standard discount percentage}\ \textit{Mercedes-Benz}}$	0.191	0.0478	4.00	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Nissan}}$	0.209	0.0779	2.69	0.01
$\beta_{\texttt{standard discount percentage}\ \textit{Opel}}$	0.316	0.0517	6.12	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Other}}$	0.154	0.0448	3.44	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Peugeot}}$	0.150	0.0474	3.17	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Renault}}$	0.182	0.0446	4.08	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Skoda}}$	0.471	0.0675	6.98	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Volkswagen}}$	0.198	0.0434	4.57	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Volvo}}$	0.306	0.0688	4.45	0.00
$\beta_{\texttt{switch quarter 1}\ \textit{BMW}}$	0.0556	0.0544	1.02	0.31
$\beta_{\texttt{switch quarter 1}\ \textit{Citroen}}$	-0.382	0.0752	-5.09	0.00
$\beta_{\texttt{switch quarter 1}\ \textit{Ford}}$	-0.166	0.0640	-2.60	0.01
$\beta_{\texttt{switch quarter 1}\ \textit{Mercedes-Benz}}$	-0.133	0.0591	-2.24	0.02
$\beta_{\texttt{switch quarter 1}\ \textit{Nissan}}$	0.000391	0.0964	0.00	1.00
$\beta_{\texttt{switch quarter 1}\ \textit{Opel}}$	-0.0908	0.0703	-1.29	0.20
$\beta_{\texttt{switch quarter 1}\ \textit{Other}}$	-0.0638	0.0581	-1.10	0.27
$\beta_{\texttt{switch quarter 1}\ \textit{Peugeot}}$	-0.276	0.0619	-4.45	0.00
$\beta_{\texttt{switch quarter 1}\ \textit{Renault}}$	-0.155	0.0606	-2.56	0.01
$\beta_{\texttt{switch quarter 1}\ \textit{Skoda}}$	0.0863	0.0808	1.07	0.29
$\beta_{\texttt{switch quarter 1}\ \textit{Volkswagen}}$	-0.195	0.0522	-3.74	0.00
$\beta_{\texttt{switch quarter 1}\ \textit{Volvo}}$	-0.00612	0.0806	-0.08	0.94
$\beta_{\texttt{switch quarter 2}\ \textit{BMW}}$	0.0258	0.0526	0.49	0.62
$\beta_{\texttt{switch quarter 2}\ \textit{Citroen}}$	-0.197	0.0720	-2.74	0.01
$\beta_{\texttt{switch quarter 2}\ \textit{Ford}}$	-0.142	0.0629	-2.26	0.02
$\beta_{\texttt{switch quarter 2}\ \textit{Mercedes-Benz}}$	-0.0735	0.0560	-1.31	0.19

$\beta_{\texttt{switch quarter 2}\ Nissan}$	0.0520	0.0919	0.57	0.57
$\beta_{\texttt{switch quarter 2}\ Opel}$	-0.00547	0.0665	-0.08	0.93
$\beta_{\texttt{switch quarter 2}\ Other}$	-0.278	0.0574	-4.84	0.00
$\beta_{\texttt{switch quarter 2}\ Peugeot}$	-0.285	0.0614	-4.65	0.00
$\beta_{\texttt{switch quarter 2}\ Renault}$	-0.368	0.0619	-5.95	0.00
$\beta_{\texttt{switch quarter 2}\ Skoda}$	0.0199	0.0799	0.25	0.80
$\beta_{\texttt{switch quarter 2}\ Volkswagen}$	-0.222	0.0509	-4.36	0.00
$\beta_{\texttt{switch quarter 2}\ Volvo}$	-0.0492	0.0786	-0.63	0.53
$\beta_{\texttt{switch quarter 3}\ BMW}$	-0.0774	0.0532	-1.45	0.15
$\beta_{\texttt{switch quarter 3}\ Citroen}$	-0.307	0.0735	-4.18	0.00
$\beta_{\texttt{switch quarter 3}\ Ford}$	-0.149	0.0611	-2.45	0.01
$\beta_{\texttt{switch quarter 3}\ Mercedes\text{-}Benz}$	-0.0251	0.0554	-0.45	0.65
$\beta_{\texttt{switch quarter 3}\ Nissan}$	-0.171	0.0941	-1.81	0.07
$\beta_{\texttt{switch quarter 3}\ Opel}$	0.0206	0.0659	0.31	0.75
$\beta_{\texttt{switch quarter 3}\ Other}$	-0.0944	0.0549	-1.72	0.09
$\beta_{\texttt{switch quarter 3}\ Peugeot}$	-0.285	0.0613	-4.65	0.00
$\beta_{\texttt{switch quarter 3}\ Renault}$	-0.383	0.0606	-6.32	0.00
$\beta_{\texttt{switch quarter 3}\ Skoda}$	-0.179	0.0810	-2.21	0.03
$\beta_{\texttt{switch quarter 3}\ Volkswagen}$	-0.298	0.0507	-5.88	0.00
$\beta_{\texttt{switch quarter 3}\ Volvo}$	-0.268	0.0818	-3.28	0.00
$\beta_{\texttt{total accessories amount}\ BMW}$	-0.0963	0.0211	-4.56	0.00
$\beta_{\texttt{total accessories amount}\ Citroen}$	0.0589	0.0301	1.96	0.05
$\beta_{\texttt{total accessories amount}\ Ford}$	0.0815	0.0248	3.29	0.00
$\beta_{\texttt{total accessories amount}\ Mercedes\text{-}Benz}$	-0.0830	0.0226	-3.68	0.00
$\beta_{\texttt{total accessories amount}\ Nissan}$	-0.0312	0.0393	-0.79	0.43
$\beta_{\texttt{total accessories amount}\ Opel}$	0.131	0.0267	4.89	0.00
$\beta_{\texttt{total accessories amount}\ Other}$	-0.187	0.0234	-7.99	0.00
$\beta_{\texttt{total accessories amount}\ Peugeot}$	-0.0748	0.0262	-2.85	0.00
$\beta_{\texttt{total accessories amount}\ Renault}$	0.0486	0.0249	1.95	0.05
$\beta_{\texttt{total accessories amount}\ Skoda}$	0.0300	0.0322	0.93	0.35
$\beta_{\texttt{total accessories amount}\ Volkswagen}$	-0.112	0.0211	-5.30	0.00
$\beta_{\texttt{total accessories amount}\ Volvo}$	-0.0331	0.0320	-1.04	0.30
$\beta_{\texttt{total options amount}\ BMW}$	-0.0750	0.0277	-2.70	0.01
$\beta_{\texttt{total options amount}\ Citroen}$	-0.187	0.0313	-5.97	0.00
$\beta_{\texttt{total options amount}\ Ford}$	-0.158	0.0286	-5.51	0.00
$\beta_{\texttt{total options amount}\ Mercedes\text{-}Benz}$	-0.0118	0.0312	-0.38	0.70
$\beta_{\texttt{total options amount}\ Nissan}$	-0.0754	0.0389	-1.94	0.05
$\beta_{\texttt{total options amount}\ Opel}$	-0.0748	0.0297	-2.52	0.01
$\beta_{\texttt{total options amount}\ Other}$	-0.0864	0.0260	-3.32	0.00
$\beta_{\texttt{total options amount}\ Peugeot}$	-0.0482	0.0287	-1.68	0.09
$\beta_{\texttt{total options amount}\ Renault}$	-0.0164	0.0285	-0.57	0.57
$\beta_{\texttt{total options amount}\ Skoda}$	-0.141	0.0381	-3.69	0.00

$\beta_{\texttt{total options amount } \textit{Volkswagen}}$	-0.134	0.0258	-5.22	0.00
$\beta_{\texttt{total options amount } \textit{Volvo}}$	-0.0668	0.0398	-1.68	0.09
$\beta_{\texttt{ufwt amount } \textit{BMW}}$	-0.0523	0.0219	-2.38	0.02
$\beta_{\texttt{ufwt amount } \textit{Citroen}}$	0.0334	0.0291	1.15	0.25
$\beta_{\texttt{ufwt amount } \textit{Ford}}$	-0.0853	0.0253	-3.37	0.00
$\beta_{\texttt{ufwt amount } \textit{Mercedes-Benz}}$	-0.124	0.0232	-5.34	0.00
$\beta_{\texttt{ufwt amount } \textit{Nissan}}$	-0.112	0.0359	-3.12	0.00
$\beta_{\texttt{ufwt amount } \textit{Opel}}$	-0.113	0.0268	-4.20	0.00
$\beta_{\texttt{ufwt amount } \textit{Other}}$	-0.129	0.0227	-5.69	0.00
$\beta_{\texttt{ufwt amount } \textit{Peugeot}}$	-0.124	0.0242	-5.10	0.00
$\beta_{\texttt{ufwt amount } \textit{Renault}}$	0.0393	0.0247	1.59	0.11
$\beta_{\texttt{ufwt amount } \textit{Skoda}}$	0.0399	0.0349	1.14	0.25
$\beta_{\texttt{ufwt amount } \textit{Volkswagen}}$	-0.0188	0.0212	-0.89	0.37
$\beta_{\texttt{ufwt amount } \textit{Volvo}}$	0.0458	0.0344	1.33	0.18
$\beta_{\texttt{vehicle segment B } \textit{BMW}}$	-0.329	0.169	-1.94	0.05
$\beta_{\texttt{vehicle segment B } \textit{Citroen}}$	0.713	0.250	2.85	0.00
$\beta_{\texttt{vehicle segment B } \textit{Ford}}$	0.421	0.180	2.34	0.02
$\beta_{\texttt{vehicle segment B } \textit{Mercedes-Benz}}$	-0.192	0.184	-1.04	0.30
$\beta_{\texttt{vehicle segment B } \textit{Nissan}}$	-0.484	0.243	-1.99	0.05
$\beta_{\texttt{vehicle segment B } \textit{Opel}}$	0.675	0.221	3.05	0.00
$\beta_{\texttt{vehicle segment B } \textit{Other}}$	0.487	0.156	3.13	0.00
$\beta_{\texttt{vehicle segment B } \textit{Peugeot}}$	0.0766	0.199	0.39	0.70
$\beta_{\texttt{vehicle segment B } \textit{Renault}}$	1.28	0.196	6.52	0.00
$\beta_{\texttt{vehicle segment B } \textit{Skoda}}$	-0.144	0.254	-0.57	0.57
$\beta_{\texttt{vehicle segment B } \textit{Volkswagen}}$	-0.0843	0.158	-0.53	0.59
$\beta_{\texttt{vehicle segment B } \textit{Volvo}}$	-0.953	0.279	-3.42	0.00
$\beta_{\texttt{vehicle segment C } \textit{BMW}}$	-0.211	0.109	-1.95	0.05
$\beta_{\texttt{vehicle segment C } \textit{Citroen}}$	0.424	0.218	1.95	0.05
$\beta_{\texttt{vehicle segment C } \textit{Ford}}$	0.496	0.136	3.64	0.00
$\beta_{\texttt{vehicle segment C } \textit{Mercedes-Benz}}$	-0.197	0.120	-1.64	0.10
$\beta_{\texttt{vehicle segment C } \textit{Nissan}}$	-0.229	0.192	-1.19	0.23
$\beta_{\texttt{vehicle segment C } \textit{Opel}}$	1.21	0.180	6.74	0.00
$\beta_{\texttt{vehicle segment C } \textit{Other}}$	0.00401	0.117	0.03	0.97
$\beta_{\texttt{vehicle segment C } \textit{Peugeot}}$	0.252	0.167	1.51	0.13
$\beta_{\texttt{vehicle segment C } \textit{Renault}}$	0.198	0.167	1.19	0.24
$\beta_{\texttt{vehicle segment C } \textit{Skoda}}$	0.551	0.179	3.08	0.00
$\beta_{\texttt{vehicle segment C } \textit{Volkswagen}}$	0.129	0.117	1.11	0.27
$\beta_{\texttt{vehicle segment C } \textit{Volvo}}$	-0.863	0.160	-5.40	0.00
$\beta_{\texttt{vehicle segment D } \textit{BMW}}$	-0.404	0.0926	-4.37	0.00
$\beta_{\texttt{vehicle segment D } \textit{Citroen}}$	0.225	0.228	0.99	0.32
$\beta_{\texttt{vehicle segment D } \textit{Ford}}$	0.0758	0.129	0.59	0.56
$\beta_{\texttt{vehicle segment D } \textit{Mercedes-Benz}}$	-0.300	0.103	-2.93	0.00

$\beta_{\texttt{vehicle segment D } \textit{Nissan}}$	-0.315	0.200	-1.58	0.11
$\beta_{\texttt{vehicle segment D } \textit{Opel}}$	0.957	0.177	5.39	0.00
$\beta_{\texttt{vehicle segment D } \textit{Other}}$	-0.431	0.111	-3.88	0.00
$\beta_{\texttt{vehicle segment D } \textit{Peugeot}}$	0.00567	0.168	0.03	0.97
$\beta_{\texttt{vehicle segment D } \textit{Renault}}$	-0.00376	0.167	-0.02	0.98
$\beta_{\texttt{vehicle segment D } \textit{Skoda}}$	0.309	0.176	1.76	0.08
$\beta_{\texttt{vehicle segment D } \textit{Volkswagen}}$	0.0562	0.108	0.52	0.60
$\beta_{\texttt{vehicle segment D } \textit{Volvo}}$	-0.732	0.137	-5.34	0.00
$\beta_{\texttt{vehicle segment E } \textit{BMW}}$	-0.486	0.102	-4.74	0.00
$\beta_{\texttt{vehicle segment E } \textit{Citroen}}$	-0.0521	0.415	-0.13	0.90
$\beta_{\texttt{vehicle segment E } \textit{Ford}}$	-0.151	0.185	-0.82	0.41
$\beta_{\texttt{vehicle segment E } \textit{Mercedes-Benz}}$	-0.406	0.113	-3.60	0.00
$\beta_{\texttt{vehicle segment E } \textit{Nissan}}$	-0.414	0.333	-1.24	0.21
$\beta_{\texttt{vehicle segment E } \textit{Opel}}$	1.06	0.219	4.82	0.00
$\beta_{\texttt{vehicle segment E } \textit{Other}}$	-0.372	0.138	-2.69	0.01
$\beta_{\texttt{vehicle segment E } \textit{Peugeot}}$	-0.265	0.281	-0.95	0.34
$\beta_{\texttt{vehicle segment E } \textit{Renault}}$	0.212	0.236	0.90	0.37
$\beta_{\texttt{vehicle segment E } \textit{Skoda}}$	0.0700	0.233	0.30	0.76
$\beta_{\texttt{vehicle segment E } \textit{Volkswagen}}$	-0.247	0.133	-1.85	0.06
$\beta_{\texttt{vehicle segment E } \textit{Volvo}}$	-0.694	0.154	-4.52	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{BMW}}$	-0.172	0.358	-0.48	0.63
$\beta_{\texttt{vehicle segment LCV } \textit{Citroen}}$	2.38	0.334	7.13	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{Ford}}$	1.73	0.278	6.23	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{Mercedes-Benz}}$	0.972	0.291	3.34	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{Nissan}}$	0.966	0.383	2.52	0.01
$\beta_{\texttt{vehicle segment LCV } \textit{Opel}}$	2.09	0.338	6.18	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{Other}}$	0.536	0.278	1.93	0.05
$\beta_{\texttt{vehicle segment LCV } \textit{Peugeot}}$	1.24	0.294	4.23	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{Renault}}$	2.03	0.294	6.92	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{Skoda}}$	1.05	0.409	2.56	0.01
$\beta_{\texttt{vehicle segment LCV } \textit{Volkswagen}}$	0.955	0.266	3.59	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{Volvo}}$	0.279	0.387	0.72	0.47
$\beta_{\texttt{vehicle segment MPV } \textit{BMW}}$	0.220	0.136	1.62	0.11
$\beta_{\texttt{vehicle segment MPV } \textit{Citroen}}$	0.395	0.197	2.00	0.05
$\beta_{\texttt{vehicle segment MPV } \textit{Ford}}$	-0.0512	0.147	-0.35	0.73
$\beta_{\texttt{vehicle segment MPV } \textit{Mercedes-Benz}}$	0.0207	0.148	0.14	0.89
$\beta_{\texttt{vehicle segment MPV } \textit{Nissan}}$	0.00443	0.229	0.02	0.98
$\beta_{\texttt{vehicle segment MPV } \textit{Opel}}$	0.685	0.206	3.32	0.00
$\beta_{\texttt{vehicle segment MPV } \textit{Other}}$	0.0355	0.147	0.24	0.81
$\beta_{\texttt{vehicle segment MPV } \textit{Peugeot}}$	-0.189	0.147	-1.28	0.20
$\beta_{\texttt{vehicle segment MPV } \textit{Renault}}$	0.404	0.177	2.29	0.02
$\beta_{\texttt{vehicle segment MPV } \textit{Skoda}}$	0.558	0.203	2.75	0.01

$\beta_{\texttt{vehicle segment MPV}\ \mathit{Volkswagen}}$	0.289	0.131	2.20	0.03
$\beta_{\texttt{vehicle segment MPV}\ \mathit{Volvo}}$	-0.333	0.190	-1.76	0.08
$\beta_{\texttt{vehicle segment Other}\ \mathit{BMW}}$	-0.0940	0.258	-0.36	0.72
$\beta_{\texttt{vehicle segment Other}\ \mathit{Citroen}}$	0.441	0.331	1.33	0.18
$\beta_{\texttt{vehicle segment Other}\ \mathit{Ford}}$	0.779	0.279	2.79	0.01
$\beta_{\texttt{vehicle segment Other}\ \mathit{Mercedes\text{-}Benz}}$	0.287	0.271	1.06	0.29
$\beta_{\texttt{vehicle segment Other}\ \mathit{Nissan}}$	-0.449	0.361	-1.24	0.21
$\beta_{\texttt{vehicle segment Other}\ \mathit{Opel}}$	0.671	0.328	2.04	0.04
$\beta_{\texttt{vehicle segment Other}\ \mathit{Other}}$	0.201	0.240	0.84	0.40
$\beta_{\texttt{vehicle segment Other}\ \mathit{Peugeot}}$	0.255	0.302	0.84	0.40
$\beta_{\texttt{vehicle segment Other}\ \mathit{Renault}}$	0.361	0.307	1.18	0.24
$\beta_{\texttt{vehicle segment Other}\ \mathit{Skoda}}$	-0.298	0.490	-0.61	0.54
$\beta_{\texttt{vehicle segment Other}\ \mathit{Volkswagen}}$	0.197	0.258	0.76	0.44
$\beta_{\texttt{vehicle segment Other}\ \mathit{Volvo}}$	-0.913	0.489	-1.87	0.06
$\beta_{\texttt{vehicle type Car}\ \mathit{BMW}}$	-0.441	0.349	-1.26	0.21
$\beta_{\texttt{vehicle type Car}\ \mathit{Citroen}}$	-2.11	0.262	-8.06	0.00
$\beta_{\texttt{vehicle type Car}\ \mathit{Ford}}$	-1.12	0.275	-4.09	0.00
$\beta_{\texttt{vehicle type Car}\ \mathit{Mercedes\text{-}Benz}}$	-0.00713	0.303	-0.02	0.98
$\beta_{\texttt{vehicle type Car}\ \mathit{Nissan}}$	-1.13	0.308	-3.67	0.00
$\beta_{\texttt{vehicle type Car}\ \mathit{Opel}}$	-0.461	0.300	-1.54	0.12
$\beta_{\texttt{vehicle type Car}\ \mathit{Other}}$	-1.59	0.264	-6.01	0.00
$\beta_{\texttt{vehicle type Car}\ \mathit{Peugeot}}$	-1.81	0.258	-7.01	0.00
$\beta_{\texttt{vehicle type Car}\ \mathit{Renault}}$	-1.81	0.257	-7.04	0.00
$\beta_{\texttt{vehicle type Car}\ \mathit{Skoda}}$	0.543	0.465	1.17	0.24
$\beta_{\texttt{vehicle type Car}\ \mathit{Volkswagen}}$	-1.32	0.263	-5.00	0.00
$\beta_{\texttt{vehicle type Car}\ \mathit{Volvo}}$	0.254	0.411	0.62	0.54

Table B.1: Overview of all estimated parameters for the multinomial logistic regression model. Each parameter name (β) is structured as follows. The first part, stated in `font`, is considered the explanatory variable. The second part of the name, stated in italics, is considered an alternative of the choice set.

Parameter	**Estimate**	**Std. Error**	***t*-stat**	***p*-value**
$\text{ASC}_{\mathit{BMW}}$	0.460	0.243	1.89	0.06
$\text{ASC}_{\mathit{Citroen}}$	-3.14	0.523	-6.01	0.00
$\text{ASC}_{\mathit{Ford}}$	-0.810	0.403	-2.01	0.04
$\text{ASC}_{\mathit{Mercedes\text{-}Benz}}$	0.250	0.224	1.12	0.26
$\text{ASC}_{\mathit{Nissan}}$	0.718	0.681	1.05	0.29
$\text{ASC}_{\mathit{Opel}}$	-2.09	0.576	-3.63	0.00
$\text{ASC}_{\mathit{Other}}$	1.98	0.367	5.41	0.00
$\text{ASC}_{\mathit{Peugeot}}$	-2.92	0.730	-4.00	0.00
$\text{ASC}_{\mathit{Renault}}$	-1.80	0.321	-5.62	0.00

ASC_{Skoda}	-2.36	0.673	-3.51	0.00
$\text{ASC}_{Volkswagen}$	0.221	0.207	1.06	0.29
ASC_{Volvo}	1.50	0.455	3.30	0.00
$\beta_{\texttt{body style APV MPV Monovolume } BMW}$	-0.0919	0.0936	-0.98	0.33
$\beta_{\texttt{body style APV MPV Monovolume } Citroen}$	0.421	0.225	1.87	0.06
$\beta_{\texttt{body style APV MPV Monovolume } Ford}$	0.448	0.156	2.86	0.00
$\beta_{\texttt{body style APV MPV Monovolume } Mercedes\text{-}Benz}$	-0.285	0.107	-2.66	0.01
$\beta_{\texttt{body style APV MPV Monovolume } Nissan}$	-0.272	0.177	-1.54	0.12
$\beta_{\texttt{body style APV MPV Monovolume } Opel}$	-0.135	0.226	-0.60	0.55
$\beta_{\texttt{body style APV MPV Monovolume } Other}$	-0.500	0.132	-3.77	0.00
$\beta_{\texttt{body style APV MPV Monovolume } Peugeot}$	0.398	0.145	2.74	0.01
$\beta_{\texttt{body style APV MPV Monovolume } Renault}$	-0.0775	0.183	-0.42	0.67
$\beta_{\texttt{body style APV MPV Monovolume } Skoda}$	-0.188	0.227	-0.83	0.41
$\beta_{\texttt{body style APV MPV Monovolume } Volkswagen}$	0.173	0.0927	1.86	0.06
$\beta_{\texttt{body style APV MPV Monovolume } Volvo}$	-0.308	0.163	-1.89	0.06
$\beta_{\texttt{body style Car Van } BMW}$	-0.261	0.242	-1.08	0.28
$\beta_{\texttt{body style Car Van } Citroen}$	-1.14	0.346	-3.29	0.00
$\beta_{\texttt{body style Car Van } Ford}$	-1.67	0.308	-5.41	0.00
$\beta_{\texttt{body style Car Van } Mercedes\text{-}Benz}$	-0.433	0.256	-1.69	0.09
$\beta_{\texttt{body style Car Van } Nissan}$	-0.898	0.316	-2.84	0.00
$\beta_{\texttt{body style Car Van } Opel}$	-1.34	0.330	-4.05	0.00
$\beta_{\texttt{body style Car Van } Other}$	-0.654	0.248	-2.64	0.01
$\beta_{\texttt{body style Car Van } Peugeot}$	-0.593	0.291	-2.04	0.04
$\beta_{\texttt{body style Car Van } Renault}$	-0.553	0.289	-1.91	0.06
$\beta_{\texttt{body style Car Van } Skoda}$	-1.05	0.390	-2.70	0.01
$\beta_{\texttt{body style Car Van } Volkswagen}$	-0.400	0.211	-1.90	0.06
$\beta_{\texttt{body style Car Van } Volvo}$	-0.632	0.357	-1.77	0.08
$\beta_{\texttt{body style Delivery Van } BMW}$	-0.218	0.230	-0.95	0.34
$\beta_{\texttt{body style Delivery Van } Citroen}$	-1.68	0.338	-4.96	0.00
$\beta_{\texttt{body style Delivery Van } Ford}$	-0.654	0.262	-2.50	0.01
$\beta_{\texttt{body style Delivery Van } Mercedes\text{-}Benz}$	0.315	0.214	1.47	0.14
$\beta_{\texttt{body style Delivery Van } Nissan}$	-0.837	0.270	-3.10	0.00
$\beta_{\texttt{body style Delivery Van } Opel}$	-0.900	0.300	-3.00	0.00
$\beta_{\texttt{body style Delivery Van } Other}$	-0.933	0.225	-4.15	0.00
$\beta_{\texttt{body style Delivery Van } Peugeot}$	-0.958	0.255	-3.76	0.00
$\beta_{\texttt{body style Delivery Van } Renault}$	-0.634	0.267	-2.38	0.02
$\beta_{\texttt{body style Delivery Van } Skoda}$	-0.654	0.347	-1.88	0.06
$\beta_{\texttt{body style Delivery Van } Volkswagen}$	-0.334	0.184	-1.81	0.07
$\beta_{\texttt{body style Delivery Van } Volvo}$	-0.0137	0.319	-0.04	0.97
$\beta_{\texttt{body style Hatchback } BMW}$	-0.0488	0.0742	-0.66	0.51
$\beta_{\texttt{body style Hatchback } Citroen}$	-0.403	0.247	-1.63	0.10
$\beta_{\texttt{body style Hatchback } Ford}$	-0.425	0.147	-2.89	0.00

$\beta_{\texttt{body style Hatchback}\ \textit{Mercedes-Benz}}$	-0.168	0.0840	-1.99	0.05
$\beta_{\texttt{body style Hatchback}\ \textit{Nissan}}$	-0.364	0.139	-2.61	0.01
$\beta_{\texttt{body style Hatchback}\ \textit{Opel}}$	-1.04	0.218	-4.77	0.00
$\beta_{\texttt{body style Hatchback}\ \textit{Other}}$	-0.341	0.106	-3.22	0.00
$\beta_{\texttt{body style Hatchback}\ \textit{Peugeot}}$	-0.222	0.192	-1.16	0.25
$\beta_{\texttt{body style Hatchback}\ \textit{Renault}}$	-0.171	0.177	-0.97	0.33
$\beta_{\texttt{body style Hatchback}\ \textit{Skoda}}$	-0.793	0.195	-4.07	0.00
$\beta_{\texttt{body style Hatchback}\ \textit{Volkswagen}}$	-0.138	0.0837	-1.64	0.10
$\beta_{\texttt{body style Hatchback}\ \textit{Volvo}}$	-0.125	0.151	-0.83	0.41
$\beta_{\texttt{body style Other}\ \textit{BMW}}$	-0.0507	0.0876	-0.58	0.56
$\beta_{\texttt{body style Other}\ \textit{Citroen}}$	-0.479	0.311	-1.54	0.12
$\beta_{\texttt{body style Other}\ \textit{Ford}}$	-0.359	0.187	-1.92	0.06
$\beta_{\texttt{body style Other}\ \textit{Mercedes-Benz}}$	-0.0756	0.0969	-0.78	0.44
$\beta_{\texttt{body style Other}\ \textit{Nissan}}$	-0.272	0.198	-1.37	0.17
$\beta_{\texttt{body style Other}\ \textit{Opel}}$	-0.380	0.221	-1.72	0.08
$\beta_{\texttt{body style Other}\ \textit{Other}}$	-0.230	0.139	-1.65	0.10
$\beta_{\texttt{body style Other}\ \textit{Peugeot}}$	-0.227	0.210	-1.08	0.28
$\beta_{\texttt{body style Other}\ \textit{Renault}}$	-0.194	0.220	-0.88	0.38
$\beta_{\texttt{body style Other}\ \textit{Skoda}}$	-0.196	0.207	-0.94	0.34
$\beta_{\texttt{body style Other}\ \textit{Volkswagen}}$	-0.0496	0.0989	-0.50	0.62
$\beta_{\texttt{body style Other}\ \textit{Volvo}}$	-0.0350	0.161	-0.22	0.83
$\beta_{\texttt{body style Sedan}\ \textit{BMW}}$	-0.0261	0.0760	-0.34	0.73
$\beta_{\texttt{body style Sedan}\ \textit{Citroen}}$	0.288	0.273	1.05	0.29
$\beta_{\texttt{body style Sedan}\ \textit{Ford}}$	0.0191	0.159	0.12	0.90
$\beta_{\texttt{body style Sedan}\ \textit{Mercedes-Benz}}$	-0.132	0.0846	-1.57	0.12
$\beta_{\texttt{body style Sedan}\ \textit{Nissan}}$	-0.0550	0.171	-0.32	0.75
$\beta_{\texttt{body style Sedan}\ \textit{Opel}}$	-0.646	0.219	-2.95	0.00
$\beta_{\texttt{body style Sedan}\ \textit{Other}}$	-0.0263	0.122	-0.22	0.83
$\beta_{\texttt{body style Sedan}\ \textit{Peugeot}}$	0.0540	0.198	0.27	0.78
$\beta_{\texttt{body style Sedan}\ \textit{Renault}}$	-0.114	0.201	-0.57	0.57
$\beta_{\texttt{body style Sedan}\ \textit{Skoda}}$	-0.219	0.197	-1.11	0.27
$\beta_{\texttt{body style Sedan}\ \textit{Volkswagen}}$	0.0439	0.0870	0.50	0.61
$\beta_{\texttt{body style Sedan}\ \textit{Volvo}}$	-0.0981	0.145	-0.68	0.50
$\beta_{\texttt{body style Stationwagon}\ \textit{BMW}}$	-0.103	0.0690	-1.49	0.14
$\beta_{\texttt{body style Stationwagon}\ \textit{Citroen}}$	-0.597	0.246	-2.42	0.02
$\beta_{\texttt{body style Stationwagon}\ \textit{Ford}}$	-0.0228	0.143	-0.16	0.87
$\beta_{\texttt{body style Stationwagon}\ \textit{Mercedes-Benz}}$	-0.149	0.0771	-1.94	0.05
$\beta_{\texttt{body style Stationwagon}\ \textit{Nissan}}$	-0.249	0.145	-1.72	0.09
$\beta_{\texttt{body style Stationwagon}\ \textit{Opel}}$	-0.640	0.212	-3.02	0.00
$\beta_{\texttt{body style Stationwagon}\ \textit{Other}}$	-0.107	0.102	-1.05	0.29
$\beta_{\texttt{body style Stationwagon}\ \textit{Peugeot}}$	0.0265	0.183	0.14	0.88
$\beta_{\texttt{body style Stationwagon}\ \textit{Renault}}$	-0.136	0.177	-0.77	0.44

$\beta_{\texttt{body style Stationwagon } \textit{Skoda}}$	-0.140	0.176	-0.80	0.43
$\beta_{\texttt{body style Stationwagon } \textit{Volkswagen}}$	0.165	0.0800	2.06	0.04
$\beta_{\texttt{body style Stationwagon } \textit{Volvo}}$	-0.0387	0.131	-0.30	0.77
$\beta_{\texttt{catalogue price } \textit{BMW}}$	0.269	0.0574	4.69	0.00
$\beta_{\texttt{catalogue price } \textit{Citroen}}$	-1.75	0.133	-13.23	0.00
$\beta_{\texttt{catalogue price } \textit{Ford}}$	-1.40	0.0955	-14.70	0.00
$\beta_{\texttt{catalogue price } \textit{Mercedes-Benz}}$	0.212	0.0606	3.50	0.00
$\beta_{\texttt{catalogue price } \textit{Nissan}}$	-0.852	0.181	-4.70	0.00
$\beta_{\texttt{catalogue price } \textit{Opel}}$	-1.75	0.113	-15.50	0.00
$\beta_{\texttt{catalogue price } \textit{Other}}$	-0.310	0.0762	-4.06	0.00
$\beta_{\texttt{catalogue price } \textit{Peugeot}}$	-1.52	0.0994	-15.32	0.00
$\beta_{\texttt{catalogue price } \textit{Renault}}$	-1.04	0.102	-10.21	0.00
$\beta_{\texttt{catalogue price } \textit{Skoda}}$	-1.14	0.119	-9.56	0.00
$\beta_{\texttt{catalogue price } \textit{Volkswagen}}$	-0.542	0.0637	-8.50	0.00
$\beta_{\texttt{catalogue price } \textit{Volvo}}$	-0.0690	0.155	-0.44	0.66
$\beta_{\texttt{client segment Corporate } \textit{BMW}}$	0.0314	0.0413	0.76	0.45
$\beta_{\texttt{client segment Corporate } \textit{Citroen}}$	0.190	0.0834	2.28	0.02
$\beta_{\texttt{client segment Corporate } \textit{Ford}}$	-0.123	0.0743	-1.65	0.10
$\beta_{\texttt{client segment Corporate } \textit{Mercedes-Benz}}$	0.00105	0.0475	0.02	0.98
$\beta_{\texttt{client segment Corporate } \textit{Nissan}}$	-0.594	0.100	-5.93	0.00
$\beta_{\texttt{client segment Corporate } \textit{Opel}}$	0.148	0.0750	1.98	0.05
$\beta_{\texttt{client segment Corporate } \textit{Other}}$	-0.161	0.0547	-2.95	0.00
$\beta_{\texttt{client segment Corporate } \textit{Peugeot}}$	0.346	0.0721	4.80	0.00
$\beta_{\texttt{client segment Corporate } \textit{Renault}}$	0.520	0.0688	7.55	0.00
$\beta_{\texttt{client segment Corporate } \textit{Skoda}}$	-0.289	0.0955	-3.03	0.00
$\beta_{\texttt{client segment Corporate } \textit{Volkswagen}}$	0.0271	0.0390	0.70	0.49
$\beta_{\texttt{client segment Corporate } \textit{Volvo}}$	-0.230	0.0716	-3.22	0.00
$\beta_{\texttt{client segment International } \textit{BMW}}$	0.211	0.0443	4.76	0.00
$\beta_{\texttt{client segment International } \textit{Citroen}}$	0.579	0.0793	7.30	0.00
$\beta_{\texttt{client segment International } \textit{Ford}}$	-0.155	0.0721	-2.15	0.03
$\beta_{\texttt{client segment International } \textit{Mercedes-Benz}}$	0.234	0.0510	4.58	0.00
$\beta_{\texttt{client segment International } \textit{Nissan}}$	-1.05	0.120	-8.79	0.00
$\beta_{\texttt{client segment International } \textit{Opel}}$	-0.0258	0.0808	-0.32	0.75
$\beta_{\texttt{client segment International } \textit{Other}}$	-0.638	0.0563	-11.33	0.00
$\beta_{\texttt{client segment International } \textit{Peugeot}}$	0.109	0.0615	1.77	0.08
$\beta_{\texttt{client segment International } \textit{Renault}}$	0.859	0.0681	12.61	0.00
$\beta_{\texttt{client segment International } \textit{Skoda}}$	-0.0749	0.0866	-0.86	0.39
$\beta_{\texttt{client segment International } \textit{Volkswagen}}$	0.0103	0.0396	0.26	0.80
$\beta_{\texttt{client segment International } \textit{Volvo}}$	-0.337	0.111	-3.04	0.00
$\beta_{\texttt{commercial discount amount } \textit{BMW}}$	0.00466	0.0134	0.35	0.73
$\beta_{\texttt{commercial discount amount } \textit{Citroen}}$	-0.0307	0.0270	-1.14	0.26
$\beta_{\texttt{commercial discount amount } \textit{Ford}}$	0.0397	0.0231	1.72	0.09

$\beta_{\texttt{commercial discount amount}\ \textit{Mercedes-Benz}}$	-0.0321	0.0144	-2.23	0.03
$\beta_{\texttt{commercial discount amount}\ \textit{Nissan}}$	0.0548	0.0246	2.22	0.03
$\beta_{\texttt{commercial discount amount}\ \textit{Opel}}$	-0.0221	0.0226	-0.98	0.33
$\beta_{\texttt{commercial discount amount}\ \textit{Other}}$	0.0116	0.0181	0.64	0.52
$\beta_{\texttt{commercial discount amount}\ \textit{Peugeot}}$	0.000337	0.0209	0.02	0.99
$\beta_{\texttt{commercial discount amount}\ \textit{Renault}}$	-0.0119	0.0258	-0.46	0.64
$\beta_{\texttt{commercial discount amount}\ \textit{Skoda}}$	-0.0128	0.0241	-0.53	0.60
$\beta_{\texttt{commercial discount amount}\ \textit{Volkswagen}}$	0.0470	0.0165	2.85	0.00
$\beta_{\texttt{commercial discount amount}\ \textit{Volvo}}$	-0.0650	0.0278	-2.34	0.02
$\beta_{\texttt{contract duration}\ \textit{BMW}}$	0.0132	0.0154	0.86	0.39
$\beta_{\texttt{contract duration}\ \textit{Citroen}}$	-0.190	0.0294	-6.45	0.00
$\beta_{\texttt{contract duration}\ \textit{Ford}}$	-0.0895	0.0277	-3.23	0.00
$\beta_{\texttt{contract duration}\ \textit{Mercedes-Benz}}$	-0.0893	0.0161	-5.55	0.00
$\beta_{\texttt{contract duration}\ \textit{Nissan}}$	-0.179	0.0301	-5.97	0.00
$\beta_{\texttt{contract duration}\ \textit{Opel}}$	-0.102	0.0296	-3.44	0.00
$\beta_{\texttt{contract duration}\ \textit{Other}}$	-0.165	0.0216	-7.65	0.00
$\beta_{\texttt{contract duration}\ \textit{Peugeot}}$	0.00325	0.0299	0.11	0.91
$\beta_{\texttt{contract duration}\ \textit{Renault}}$	-0.0541	0.0242	-2.24	0.03
$\beta_{\texttt{contract duration}\ \textit{Skoda}}$	0.123	0.0466	2.64	0.01
$\beta_{\texttt{contract duration}\ \textit{Volkswagen}}$	0.0395	0.0158	2.50	0.01
$\beta_{\texttt{contract duration}\ \textit{Volvo}}$	-0.0412	0.0406	-1.02	0.31
$\beta_{\texttt{country Belgium}\ \textit{BMW}}$	0.113	0.0601	1.88	0.06
$\beta_{\texttt{country Belgium}\ \textit{Citroen}}$	0.100	0.148	0.68	0.50
$\beta_{\texttt{country Belgium}\ \textit{Ford}}$	-0.763	0.116	-6.55	0.00
$\beta_{\texttt{country Belgium}\ \textit{Mercedes-Benz}}$	0.0332	0.0644	0.52	0.61
$\beta_{\texttt{country Belgium}\ \textit{Nissan}}$	-0.427	0.166	-2.58	0.01
$\beta_{\texttt{country Belgium}\ \textit{Opel}}$	-1.16	0.167	-6.95	0.00
$\beta_{\texttt{country Belgium}\ \textit{Other}}$	-0.142	0.0815	-1.75	0.08
$\beta_{\texttt{country Belgium}\ \textit{Peugeot}}$	-0.0608	0.202	-0.30	0.76
$\beta_{\texttt{country Belgium}\ \textit{Renault}}$	-0.343	0.111	-3.10	0.00
$\beta_{\texttt{country Belgium}\ \textit{Skoda}}$	-0.626	0.142	-4.42	0.00
$\beta_{\texttt{country Belgium}\ \textit{Volkswagen}}$	0.0468	0.0648	0.72	0.47
$\beta_{\texttt{country Belgium}\ \textit{Volvo}}$	0.345	0.108	3.20	0.00
$\beta_{\texttt{country France}\ \textit{BMW}}$	-0.461	0.0748	-6.15	0.00
$\beta_{\texttt{country France}\ \textit{Citroen}}$	1.50	0.152	9.86	0.00
$\beta_{\texttt{country France}\ \textit{Ford}}$	-0.114	0.118	-0.96	0.34
$\beta_{\texttt{country France}\ \textit{Mercedes-Benz}}$	-0.555	0.0837	-6.63	0.00
$\beta_{\texttt{country France}\ \textit{Nissan}}$	-0.161	0.159	-1.01	0.31
$\beta_{\texttt{country France}\ \textit{Opel}}$	-0.863	0.328	-2.63	0.01
$\beta_{\texttt{country France}\ \textit{Other}}$	-0.525	0.0934	-5.62	0.00
$\beta_{\texttt{country France}\ \textit{Peugeot}}$	2.22	0.428	5.19	0.00
$\beta_{\texttt{country France}\ \textit{Renault}}$	1.26	0.114	11.13	0.00

$\beta_{\texttt{country France}\ \textit{Skoda}}$	-1.14	0.309	-3.69	0.00
$\beta_{\texttt{country France}\ \textit{Volkswagen}}$	0.467	0.0714	6.55	0.00
$\beta_{\texttt{country France}\ \textit{Volvo}}$	-0.791	0.169	-4.69	0.00
$\beta_{\texttt{country Germany}\ \textit{BMW}}$	0.0452	0.0741	0.61	0.54
$\beta_{\texttt{country Germany}\ \textit{Citroen}}$	-0.175	0.198	-0.88	0.38
$\beta_{\texttt{country Germany}\ \textit{Ford}}$	0.617	0.236	2.61	0.01
$\beta_{\texttt{country Germany}\ \textit{Mercedes-Benz}}$	0.141	0.0764	1.85	0.06
$\beta_{\texttt{country Germany}\ \textit{Nissan}}$	-2.13	0.325	-6.56	0.00
$\beta_{\texttt{country Germany}\ \textit{Opel}}$	-0.728	0.226	-3.22	0.00
$\beta_{\texttt{country Germany}\ \textit{Other}}$	-1.54	0.123	-12.55	0.00
$\beta_{\texttt{country Germany}\ \textit{Peugeot}}$	-1.44	0.383	-3.76	0.00
$\beta_{\texttt{country Germany}\ \textit{Renault}}$	-1.57	0.169	-9.32	0.00
$\beta_{\texttt{country Germany}\ \textit{Skoda}}$	-0.855	0.200	-4.27	0.00
$\beta_{\texttt{country Germany}\ \textit{Volkswagen}}$	0.246	0.0747	3.28	0.00
$\beta_{\texttt{country Germany}\ \textit{Volvo}}$	-1.10	0.140	-7.86	0.00
$\beta_{\texttt{country Italy}\ \textit{BMW}}$	-0.0212	0.0690	-0.31	0.76
$\beta_{\texttt{country Italy}\ \textit{Citroen}}$	0.838	0.161	5.21	0.00
$\beta_{\texttt{country Italy}\ \textit{Ford}}$	0.458	0.136	3.38	0.00
$\beta_{\texttt{country Italy}\ \textit{Mercedes-Benz}}$	-0.393	0.0881	-4.46	0.00
$\beta_{\texttt{country Italy}\ \textit{Nissan}}$	0.625	0.131	4.77	0.00
$\beta_{\texttt{country Italy}\ \textit{Opel}}$	-0.381	0.170	-2.24	0.02
$\beta_{\texttt{country Italy}\ \textit{Other}}$	1.01	0.0935	10.80	0.00
$\beta_{\texttt{country Italy}\ \textit{Peugeot}}$	1.42	0.300	4.73	0.00
$\beta_{\texttt{country Italy}\ \textit{Renault}}$	0.277	0.124	2.24	0.03
$\beta_{\texttt{country Italy}\ \textit{Skoda}}$	-0.478	0.173	-2.76	0.01
$\beta_{\texttt{country Italy}\ \textit{Volkswagen}}$	0.355	0.0756	4.70	0.00
$\beta_{\texttt{country Italy}\ \textit{Volvo}}$	0.386	0.198	1.96	0.05
$\beta_{\texttt{country Norway}\ \textit{BMW}}$	-0.876	0.227	-3.86	0.00
$\beta_{\texttt{country Norway}\ \textit{Citroen}}$	6.33	0.534	11.87	0.00
$\beta_{\texttt{country Norway}\ \textit{Ford}}$	5.74	0.372	15.42	0.00
$\beta_{\texttt{country Norway}\ \textit{Mercedes-Benz}}$	-0.893	0.245	-3.65	0.00
$\beta_{\texttt{country Norway}\ \textit{Nissan}}$	3.52	0.735	4.80	0.00
$\beta_{\texttt{country Norway}\ \textit{Opel}}$	5.49	0.528	10.41	0.00
$\beta_{\texttt{country Norway}\ \textit{Other}}$	1.29	0.303	4.27	0.00
$\beta_{\texttt{country Norway}\ \textit{Peugeot}}$	7.39	0.484	15.27	0.00
$\beta_{\texttt{country Norway}\ \textit{Renault}}$	2.87	0.404	7.10	0.00
$\beta_{\texttt{country Norway}\ \textit{Skoda}}$	5.19	0.496	10.45	0.00
$\beta_{\texttt{country Norway}\ \textit{Volkswagen}}$	3.23	0.283	11.41	0.00
$\beta_{\texttt{country Norway}\ \textit{Volvo}}$	0.515	0.576	0.90	0.37
$\beta_{\texttt{country Other}\ \textit{BMW}}$	-0.208	0.0789	-2.64	0.01
$\beta_{\texttt{country Other}\ \textit{Citroen}}$	-2.37	0.589	-4.02	0.00
$\beta_{\texttt{country Other}\ \textit{Ford}}$	0.569	0.137	4.15	0.00

$\beta_{\texttt{country Other } \textit{Mercedes-Benz}}$	-0.466	0.0933	-5.00	0.00
$\beta_{\texttt{country Other } \textit{Nissan}}$	0.138	0.177	0.78	0.44
$\beta_{\texttt{country Other } \textit{Opel}}$	-0.580	0.320	-1.82	0.07
$\beta_{\texttt{country Other } \textit{Other}}$	0.233	0.108	2.15	0.03
$\beta_{\texttt{country Other } \textit{Peugeot}}$	0.748	0.201	3.71	0.00
$\beta_{\texttt{country Other } \textit{Renault}}$	-1.03	0.214	-4.82	0.00
$\beta_{\texttt{country Other } \textit{Skoda}}$	0.985	0.192	5.13	0.00
$\beta_{\texttt{country Other } \textit{Volkswagen}}$	0.936	0.0905	10.35	0.00
$\beta_{\texttt{country Other } \textit{Volvo}}$	0.638	0.130	4.90	0.00
$\beta_{\texttt{country Spain } \textit{BMW}}$	-0.299	0.101	-2.96	0.00
$\beta_{\texttt{country Spain } \textit{Citroen}}$	-0.417	0.201	-2.08	0.04
$\beta_{\texttt{country Spain } \textit{Ford}}$	-0.0956	0.135	-0.71	0.48
$\beta_{\texttt{country Spain } \textit{Mercedes-Benz}}$	-0.521	0.114	-4.57	0.00
$\beta_{\texttt{country Spain } \textit{Nissan}}$	0.497	0.166	2.99	0.00
$\beta_{\texttt{country Spain } \textit{Opel}}$	-0.176	0.188	-0.94	0.35
$\beta_{\texttt{country Spain } \textit{Other}}$	0.819	0.115	7.11	0.00
$\beta_{\texttt{country Spain } \textit{Peugeot}}$	1.62	0.332	4.89	0.00
$\beta_{\texttt{country Spain } \textit{Renault}}$	1.35	0.130	10.41	0.00
$\beta_{\texttt{country Spain } \textit{Skoda}}$	-1.00	0.268	-3.73	0.00
$\beta_{\texttt{country Spain } \textit{Volkswagen}}$	0.816	0.0973	8.39	0.00
$\beta_{\texttt{country Spain } \textit{Volvo}}$	0.186	0.218	0.86	0.39
$\beta_{\texttt{fuel type Diesel } \textit{BMW}}$	-0.0769	0.0539	-1.43	0.15
$\beta_{\texttt{fuel type Diesel } \textit{Citroen}}$	0.344	0.142	2.42	0.02
$\beta_{\texttt{fuel type Diesel } \textit{Ford}}$	0.277	0.0866	3.20	0.00
$\beta_{\texttt{fuel type Diesel } \textit{Mercedes-Benz}}$	-0.126	0.0582	-2.17	0.03
$\beta_{\texttt{fuel type Diesel } \textit{Nissan}}$	-0.385	0.0861	-4.47	0.00
$\beta_{\texttt{fuel type Diesel } \textit{Opel}}$	0.396	0.0886	4.47	0.00
$\beta_{\texttt{fuel type Diesel } \textit{Other}}$	-0.387	0.0657	-5.90	0.00
$\beta_{\texttt{fuel type Diesel } \textit{Peugeot}}$	0.829	0.150	5.54	0.00
$\beta_{\texttt{fuel type Diesel } \textit{Renault}}$	0.858	0.135	6.36	0.00
$\beta_{\texttt{fuel type Diesel } \textit{Skoda}}$	0.323	0.114	2.84	0.00
$\beta_{\texttt{fuel type Diesel } \textit{Volkswagen}}$	0.103	0.0572	1.81	0.07
$\beta_{\texttt{fuel type Diesel } \textit{Volvo}}$	-0.253	0.0980	-2.58	0.01
$\beta_{\texttt{lease type Financial lease } \textit{BMW}}$	-0.165	0.0803	-2.06	0.04
$\beta_{\texttt{lease type Financial lease } \textit{Citroen}}$	0.674	0.155	4.36	0.00
$\beta_{\texttt{lease type Financial lease } \textit{Ford}}$	0.738	0.136	5.44	0.00
$\beta_{\texttt{lease type Financial lease } \textit{Mercedes-Benz}}$	0.401	0.0728	5.50	0.00
$\beta_{\texttt{lease type Financial lease } \textit{Nissan}}$	-1.15	0.283	-4.07	0.00
$\beta_{\texttt{lease type Financial lease } \textit{Opel}}$	0.496	0.147	3.39	0.00
$\beta_{\texttt{lease type Financial lease } \textit{Other}}$	-0.814	0.140	-5.80	0.00
$\beta_{\texttt{lease type Financial lease } \textit{Peugeot}}$	-0.295	0.205	-1.44	0.15
$\beta_{\texttt{lease type Financial lease } \textit{Renault}}$	-0.353	0.173	-2.04	0.04

$\beta_{\texttt{lease type Financial lease}\ Skoda}$	0.261	0.195	1.34	0.18
$\beta_{\texttt{lease type Financial lease}\ Volkswagen}$	0.0438	0.0851	0.51	0.61
$\beta_{\texttt{lease type Financial lease}\ Volvo}$	-1.07	0.196	-5.43	0.00
$\beta_{\texttt{make Audi}\ BMW}$	-0.565	0.0732	-7.72	0.00
$\beta_{\texttt{make Audi}\ Citroen}$	-0.540	0.443	-1.22	0.22
$\beta_{\texttt{make Audi}\ Ford}$	-1.56	0.155	-10.06	0.00
$\beta_{\texttt{make Audi}\ Mercedes\text{-}Benz}$	-0.607	0.0767	-7.92	0.00
$\beta_{\texttt{make Audi}\ Nissan}$	-1.51	0.290	-5.22	0.00
$\beta_{\texttt{make Audi}\ Opel}$	-0.855	0.226	-3.79	0.00
$\beta_{\texttt{make Audi}\ Other}$	-1.55	0.220	-7.03	0.00
$\beta_{\texttt{make Audi}\ Peugeot}$	-0.729	0.230	-3.16	0.00
$\beta_{\texttt{make Audi}\ Renault}$	-1.91	0.188	-10.13	0.00
$\beta_{\texttt{make Audi}\ Skoda}$	-0.847	0.149	-5.69	0.00
$\beta_{\texttt{make Audi}\ Volkswagen}$	-0.768	0.0828	-9.28	0.00
$\beta_{\texttt{make Audi}\ Volvo}$	-2.69	0.131	-20.51	0.00
$\beta_{\texttt{make BMW}\ BMW}$	0.549	0.0727	7.55	0.00
$\beta_{\texttt{make BMW}\ Citroen}$	0.252	0.449	0.56	0.58
$\beta_{\texttt{make BMW}\ Ford}$	-0.913	0.192	-4.75	0.00
$\beta_{\texttt{make BMW}\ Mercedes\text{-}Benz}$	-0.0183	0.0706	-0.26	0.80
$\beta_{\texttt{make BMW}\ Nissan}$	-0.517	0.361	-1.43	0.15
$\beta_{\texttt{make BMW}\ Opel}$	0.640	0.293	2.18	0.03
$\beta_{\texttt{make BMW}\ Other}$	-0.981	0.214	-4.58	0.00
$\beta_{\texttt{make BMW}\ Peugeot}$	0.0373	0.227	0.16	0.87
$\beta_{\texttt{make BMW}\ Renault}$	-0.951	0.190	-5.00	0.00
$\beta_{\texttt{make BMW}\ Skoda}$	-0.406	0.164	-2.48	0.01
$\beta_{\texttt{make BMW}\ Volkswagen}$	-0.328	0.0745	-4.41	0.00
$\beta_{\texttt{make BMW}\ Volvo}$	-2.13	0.142	-15.01	0.00
$\beta_{\texttt{make Citroen}\ BMW}$	-0.0195	0.116	-0.17	0.87
$\beta_{\texttt{make Citroen}\ Citroen}$	3.27	0.428	7.66	0.00
$\beta_{\texttt{make Citroen}\ Ford}$	0.474	0.248	1.91	0.06
$\beta_{\texttt{make Citroen}\ Mercedes\text{-}Benz}$	0.160	0.121	1.32	0.19
$\beta_{\texttt{make Citroen}\ Nissan}$	0.616	0.356	1.73	0.08
$\beta_{\texttt{make Citroen}\ Opel}$	1.51	0.224	6.74	0.00
$\beta_{\texttt{make Citroen}\ Other}$	0.272	0.247	1.10	0.27
$\beta_{\texttt{make Citroen}\ Peugeot}$	2.08	0.271	7.68	0.00
$\beta_{\texttt{make Citroen}\ Renault}$	0.836	0.184	4.54	0.00
$\beta_{\texttt{make Citroen}\ Skoda}$	0.897	0.252	3.57	0.00
$\beta_{\texttt{make Citroen}\ Volkswagen}$	0.104	0.102	1.02	0.31
$\beta_{\texttt{make Citroen}\ Volvo}$	-1.09	0.205	-5.30	0.00
$\beta_{\texttt{make Ford}\ BMW}$	0.00275	0.0886	0.03	0.98
$\beta_{\texttt{make Ford}\ Citroen}$	2.08	0.427	4.88	0.00
$\beta_{\texttt{make Ford}\ Ford}$	2.51	0.174	14.44	0.00

$\beta_{\texttt{make Ford}}$ *Mercedes-Benz*	0.0924	0.0916	1.01	0.31
$\beta_{\texttt{make Ford}}$ *Nissan*	0.724	0.387	1.87	0.06
$\beta_{\texttt{make Ford}}$ *Opel*	1.35	0.260	5.21	0.00
$\beta_{\texttt{make Ford}}$ *Other*	0.150	0.231	0.65	0.51
$\beta_{\texttt{make Ford}}$ *Peugeot*	1.89	0.213	8.86	0.00
$\beta_{\texttt{make Ford}}$ *Renault*	0.456	0.179	2.55	0.01
$\beta_{\texttt{make Ford}}$ *Skoda*	0.784	0.369	2.13	0.03
$\beta_{\texttt{make Ford}}$ *Volkswagen*	0.131	0.0843	1.55	0.12
$\beta_{\texttt{make Ford}}$ *Volvo*	-1.19	0.200	-5.92	0.00
$\beta_{\texttt{make Mercedes-Benz}}$ *BMW*	0.227	0.0751	3.02	0.00
$\beta_{\texttt{make Mercedes-Benz}}$ *Citroen*	0.238	0.453	0.52	0.60
$\beta_{\texttt{make Mercedes-Benz}}$ *Ford*	-0.894	0.189	-4.74	0.00
$\beta_{\texttt{make Mercedes-Benz}}$ *Mercedes-Benz*	0.907	0.0989	9.18	0.00
$\beta_{\texttt{make Mercedes-Benz}}$ *Nissan*	-0.331	0.355	-0.93	0.35
$\beta_{\texttt{make Mercedes-Benz}}$ *Opel*	0.0843	0.245	0.34	0.73
$\beta_{\texttt{make Mercedes-Benz}}$ *Other*	-0.674	0.225	-3.00	0.00
$\beta_{\texttt{make Mercedes-Benz}}$ *Peugeot*	0.341	0.289	1.18	0.24
$\beta_{\texttt{make Mercedes-Benz}}$ *Renault*	-0.712	0.205	-3.48	0.00
$\beta_{\texttt{make Mercedes-Benz}}$ *Skoda*	-0.335	0.185	-1.80	0.07
$\beta_{\texttt{make Mercedes-Benz}}$ *Volkswagen*	0.0639	0.0793	0.81	0.42
$\beta_{\texttt{make Mercedes-Benz}}$ *Volvo*	-1.78	0.151	-11.82	0.00
$\beta_{\texttt{make Nissan}}$ *BMW*	0.0289	0.146	0.20	0.84
$\beta_{\texttt{make Nissan}}$ *Citroen*	1.25	0.485	2.57	0.01
$\beta_{\texttt{make Nissan}}$ *Ford*	0.677	0.208	3.25	0.00
$\beta_{\texttt{make Nissan}}$ *Mercedes-Benz*	0.0154	0.152	0.10	0.92
$\beta_{\texttt{make Nissan}}$ *Nissan*	2.04	0.497	4.10	0.00
$\beta_{\texttt{make Nissan}}$ *Opel*	1.25	0.314	3.99	0.00
$\beta_{\texttt{make Nissan}}$ *Other*	0.742	0.196	3.78	0.00
$\beta_{\texttt{make Nissan}}$ *Peugeot*	1.03	0.275	3.75	0.00
$\beta_{\texttt{make Nissan}}$ *Renault*	0.248	0.242	1.02	0.31
$\beta_{\texttt{make Nissan}}$ *Skoda*	0.767	0.270	2.84	0.00
$\beta_{\texttt{make Nissan}}$ *Volkswagen*	0.389	0.131	2.97	0.00
$\beta_{\texttt{make Nissan}}$ *Volvo*	-1.30	0.438	-2.97	0.00
$\beta_{\texttt{make Opel}}$ *BMW*	-0.0993	0.0800	-1.24	0.21
$\beta_{\texttt{make Opel}}$ *Citroen*	1.06	0.437	2.42	0.02
$\beta_{\texttt{make Opel}}$ *Ford*	0.285	0.287	0.99	0.32
$\beta_{\texttt{make Opel}}$ *Mercedes-Benz*	-0.273	0.0899	-3.04	0.00
$\beta_{\texttt{make Opel}}$ *Nissan*	0.196	0.403	0.49	0.63
$\beta_{\texttt{make Opel}}$ *Opel*	2.66	0.322	8.26	0.00
$\beta_{\texttt{make Opel}}$ *Other*	-0.419	0.235	-1.79	0.07
$\beta_{\texttt{make Opel}}$ *Peugeot*	1.56	0.212	7.33	0.00
$\beta_{\texttt{make Opel}}$ *Renault*	-0.00513	0.181	-0.03	0.98

$\beta_{\texttt{make Opel}\ Skoda}$	0.414	0.324	1.28	0.20
$\beta_{\texttt{make Opel}\ Volkswagen}$	-0.139	0.0801	-1.73	0.08
$\beta_{\texttt{make Opel}\ Volvo}$	-1.94	0.234	-8.30	0.00
$\beta_{\texttt{make Other}\ BMW}$	0.152	0.0795	1.92	0.06
$\beta_{\texttt{make Other}\ Citroen}$	1.43	0.428	3.35	0.00
$\beta_{\texttt{make Other}\ Ford}$	0.347	0.156	2.23	0.03
$\beta_{\texttt{make Other}\ Mercedes\text{-}Benz}$	0.0714	0.0863	0.83	0.41
$\beta_{\texttt{make Other}\ Nissan}$	0.855	0.263	3.25	0.00
$\beta_{\texttt{make Other}\ Opel}$	1.23	0.234	5.25	0.00
$\beta_{\texttt{make Other}\ Other}$	1.12	0.263	4.24	0.00
$\beta_{\texttt{make Other}\ Peugeot}$	1.02	0.230	4.43	0.00
$\beta_{\texttt{make Other}\ Renault}$	0.631	0.168	3.76	0.00
$\beta_{\texttt{make Other}\ Skoda}$	0.919	0.150	6.13	0.00
$\beta_{\texttt{make Other}\ Volkswagen}$	0.137	0.0801	1.71	0.09
$\beta_{\texttt{make Other}\ Volvo}$	-1.27	0.351	-3.62	0.00
$\beta_{\texttt{make Peugeot}\ BMW}$	-0.0967	0.0927	-1.04	0.30
$\beta_{\texttt{make Peugeot}\ Citroen}$	1.77	0.426	4.16	0.00
$\beta_{\texttt{make Peugeot}\ Ford}$	0.752	0.191	3.94	0.00
$\beta_{\texttt{make Peugeot}\ Mercedes\text{-}Benz}$	-0.182	0.102	-1.77	0.08
$\beta_{\texttt{make Peugeot}\ Nissan}$	0.279	0.372	0.75	0.45
$\beta_{\texttt{make Peugeot}\ Opel}$	1.17	0.211	5.56	0.00
$\beta_{\texttt{make Peugeot}\ Other}$	-0.193	0.241	-0.80	0.43
$\beta_{\texttt{make Peugeot}\ Peugeot}$	2.36	0.272	8.66	0.00
$\beta_{\texttt{make Peugeot}\ Renault}$	0.610	0.173	3.52	0.00
$\beta_{\texttt{make Peugeot}\ Skoda}$	0.642	0.260	2.47	0.01
$\beta_{\texttt{make Peugeot}\ Volkswagen}$	0.00464	0.0848	0.05	0.96
$\beta_{\texttt{make Peugeot}\ Volvo}$	-1.83	0.271	-6.77	0.00
$\beta_{\texttt{make Renault}\ BMW}$	-0.00197	0.0940	-0.02	0.98
$\beta_{\texttt{make Renault}\ Citroen}$	2.49	0.424	5.87	0.00
$\beta_{\texttt{make Renault}\ Ford}$	0.501	0.178	2.81	0.00
$\beta_{\texttt{make Renault}\ Mercedes\text{-}Benz}$	-0.00207	0.101	-0.02	0.98
$\beta_{\texttt{make Renault}\ Nissan}$	0.958	0.373	2.57	0.01
$\beta_{\texttt{make Renault}\ Opel}$	1.29	0.218	5.93	0.00
$\beta_{\texttt{make Renault}\ Other}$	0.412	0.233	1.77	0.08
$\beta_{\texttt{make Renault}\ Peugeot}$	1.55	0.263	5.90	0.00
$\beta_{\texttt{make Renault}\ Renault}$	1.99	0.170	11.74	0.00
$\beta_{\texttt{make Renault}\ Skoda}$	0.236	0.273	0.86	0.39
$\beta_{\texttt{make Renault}\ Volkswagen}$	0.0558	0.0871	0.64	0.52
$\beta_{\texttt{make Renault}\ Volvo}$	-0.959	0.220	-4.36	0.00
$\beta_{\texttt{make Skoda}\ BMW}$	-0.00327	0.117	-0.03	0.98
$\beta_{\texttt{make Skoda}\ Citroen}$	0.888	0.536	1.66	0.10
$\beta_{\texttt{make Skoda}\ Ford}$	0.195	0.322	0.60	0.55

$\beta_{\texttt{make Skoda}\ \textit{Mercedes-Benz}}$	0.120	0.120	1.00	0.32
$\beta_{\texttt{make Skoda}\ \textit{Nissan}}$	0.670	0.401	1.67	0.09
$\beta_{\texttt{make Skoda}\ \textit{Opel}}$	0.719	0.280	2.57	0.01
$\beta_{\texttt{make Skoda}\ \textit{Other}}$	0.127	0.252	0.51	0.61
$\beta_{\texttt{make Skoda}\ \textit{Peugeot}}$	0.700	0.280	2.50	0.01
$\beta_{\texttt{make Skoda}\ \textit{Renault}}$	-0.340	0.283	-1.20	0.23
$\beta_{\texttt{make Skoda}\ \textit{Skoda}}$	2.39	0.208	11.48	0.00
$\beta_{\texttt{make Skoda}\ \textit{Volkswagen}}$	0.148	0.106	1.40	0.16
$\beta_{\texttt{make Skoda}\ \textit{Volvo}}$	-1.47	0.236	-6.22	0.00
$\beta_{\texttt{make Volkswagen}\ \textit{BMW}}$	-0.115	0.0679	-1.69	0.09
$\beta_{\texttt{make Volkswagen}\ \textit{Citroen}}$	0.381	0.425	0.90	0.37
$\beta_{\texttt{make Volkswagen}\ \textit{Ford}}$	-0.399	0.137	-2.90	0.00
$\beta_{\texttt{make Volkswagen}\ \textit{Mercedes-Benz}}$	-0.0295	0.0725	-0.41	0.68
$\beta_{\texttt{make Volkswagen}\ \textit{Nissan}}$	-0.417	0.352	-1.19	0.24
$\beta_{\texttt{make Volkswagen}\ \textit{Opel}}$	0.109	0.202	0.54	0.59
$\beta_{\texttt{make Volkswagen}\ \textit{Other}}$	-0.879	0.216	-4.06	0.00
$\beta_{\texttt{make Volkswagen}\ \textit{Peugeot}}$	0.165	0.221	0.74	0.46
$\beta_{\texttt{make Volkswagen}\ \textit{Renault}}$	-0.808	0.172	-4.71	0.00
$\beta_{\texttt{make Volkswagen}\ \textit{Skoda}}$	0.167	0.141	1.19	0.24
$\beta_{\texttt{make Volkswagen}\ \textit{Volkswagen}}$	0.843	0.0864	9.75	0.00
$\beta_{\texttt{make Volkswagen}\ \textit{Volvo}}$	-2.03	0.131	-15.54	0.00
$\beta_{\texttt{mileage month}\ \textit{BMW}}$	-0.0309	0.0154	-2.01	0.04
$\beta_{\texttt{mileage month}\ \textit{Citroen}}$	0.0679	0.0296	2.29	0.02
$\beta_{\texttt{mileage month}\ \textit{Ford}}$	0.162	0.0333	4.86	0.00
$\beta_{\texttt{mileage month}\ \textit{Mercedes-Benz}}$	-0.0751	0.0161	-4.66	0.00
$\beta_{\texttt{mileage month}\ \textit{Nissan}}$	-0.0312	0.0297	-1.05	0.29
$\beta_{\texttt{mileage month}\ \textit{Opel}}$	0.146	0.0326	4.47	0.00
$\beta_{\texttt{mileage month}\ \textit{Other}}$	-0.0796	0.0225	-3.54	0.00
$\beta_{\texttt{mileage month}\ \textit{Peugeot}}$	0.212	0.0283	7.50	0.00
$\beta_{\texttt{mileage month}\ \textit{Renault}}$	0.140	0.0249	5.64	0.00
$\beta_{\texttt{mileage month}\ \textit{Skoda}}$	0.399	0.0473	8.43	0.00
$\beta_{\texttt{mileage month}\ \textit{Volkswagen}}$	0.130	0.0168	7.73	0.00
$\beta_{\texttt{mileage month}\ \textit{Volvo}}$	0.00708	0.0304	0.23	0.82
$\beta_{\texttt{standard discount percentage}\ \textit{BMW}}$	-0.0214	0.0309	-0.69	0.49
$\beta_{\texttt{standard discount percentage}\ \textit{Citroen}}$	0.198	0.0436	4.53	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Ford}}$	0.0956	0.0413	2.31	0.02
$\beta_{\texttt{standard discount percentage}\ \textit{Mercedes-Benz}}$	0.114	0.0317	3.59	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Nissan}}$	0.143	0.0529	2.70	0.01
$\beta_{\texttt{standard discount percentage}\ \textit{Opel}}$	0.238	0.0525	4.53	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Other}}$	0.118	0.0362	3.27	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Peugeot}}$	0.102	0.0383	2.65	0.01
$\beta_{\texttt{standard discount percentage}\ \textit{Renault}}$	0.134	0.0369	3.64	0.00

$\beta_{\texttt{standard discount percentage } Skoda}$	0.370	0.0797	4.64	0.00
$\beta_{\texttt{standard discount percentage } Volkswagen}$	0.117	0.0294	3.99	0.00
$\beta_{\texttt{standard discount percentage } Volvo}$	0.182	0.0547	3.32	0.00
$\beta_{\texttt{switch quarter 1 } BMW}$	0.0280	0.0331	0.84	0.40
$\beta_{\texttt{switch quarter 1 } Citroen}$	-0.337	0.0699	-4.82	0.00
$\beta_{\texttt{switch quarter 1 } Ford}$	-0.126	0.0535	-2.36	0.02
$\beta_{\texttt{switch quarter 1 } Mercedes\text{-}Benz}$	-0.0806	0.0364	-2.21	0.03
$\beta_{\texttt{switch quarter 1 } Nissan}$	0.00539	0.0673	0.08	0.94
$\beta_{\texttt{switch quarter 1 } Opel}$	-0.0562	0.0592	-0.95	0.34
$\beta_{\texttt{switch quarter 1 } Other}$	-0.0167	0.0478	-0.35	0.73
$\beta_{\texttt{switch quarter 1 } Peugeot}$	-0.211	0.0576	-3.67	0.00
$\beta_{\texttt{switch quarter 1 } Renault}$	-0.111	0.0541	-2.05	0.04
$\beta_{\texttt{switch quarter 1 } Skoda}$	0.0939	0.0747	1.26	0.21
$\beta_{\texttt{switch quarter 1 } Volkswagen}$	-0.110	0.0342	-3.22	0.00
$\beta_{\texttt{switch quarter 1 } Volvo}$	-0.00556	0.0615	-0.09	0.93
$\beta_{\texttt{switch quarter 2 } BMW}$	0.0112	0.0320	0.35	0.73
$\beta_{\texttt{switch quarter 2 } Citroen}$	-0.151	0.0670	-2.26	0.02
$\beta_{\texttt{switch quarter 2 } Ford}$	-0.0976	0.0529	-1.84	0.07
$\beta_{\texttt{switch quarter 2 } Mercedes\text{-}Benz}$	-0.0491	0.0343	-1.43	0.15
$\beta_{\texttt{switch quarter 2 } Nissan}$	-0.0156	0.0750	-0.21	0.84
$\beta_{\texttt{switch quarter 2 } Opel}$	0.00980	0.0606	0.16	0.87
$\beta_{\texttt{switch quarter 2 } Other}$	-0.205	0.0498	-4.10	0.00
$\beta_{\texttt{switch quarter 2 } Peugeot}$	-0.218	0.0580	-3.76	0.00
$\beta_{\texttt{switch quarter 2 } Renault}$	-0.320	0.0559	-5.73	0.00
$\beta_{\texttt{switch quarter 2 } Skoda}$	0.0428	0.0701	0.61	0.54
$\beta_{\texttt{switch quarter 2 } Volkswagen}$	-0.133	0.0334	-4.00	0.00
$\beta_{\texttt{switch quarter 2 } Volvo}$	-0.0737	0.0620	-1.19	0.23
$\beta_{\texttt{switch quarter 3 } BMW}$	-0.0516	0.0325	-1.59	0.11
$\beta_{\texttt{switch quarter 3 } Citroen}$	-0.251	0.0685	-3.67	0.00
$\beta_{\texttt{switch quarter 3 } Ford}$	-0.102	0.0514	-1.99	0.05
$\beta_{\texttt{switch quarter 3 } Mercedes\text{-}Benz}$	-0.0192	0.0339	-0.57	0.57
$\beta_{\texttt{switch quarter 3 } Nissan}$	-0.0927	0.0641	-1.45	0.15
$\beta_{\texttt{switch quarter 3 } Opel}$	0.0291	0.0653	0.45	0.66
$\beta_{\texttt{switch quarter 3 } Other}$	-0.0676	0.0459	-1.47	0.14
$\beta_{\texttt{switch quarter 3 } Peugeot}$	-0.200	0.0603	-3.32	0.00
$\beta_{\texttt{switch quarter 3 } Renault}$	-0.326	0.0545	-5.99	0.00
$\beta_{\texttt{switch quarter 3 } Skoda}$	-0.112	0.0687	-1.63	0.10
$\beta_{\texttt{switch quarter 3 } Volkswagen}$	-0.178	0.0345	-5.15	0.00
$\beta_{\texttt{switch quarter 3 } Volvo}$	-0.172	0.0634	-2.71	0.01
$\beta_{\texttt{total accessories amount } BMW}$	-0.0566	0.0136	-4.17	0.00
$\beta_{\texttt{total accessories amount } Citroen}$	0.101	0.0287	3.51	0.00
$\beta_{\texttt{total accessories amount } Ford}$	0.113	0.0213	5.32	0.00

$\beta_{\texttt{total accessories amount}\ \textit{Mercedes-Benz}}$	-0.0501	0.0142	-3.54	0.00
$\beta_{\texttt{total accessories amount}\ \textit{Nissan}}$	-0.0357	0.0304	-1.17	0.24
$\beta_{\texttt{total accessories amount}\ \textit{Opel}}$	0.149	0.0268	5.54	0.00
$\beta_{\texttt{total accessories amount}\ \textit{Other}}$	-0.133	0.0217	-6.15	0.00
$\beta_{\texttt{total accessories amount}\ \textit{Peugeot}}$	-0.0195	0.0304	-0.64	0.52
$\beta_{\texttt{total accessories amount}\ \textit{Renault}}$	0.0893	0.0230	3.88	0.00
$\beta_{\texttt{total accessories amount}\ \textit{Skoda}}$	0.0736	0.0290	2.54	0.01
$\beta_{\texttt{total accessories amount}\ \textit{Volkswagen}}$	-0.0700	0.0138	-5.06	0.00
$\beta_{\texttt{total accessories amount}\ \textit{Volvo}}$	-0.0406	0.0285	-1.43	0.15
$\beta_{\texttt{total options amount}\ \textit{BMW}}$	-0.0453	0.0173	-2.62	0.01
$\beta_{\texttt{total options amount}\ \textit{Citroen}}$	-0.159	0.0279	-5.69	0.00
$\beta_{\texttt{total options amount}\ \textit{Ford}}$	-0.116	0.0269	-4.32	0.00
$\beta_{\texttt{total options amount}\ \textit{Mercedes-Benz}}$	-0.00824	0.0192	-0.43	0.67
$\beta_{\texttt{total options amount}\ \textit{Nissan}}$	-0.0500	0.0272	-1.84	0.07
$\beta_{\texttt{total options amount}\ \textit{Opel}}$	-0.0524	0.0251	-2.09	0.04
$\beta_{\texttt{total options amount}\ \textit{Other}}$	-0.0567	0.0210	-2.71	0.01
$\beta_{\texttt{total options amount}\ \textit{Peugeot}}$	-0.0212	0.0236	-0.90	0.37
$\beta_{\texttt{total options amount}\ \textit{Renault}}$	0.0131	0.0247	0.53	0.60
$\beta_{\texttt{total options amount}\ \textit{Skoda}}$	-0.109	0.0321	-3.40	0.00
$\beta_{\texttt{total options amount}\ \textit{Volkswagen}}$	-0.0824	0.0175	-4.72	0.00
$\beta_{\texttt{total options amount}\ \textit{Volvo}}$	-0.0408	0.0279	-1.46	0.14
$\beta_{\texttt{ufwt amount}\ \textit{BMW}}$	-0.0328	0.0136	-2.41	0.02
$\beta_{\texttt{ufwt amount}\ \textit{Citroen}}$	0.0559	0.0274	2.04	0.04
$\beta_{\texttt{ufwt amount}\ \textit{Ford}}$	-0.0689	0.0212	-3.25	0.00
$\beta_{\texttt{ufwt amount}\ \textit{Mercedes-Benz}}$	-0.0755	0.0152	-4.96	0.00
$\beta_{\texttt{ufwt amount}\ \textit{Nissan}}$	-0.0947	0.0251	-3.78	0.00
$\beta_{\texttt{ufwt amount}\ \textit{Opel}}$	-0.101	0.0231	-4.35	0.00
$\beta_{\texttt{ufwt amount}\ \textit{Other}}$	-0.101	0.0194	-5.19	0.00
$\beta_{\texttt{ufwt amount}\ \textit{Peugeot}}$	-0.0930	0.0258	-3.61	0.00
$\beta_{\texttt{ufwt amount}\ \textit{Renault}}$	0.0610	0.0225	2.71	0.01
$\beta_{\texttt{ufwt amount}\ \textit{Skoda}}$	0.0369	0.0362	1.02	0.31
$\beta_{\texttt{ufwt amount}\ \textit{Volkswagen}}$	-0.0153	0.0132	-1.16	0.25
$\beta_{\texttt{ufwt amount}\ \textit{Volvo}}$	0.00821	0.0330	0.25	0.80
$\beta_{\texttt{vehicle segment B}\ \textit{BMW}}$	-0.192	0.103	-1.86	0.06
$\beta_{\texttt{vehicle segment B}\ \textit{Citroen}}$	0.798	0.237	3.37	0.00
$\beta_{\texttt{vehicle segment B}\ \textit{Ford}}$	0.464	0.152	3.04	0.00
$\beta_{\texttt{vehicle segment B}\ \textit{Mercedes-Benz}}$	-0.0980	0.112	-0.88	0.38
$\beta_{\texttt{vehicle segment B}\ \textit{Nissan}}$	-0.0751	0.196	-0.38	0.70
$\beta_{\texttt{vehicle segment B}\ \textit{Opel}}$	0.668	0.207	3.22	0.00
$\beta_{\texttt{vehicle segment B}\ \textit{Other}}$	0.466	0.131	3.54	0.00
$\beta_{\texttt{vehicle segment B}\ \textit{Peugeot}}$	0.173	0.178	0.97	0.33
$\beta_{\texttt{vehicle segment B}\ \textit{Renault}}$	1.36	0.178	7.65	0.00

$\beta_{\texttt{vehicle segment B}\ \textit{Skoda}}$	0.0861	0.267	0.32	0.75
$\beta_{\texttt{vehicle segment B}\ \textit{Volkswagen}}$	-0.0529	0.0982	-0.54	0.59
$\beta_{\texttt{vehicle segment B}\ \textit{Volvo}}$	-0.415	0.248	-1.67	0.09
$\beta_{\texttt{vehicle segment C}\ \textit{BMW}}$	-0.119	0.0665	-1.78	0.07
$\beta_{\texttt{vehicle segment C}\ \textit{Citroen}}$	0.476	0.211	2.25	0.02
$\beta_{\texttt{vehicle segment C}\ \textit{Ford}}$	0.516	0.115	4.49	0.00
$\beta_{\texttt{vehicle segment C}\ \textit{Mercedes-Benz}}$	-0.111	0.0730	-1.52	0.13
$\beta_{\texttt{vehicle segment C}\ \textit{Nissan}}$	-0.169	0.131	-1.29	0.20
$\beta_{\texttt{vehicle segment C}\ \textit{Opel}}$	1.11	0.197	5.62	0.00
$\beta_{\texttt{vehicle segment C}\ \textit{Other}}$	-0.0490	0.103	-0.48	0.63
$\beta_{\texttt{vehicle segment C}\ \textit{Peugeot}}$	0.343	0.174	1.98	0.05
$\beta_{\texttt{vehicle segment C}\ \textit{Renault}}$	0.243	0.158	1.54	0.12
$\beta_{\texttt{vehicle segment C}\ \textit{Skoda}}$	0.637	0.157	4.05	0.00
$\beta_{\texttt{vehicle segment C}\ \textit{Volkswagen}}$	0.0810	0.0722	1.12	0.26
$\beta_{\texttt{vehicle segment C}\ \textit{Volvo}}$	-0.541	0.161	-3.36	0.00
$\beta_{\texttt{vehicle segment D}\ \textit{BMW}}$	-0.241	0.0584	-4.12	0.00
$\beta_{\texttt{vehicle segment D}\ \textit{Citroen}}$	0.318	0.224	1.42	0.16
$\beta_{\texttt{vehicle segment D}\ \textit{Ford}}$	0.172	0.114	1.51	0.13
$\beta_{\texttt{vehicle segment D}\ \textit{Mercedes-Benz}}$	-0.183	0.0630	-2.91	0.00
$\beta_{\texttt{vehicle segment D}\ \textit{Nissan}}$	-0.301	0.138	-2.18	0.03
$\beta_{\texttt{vehicle segment D}\ \textit{Opel}}$	0.893	0.193	4.63	0.00
$\beta_{\texttt{vehicle segment D}\ \textit{Other}}$	-0.356	0.0911	-3.90	0.00
$\beta_{\texttt{vehicle segment D}\ \textit{Peugeot}}$	0.132	0.166	0.80	0.43
$\beta_{\texttt{vehicle segment D}\ \textit{Renault}}$	0.0822	0.160	0.52	0.61
$\beta_{\texttt{vehicle segment D}\ \textit{Skoda}}$	0.409	0.150	2.73	0.01
$\beta_{\texttt{vehicle segment D}\ \textit{Volkswagen}}$	0.0453	0.0670	0.68	0.50
$\beta_{\texttt{vehicle segment D}\ \textit{Volvo}}$	-0.509	0.122	-4.17	0.00
$\beta_{\texttt{vehicle segment E}\ \textit{BMW}}$	-0.298	0.0649	-4.60	0.00
$\beta_{\texttt{vehicle segment E}\ \textit{Citroen}}$	-0.00500	0.419	-0.01	0.99
$\beta_{\texttt{vehicle segment E}\ \textit{Ford}}$	-0.0318	0.163	-0.19	0.85
$\beta_{\texttt{vehicle segment E}\ \textit{Mercedes-Benz}}$	-0.256	0.0701	-3.65	0.00
$\beta_{\texttt{vehicle segment E}\ \textit{Nissan}}$	-0.298	0.211	-1.41	0.16
$\beta_{\texttt{vehicle segment E}\ \textit{Opel}}$	0.961	0.249	3.85	0.00
$\beta_{\texttt{vehicle segment E}\ \textit{Other}}$	-0.242	0.113	-2.14	0.03
$\beta_{\texttt{vehicle segment E}\ \textit{Peugeot}}$	-0.0347	0.278	-0.12	0.90
$\beta_{\texttt{vehicle segment E}\ \textit{Renault}}$	0.285	0.230	1.24	0.21
$\beta_{\texttt{vehicle segment E}\ \textit{Skoda}}$	0.206	0.200	1.03	0.30
$\beta_{\texttt{vehicle segment E}\ \textit{Volkswagen}}$	-0.123	0.0834	-1.47	0.14
$\beta_{\texttt{vehicle segment E}\ \textit{Volvo}}$	-0.372	0.150	-2.47	0.01
$\beta_{\texttt{vehicle segment LCV}\ \textit{BMW}}$	-0.0980	0.215	-0.46	0.65
$\beta_{\texttt{vehicle segment LCV}\ \textit{Citroen}}$	2.10	0.284	7.39	0.00
$\beta_{\texttt{vehicle segment LCV}\ \textit{Ford}}$	1.39	0.225	6.16	0.00

$\beta_{\texttt{vehicle segment LCV } \textit{Mercedes-Benz}}$	0.591	0.183	3.23	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{Nissan}}$	0.508	0.281	1.81	0.07
$\beta_{\texttt{vehicle segment LCV } \textit{Opel}}$	1.68	0.306	5.48	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{Other}}$	0.276	0.208	1.33	0.18
$\beta_{\texttt{vehicle segment LCV } \textit{Peugeot}}$	0.970	0.223	4.35	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{Renault}}$	1.78	0.236	7.53	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{Skoda}}$	0.881	0.325	2.71	0.01
$\beta_{\texttt{vehicle segment LCV } \textit{Volkswagen}}$	0.630	0.165	3.81	0.00
$\beta_{\texttt{vehicle segment LCV } \textit{Volvo}}$	0.172	0.254	0.68	0.50
$\beta_{\texttt{vehicle segment MPV } \textit{BMW}}$	0.131	0.0820	1.60	0.11
$\beta_{\texttt{vehicle segment MPV } \textit{Citroen}}$	0.334	0.184	1.81	0.07
$\beta_{\texttt{vehicle segment MPV } \textit{Ford}}$	-0.100	0.121	-0.83	0.41
$\beta_{\texttt{vehicle segment MPV } \textit{Mercedes-Benz}}$	0.00561	0.0891	0.06	0.95
$\beta_{\texttt{vehicle segment MPV } \textit{Nissan}}$	-0.0481	0.159	-0.30	0.76
$\beta_{\texttt{vehicle segment MPV } \textit{Opel}}$	0.487	0.213	2.28	0.02
$\beta_{\texttt{vehicle segment MPV } \textit{Other}}$	-0.0523	0.120	-0.44	0.66
$\beta_{\texttt{vehicle segment MPV } \textit{Peugeot}}$	-0.237	0.126	-1.87	0.06
$\beta_{\texttt{vehicle segment MPV } \textit{Renault}}$	0.349	0.162	2.15	0.03
$\beta_{\texttt{vehicle segment MPV } \textit{Skoda}}$	0.451	0.187	2.42	0.02
$\beta_{\texttt{vehicle segment MPV } \textit{Volkswagen}}$	0.177	0.0805	2.20	0.03
$\beta_{\texttt{vehicle segment MPV } \textit{Volvo}}$	-0.270	0.147	-1.84	0.07
$\beta_{\texttt{vehicle segment Other } \textit{BMW}}$	-0.0466	0.156	-0.30	0.76
$\beta_{\texttt{vehicle segment Other } \textit{Citroen}}$	0.446	0.300	1.49	0.14
$\beta_{\texttt{vehicle segment Other } \textit{Ford}}$	0.660	0.254	2.60	0.01
$\beta_{\texttt{vehicle segment Other } \textit{Mercedes-Benz}}$	0.185	0.164	1.13	0.26
$\beta_{\texttt{vehicle segment Other } \textit{Nissan}}$	-0.209	0.256	-0.82	0.41
$\beta_{\texttt{vehicle segment Other } \textit{Opel}}$	0.584	0.294	1.98	0.05
$\beta_{\texttt{vehicle segment Other } \textit{Other}}$	0.150	0.189	0.79	0.43
$\beta_{\texttt{vehicle segment Other } \textit{Peugeot}}$	0.244	0.250	0.98	0.33
$\beta_{\texttt{vehicle segment Other } \textit{Renault}}$	0.372	0.271	1.37	0.17
$\beta_{\texttt{vehicle segment Other } \textit{Skoda}}$	-0.0615	0.399	-0.15	0.88
$\beta_{\texttt{vehicle segment Other } \textit{Volkswagen}}$	0.123	0.164	0.75	0.46
$\beta_{\texttt{vehicle segment Other } \textit{Volvo}}$	-0.607	0.370	-1.64	0.10
$\beta_{\texttt{vehicle type Car } \textit{BMW}}$	-0.225	0.209	-1.08	0.28
$\beta_{\texttt{vehicle type Car } \textit{Citroen}}$	-1.67	0.189	-8.82	0.00
$\beta_{\texttt{vehicle type Car } \textit{Ford}}$	-0.781	0.207	-3.77	0.00
$\beta_{\texttt{vehicle type Car } \textit{Mercedes-Benz}}$	0.0555	0.183	0.30	0.76
$\beta_{\texttt{vehicle type Car } \textit{Nissan}}$	-0.812	0.218	-3.73	0.00
$\beta_{\texttt{vehicle type Car } \textit{Opel}}$	-0.213	0.289	-0.74	0.46
$\beta_{\texttt{vehicle type Car } \textit{Other}}$	-1.08	0.191	-5.65	0.00
$\beta_{\texttt{vehicle type Car } \textit{Peugeot}}$	-1.34	0.185	-7.26	0.00
$\beta_{\texttt{vehicle type Car } \textit{Renault}}$	-1.36	0.182	-7.48	0.00

$\beta_{\texttt{vehicle type Car}\ \textit{Skoda}}$	0.671	0.527	1.27	0.20
$\beta_{\texttt{vehicle type Car}\ \textit{Volkswagen}}$	-0.745	0.171	-4.36	0.00
$\beta_{\texttt{vehicle type Car}\ \textit{Volvo}}$	0.0799	0.351	0.23	0.82

Table B.2: Overview of all estimated parameters for the nested multinomial logistic regression model. Each parameter name (β) is structured as follows. The first part, stated in `font`, is considered the explanatory variable. The second part of the name, stated in italics, is considered an alternative of the choice set.

Parameter	**Estimate**	**Std. Error**	***t*-stat**	***p*-value**
ASC_{BMW}	0.583	0.276	2.11	0.03
$\text{ASC}_{Citroen}$	-1.95	0.318	-6.12	0.00
ASC_{Ford}	-0.243	0.250	-0.97	0.33
$\text{ASC}_{Mercedes\text{-}Benz}$	0.448	0.249	1.80	0.07
ASC_{Nissan}	0.487	0.318	1.53	0.13
ASC_{Opel}	-0.981	0.269	-3.64	0.00
ASC_{Other}	1.62	0.232	7.00	0.00
$\text{ASC}_{Peugeot}$	-1.81	0.316	-5.73	0.00
$\text{ASC}_{Renault}$	-1.26	0.382	-3.29	0.00
ASC_{Skoda}	-1.16	0.292	-3.97	0.00
$\text{ASC}_{Volkswagen}$	-0.0609	0.256	-0.24	0.81
ASC_{Volvo}	1.14	0.346	3.30	0.00
$\beta_{\texttt{body style APV MPV Monovolume}\ \textit{BMW}}$	-0.106	0.0980	-1.08	0.28
$\beta_{\texttt{body style APV MPV Monovolume}\ \textit{Citroen}}$	0.298	0.156	1.91	0.06
$\beta_{\texttt{body style APV MPV Monovolume}\ \textit{Ford}}$	0.294	0.120	2.46	0.01
$\beta_{\texttt{body style APV MPV Monovolume}\ \textit{Mercedes-Benz}}$	-0.302	0.106	-2.86	0.00
$\beta_{\texttt{body style APV MPV Monovolume}\ \textit{Nissan}}$	-0.156	0.138	-1.13	0.26
$\beta_{\texttt{body style APV MPV Monovolume}\ \textit{Opel}}$	0.0133	0.140	0.09	0.92
$\beta_{\texttt{body style APV MPV Monovolume}\ \textit{Other}}$	-0.401	0.111	-3.61	0.00
$\beta_{\texttt{body style APV MPV Monovolume}\ \textit{Peugeot}}$	0.241	0.105	2.28	0.02
$\beta_{\texttt{body style APV MPV Monovolume}\ \textit{Renault}}$	-0.108	0.142	-0.76	0.45
$\beta_{\texttt{body style APV MPV Monovolume}\ \textit{Skoda}}$	-0.0639	0.140	-0.46	0.65
$\beta_{\texttt{body style APV MPV Monovolume}\ \textit{Volkswagen}}$	0.244	0.105	2.33	0.02
$\beta_{\texttt{body style APV MPV Monovolume}\ \textit{Volvo}}$	-0.161	0.142	-1.13	0.26
$\beta_{\texttt{body style Car Van}\ \textit{BMW}}$	-0.287	0.311	-0.92	0.36
$\beta_{\texttt{body style Car Van}\ \textit{Citroen}}$	-0.663	0.309	-2.15	0.03
$\beta_{\texttt{body style Car Van}\ \textit{Ford}}$	-1.32	0.278	-4.75	0.00
$\beta_{\texttt{body style Car Van}\ \textit{Mercedes-Benz}}$	-0.469	0.307	-1.53	0.13
$\beta_{\texttt{body style Car Van}\ \textit{Nissan}}$	-0.508	0.297	-1.71	0.09
$\beta_{\texttt{body style Car Van}\ \textit{Opel}}$	-0.802	0.285	-2.82	0.00
$\beta_{\texttt{body style Car Van}\ \textit{Other}}$	-0.448	0.258	-1.74	0.08
$\beta_{\texttt{body style Car Van}\ \textit{Peugeot}}$	-0.429	0.268	-1.60	0.11

$\beta_{\texttt{body style Car Van}\ \textit{Renault}}$	-0.400	0.283	-1.41	0.16
$\beta_{\texttt{body style Car Van}\ \textit{Skoda}}$	-0.467	0.302	-1.55	0.12
$\beta_{\texttt{body style Car Van}\ \textit{Volkswagen}}$	-0.410	0.249	-1.65	0.10
$\beta_{\texttt{body style Car Van}\ \textit{Volvo}}$	-0.435	0.368	-1.18	0.24
$\beta_{\texttt{body style Delivery Van}\ \textit{BMW}}$	-0.253	0.250	-1.01	0.31
$\beta_{\texttt{body style Delivery Van}\ \textit{Citroen}}$	-1.28	0.288	-4.46	0.00
$\beta_{\texttt{body style Delivery Van}\ \textit{Ford}}$	-0.599	0.235	-2.55	0.01
$\beta_{\texttt{body style Delivery Van}\ \textit{Mercedes-Benz}}$	0.188	0.237	0.79	0.43
$\beta_{\texttt{body style Delivery Van}\ \textit{Nissan}}$	-0.703	0.260	-2.70	0.01
$\beta_{\texttt{body style Delivery Van}\ \textit{Opel}}$	-0.617	0.245	-2.52	0.01
$\beta_{\texttt{body style Delivery Van}\ \textit{Other}}$	-0.908	0.226	-4.02	0.00
$\beta_{\texttt{body style Delivery Van}\ \textit{Peugeot}}$	-0.871	0.236	-3.70	0.00
$\beta_{\texttt{body style Delivery Van}\ \textit{Renault}}$	-0.634	0.257	-2.47	0.01
$\beta_{\texttt{body style Delivery Van}\ \textit{Skoda}}$	-0.453	0.272	-1.67	0.09
$\beta_{\texttt{body style Delivery Van}\ \textit{Volkswagen}}$	-0.446	0.217	-2.05	0.04
$\beta_{\texttt{body style Delivery Van}\ \textit{Volvo}}$	0.0937	0.274	0.34	0.73
$\beta_{\texttt{body style Hatchback}\ \textit{BMW}}$	-0.0871	0.0801	-1.09	0.28
$\beta_{\texttt{body style Hatchback}\ \textit{Citroen}}$	-0.279	0.177	-1.58	0.11
$\beta_{\texttt{body style Hatchback}\ \textit{Ford}}$	-0.332	0.118	-2.81	0.00
$\beta_{\texttt{body style Hatchback}\ \textit{Mercedes-Benz}}$	-0.204	0.0861	-2.37	0.02
$\beta_{\texttt{body style Hatchback}\ \textit{Nissan}}$	-0.216	0.121	-1.78	0.07
$\beta_{\texttt{body style Hatchback}\ \textit{Opel}}$	-0.610	0.134	-4.56	0.00
$\beta_{\texttt{body style Hatchback}\ \textit{Other}}$	-0.256	0.0950	-2.69	0.01
$\beta_{\texttt{body style Hatchback}\ \textit{Peugeot}}$	-0.145	0.125	-1.17	0.24
$\beta_{\texttt{body style Hatchback}\ \textit{Renault}}$	-0.139	0.153	-0.91	0.36
$\beta_{\texttt{body style Hatchback}\ \textit{Skoda}}$	-0.458	0.132	-3.48	0.00
$\beta_{\texttt{body style Hatchback}\ \textit{Volkswagen}}$	-0.153	0.0996	-1.54	0.12
$\beta_{\texttt{body style Hatchback}\ \textit{Volvo}}$	-0.0306	0.116	-0.26	0.79
$\beta_{\texttt{body style Other}\ \textit{BMW}}$	-0.0759	0.0893	-0.85	0.40
$\beta_{\texttt{body style Other}\ \textit{Citroen}}$	-0.281	0.223	-1.26	0.21
$\beta_{\texttt{body style Other}\ \textit{Ford}}$	-0.320	0.148	-2.16	0.03
$\beta_{\texttt{body style Other}\ \textit{Mercedes-Benz}}$	-0.134	0.0959	-1.40	0.16
$\beta_{\texttt{body style Other}\ \textit{Nissan}}$	-0.121	0.170	-0.71	0.48
$\beta_{\texttt{body style Other}\ \textit{Opel}}$	-0.136	0.153	-0.89	0.37
$\beta_{\texttt{body style Other}\ \textit{Other}}$	-0.201	0.119	-1.69	0.09
$\beta_{\texttt{body style Other}\ \textit{Peugeot}}$	-0.162	0.159	-1.02	0.31
$\beta_{\texttt{body style Other}\ \textit{Renault}}$	-0.178	0.184	-0.97	0.33
$\beta_{\texttt{body style Other}\ \textit{Skoda}}$	-0.0661	0.141	-0.47	0.64
$\beta_{\texttt{body style Other}\ \textit{Volkswagen}}$	-0.119	0.119	-1.00	0.32
$\beta_{\texttt{body style Other}\ \textit{Volvo}}$	0.0279	0.131	0.21	0.83
$\beta_{\texttt{body style Sedan}\ \textit{BMW}}$	-0.0482	0.0794	-0.61	0.54
$\beta_{\texttt{body style Sedan}\ \textit{Citroen}}$	0.208	0.194	1.07	0.28

$\beta_{\texttt{body style Sedan } Ford}$	0.00472	0.123	0.04	0.97
$\beta_{\texttt{body style Sedan } Mercedes\text{-}Benz}$	-0.163	0.0854	-1.91	0.06
$\beta_{\texttt{body style Sedan } Nissan}$	0.0531	0.150	0.35	0.72
$\beta_{\texttt{body style Sedan } Opel}$	-0.308	0.138	-2.24	0.03
$\beta_{\texttt{body style Sedan } Other}$	0.0133	0.101	0.13	0.90
$\beta_{\texttt{body style Sedan } Peugeot}$	0.0748	0.137	0.55	0.58
$\beta_{\texttt{body style Sedan } Renault}$	-0.0736	0.169	-0.43	0.66
$\beta_{\texttt{body style Sedan } Skoda}$	-0.0473	0.134	-0.35	0.72
$\beta_{\texttt{body style Sedan } Volkswagen}$	0.0651	0.103	0.63	0.53
$\beta_{\texttt{body style Sedan } Volvo}$	-0.0683	0.122	-0.56	0.57
$\beta_{\texttt{body style Stationwagon } BMW}$	-0.121	0.0752	-1.61	0.11
$\beta_{\texttt{body style Stationwagon } Citroen}$	-0.225	0.179	-1.26	0.21
$\beta_{\texttt{body style Stationwagon } Ford}$	-0.0196	0.111	-0.18	0.86
$\beta_{\texttt{body style Stationwagon } Mercedes\text{-}Benz}$	-0.187	0.0803	-2.33	0.02
$\beta_{\texttt{body style Stationwagon } Nissan}$	-0.132	0.121	-1.09	0.28
$\beta_{\texttt{body style Stationwagon } Opel}$	-0.304	0.128	-2.37	0.02
$\beta_{\texttt{body style Stationwagon } Other}$	-0.0699	0.0877	-0.80	0.43
$\beta_{\texttt{body style Stationwagon } Peugeot}$	0.0597	0.124	0.48	0.63
$\beta_{\texttt{body style Stationwagon } Renault}$	-0.0667	0.151	-0.44	0.66
$\beta_{\texttt{body style Stationwagon } Skoda}$	-0.00987	0.120	-0.08	0.93
$\beta_{\texttt{body style Stationwagon } Volkswagen}$	0.175	0.0948	1.85	0.06
$\beta_{\texttt{body style Stationwagon } Volvo}$	0.00271	0.106	0.03	0.98
$\beta_{\texttt{catalogue price } BMW}$	0.280	0.0846	3.32	0.00
$\beta_{\texttt{catalogue price } Citroen}$	-1.32	0.104	-12.67	0.00
$\beta_{\texttt{catalogue price } Ford}$	-1.03	0.0805	-12.80	0.00
$\beta_{\texttt{catalogue price } Mercedes\text{-}Benz}$	0.230	0.0777	2.96	0.00
$\beta_{\texttt{catalogue price } Nissan}$	-0.603	0.0973	-6.20	0.00
$\beta_{\texttt{catalogue price } Opel}$	-1.21	0.0857	-14.11	0.00
$\beta_{\texttt{catalogue price } Other}$	-0.175	0.0630	-2.78	0.01
$\beta_{\texttt{catalogue price } Peugeot}$	-1.07	0.0853	-12.52	0.00
$\beta_{\texttt{catalogue price } Renault}$	-0.814	0.0837	-9.72	0.00
$\beta_{\texttt{catalogue price } Skoda}$	-0.694	0.0876	-7.93	0.00
$\beta_{\texttt{catalogue price } Volkswagen}$	-0.589	0.0672	-8.76	0.00
$\beta_{\texttt{catalogue price } Volvo}$	0.153	0.0954	1.60	0.11
$\beta_{\texttt{client segment Corporate } BMW}$	0.0400	0.0456	0.88	0.38
$\beta_{\texttt{client segment Corporate } Citroen}$	0.229	0.0660	3.47	0.00
$\beta_{\texttt{client segment Corporate } Ford}$	-0.0886	0.0490	-1.81	0.07
$\beta_{\texttt{client segment Corporate } Mercedes\text{-}Benz}$	0.0146	0.0466	0.31	0.75
$\beta_{\texttt{client segment Corporate } Nissan}$	-0.488	0.0658	-7.42	0.00
$\beta_{\texttt{client segment Corporate } Opel}$	0.133	0.0549	2.42	0.02
$\beta_{\texttt{client segment Corporate } Other}$	-0.0787	0.0442	-1.78	0.07
$\beta_{\texttt{client segment Corporate } Peugeot}$	0.277	0.0471	5.87	0.00

$\beta_{\texttt{client segment Corporate } \textit{Renault}}$	0.421	0.0549	7.66	0.00
$\beta_{\texttt{client segment Corporate } \textit{Skoda}}$	-0.152	0.0605	-2.52	0.01
$\beta_{\texttt{client segment Corporate } \textit{Volkswagen}}$	0.0601	0.0462	1.30	0.19
$\beta_{\texttt{client segment Corporate } \textit{Volvo}}$	-0.159	0.0633	-2.51	0.01
$\beta_{\texttt{client segment International } \textit{BMW}}$	0.226	0.0670	3.38	0.00
$\beta_{\texttt{client segment International } \textit{Citroen}}$	0.488	0.0724	6.74	0.00
$\beta_{\texttt{client segment International } \textit{Ford}}$	-0.0587	0.0512	-1.15	0.25
$\beta_{\texttt{client segment International } \textit{Mercedes-Benz}}$	0.235	0.0609	3.85	0.00
$\beta_{\texttt{client segment International } \textit{Nissan}}$	-0.844	0.0826	-10.22	0.00
$\beta_{\texttt{client segment International } \textit{Opel}}$	0.0472	0.0630	0.75	0.45
$\beta_{\texttt{client segment International } \textit{Other}}$	-0.424	0.0492	-8.63	0.00
$\beta_{\texttt{client segment International } \textit{Peugeot}}$	0.105	0.0498	2.12	0.03
$\beta_{\texttt{client segment International } \textit{Renault}}$	0.706	0.0599	11.80	0.00
$\beta_{\texttt{client segment International } \textit{Skoda}}$	0.0454	0.0662	0.69	0.49
$\beta_{\texttt{client segment International } \textit{Volkswagen}}$	0.0647	0.0496	1.30	0.19
$\beta_{\texttt{client segment International } \textit{Volvo}}$	-0.152	0.0679	-2.23	0.03
$\beta_{\texttt{commercial discount amount } \textit{BMW}}$	0.00175	0.0148	0.12	0.91
$\beta_{\texttt{commercial discount amount } \textit{Citroen}}$	0.0226	0.0216	1.04	0.30
$\beta_{\texttt{commercial discount amount } \textit{Ford}}$	0.0408	0.0175	2.34	0.02
$\beta_{\texttt{commercial discount amount } \textit{Mercedes-Benz}}$	-0.0283	0.0146	-1.93	0.05
$\beta_{\texttt{commercial discount amount } \textit{Nissan}}$	0.0486	0.0215	2.26	0.02
$\beta_{\texttt{commercial discount amount } \textit{Opel}}$	-0.00753	0.0158	-0.48	0.63
$\beta_{\texttt{commercial discount amount } \textit{Other}}$	0.0166	0.0149	1.11	0.27
$\beta_{\texttt{commercial discount amount } \textit{Peugeot}}$	0.00716	0.0165	0.43	0.66
$\beta_{\texttt{commercial discount amount } \textit{Renault}}$	0.00817	0.0203	0.40	0.69
$\beta_{\texttt{commercial discount amount } \textit{Skoda}}$	0.000739	0.0194	0.04	0.97
$\beta_{\texttt{commercial discount amount } \textit{Volkswagen}}$	0.0623	0.0161	3.86	0.00
$\beta_{\texttt{commercial discount amount } \textit{Volvo}}$	-0.0684	0.0188	-3.64	0.00
$\beta_{\texttt{contract duration } \textit{BMW}}$	0.00694	0.0161	0.43	0.67
$\beta_{\texttt{contract duration } \textit{Citroen}}$	-0.114	0.0250	-4.55	0.00
$\beta_{\texttt{contract duration } \textit{Ford}}$	-0.0729	0.0208	-3.51	0.00
$\beta_{\texttt{contract duration } \textit{Mercedes-Benz}}$	-0.0810	0.0170	-4.76	0.00
$\beta_{\texttt{contract duration } \textit{Nissan}}$	-0.133	0.0259	-5.14	0.00
$\beta_{\texttt{contract duration } \textit{Opel}}$	-0.0688	0.0180	-3.81	0.00
$\beta_{\texttt{contract duration } \textit{Other}}$	-0.127	0.0201	-6.30	0.00
$\beta_{\texttt{contract duration } \textit{Peugeot}}$	0.00575	0.0186	0.31	0.76
$\beta_{\texttt{contract duration } \textit{Renault}}$	-0.0287	0.0221	-1.30	0.19
$\beta_{\texttt{contract duration } \textit{Skoda}}$	0.0845	0.0230	3.67	0.00
$\beta_{\texttt{contract duration } \textit{Volkswagen}}$	0.0479	0.0187	2.56	0.01
$\beta_{\texttt{contract duration } \textit{Volvo}}$	0.00227	0.0235	0.10	0.92
$\beta_{\texttt{country Belgium } \textit{BMW}}$	0.116	0.0603	1.92	0.06
$\beta_{\texttt{country Belgium } \textit{Citroen}}$	0.245	0.106	2.32	0.02

$\beta_{\texttt{country Belgium} \textit{ Ford}}$	-0.342	0.0832	-4.11	0.00
$\beta_{\texttt{country Belgium} \textit{ Mercedes-Benz}}$	0.0326	0.0629	0.52	0.60
$\beta_{\texttt{country Belgium} \textit{ Nissan}}$	-0.340	0.111	-3.06	0.00
$\beta_{\texttt{country Belgium} \textit{ Opel}}$	-0.724	0.0825	-8.78	0.00
$\beta_{\texttt{country Belgium} \textit{ Other}}$	-0.110	0.0703	-1.56	0.12
$\beta_{\texttt{country Belgium} \textit{ Peugeot}}$	0.156	0.0992	1.58	0.12
$\beta_{\texttt{country Belgium} \textit{ Renault}}$	-0.177	0.0857	-2.07	0.04
$\beta_{\texttt{country Belgium} \textit{ Skoda}}$	-0.334	0.0920	-3.63	0.00
$\beta_{\texttt{country Belgium} \textit{ Volkswagen}}$	0.0535	0.0756	0.71	0.48
$\beta_{\texttt{country Belgium} \textit{ Volvo}}$	0.337	0.0899	3.74	0.00
$\beta_{\texttt{country France} \textit{ BMW}}$	-0.489	0.146	-3.34	0.00
$\beta_{\texttt{country France} \textit{ Citroen}}$	1.33	0.102	12.97	0.00
$\beta_{\texttt{country France} \textit{ Ford}}$	0.0104	0.103	0.10	0.92
$\beta_{\texttt{country France} \textit{ Mercedes-Benz}}$	-0.576	0.133	-4.34	0.00
$\beta_{\texttt{country France} \textit{ Nissan}}$	-0.0606	0.121	-0.50	0.62
$\beta_{\texttt{country France} \textit{ Opel}}$	-0.583	0.126	-4.63	0.00
$\beta_{\texttt{country France} \textit{ Other}}$	-0.371	0.0976	-3.80	0.00
$\beta_{\texttt{country France} \textit{ Peugeot}}$	1.76	0.126	13.94	0.00
$\beta_{\texttt{country France} \textit{ Renault}}$	1.09	0.115	9.50	0.00
$\beta_{\texttt{country France} \textit{ Skoda}}$	-0.794	0.145	-5.47	0.00
$\beta_{\texttt{country France} \textit{ Volkswagen}}$	0.656	0.104	6.30	0.00
$\beta_{\texttt{country France} \textit{ Volvo}}$	-0.678	0.137	-4.96	0.00
$\beta_{\texttt{country Germany} \textit{ BMW}}$	0.0736	0.0746	0.99	0.32
$\beta_{\texttt{country Germany} \textit{ Citroen}}$	0.215	0.149	1.45	0.15
$\beta_{\texttt{country Germany} \textit{ Ford}}$	0.688	0.104	6.64	0.00
$\beta_{\texttt{country Germany} \textit{ Mercedes-Benz}}$	0.146	0.0766	1.91	0.06
$\beta_{\texttt{country Germany} \textit{ Nissan}}$	-1.56	0.218	-7.17	0.00
$\beta_{\texttt{country Germany} \textit{ Opel}}$	-0.386	0.0868	-4.45	0.00
$\beta_{\texttt{country Germany} \textit{ Other}}$	-1.11	0.118	-9.43	0.00
$\beta_{\texttt{country Germany} \textit{ Peugeot}}$	-1.22	0.253	-4.82	0.00
$\beta_{\texttt{country Germany} \textit{ Renault}}$	-1.16	FIXED		
$\beta_{\texttt{country Germany} \textit{ Skoda}}$	-0.482	0.113	-4.27	0.00
$\beta_{\texttt{country Germany} \textit{ Volkswagen}}$	0.353	0.0857	4.12	0.00
$\beta_{\texttt{country Germany} \textit{ Volvo}}$	-0.647	0.122	-5.31	0.00
$\beta_{\texttt{country Italy} \textit{ BMW}}$	-0.0524	0.0863	-0.61	0.54
$\beta_{\texttt{country Italy} \textit{ Citroen}}$	0.844	0.109	7.75	0.00
$\beta_{\texttt{country Italy} \textit{ Ford}}$	0.431	0.0801	5.38	0.00
$\beta_{\texttt{country Italy} \textit{ Mercedes-Benz}}$	-0.415	0.0980	-4.24	0.00
$\beta_{\texttt{country Italy} \textit{ Nissan}}$	0.512	0.0998	5.13	0.00
$\beta_{\texttt{country Italy} \textit{ Opel}}$	-0.234	0.0850	-2.75	0.01
$\beta_{\texttt{country Italy} \textit{ Other}}$	0.841	0.0794	10.58	0.00
$\beta_{\texttt{country Italy} \textit{ Peugeot}}$	1.19	0.111	10.72	0.00

$\beta_{\texttt{country Italy}\ \textit{Renault}}$	0.320	0.0853	3.75	0.00
$\beta_{\texttt{country Italy}\ \textit{Skoda}}$	-0.341	0.123	-2.78	0.01
$\beta_{\texttt{country Italy}\ \textit{Volkswagen}}$	0.397	0.0873	4.55	0.00
$\beta_{\texttt{country Italy}\ \textit{Volvo}}$	0.164	0.119	1.37	0.17
$\beta_{\texttt{country Norway}\ \textit{BMW}}$	-0.924	0.353	-2.62	0.01
$\beta_{\texttt{country Norway}\ \textit{Citroen}}$	5.28	0.397	13.33	0.00
$\beta_{\texttt{country Norway}\ \textit{Ford}}$	4.55	0.322	14.15	0.00
$\beta_{\texttt{country Norway}\ \textit{Mercedes-Benz}}$	-0.925	0.326	-2.84	0.00
$\beta_{\texttt{country Norway}\ \textit{Nissan}}$	2.69	0.389	6.90	0.00
$\beta_{\texttt{country Norway}\ \textit{Opel}}$	4.07	0.335	12.17	0.00
$\beta_{\texttt{country Norway}\ \textit{Other}}$	0.893	0.247	3.61	0.00
$\beta_{\texttt{country Norway}\ \textit{Peugeot}}$	5.61	0.382	14.70	0.00
$\beta_{\texttt{country Norway}\ \textit{Renault}}$	2.46	0.306	8.06	0.00
$\beta_{\texttt{country Norway}\ \textit{Skoda}}$	3.46	0.352	9.81	0.00
$\beta_{\texttt{country Norway}\ \textit{Volkswagen}}$	3.65	0.271	13.48	0.00
$\beta_{\texttt{country Norway}\ \textit{Volvo}}$	-0.221	0.378	-0.58	0.56
$\beta_{\texttt{country Other}\ \textit{BMW}}$	-0.216	0.111	-1.94	0.05
$\beta_{\texttt{country Other}\ \textit{Citroen}}$	-1.10	0.441	-2.49	0.01
$\beta_{\texttt{country Other}\ \textit{Ford}}$	0.600	0.102	5.86	0.00
$\beta_{\texttt{country Other}\ \textit{Mercedes-Benz}}$	-0.444	0.113	-3.91	0.00
$\beta_{\texttt{country Other}\ \textit{Nissan}}$	0.0510	0.145	0.35	0.72
$\beta_{\texttt{country Other}\ \textit{Opel}}$	-0.303	0.112	-2.71	0.01
$\beta_{\texttt{country Other}\ \textit{Other}}$	0.125	0.0921	1.36	0.17
$\beta_{\texttt{country Other}\ \textit{Peugeot}}$	0.689	0.145	4.73	0.00
$\beta_{\texttt{country Other}\ \textit{Renault}}$	-0.655	0.139	-4.72	0.00
$\beta_{\texttt{country Other}\ \textit{Skoda}}$	0.768	0.110	7.00	0.00
$\beta_{\texttt{country Other}\ \textit{Volkswagen}}$	1.09	0.0992	10.98	0.00
$\beta_{\texttt{country Other}\ \textit{Volvo}}$	0.440	0.113	3.91	0.00
$\beta_{\texttt{country Spain}\ \textit{BMW}}$	-0.272	0.165	-1.65	0.10
$\beta_{\texttt{country Spain}\ \textit{Citroen}}$	0.202	0.169	1.20	0.23
$\beta_{\texttt{country Spain}\ \textit{Ford}}$	0.209	0.115	1.81	0.07
$\beta_{\texttt{country Spain}\ \textit{Mercedes-Benz}}$	-0.500	0.159	-3.15	0.00
$\beta_{\texttt{country Spain}\ \textit{Nissan}}$	0.550	0.137	4.03	0.00
$\beta_{\texttt{country Spain}\ \textit{Opel}}$	0.108	0.112	0.97	0.33
$\beta_{\texttt{country Spain}\ \textit{Other}}$	0.841	0.0946	8.90	0.00
$\beta_{\texttt{country Spain}\ \textit{Peugeot}}$	1.52	0.128	11.86	0.00
$\beta_{\texttt{country Spain}\ \textit{Renault}}$	1.32	0.136	9.72	0.00
$\beta_{\texttt{country Spain}\ \textit{Skoda}}$	-0.545	0.155	-3.51	0.00
$\beta_{\texttt{country Spain}\ \textit{Volkswagen}}$	1.01	0.117	8.62	0.00
$\beta_{\texttt{country Spain}\ \textit{Volvo}}$	0.162	0.167	0.97	0.33
$\beta_{\texttt{fuel type Diesel}\ \textit{BMW}}$	-0.0710	0.0547	-1.30	0.19
$\beta_{\texttt{fuel type Diesel}\ \textit{Citroen}}$	0.275	0.103	2.67	0.01

$\beta_{\texttt{fuel type Diesel}\ \textit{Ford}}$	0.209	0.0688	3.05	0.00
$\beta_{\texttt{fuel type Diesel}\ \textit{Mercedes-Benz}}$	-0.137	0.0569	-2.41	0.02
$\beta_{\texttt{fuel type Diesel}\ \textit{Nissan}}$	-0.359	0.0792	-4.53	0.00
$\beta_{\texttt{fuel type Diesel}\ \textit{Opel}}$	0.277	0.0667	4.16	0.00
$\beta_{\texttt{fuel type Diesel}\ \textit{Other}}$	-0.355	0.0597	-5.95	0.00
$\beta_{\texttt{fuel type Diesel}\ \textit{Peugeot}}$	0.560	0.0917	6.11	0.00
$\beta_{\texttt{fuel type Diesel}\ \textit{Renault}}$	0.665	0.117	5.69	0.00
$\beta_{\texttt{fuel type Diesel}\ \textit{Skoda}}$	0.223	0.0750	2.98	0.00
$\beta_{\texttt{fuel type Diesel}\ \textit{Volkswagen}}$	0.132	0.0645	2.05	0.04
$\beta_{\texttt{fuel type Diesel}\ \textit{Volvo}}$	-0.201	0.0835	-2.41	0.02
$\beta_{\texttt{lease type Financial lease}\ \textit{BMW}}$	-0.0574	0.0962	-0.60	0.55
$\beta_{\texttt{lease type Financial lease}\ \textit{Citroen}}$	0.527	0.147	3.59	0.00
$\beta_{\texttt{lease type Financial lease}\ \textit{Ford}}$	0.666	0.113	5.88	0.00
$\beta_{\texttt{lease type Financial lease}\ \textit{Mercedes-Benz}}$	0.460	0.0917	5.02	0.00
$\beta_{\texttt{lease type Financial lease}\ \textit{Nissan}}$	-0.770	0.249	-3.10	0.00
$\beta_{\texttt{lease type Financial lease}\ \textit{Opel}}$	0.432	0.114	3.79	0.00
$\beta_{\texttt{lease type Financial lease}\ \textit{Other}}$	-0.501	0.124	-4.03	0.00
$\beta_{\texttt{lease type Financial lease}\ \textit{Peugeot}}$	-0.199	0.140	-1.42	0.16
$\beta_{\texttt{lease type Financial lease}\ \textit{Renault}}$	-0.262	0.161	-1.62	0.10
$\beta_{\texttt{lease type Financial lease}\ \textit{Skoda}}$	0.270	0.130	2.07	0.04
$\beta_{\texttt{lease type Financial lease}\ \textit{Volkswagen}}$	-0.00378	0.115	-0.03	0.97
$\beta_{\texttt{lease type Financial lease}\ \textit{Volvo}}$	-0.628	0.170	-3.70	0.00
$\beta_{\texttt{make Audi}\ \textit{BMW}}$	-0.541	0.105	-5.14	0.00
$\beta_{\texttt{make Audi}\ \textit{Citroen}}$	-0.764	0.263	-2.91	0.00
$\beta_{\texttt{make Audi}\ \textit{Ford}}$	-1.18	0.142	-8.29	0.00
$\beta_{\texttt{make Audi}\ \textit{Mercedes-Benz}}$	-0.626	0.0947	-6.62	0.00
$\beta_{\texttt{make Audi}\ \textit{Nissan}}$	-1.34	0.141	-9.55	0.00
$\beta_{\texttt{make Audi}\ \textit{Opel}}$	-0.661	0.134	-4.95	0.00
$\beta_{\texttt{make Audi}\ \textit{Other}}$	-1.13	FIXED		
$\beta_{\texttt{make Audi}\ \textit{Peugeot}}$	-0.459	0.167	-2.74	0.01
$\beta_{\texttt{make Audi}\ \textit{Renault}}$	-1.66	0.130	-12.76	0.00
$\beta_{\texttt{make Audi}\ \textit{Skoda}}$	-0.531	0.106	-5.02	0.00
$\beta_{\texttt{make Audi}\ \textit{Volkswagen}}$	-0.868	0.0665	-13.05	0.00
$\beta_{\texttt{make Audi}\ \textit{Volvo}}$	-2.25	FIXED		
$\beta_{\texttt{make BMW}\ \textit{BMW}}$	0.572	0.156	3.65	0.00
$\beta_{\texttt{make BMW}\ \textit{Citroen}}$	-0.0402	0.279	-0.14	0.89
$\beta_{\texttt{make BMW}\ \textit{Ford}}$	-0.599	0.133	-4.50	0.00
$\beta_{\texttt{make BMW}\ \textit{Mercedes-Benz}}$	0.0387	0.134	0.29	0.77
$\beta_{\texttt{make BMW}\ \textit{Nissan}}$	-0.341	0.199	-1.72	0.09
$\beta_{\texttt{make BMW}\ \textit{Opel}}$	0.550	0.152	3.62	0.00
$\beta_{\texttt{make BMW}\ \textit{Other}}$	-0.549	0.110	-4.99	0.00
$\beta_{\texttt{make BMW}\ \textit{Peugeot}}$	0.158	0.191	0.82	0.41

$\beta_{\texttt{make BMW}\ Renault}$	-0.755	0.186	-4.05	0.00
$\beta_{\texttt{make BMW}\ Skoda}$	-0.0488	0.127	-0.38	0.70
$\beta_{\texttt{make BMW}\ Volkswagen}$	-0.498	0.142	-3.52	0.00
$\beta_{\texttt{make BMW}\ Volvo}$	-1.68	0.0609	-27.65	0.00
$\beta_{\texttt{make Citroen}\ BMW}$	0.0533	0.128	0.42	0.68
$\beta_{\texttt{make Citroen}\ Citroen}$	2.15	0.296	7.28	0.00
$\beta_{\texttt{make Citroen}\ Ford}$	0.301	0.159	1.89	0.06
$\beta_{\texttt{make Citroen}\ Mercedes\text{-}Benz}$	0.201	0.141	1.43	0.15
$\beta_{\texttt{make Citroen}\ Nissan}$	0.436	0.241	1.81	0.07
$\beta_{\texttt{make Citroen}\ Opel}$	0.843	0.158	5.34	0.00
$\beta_{\texttt{make Citroen}\ Other}$	0.392	0.172	2.28	0.02
$\beta_{\texttt{make Citroen}\ Peugeot}$	1.54	0.218	7.07	0.00
$\beta_{\texttt{make Citroen}\ Renault}$	0.663	0.164	4.04	0.00
$\beta_{\texttt{make Citroen}\ Skoda}$	0.635	0.156	4.08	0.00
$\beta_{\texttt{make Citroen}\ Volkswagen}$	0.469	0.152	3.09	0.00
$\beta_{\texttt{make Citroen}\ Volvo}$	-1.05	0.124	-8.45	0.00
$\beta_{\texttt{make Ford}\ BMW}$	0.169	0.117	1.44	0.15
$\beta_{\texttt{make Ford}\ Citroen}$	1.16	0.297	3.92	0.00
$\beta_{\texttt{make Ford}\ Ford}$	1.77	0.210	8.43	0.00
$\beta_{\texttt{make Ford}\ Mercedes\text{-}Benz}$	0.234	0.125	1.87	0.06
$\beta_{\texttt{make Ford}\ Nissan}$	0.425	0.260	1.63	0.10
$\beta_{\texttt{make Ford}\ Opel}$	0.734	0.167	4.38	0.00
$\beta_{\texttt{make Ford}\ Other}$	0.181	0.193	0.94	0.35
$\beta_{\texttt{make Ford}\ Peugeot}$	1.34	0.236	5.67	0.00
$\beta_{\texttt{make Ford}\ Renault}$	0.319	0.207	1.54	0.12
$\beta_{\texttt{make Ford}\ Skoda}$	0.534	0.148	3.60	0.00
$\beta_{\texttt{make Ford}\ Volkswagen}$	0.250	0.162	1.54	0.12
$\beta_{\texttt{make Ford}\ Volvo}$	-1.24	0.122	-10.20	0.00
$\beta_{\texttt{make Mercedes-Benz}\ BMW}$	0.438	0.218	2.01	0.04
$\beta_{\texttt{make Mercedes-Benz}\ Citroen}$	0.164	0.299	0.55	0.58
$\beta_{\texttt{make Mercedes-Benz}\ Ford}$	-0.383	0.155	-2.47	0.01
$\beta_{\texttt{make Mercedes-Benz}\ Mercedes\text{-}Benz}$	0.968	0.210	4.61	0.00
$\beta_{\texttt{make Mercedes-Benz}\ Nissan}$	-0.0520	0.256	-0.20	0.84
$\beta_{\texttt{make Mercedes-Benz}\ Opel}$	0.296	0.198	1.50	0.13
$\beta_{\texttt{make Mercedes-Benz}\ Other}$	-0.132	0.181	-0.73	0.47
$\beta_{\texttt{make Mercedes-Benz}\ Peugeot}$	0.573	0.244	2.35	0.02
$\beta_{\texttt{make Mercedes-Benz}\ Renault}$	-0.364	0.220	-1.65	0.10
$\beta_{\texttt{make Mercedes-Benz}\ Skoda}$	0.168	0.165	1.02	0.31
$\beta_{\texttt{make Mercedes-Benz}\ Volkswagen}$	0.00646	0.165	0.04	0.97
$\beta_{\texttt{make Mercedes-Benz}\ Volvo}$	-1.23	0.146	-8.44	0.00
$\beta_{\texttt{make Nissan}\ BMW}$	0.0623	0.144	0.43	0.67
$\beta_{\texttt{make Nissan}\ Citroen}$	0.846	0.307	2.75	0.01

$\beta_{\texttt{make Nissan}\ \textit{Ford}}$	0.603	0.178	3.39	0.00
$\beta_{\texttt{make Nissan}\ \textit{Mercedes-Benz}}$	0.0276	0.149	0.19	0.85
$\beta_{\texttt{make Nissan}\ \textit{Nissan}}$	1.92	0.258	7.44	0.00
$\beta_{\texttt{make Nissan}\ \textit{Opel}}$	0.809	0.188	4.31	0.00
$\beta_{\texttt{make Nissan}\ \textit{Other}}$	0.901	0.167	5.40	0.00
$\beta_{\texttt{make Nissan}\ \textit{Peugeot}}$	0.937	0.223	4.20	0.00
$\beta_{\texttt{make Nissan}\ \textit{Renault}}$	0.350	0.207	1.69	0.09
$\beta_{\texttt{make Nissan}\ \textit{Skoda}}$	0.623	0.189	3.30	0.00
$\beta_{\texttt{make Nissan}\ \textit{Volkswagen}}$	0.590	0.149	3.95	0.00
$\beta_{\texttt{make Nissan}\ \textit{Volvo}}$	-1.20	0.215	-5.60	0.00
$\beta_{\texttt{make Opel}\ \textit{BMW}}$	0.00710	0.111	0.06	0.95
$\beta_{\texttt{make Opel}\ \textit{Citroen}}$	0.353	0.269	1.31	0.19
$\beta_{\texttt{make Opel}\ \textit{Ford}}$	0.0436	0.124	0.35	0.72
$\beta_{\texttt{make Opel}\ \textit{Mercedes-Benz}}$	-0.200	0.111	-1.80	0.07
$\beta_{\texttt{make Opel}\ \textit{Nissan}}$	-0.0536	0.229	-0.23	0.81
$\beta_{\texttt{make Opel}\ \textit{Opel}}$	1.59	0.157	10.11	0.00
$\beta_{\texttt{make Opel}\ \textit{Other}}$	-0.372	0.139	-2.68	0.01
$\beta_{\texttt{make Opel}\ \textit{Peugeot}}$	1.07	0.203	5.27	0.00
$\beta_{\texttt{make Opel}\ \textit{Renault}}$	-0.184	0.171	-1.08	0.28
$\beta_{\texttt{make Opel}\ \textit{Skoda}}$	0.263	0.126	2.08	0.04
$\beta_{\texttt{make Opel}\ \textit{Volkswagen}}$	-0.171	0.135	-1.26	0.21
$\beta_{\texttt{make Opel}\ \textit{Volvo}}$	-1.92	0.0750	-25.64	0.00
$\beta_{\texttt{make Other}\ \textit{BMW}}$	0.165	0.0852	1.94	0.05
$\beta_{\texttt{make Other}\ \textit{Citroen}}$	0.750	0.267	2.81	0.00
$\beta_{\texttt{make Other}\ \textit{Ford}}$	0.233	0.135	1.72	0.08
$\beta_{\texttt{make Other}\ \textit{Mercedes-Benz}}$	0.0560	0.0909	0.62	0.54
$\beta_{\texttt{make Other}\ \textit{Nissan}}$	0.666	0.195	3.41	0.00
$\beta_{\texttt{make Other}\ \textit{Opel}}$	0.623	0.126	4.94	0.00
$\beta_{\texttt{make Other}\ \textit{Other}}$	1.14	0.159	7.21	0.00
$\beta_{\texttt{make Other}\ \textit{Peugeot}}$	0.813	0.185	4.39	0.00
$\beta_{\texttt{make Other}\ \textit{Renault}}$	0.577	0.133	4.33	0.00
$\beta_{\texttt{make Other}\ \textit{Skoda}}$	0.640	0.117	5.49	0.00
$\beta_{\texttt{make Other}\ \textit{Volkswagen}}$	0.146	0.105	1.39	0.16
$\beta_{\texttt{make Other}\ \textit{Volvo}}$	-1.29	0.0650	-19.81	0.00
$\beta_{\texttt{make Peugeot}\ \textit{BMW}}$	-0.0186	0.102	-0.18	0.85
$\beta_{\texttt{make Peugeot}\ \textit{Citroen}}$	0.865	0.268	3.23	0.00
$\beta_{\texttt{make Peugeot}\ \textit{Ford}}$	0.453	0.143	3.18	0.00
$\beta_{\texttt{make Peugeot}\ \textit{Mercedes-Benz}}$	-0.131	0.111	-1.18	0.24
$\beta_{\texttt{make Peugeot}\ \textit{Nissan}}$	0.197	0.220	0.90	0.37
$\beta_{\texttt{make Peugeot}\ \textit{Opel}}$	0.571	0.144	3.96	0.00
$\beta_{\texttt{make Peugeot}\ \textit{Other}}$	0.0318	0.138	0.23	0.82
$\beta_{\texttt{make Peugeot}\ \textit{Peugeot}}$	1.70	0.219	7.78	0.00

$\beta_{\texttt{make Peugeot } Renault}$	0.411	0.156	2.63	0.01
$\beta_{\texttt{make Peugeot } Skoda}$	0.382	0.131	2.92	0.00
$\beta_{\texttt{make Peugeot } Volkswagen}$	0.105	0.127	0.83	0.41
$\beta_{\texttt{make Peugeot } Volvo}$	-1.72	0.100	-17.18	0.00
$\beta_{\texttt{make Renault } BMW}$	0.0285	0.0976	0.29	0.77
$\beta_{\texttt{make Renault } Citroen}$	1.54	0.266	5.81	0.00
$\beta_{\texttt{make Renault } Ford}$	0.348	0.123	2.83	0.00
$\beta_{\texttt{make Renault } Mercedes\text{-}Benz}$	0.000752	0.0987	0.01	0.99
$\beta_{\texttt{make Renault } Nissan}$	0.732	0.195	3.74	0.00
$\beta_{\texttt{make Renault } Opel}$	0.675	0.131	5.14	0.00
$\beta_{\texttt{make Renault } Other}$	0.541	0.0996	5.43	0.00
$\beta_{\texttt{make Renault } Peugeot}$	1.19	0.187	6.35	0.00
$\beta_{\texttt{make Renault } Renault}$	1.66	0.138	12.04	0.00
$\beta_{\texttt{make Renault } Skoda}$	0.150	0.128	1.17	0.24
$\beta_{\texttt{make Renault } Volkswagen}$	0.364	0.105	3.45	0.00
$\beta_{\texttt{make Renault } Volvo}$	-0.982	0.133	-7.40	0.00
$\beta_{\texttt{make Skoda } BMW}$	0.162	0.139	1.17	0.24
$\beta_{\texttt{make Skoda } Citroen}$	0.436	0.337	1.30	0.20
$\beta_{\texttt{make Skoda } Ford}$	0.0426	0.160	0.27	0.79
$\beta_{\texttt{make Skoda } Mercedes\text{-}Benz}$	0.225	0.141	1.59	0.11
$\beta_{\texttt{make Skoda } Nissan}$	0.412	0.256	1.61	0.11
$\beta_{\texttt{make Skoda } Opel}$	0.296	0.185	1.60	0.11
$\beta_{\texttt{make Skoda } Other}$	0.184	0.174	1.06	0.29
$\beta_{\texttt{make Skoda } Peugeot}$	0.472	0.227	2.08	0.04
$\beta_{\texttt{make Skoda } Renault}$	-0.305	0.249	-1.23	0.22
$\beta_{\texttt{make Skoda } Skoda}$	1.68	0.173	9.67	0.00
$\beta_{\texttt{make Skoda } Volkswagen}$	0.147	0.162	0.91	0.36
$\beta_{\texttt{make Skoda } Volvo}$	-1.44	0.148	-9.74	0.00
$\beta_{\texttt{make Volkswagen } BMW}$	-0.0868	0.0730	-1.19	0.23
$\beta_{\texttt{make Volkswagen } Citroen}$	0.671	0.259	2.60	0.01
$\beta_{\texttt{make Volkswagen } Ford}$	-0.0749	0.0975	-0.77	0.44
$\beta_{\texttt{make Volkswagen } Mercedes\text{-}Benz}$	-0.0795	0.0587	-1.36	0.18
$\beta_{\texttt{make Volkswagen } Nissan}$	-0.222	0.168	-1.32	0.19
$\beta_{\texttt{make Volkswagen } Opel}$	0.151	0.119	1.27	0.20
$\beta_{\texttt{make Volkswagen } Other}$	-0.399	0.0574	-6.95	0.00
$\beta_{\texttt{make Volkswagen } Peugeot}$	0.465	0.168	2.76	0.01
$\beta_{\texttt{make Volkswagen } Renault}$	-0.416	0.115	-3.61	0.00
$\beta_{\texttt{make Volkswagen } Skoda}$	0.388	0.0870	4.45	0.00
$\beta_{\texttt{make Volkswagen } Volkswagen}$	1.16	0.0699	16.66	0.00
$\beta_{\texttt{make Volkswagen } Volvo}$	-1.60	0.0614	-26.07	0.00
$\beta_{\texttt{mileage month } BMW}$	-0.0338	0.0169	-2.01	0.04
$\beta_{\texttt{mileage month } Citroen}$	0.0648	0.0225	2.88	0.00

$\beta_{\texttt{mileage month}\ \textit{Ford}}$	0.0998	0.0191	5.23	0.00
$\beta_{\texttt{mileage month}\ \textit{Mercedes-Benz}}$	-0.0660	0.0174	-3.80	0.00
$\beta_{\texttt{mileage month}\ \textit{Nissan}}$	-0.0304	0.0260	-1.17	0.24
$\beta_{\texttt{mileage month}\ \textit{Opel}}$	0.0981	0.0201	4.88	0.00
$\beta_{\texttt{mileage month}\ \textit{Other}}$	-0.0718	0.0197	-3.64	0.00
$\beta_{\texttt{mileage month}\ \textit{Peugeot}}$	0.151	0.0203	7.45	0.00
$\beta_{\texttt{mileage month}\ \textit{Renault}}$	0.115	0.0208	5.53	0.00
$\beta_{\texttt{mileage month}\ \textit{Skoda}}$	0.253	0.0235	10.80	0.00
$\beta_{\texttt{mileage month}\ \textit{Volkswagen}}$	0.145	0.0167	8.68	0.00
$\beta_{\texttt{mileage month}\ \textit{Volvo}}$	0.00703	0.0242	0.29	0.77
$\beta_{\texttt{standard discount percentage}\ \textit{BMW}}$	-0.0264	0.0313	-0.84	0.40
$\beta_{\texttt{standard discount percentage}\ \textit{Citroen}}$	0.133	0.0357	3.73	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Ford}}$	0.0529	0.0312	1.70	0.09
$\beta_{\texttt{standard discount percentage}\ \textit{Mercedes-Benz}}$	0.0766	0.0307	2.50	0.01
$\beta_{\texttt{standard discount percentage}\ \textit{Nissan}}$	0.108	0.0426	2.53	0.01
$\beta_{\texttt{standard discount percentage}\ \textit{Opel}}$	0.164	0.0320	5.10	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Other}}$	0.0870	0.0317	2.75	0.01
$\beta_{\texttt{standard discount percentage}\ \textit{Peugeot}}$	0.0608	0.0317	1.92	0.05
$\beta_{\texttt{standard discount percentage}\ \textit{Renault}}$	0.0855	0.0312	2.74	0.01
$\beta_{\texttt{standard discount percentage}\ \textit{Skoda}}$	0.258	0.0418	6.18	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Volkswagen}}$	0.112	0.0307	3.65	0.00
$\beta_{\texttt{standard discount percentage}\ \textit{Volvo}}$	0.139	0.0433	3.21	0.00
$\beta_{\texttt{switch quarter 1}\ \textit{BMW}}$	0.0288	0.0337	0.85	0.39
$\beta_{\texttt{switch quarter 1}\ \textit{Citroen}}$	-0.300	0.0533	-5.62	0.00
$\beta_{\texttt{switch quarter 1}\ \textit{Ford}}$	-0.113	0.0417	-2.72	0.01
$\beta_{\texttt{switch quarter 1}\ \textit{Mercedes-Benz}}$	-0.0633	0.0354	-1.79	0.07
$\beta_{\texttt{switch quarter 1}\ \textit{Nissan}}$	-0.0166	0.0585	-0.28	0.78
$\beta_{\texttt{switch quarter 1}\ \textit{Opel}}$	-0.0792	0.0422	-1.88	0.06
$\beta_{\texttt{switch quarter 1}\ \textit{Other}}$	-0.0240	0.0398	-0.60	0.55
$\beta_{\texttt{switch quarter 1}\ \textit{Peugeot}}$	-0.174	0.0414	-4.21	0.00
$\beta_{\texttt{switch quarter 1}\ \textit{Renault}}$	-0.120	0.0451	-2.67	0.01
$\beta_{\texttt{switch quarter 1}\ \textit{Skoda}}$	0.0515	0.0474	1.09	0.28
$\beta_{\texttt{switch quarter 1}\ \textit{Volkswagen}}$	-0.151	0.0372	-4.04	0.00
$\beta_{\texttt{switch quarter 1}\ \textit{Volvo}}$	-0.00899	0.0524	-0.17	0.86
$\beta_{\texttt{switch quarter 2}\ \textit{BMW}}$	0.0123	0.0329	0.37	0.71
$\beta_{\texttt{switch quarter 2}\ \textit{Citroen}}$	-0.147	0.0500	-2.93	0.00
$\beta_{\texttt{switch quarter 2}\ \textit{Ford}}$	-0.0961	0.0404	-2.38	0.02
$\beta_{\texttt{switch quarter 2}\ \textit{Mercedes-Benz}}$	-0.0396	0.0340	-1.16	0.24
$\beta_{\texttt{switch quarter 2}\ \textit{Nissan}}$	-0.000860	0.0566	-0.02	0.99
$\beta_{\texttt{switch quarter 2}\ \textit{Opel}}$	-0.0327	0.0405	-0.81	0.42
$\beta_{\texttt{switch quarter 2}\ \textit{Other}}$	-0.183	0.0395	-4.64	0.00
$\beta_{\texttt{switch quarter 2}\ \textit{Peugeot}}$	-0.176	0.0409	-4.30	0.00

$\beta_{\texttt{switch quarter 2}\ \textit{Renault}}$	-0.282	0.0449	-6.28	0.00
$\beta_{\texttt{switch quarter 2}\ \textit{Skoda}}$	0.00668	0.0464	0.14	0.89
$\beta_{\texttt{switch quarter 2}\ \textit{Volkswagen}}$	-0.161	0.0359	-4.49	0.00
$\beta_{\texttt{switch quarter 2}\ \textit{Volvo}}$	-0.0430	0.0505	-0.85	0.39
$\beta_{\texttt{switch quarter 3}\ \textit{BMW}}$	-0.0410	0.0324	-1.27	0.21
$\beta_{\texttt{switch quarter 3}\ \textit{Citroen}}$	-0.223	0.0504	-4.43	0.00
$\beta_{\texttt{switch quarter 3}\ \textit{Ford}}$	-0.110	0.0397	-2.77	0.01
$\beta_{\texttt{switch quarter 3}\ \textit{Mercedes-Benz}}$	-0.0143	0.0337	-0.42	0.67
$\beta_{\texttt{switch quarter 3}\ \textit{Nissan}}$	-0.107	0.0569	-1.89	0.06
$\beta_{\texttt{switch quarter 3}\ \textit{Opel}}$	-0.0295	0.0402	-0.73	0.46
$\beta_{\texttt{switch quarter 3}\ \textit{Other}}$	-0.0731	0.0385	-1.90	0.06
$\beta_{\texttt{switch quarter 3}\ \textit{Peugeot}}$	-0.172	0.0410	-4.20	0.00
$\beta_{\texttt{switch quarter 3}\ \textit{Renault}}$	-0.299	0.0453	-6.59	0.00
$\beta_{\texttt{switch quarter 3}\ \textit{Skoda}}$	-0.130	0.0481	-2.69	0.01
$\beta_{\texttt{switch quarter 3}\ \textit{Volkswagen}}$	-0.232	0.0363	-6.39	0.00
$\beta_{\texttt{switch quarter 3}\ \textit{Volvo}}$	-0.156	0.0515	-3.02	0.00
$\beta_{\texttt{total accessories amount}\ \textit{BMW}}$	-0.0538	0.0139	-3.88	0.00
$\beta_{\texttt{total accessories amount}\ \textit{Citroen}}$	0.0392	0.0210	1.86	0.06
$\beta_{\texttt{total accessories amount}\ \textit{Ford}}$	0.0613	0.0164	3.74	0.00
$\beta_{\texttt{total accessories amount}\ \textit{Mercedes-Benz}}$	-0.0479	0.0137	-3.49	0.00
$\beta_{\texttt{total accessories amount}\ \textit{Nissan}}$	-0.0520	0.0242	-2.15	0.03
$\beta_{\texttt{total accessories amount}\ \textit{Opel}}$	0.0830	0.0163	5.10	0.00
$\beta_{\texttt{total accessories amount}\ \textit{Other}}$	-0.136	0.0182	-7.49	0.00
$\beta_{\texttt{total accessories amount}\ \textit{Peugeot}}$	-0.0501	0.0165	-3.03	0.00
$\beta_{\texttt{total accessories amount}\ \textit{Renault}}$	0.0332	0.0188	1.76	0.08
$\beta_{\texttt{total accessories amount}\ \textit{Skoda}}$	0.0273	0.0191	1.43	0.15
$\beta_{\texttt{total accessories amount}\ \textit{Volkswagen}}$	-0.0652	0.0142	-4.60	0.00
$\beta_{\texttt{total accessories amount}\ \textit{Volvo}}$	-0.0367	0.0203	-1.81	0.07
$\beta_{\texttt{total options amount}\ \textit{BMW}}$	-0.0446	0.0166	-2.68	0.01
$\beta_{\texttt{total options amount}\ \textit{Citroen}}$	-0.148	0.0228	-6.49	0.00
$\beta_{\texttt{total options amount}\ \textit{Ford}}$	-0.102	0.0197	-5.19	0.00
$\beta_{\texttt{total options amount}\ \textit{Mercedes-Benz}}$	-0.00731	0.0180	-0.41	0.68
$\beta_{\texttt{total options amount}\ \textit{Nissan}}$	-0.0441	0.0232	-1.90	0.06
$\beta_{\texttt{total options amount}\ \textit{Opel}}$	-0.0426	0.0181	-2.36	0.02
$\beta_{\texttt{total options amount}\ \textit{Other}}$	-0.0484	0.0176	-2.76	0.01
$\beta_{\texttt{total options amount}\ \textit{Peugeot}}$	-0.0336	0.0188	-1.79	0.07
$\beta_{\texttt{total options amount}\ \textit{Renault}}$	-0.00579	0.0191	-0.30	0.76
$\beta_{\texttt{total options amount}\ \textit{Skoda}}$	-0.0822	0.0226	-3.64	0.00
$\beta_{\texttt{total options amount}\ \textit{Volkswagen}}$	-0.104	0.0179	-5.82	0.00
$\beta_{\texttt{total options amount}\ \textit{Volvo}}$	-0.0373	0.0252	-1.48	0.14
$\beta_{\texttt{ufwt amount}\ \textit{BMW}}$	-0.0353	0.0164	-2.15	0.03
$\beta_{\texttt{ufwt amount}\ \textit{Citroen}}$	0.0254	0.0231	1.10	0.27

$\beta_{\texttt{ufwt amount}\ \textit{Ford}}$	-0.0481	0.0179	-2.68	0.01
$\beta_{\texttt{ufwt amount}\ \textit{Mercedes-Benz}}$	-0.0714	0.0171	-4.18	0.00
$\beta_{\texttt{ufwt amount}\ \textit{Nissan}}$	-0.0755	0.0236	-3.20	0.00
$\beta_{\texttt{ufwt amount}\ \textit{Opel}}$	-0.0732	0.0181	-4.05	0.00
$\beta_{\texttt{ufwt amount}\ \textit{Other}}$	-0.0822	0.0184	-4.46	0.00
$\beta_{\texttt{ufwt amount}\ \textit{Peugeot}}$	-0.0705	0.0168	-4.21	0.00
$\beta_{\texttt{ufwt amount}\ \textit{Renault}}$	0.0529	0.0217	2.44	0.01
$\beta_{\texttt{ufwt amount}\ \textit{Skoda}}$	0.0155	0.0203	0.76	0.45
$\beta_{\texttt{ufwt amount}\ \textit{Volkswagen}}$	0.00823	0.0177	0.46	0.64
$\beta_{\texttt{ufwt amount}\ \textit{Volvo}}$	0.0225	0.0215	1.05	0.29
$\beta_{\texttt{vehicle segment B}\ \textit{BMW}}$	-0.124	0.103	-1.20	0.23
$\beta_{\texttt{vehicle segment B}\ \textit{Citroen}}$	0.527	0.172	3.06	0.00
$\beta_{\texttt{vehicle segment B}\ \textit{Ford}}$	0.302	0.118	2.56	0.01
$\beta_{\texttt{vehicle segment B}\ \textit{Mercedes-Benz}}$	-0.0403	0.108	-0.37	0.71
$\beta_{\texttt{vehicle segment B}\ \textit{Nissan}}$	-0.114	0.150	-0.76	0.45
$\beta_{\texttt{vehicle segment B}\ \textit{Opel}}$	0.355	0.124	2.87	0.00
$\beta_{\texttt{vehicle segment B}\ \textit{Other}}$	0.414	0.101	4.09	0.00
$\beta_{\texttt{vehicle segment B}\ \textit{Peugeot}}$	0.0722	0.130	0.55	0.58
$\beta_{\texttt{vehicle segment B}\ \textit{Renault}}$	1.09	0.145	7.53	0.00
$\beta_{\texttt{vehicle segment B}\ \textit{Skoda}}$	-0.0199	0.147	-0.14	0.89
$\beta_{\texttt{vehicle segment B}\ \textit{Volkswagen}}$	0.107	0.114	0.94	0.35
$\beta_{\texttt{vehicle segment B}\ \textit{Volvo}}$	-0.370	0.176	-2.10	0.04
$\beta_{\texttt{vehicle segment C}\ \textit{BMW}}$	-0.102	0.0689	-1.48	0.14
$\beta_{\texttt{vehicle segment C}\ \textit{Citroen}}$	0.280	0.153	1.83	0.07
$\beta_{\texttt{vehicle segment C}\ \textit{Ford}}$	0.374	0.0918	4.07	0.00
$\beta_{\texttt{vehicle segment C}\ \textit{Mercedes-Benz}}$	-0.0837	0.0724	-1.16	0.25
$\beta_{\texttt{vehicle segment C}\ \textit{Nissan}}$	-0.187	0.110	-1.70	0.09
$\beta_{\texttt{vehicle segment C}\ \textit{Opel}}$	0.684	0.101	6.80	0.00
$\beta_{\texttt{vehicle segment C}\ \textit{Other}}$	-0.0172	0.0785	-0.22	0.83
$\beta_{\texttt{vehicle segment C}\ \textit{Peugeot}}$	0.208	0.111	1.87	0.06
$\beta_{\texttt{vehicle segment C}\ \textit{Renault}}$	0.193	0.137	1.41	0.16
$\beta_{\texttt{vehicle segment C}\ \textit{Skoda}}$	0.341	0.104	3.26	0.00
$\beta_{\texttt{vehicle segment C}\ \textit{Volkswagen}}$	0.163	0.0852	1.91	0.06
$\beta_{\texttt{vehicle segment C}\ \textit{Volvo}}$	-0.501	0.104	-4.81	0.00
$\beta_{\texttt{vehicle segment D}\ \textit{BMW}}$	-0.236	0.0691	-3.41	0.00
$\beta_{\texttt{vehicle segment D}\ \textit{Citroen}}$	0.230	0.156	1.47	0.14
$\beta_{\texttt{vehicle segment D}\ \textit{Ford}}$	0.0994	0.0854	1.16	0.24
$\beta_{\texttt{vehicle segment D}\ \textit{Mercedes-Benz}}$	-0.169	0.0695	-2.44	0.01
$\beta_{\texttt{vehicle segment D}\ \textit{Nissan}}$	-0.282	0.114	-2.47	0.01
$\beta_{\texttt{vehicle segment D}\ \textit{Opel}}$	0.536	0.0988	5.42	0.00
$\beta_{\texttt{vehicle segment D}\ \textit{Other}}$	-0.278	0.0788	-3.53	0.00
$\beta_{\texttt{vehicle segment D}\ \textit{Peugeot}}$	0.0617	0.112	0.55	0.58

$\beta_{\texttt{vehicle segment D } Renault}$	0.0684	0.141	0.49	0.63
$\beta_{\texttt{vehicle segment D } Skoda}$	0.170	0.0994	1.71	0.09
$\beta_{\texttt{vehicle segment D } Volkswagen}$	0.144	0.0833	1.73	0.08
$\beta_{\texttt{vehicle segment D } Volvo}$	-0.437	0.0933	-4.68	0.00
$\beta_{\texttt{vehicle segment E } BMW}$	-0.308	0.0856	-3.60	0.00
$\beta_{\texttt{vehicle segment E } Citroen}$	0.0406	0.257	0.16	0.87
$\beta_{\texttt{vehicle segment E } Ford}$	-0.0929	0.133	-0.70	0.48
$\beta_{\texttt{vehicle segment E } Mercedes\text{-}Benz}$	-0.258	0.0841	-3.07	0.00
$\beta_{\texttt{vehicle segment E } Nissan}$	-0.354	0.186	-1.90	0.06
$\beta_{\texttt{vehicle segment E } Opel}$	0.609	0.125	4.87	0.00
$\beta_{\texttt{vehicle segment E } Other}$	-0.236	0.0987	-2.39	0.02
$\beta_{\texttt{vehicle segment E } Peugeot}$	-0.129	0.196	-0.66	0.51
$\beta_{\texttt{vehicle segment E } Renault}$	0.192	0.191	1.00	0.31
$\beta_{\texttt{vehicle segment E } Skoda}$	0.0310	0.129	0.24	0.81
$\beta_{\texttt{vehicle segment E } Volkswagen}$	-0.147	0.102	-1.44	0.15
$\beta_{\texttt{vehicle segment E } Volvo}$	-0.409	0.106	-3.85	0.00
$\beta_{\texttt{vehicle segment LCV } BMW}$	-0.0645	0.241	-0.27	0.79
$\beta_{\texttt{vehicle segment LCV } Citroen}$	1.63	0.266	6.13	0.00
$\beta_{\texttt{vehicle segment LCV } Ford}$	1.25	0.220	5.66	0.00
$\beta_{\texttt{vehicle segment LCV } Mercedes\text{-}Benz}$	0.583	0.222	2.62	0.01
$\beta_{\texttt{vehicle segment LCV } Nissan}$	0.505	0.247	2.05	0.04
$\beta_{\texttt{vehicle segment LCV } Opel}$	1.26	0.221	5.70	0.00
$\beta_{\texttt{vehicle segment LCV } Other}$	0.382	0.206	1.86	0.06
$\beta_{\texttt{vehicle segment LCV } Peugeot}$	0.896	0.223	4.02	0.00
$\beta_{\texttt{vehicle segment LCV } Renault}$	1.63	0.227	7.16	0.00
$\beta_{\texttt{vehicle segment LCV } Skoda}$	0.572	0.263	2.18	0.03
$\beta_{\texttt{vehicle segment LCV } Volkswagen}$	0.768	0.204	3.77	0.00
$\beta_{\texttt{vehicle segment LCV } Volvo}$	0.302	0.295	1.02	0.31
$\beta_{\texttt{vehicle segment MPV } BMW}$	0.111	0.0796	1.40	0.16
$\beta_{\texttt{vehicle segment MPV } Citroen}$	0.354	0.129	2.74	0.01
$\beta_{\texttt{vehicle segment MPV } Ford}$	0.0208	0.0941	0.22	0.82
$\beta_{\texttt{vehicle segment MPV } Mercedes\text{-}Benz}$	0.0231	0.0840	0.27	0.78
$\beta_{\texttt{vehicle segment MPV } Nissan}$	0.00950	0.127	0.07	0.94
$\beta_{\texttt{vehicle segment MPV } Opel}$	0.357	0.112	3.19	0.00
$\beta_{\texttt{vehicle segment MPV } Other}$	0.0367	0.0938	0.39	0.70
$\beta_{\texttt{vehicle segment MPV } Peugeot}$	-0.0752	0.0906	-0.83	0.41
$\beta_{\texttt{vehicle segment MPV } Renault}$	0.416	0.128	3.26	0.00
$\beta_{\texttt{vehicle segment MPV } Skoda}$	0.294	0.118	2.48	0.01
$\beta_{\texttt{vehicle segment MPV } Volkswagen}$	0.267	0.0906	2.95	0.00
$\beta_{\texttt{vehicle segment MPV } Volvo}$	-0.187	0.128	-1.46	0.14
$\beta_{\texttt{vehicle segment Other } BMW}$	-0.0136	0.158	-0.09	0.93
$\beta_{\texttt{vehicle segment Other } Citroen}$	0.220	0.247	0.89	0.37

$\beta_{\texttt{vehicle segment Other } \textit{Ford}}$	0.548	0.186	2.95	0.00
$\beta_{\texttt{vehicle segment Other } \textit{Mercedes-Benz}}$	0.211	0.160	1.32	0.19
$\beta_{\texttt{vehicle segment Other } \textit{Nissan}}$	-0.220	0.216	-1.02	0.31
$\beta_{\texttt{vehicle segment Other } \textit{Opel}}$	0.368	0.192	1.92	0.06
$\beta_{\texttt{vehicle segment Other } \textit{Other}}$	0.156	0.163	0.96	0.34
$\beta_{\texttt{vehicle segment Other } \textit{Peugeot}}$	0.125	0.195	0.64	0.52
$\beta_{\texttt{vehicle segment Other } \textit{Renault}}$	0.386	0.223	1.73	0.08
$\beta_{\texttt{vehicle segment Other } \textit{Skoda}}$	-0.0444	0.265	-0.17	0.87
$\beta_{\texttt{vehicle segment Other } \textit{Volkswagen}}$	0.251	0.185	1.35	0.18
$\beta_{\texttt{vehicle segment Other } \textit{Volvo}}$	-0.556	0.320	-1.74	0.08
$\beta_{\texttt{vehicle type Car } \textit{BMW}}$	-0.403	0.239	-1.69	0.09
$\beta_{\texttt{vehicle type Car } \textit{Citroen}}$	-1.57	0.189	-8.32	0.00
$\beta_{\texttt{vehicle type Car } \textit{Ford}}$	-0.828	0.190	-4.36	0.00
$\beta_{\texttt{vehicle type Car } \textit{Mercedes-Benz}}$	-0.163	0.230	-0.71	0.48
$\beta_{\texttt{vehicle type Car } \textit{Nissan}}$	-0.868	0.212	-4.09	0.00
$\beta_{\texttt{vehicle type Car } \textit{Opel}}$	-0.421	0.197	-2.14	0.03
$\beta_{\texttt{vehicle type Car } \textit{Other}}$	-1.17	0.191	-6.12	0.00
$\beta_{\texttt{vehicle type Car } \textit{Peugeot}}$	-1.25	0.181	-6.91	0.00
$\beta_{\texttt{vehicle type Car } \textit{Renault}}$	-1.33	0.183	-7.30	0.00
$\beta_{\texttt{vehicle type Car } \textit{Skoda}}$	0.0824	0.256	0.32	0.75
$\beta_{\texttt{vehicle type Car } \textit{Volkswagen}}$	-1.01	0.188	-5.36	0.00
$\beta_{\texttt{vehicle type Car } \textit{Volvo}}$	0.124	0.304	0.41	0.68

Table B.3: Overview of all estimated parameters for the cross-nested multinomial logistic regression model. Each parameter name (β) is structured as follows. The first part, stated in `font` , is considered the explanatory variable. The second part of the name, stated in italics, is considered an alternative of the choice set.

	#Obs.	**MNL**	**n-MNL**	**cn-MNL**	**Embed.**	**One-Hot**
Audi	553	46.65%	46.29%	**47.20%**	44.30%	45.21%
BMW	567	39.15%	39.33%	39.15%	**41.62%**	40.92%
Citroen	307	25.08%	24.43%	24.43%	23.45%	**27.69%**
Ford	621	57.17%	56.84%	57.17%	56.36%	**57.49%**
Mercedes-Benz	568	**45.95%**	45.77%	45.77%	44.37%	45.77%
Nissan	187	12.30%	12.30%	11.23%	42.25%	**46.52%**
Opel	303	42.90%	41.91%	42.57%	**43.56%**	42.57%
Other	802	45.76%	45.76%	45.39%	**54.61%**	45.89%
Peugeot	831	36.10%	36.22%	37.42%	**43.68%**	41.40%
Renault	906	69.65%	69.21%	**69.98%**	67.11%	68.10%
Skoda	232	41.38%	40.95%	41.81%	42.67%	**43.53%**
Volkswagen	847	63.28%	63.64%	63.75%	**63.99%**	62.46%
Volvo	172	**41.86%**	35.47%	39.53%	38.95%	38.95%

Table B.4: Performance breakdown of all models on test data. The rows indicate the alternatives of the choice set, the columns indicate the achieved accuracy per model per alternative of the choice set. The first column states the number of observations of the corresponding alternative in the test data. Per alternative, the best performing model is indicated by a bold accuracy score.