Vrije Universiteit Amsterdam

Master Thesis

Airplane Seat pricing: Prediction and Optimization based on historical data

Author: Tim von Luecken (2737512)

1st supervisor:	Prof. dr. Ger Koole	
Industry supervisor:	Kevin Duijndam	(KLM AF)
2nd reader:	dr. Christian Bick	

A thesis submitted in fulfillment of the requirements for the VU Master of Science degree in Business Analytics

July 13, 2023

Abstract

Context. This project is conducted at the Air France KLM group in the field of aircraft seat pricing and is using a machine learning approach to predict price elasticity and optimize revenue for seats in the Economy Comfort section.

Goal. The goal is to assess how currently available historical data could be utilized to optimize the Economy Comfort class seat prices on passenger level and to evaluate the contribution of passenger information within the optimization.

Method. To achieve this, the available data is cleaned and labeled, new features are build and their influence on the purchase decision is evaluated. The data is used as input for various classification models with different parameters, which are compared based on their prediction bias, price sensitivity, model calibration and their ability to increase revenue using traveler related information. The predicted price elasticity per customer is used to optimize prices on passenger level, flight level and globally. The resulting revenues are compared with the actual revenue and among each other.

Results. Most of the tree boosting models tested achieve similar results. When the maximal discount applicable on the seat price is 12.5%, the passenger level optimization is expected to increase the true revenue utilizing any of the boosting models by approximately 19-20.5% on intercontinental and 14-15.4% on medium haul flights. As this result heavily relies on the prices learned by the model which were mostly unknown and had to be estimated, this result is not guaranteed, but could be proven by tracking all seat prices offered at different times to the passengers. The optimization on passenger level is on average 3.37% higher than flight level optimization on intercontinental and 4.53% higher on medium haul flights.

Conclusions. This result leads to the final conclusion, that passenger data can be used to increase revenue in seat pricing. Next to other interesting adaptations of this approach, the exact framework should be tested with the actual observed prices and is expected to give valuable pricing directions. This project is a very promising framework of how passenger data can generate more value in seat pricing.

Acknowledgements

I would like to express my gratitude and appreciation to my supervisors Ger Koole (Vrije Universiteit) and Kevin Duijmdam (Air France KLM), for their invaluable assistance during the process of the project. Their insightful feedback and critical questions greatly contributed to shaping the direction of this thesis. With regular meetings and quick correspondence, the project could be carried out in an organised and interruption-free manner. Furthermore, I wold like to extend my sincere gratitude to the AFKL Performance Insights and Analytic team, for many ideas and recommendations throughout the thesis writing process. I would like to thank Nora Cselotei (Air France KLM) in particular for extensive introductions into the data bases and her fundamental knowledge about the seat products, which have greatly aided the project's early start. Finally, I want to thank Christian Bick (Vrije Universiteit) for being my second reader and Annemieke van Goor-Balk (Vrije Universiteit) for an excellent coordination of the entire internship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Air France-KLM The Air France-KLM Group (AFKL) is a major player in aviation, operating flights to 145 destinations with a fleet of 110 carriers and a total revenue of [redacted] euros in 2022. The airline holding company is the result of the merger of the airlines Air France and KLM in 2004 and has its head quarters in Paris and Amstelveen.
Next to the transportation service, the group offers many more products to its customers, depending on the comfort and experience desired. The so called *Ancillaries*, which include side products like seating, baggage, meals, etc., made up [redacted] of the total revenue in 2022. By revenue, Baggage [redacted], Seating [redacted] and Upgrades [redacted] are the three main categories, with Economy Comfort accounting for [redacted] of the Seating revenue. All KLM-operated flights provide Economy Comfort (EC) seating, which is the most expensive section and located in the Economy cabin's front.

Revenue management Revenue management began in the late 1970s with the airline deregulation act in the United States. After 40 years of operating on fixed routes at controlled rates, the airline sector was suddenly able to make all of these decisions on its own, causing significant confusion at first. In the years that followed, revenue management strategies evolved as a requirement for survival in an increasingly competitive industry (Cross [9]). Revenue management aims to find the right price for the right product for the right customer at the right place and the right time.

Seat pricing at AFKL It is worth noting that there is a fundamental difference between *ticket price* and *seat price*, because the ticket guarantees a seat on the flight, but passengers can furthermore choose to purchase a specific seat. For the ticket prices the current revenue management model for AFKL has been in place since 2018, dynamically modifying prices using enhanced forecasting models. These models rely on a large quantity of historical data and have a grasp of the

passengers' price sensitivity. So far, this has not been the case for seat prices. Until now, base seat prices have been assigned to all different seat types, based on the destination, the month and whether it is an incoming (inbound) or outbound flight. In addition to this base price, there are simple business rules that allow seats in certain sections to be discounted by up to 10% based on the days until departure and the load factor of the flight. The load factor gives the number of sold tickets in relation to the capacity of a specific flight. Additionally there are very big last minute discounts at check-in and general discounts granted to passengers of specific levels of the frequent flyer program.

1.2 Motivation

EC seats cost an average of [redacted] on intercontinental flights and [redacted] on medium haul flights. In 2022, the EC section had a load factor of [redacted] on intercontinental flights and [redacted] on medium haul flights compared to an overall load factor of [redacted] and [redacted] respectively. Because many passengers are upgraded for free, the paid load factor is substantially lower, at [redacted] on intercontinental and [redacted] on medium haul flights. KLM operated [redacted] intercontinental flights with an average EC capacity of [redacted] seats and [redacted] medium haul flights with [redacted] seats in the EC section in 2022. Assuming a target load factor of 95% and that any seat can be sold at a 50% discount on the average prices, the maximum potential additional revenue from intercontinental and medium haul passengers is [redacted] and [redacted] respectively. Note, that this is a very high-level boundary which does not consider if customers would actually buy these seats at a 50% discount. Gaining a better understanding of the *willingness to pay* (wtp) for EC seats under particular conditions can contribute to a pricing strategy that can recover this missed revenue.

1.3 Problem Statement

The issue with selecting seat prices is that there is insufficient historical data to estimate the passengers' willingness to pay, and select the price that optimizes the expected revenue. This is because prices were not widely varied in the past, but rather were consistent across all flights on the same route. Furthermore, the data only provides limited information about the travelers. Meeting demand with the optimal prices provides a challenge without a clear understanding of the customer willingness to pay, and leaves many unused revenue opportunities. The problem raises related sub questions that will be investigated during the course of this thesis:

i. What features are important when predicting if a customer is going to buy a seat?

ii. What models are suitable for the classification of customers who buy or not buy a seat?

iii. Can the total revenue be increased by using the price elasticity derived from the classification models?

iv. What is the added value of passenger information in this framework?

1.4 Scope

Kevin Duijndam, the supervisor of this study from AFKL side, is currently investigating how reinforcement learning can be used to improve pricing decisions by learning the willingness to pay of passengers for seats on flight level in a Bayesian learning framework. This framework needs a prior estimation of the price elasticity function and learns the function parameters on flight level, without making use of any already existing historical data.
The scope of this study is to explicitly focus on the available historical data and determine whether it is possible to use this data to predict the price elasticity on passenger level rather than on flight level. This research will be limited to seats in the Economy Comfort (EC) section, and will hence only use data from KLM-operated flights. This means that each time a seat price is mentioned, it is referring to a seat in the EC section.

1.5 Aim and Objectives

The research question of this project is: How can the current historical data of KLM be used to optimize Economy Comfort seat prices on passenger level and what is the contribution of traveler information in this approach? To answer this question, the study is split in three parts:

1. Data analysis and feature engineering: Exploring the influence of various features on the purchasing decision of passengers.

2. Predictive modelling: Building a classification model that can be used to predict logical price elasticity curves on passenger level.

3. Optimization: Using the estimated price elasticity to optimize the seat prices and evaluate the results also with regards to the implications on the prices used by KLM at the moment.

The features discovered in the data analysis will be assessed in an exploratory data analysis and compared based on their permutation importance. Together these methods will give insights into the features' contributions to the passengers' purchase decision. The predictive models' prediction bias and calibration will be evaluated in order to determine the reliability of the price elasticity curves in practice. Finally, the optimized revenues of the various models will be compared to the true revenues. Several key research objectives are needed to support the study, in particular:

i. Identification and ranking of features relevant to the passengers purchasing decision.

ii. Comparison and evaluation of suitable classification algorithms.

iii. A pricing optimization approach that uses the price sensitivity of the classification model.

iv. An estimation of how much the revenue can be improved by applying this framework and how reliable this result is.

v. Assessment of the additional value of utilizing detailed passenger information.

1.6 Outline

After the introduction, the project starts by evaluating the related work in Chapter 2, giving a brief overview of the demand prediction domain with the most recent relevant papers and the evolution of the most successful classification algorithms. In Chapter 3, the data is presented, prepared, explored and various features like the price feature are created. Chapter 4 introduces all methods of classification and optimization used followed by some practical implementation topics in Chapter 5. The results are presented in Chapter 6 and its limitations and implications discussed in Chapter 7. Lastly, the conclusion and ideas for future work are described in Chapter 8.

Chapter 2

Related Work

This section contains a review of the most pertinent literature. For this purpose, the chapter is divided into two sections. The first Section *Demand prediction & dynamic pricing* 2.1 covers the concepts and motivations behind demand learning with special emphasis on perishable and limited products such as airline seats. This section begins with an overview of the core aims of demand prediction and dynamic pricing followed by a discussion of various possible techniques to achieving them. Following the generic ideas, two interesting articles dealing specifically with aviation and their success stories are presented. The second Section *Predictive Classification Modelling* 2.2 provides a brief history of the most prominent classification algorithms as well as an explanation of their benefits. In Chapter 4 a more extensive explanation and comparison of the algorithms is offered.

2.1 Demand prediction & dynamic pricing

Demand learning is an important part of the pricing process of every company. Most of the times, the market demands are unknown before the start of the selling season, especially as customer preferences are shifting more and more rapidly (Vinokurova [23]). In the sale of perishable or seasonal products, the understanding of customer preferences is particularly critical. These products usually are available in limited capacity and not possible to sell after expiration (Yang et al. [24]), just like seats on a flight. The goal of dynamic pricing is to optimize the revenue by setting dynamic prices of a given stock of items when demand is probabilistic and price sensitive (Gallego and Talebian [12]). In order to predict demand, the content to be learned mainly includes customer arrival rate and distribution of customer willingness to pay (wtp) (Yang et al. [24]).
In 2012, Gallego et al. [12] proposed a rollig horizon framework to choose prices dynamically while simultaneously learning unknown parameters for the arrival rate and core value of the customers

for different seat locations at an event as the sales evolve. The assumption is, that all seats share an unknown core value which then commands a known premium or discount relative to the core value. For the arrival rate, a Bayesian updating approach is applied with Poisson arrival and Gamma distribution as prior for the arrival rate at each time. The core value is estimated using makimum likelihood and the pricing is solved using dynamic programming.

Talebian et al. [21] proposed a dynamic programming model for simultaneously making assortment and pricing decisions for perishable and seasonal products, which uses demand learning based on Bayesian updates. They show that it is profitable for the retailer to use price reductions early in the sales season to accelerate demand learning.

Yang et al. [24] developed in 2022 an agile revenue management framework that combines an LSTM neural network with a Markov decision process (MDP) pricing model. The LSTM model is trained by historical data with already known willingness to pay, and learns to estimate the parameters of the WTP distributions. Like this the WTP can be predicted early in the selling season and revenue optimized utilizing the MDP.

According to Perera et al. [19], there are two main challenges when measuring air passenger price elasticity and implement it with revenue management. First, the required level of granularity required in a dataset to conduct such an analysis is demanding and often difficult for academics to acquire, as disaggregated passenger ticket and booking data is proprietary and of high commercial value. Second, price and passenger demand are endogenously determined variables and have a simultaneous relationship, so that prices are determined by expected passenger numbers and passenger numbers are strongly influenced by the prices (Perera and Tan [19]). The authors have access to high granularity flight data and are combining revenue accounting coupons, booking and ticketing data. They overcome the endogeneity bias by using flow traffic passengers, which continue their trip after the specific flight and are as such uncorrelated with demand shocks in the origin and destination market of interest. With the 2-Stage Least Squares (2SLS) model, they find a demand elasticity consistent with industry observations.

In 2023, Thirumuruganathan et al. [22] developed a price elasticity model (PREM) to identify customers likely to purchase an upgrade offer from economy to premium class and achieve an increase of 37.2% accepted offers. PREM uses denoising autoencoders to obtain embedding-based representations and reduce the feature space. It then performs a cost-sensitivity classification, identifying customers who are more likely to accept an upgrade-offer, while penalizing False-Negatives. With three different binary autoencoders, the customers are classified into their personalized upgrade price bucket, which is either a low, middle or high offer. Lastly, the revenue-maximizer sends upgrade offers to the customers by maximizing the expected revenue.

2.2 Predictive Classification Modelling

Typical machine learning application are building a non-parametric regression or classification on a given data set, to make predictions about the future. These machine learning frameworks can rely on a single strong predictor model, or alternatively use an ensemble of various weaker models to obtain a strong prediction. Popular examples of such ensemble techniques are random forest and neural network ensembles (Hansen and Salamon [13]), which have found many successful applications in different domains (Natekin and Knoll [18]). While ensemble techniques like random forest rely on simply averaging all model outcomes in the ensemble, the family of boosting algorithms is based on the idea of sequentially adding models to the ensemble, training each new model on the errors of the previous one. These boosting ensembles were finally connected with a gradient-descent based formulation around 2000 (Freund and Schapire [10], Friedman et al. [11]). This formulation and the corresponding models were called the gradient boosting machines (Natekin and Knoll [18]).

2.2.1 Classification models

Very recent ensemble methods which rely on gradient boosting include eXtreme Gradient Boosting (XGBoost) introduced by Chen and Guestrin in 2016 [8], Light Gradient Boosting Machines (LightGBM) proposed in 2017 by Ke et al. [16] and Categorical Boosting (CATBoost) from 2018 by Prokhorenkova et al. [20]. XGBoost has consitently been placed in the winning algorithms for various Kaggle competitions (Data mining competitions). CATBoost is promising with regards to generalization accuracy, whereas LightGBM is known for its training speed (Bentéjac et al. [5]).

Chapter 3

Data Description and Preparation

The aim of this chapter is to provide an overview of the raw data, summarize the required pre-processing steps and develop an understanding of the created features. The first section of this chapter will concentrate on the data set at hand, providing a comprehensive understanding of the data bases and the extraction process as well as a preliminary review of all variables in the raw data that were utilized later (Section 3.1). Section 3.2 contains all pre-processing steps, feature engineering and will define all variables that will be used as input for the classification models. The pricing feature will be addressed separately from the other features in Section 3.3, which will discuss the procedure of estimating the missing prices. Finally, in Section 3.4, this chapter will finish with an exploratory data analysis of the most relevant variables.

3.1 Overview Data set

Data bases The data was extracted from the operational performance analysis data base of KLM called *Opera*. Opera is a backward looking application that is used for operational analysis and contains all data on passenger level. Additionally to the operational data base, data from the sales database *Monet* was extracted, to gain insights on the ticket prices paid by the passengers as well as additional connection information. Monet is also backward looking and contains the flown revenue. The data bases are part of the decision support toolbox, which was developed by and for the pricing teams, but is also used by other departments like the cargo, distribution and offer management. Both data bases aggregate various different data sources. This data is relevant for this research, because it contains various information on the passengers and flights as well as the seat the passenger eventually bought which will be used in training and evaluating the classification models.

Data sets extracted The flight data used for the data analysis and later for the model training and testing was collected between the the 16.01.2023 and the 12.03.2023 and contains all flight data of KLM-operated flights on passenger level. The data set was split by Revenue Group into two data sets: **Medium haul ds** contains all *medium haul* flights in the six weeks from 16.01 to 26.02.2023 and **Intercontinental ds** comprises all *Intercontinental* and *North Atlantic JV* flights in the eight weeks from 16.01 to 12.03.2023. Intercontinental and North Atlantic JV are separate revenue groups because AFKL has a joint venture with Delta Airlines which concerns all flights to and from North America. The Medium haul data set covers a shorter time period as it contains significantly more flights and passengers than the Intercontinental data set. A brief overview of the two data sets is given in Table 3.1. A *data point* is considered one row in the data, which represents a passenger with a corresponding flight.

Data set name	Data points (Raw)	Data points filtered	Period
Intercontinental ds	[redacted]	1,117,445	16.01 - 12.03.2023
Medium haul ds	[redacted]	1,585,090	16.01 - 26.02.2023

Table 3.1: Overview Data sets

Joining operational and sales data Joining the extracted data from Opera and Monet provided a small challenge, as this is never done in practice. When a passenger buys a flight, one gets a First Serial Number for their purchase, meaning that this number is the same for all flights included in their journey which might be multiple legs or even a return flight. The single flights on a ticket are then enumerated as Coupons, and the number of the Coupon is appended to the First Serial Number to form the Serial Number. In Monet the numbers are part of the data, in Opera however only a Ticket number is given, which corresponded to the First Serial Number. After converting all numbers to floats, the join could be performed on the First Serial Number and Ticket Number, each combined with the Flight Number which makes the First Serial Number unique.

Variables in raw data In tables 3.2, 3.3, 3.4 and 3.5, an overview of all variables that were considered or used in any way is given as they appeared in the downloaded data set. Most of the data comes from the operational database Opera, but those features joined from the sales database Monet are marked with an (M) behind the name. Explanations of the features will be given as they are used in the following sections.

Variable	Type	Range	Unknown
Booked Cabin	Ordinal	{2. Business, 4. Economy}	0%
Flown Cabin	Ordinal	{1. First, 2. Business, 4. Economy}	0%
Flown Section	Ordinal	{Business, Economy Comfort, Economy Extra Legroom, Economy Front, Economy Standard (SSS), Economy Restricted, First}	9%
Flown Row	Nominal	{R01,...,R73}	13%
Flown Type	Nominal	{Window, Aisle, Middle}	9%
Flown Seat	Nominal	{01A,...,74D}	13%
Seat Realization RES	Nominal	{Relocated, Realized, No Request}	0%
Seat Reservation Type	Nominal	{Non Reserved, Free Seat Selection, ASR Group, ASRW, Paid Seat Selection, ASR paid}	6%
Seat Paidness Type	Nominal	{Paid Normal, Paid Miles, Paid Upgrade, Lucky, Flying Blue Silver/ Gold/ Platinum, Branded Fare, Downgrade {Section}, Restricted}	13%
Revenue Group	Nominal	{Medium haul, Intercontinental, North Atlantic JV}	0%
Region	Nominal	{Europe, US, Canada, Asia, ...}	0%
Complex Group	Nominal	{Spokes West, Spokes East, Hubs, NYC, Canada, Mexico, Gulf,...}	0%
Complex	Nominal	{Spokes West, Spokes East, Hubs, NYC, Canada, Mexico, Gulf, Brazil, Oil Africa, West Africa,...}	0%
Flight	Nominal	{KL0601_AMSLAX,...}	0%
Subline	Nominal	{AMSLAX, AMSRUH/DMM, ...}	0%
Flight Segment	Nominal	{KL0601_AMSLAX, KL0423_AMSRUH, ...}	0%
Day	Ordinal	{01/16/2023,... 03/12/2023}	0%
Scheduled Time Departure	Nominal	{...}	0%
Scheduled Time Arrival	Nominal	{...}	0%
Aircraft Subtype	Nominal	{781, 772, 789, 77W, 333, 332, 73J}	1.1%
Aircraft Detailed Subtype	Nominal	{781C, 772F, 789A, 77WC, 333B, 332L, 781A, 789B, 73JC}	0.1%
Trip Type	Nominal	{Unk, Out, In, Single}	1%
Point of Origin	Nominal	{Rest, Norway, Austria, ...}	1.7%

Table 3.2: Categorical Variables in raw data set (1)

Variable	Type	Range	Unknown
Saturday Stay	Nominal	{Yes, No, Unknown}	1%
Weekend Stay	Nominal	{Yes, No, Unknown}	1%
PNR	Nominal	{Oz2FAC, ...}	7.5%
Booking Agent Channel	Nominal	{Unknown, A - Business, B - Speciality Sales, G - Generalist, N - E Commerce, L - Offline Leisure, V - Call Center, I - Online Leisure, P - City Ticket Office, R - Others, Q - Airport Ticket Office, Y - Other Airlines, W - Corporate Desk, S - Swipe, E - Ticket Issued Center}	14.3%
Direct/Indirect - Online/Offline	Nominal	{Direct - Online, ..., Indirect - Offline}	14.3%
Sky Priority	Nominal	{Sky Priority, No Sky Priority}	0%
FF Level	Ordinal	{None, Non-FB Program, Non-Skyteam, Explorer, Silver, Gold, Platinum}	7.9%
Corporate Level	Ordinal	{Unleveled, A,B,C,D,E}	7.3%
Paid Option Touchpoint	Nominal	{None, Mobile, Blueweb EBT, Blueweb MyTrip, Deltamatic, Multiple, Indirect Online, Indirect Offline, Direct Online other, Direct Offline Other, Call Center, City Ticket Office, Airport Ticket Office, Blueweb ICI, Local Kiosk,...}	0%
Paid Option EMD Segmentation	Nominal	{None, Direct Online, Multiple, Indirect Online, Indirect Offline, Direct Offline}	1%
Paid Options	Nominal	{None, Economy Comfort, Extra Legroom, Multiple, Standard Seat, Front Seat, Duo Seat, Standard Seat Premium Economy}	0%
Branded Fares	Nominal	{Standard, No Brand, Flex, Light, Comfort Plus}	17.1%
Farelogix Seat Merch. (NOM)	Nominal	{No EMD, NOMP, No Farelogix, NOMB, NOMP-Disc}	0.1%
Gender	Nominal	{Male, Female}	7.2%
Nationality	Nominal	{NO, AT, US, ...}	15.2%
Ticket Number	Nominal	{1117120474, ...}	7.2%

Table 3.3: Categorical Variables in raw data set(2)

Variable	Type	Range	Unknown
Connecting Hub Carrier	Nominal	{AF, KL}	0%
Hub Connecting Time	Ordinal	{00:05, . . . , 24:00}	0%
Connecting Before Flight (M)	Nominal	{KL0601_AMSLAX,. . . }	0%
Connecting After Flight (M)	Nominal	{KL0601_AMSLAX,. . . }	0%
Travel Motive (M)	Nominal	{Leisure, One Way, Business}	0%
Before Connection Time (M)	Ordinal	{0:00,. . . , 24:00}	0%
After Connection Time (M)	Ordinal	{0:00,. . . , 24:00}	0%

Table 3.4: Categorical Variables in raw data set(3)

Variable	Type	Range	Unknown
Length of Stay	Discrete	{0,1,2,. . . ,30+}	0%
PNR Size	Discrete	{1,2,. . . , 10+}	7.2%
Age	Discrete	{0,1,. . . , 99}	12.5%
Baggage Amount	Discrete	{0,1,2,3+}	7.2%
Booking Lead Time	Discrete	{0,1,. . . , 90+}	1.8%
Flight Capacity	Discrete	{0,8,12,. . . , 306}	0%
Cabin Capacity	Discrete	{0,8,12,. . . , 374}	0%
Section capacity	Discrete	{0,1,2,. . . , 104}	0%
Economy Comfort (Rev)	Continuous	{0,. . . , 1076}	0%
Ticket price (M)	Continuous	{0,. . . , 15.350}	0%

Table 3.5: Numerical Variables in raw data set

3.2 Data Preparation

3.2.1 Filtering

The aim of the clean data set is to clearly identify and characterize all passengers who bought an EC seat and and those who decided not to buy an EC seat. While unknown data and outliers will be handled in the following section, it is necessary to ensure that all passengers can be categorized into buyers and non-buyers. Customers with the frequent flyer status *Platinum* can choose an EC seat for free, hence do not pay with money but rather with their loyalty (which they collect via the *Frequent Flyer (FF)* loyalty program). In terms of available seat capacity the Platinum members count just like all passengers who pay for their seats, but as the main goal of this project is to perform pricing optimization on the seats, a passenger class that always gets the seats for free will be misleading in the classification task and is excluded from the data.

3.2.2 Unknown Data/ Outliers

Many of the unknown values in the collected data can be explained and are not due to faulty measuring or other errors. Most prices of seats are given per flight leg, however there are cases where passengers assume a seat on a segment flight and have a stop over in another airport along the way. For these passengers, the seat price is given for the whole segment, as one remains seated during the stop over and does not change the airplane, but as the data is reported per leg, it appears as if the passenger pays the same price for both flight legs, although the price covered the seat on the entire segment. As it is unclear how to split this price among the legs and the passengers also do not have the option to book seats for the flight legs separately, all these cases were combined as single legs. For example a flight to Entebbe, Uganda (EBB) is not possible on a direct route, but has to make a stop in either Kigali, Rwanda (KGL) or Kilimanjaro, Tanzania (JRO). All passengers flying to Uganda over e.g. Kilimanjaro will not appear in two different flight legs AMS/JRO, JRO/EBB anymore, but instead the flight length will be aggregated and the passenger will now appear with only the flight segment AMS/EBB.

Frequent Flyers If a value in the `FF Level` feature was not known, that was most likely because the person was not logged in with the user account when buying the seat, but only with the booking number. Because frequent flyers would usually like to collect points and are most likely logged in automatically, all unknown flyer levels were converted to *None*.

Booking number If there was no `PNR` assigned to a data point, it was not a passenger but contained general information about a flight. These rows were dropped, after the information was extracted and joined to the respective passengers (see Section 3.2.3).

Age All unknown Ages where drawn from the overall normal distribution of the age of all passengers.

3.2.3 Feature Engineering

In the following paragraphs all features that were not originally included in the raw data are briefly explained along with their creation process.

Business traveller The travel motive of the passengers is identified using the features `Weekend Stay` or `Sunday Stay` and the `Length of Stay`. On medium haul flights a passenger is considered a Business traveller if the trip is shorter than 3 days and not over the weekend. Business travellers on Long Haul flights are staying less than 6 days excluding a Sunday. On one-way trips the length of stay is unknown and can as such not be considered a business trip according to the aforementioned rules. As these rules are very strict and certainly do not detect all business travellers, the following features were examined additionally: `Booking Agent Channel`, `Corporate level` and `Corporate Flag`. All passengers that booked their tickets in the `Booking Agent Channel` category *A - Business* or *W - Corporate Desk*, as well as all passengers that were flagged as corporate or were assigned a corporate level were moreover identified as business travellers.

Child The binary feature `Child` indicates whether a passenger is younger than 18.

Male To obtain the feature `Male`, the feature `Gender` was modified into a binary feature indicating whether a passenger is male.

Connections Various different features were generated from information on the connections. The feature `Part of Connection` is a binary feature. A flight is part of a connection if the flight has a preceding or succeeding flight recorded, (`Connecting After Flight` and `Connecting Before Flight`), if it has a connecting hub Carrier (`Connecting Hub Carrier`) and if it has a connection time assigned (`Hub Connecting Time`). The binary feature `After Connection` specifies whether there is a connecting flight waiting after the current flight. It is assumed that a connecting flight is waiting if there is a connection time given by the feature `After Connecting Time`. The additional feature `Connection Time` is constructed by merging the features `After Connecting Time` and `Before Connecting Time` to get all connection times regardless when they occur. The

feature `After Connection Time` is kept and used, but was renamed to `Connection Time (A)`. All connection times are converted to integer values of the time in minutes.

Weekend Trip The feature `Weekend Trip` was build by combining the features `Trip Type` and `Weekend Stay`. `Trip Type` has the attributes *In, Out, Unknown* and *Single*, where *In, Out* mean the flight is inbound or outbound, so the start or end of the trip. If only one flight was booked on a passenger number, the value will be *Single*, and if more flights are booked, but it is a round trip and not clear where the trip starts and ends, the feature is set to *Unknown*. The feature `Weekend Stay` simply states if the trip includes a weekend or not, which is never known for *Single* trips.

Origin/Booking comp The `Point of Booking` gives the country where the respective passenger number (PNR) was created whereas the `Point of Origin` gives the country of the first station of the trip. This feature compares whether the two locations coincide or differ.

Length of Flight The duration of a flight was not given in the operational data directly. The departure and arrival times of the flights were given in local time which required the conversion into the same time zone. By using a separate table containing all airport codes with corresponding time zone identifier, the time zones could be joined to the arrival and departure airport of the flights. After converting the times into the respective time zones, the length of the flight was calculated in minutes.

Arrival Time & Departure Time The arrival and departure time was already given as a timestamp of the respective local time in the raw data. As this data is not simply categorical, but ordinal, the times were encoded into quarters of the day, making it a numerical feature. For instance 1 am was encoded to 4, as it is the fourth quarter of the day. Additionally, these features were binned into *morning, midday, afternoon, evening* and *night*. All times between 10 pm and 4 am are considered *night* times, followed by *morning* between 4 am and 9 am. *Midday* was considered everything from 9 am to 1 pm, when the *afternoon* starts which ends at 4 pm. Finally the *evening* times were set from 4 pm to 10 pm.

Overnight A flight was considered an over night flight if the local departure time was earlier than 12 am and the arrival time later than 2 am the next day. This feature was not created for medium haul flights.

Section Capacities The capacity of the EC section underlies an additional challenge. Most of the medium haul flights have a so called *movable curtain* between the Business and Economy Cabin. This curtain is fixed 72 hours before departure and adjusts not only the number of seats in the Business and Economy Comfort Section, but might also change the total number of available seats on the aircraft. In bigger aircrafts with a middle row, the middle seats are blocked if the entire row lies within the Business Section, but is available in the Economy Comfort section. In the raw data, there is one dummy row (with no PNR number as mentioned previously) added for every Section on every flight which gives the section capacity on that specific flight. Grouping by the unique flights, the capacities for every section and the number of total Cabin seats were extracted, which also revealed the total capacity of the aircraft. In this way, the features `Total capacity`, `Economy capacity` and `Economy Comfort capacity` were created. Afterwards, the dummy rows were removed from the data set.

Label creation A label is required for each data point in supervised learning classification models. In this situation, the label will indicate whether or not the associated passenger purchased a seat in the EC section. The `Economy Comfort (Rev)` feature provides information about how much clients spent on EC seats. If this sum is larger than zero, the customer purchased an EC seat, regardless of what happens later, such as downgrades, relocation, or upgrades. Not everyone who paid for an EC seat flew in this section, but to distinguish between those who intended to buy the seat and those who did not, labeling all customers who paid as buyers and all others as non-buyers is adequate.

Normalization by Capacity As some aircrafts hold significantly more seats of a specific section or seat type than others, the data points have to be normalized accordingly. A passenger buying a seat in the EC section would not count as 1, but was divided by the EC section capacity of the respective flight. Just in the same way, the passengers not buying a seat were divided by the total capacity of the flight leaving out the EC section. The feature `Economy Comfort paid norm` holds the capacity weighted count of each passenger and will be used in the data exploration Section 3.4.

3.2.4 Selected features

Following the previously mentioned filtering, cleaning and engineering sections, an overview of all the features used as model inputs along with their ranges is given in Table 3.6. All features that were not created but retained as they appeared in the raw data are briefly discussed; an explanation of all other features can be found in the previous section (3.2.3). Note, that the features `Arrival`

`Time` and `Departure Time` were later excluded from the modelling. Three categories can be used to group the features:

Passenger information The variables `Age`, `Child` and `Male` provide details on the traveller.

Booking information Following the passenger details, there is specific information about the booking, which consist of the features: `Baggage Amount; PNR Size`, which represents the number of people in the booking; `Weekend Trip`; `Business traveller`; `Direct/Indirect - Online/Offline`; `Booking Lead Time` which gives the time of the ticket purchase in days before departure, and `Origin/Booking comp`. Note that while there can be more than one person in a booking with the same passenger number (PNR), each of the passengers is presented individually in the data. The feature `Direct/Indirect - Online/Offline` indicates where the ticket was bought. An example for a *Direct - Offline* channel is a ticket issued center or a city ticket office. A *Direct - Online* channel is for example any E-commerce platform like the website or app. Indirect channels are other travel agencies or airlines, where passengers also have the option to buy either online via a website or in the travel shop.

Flight information Lastly, some features are used to describe the flight itself, implying that these values are most likely the same for all passengers on the trip. This includes information about the origin and destination (`Point of Origin, Complex Group`) and details about the connecting flights such as `Part of Connection, After Connection` and `Connection Time` as well as the additional insights `Flight length, Arrival Time, Departure Time, Overnight` and `Economy Comfort capacity`. The feature `Point of Origin` already exists in the raw data, and denotes the starting country of the journey. Although medium haul flights are mostly within Europe and do not reach as far as Canada for example, the `Point of Origin` in the intercontinental and medium haul data sets still contain all countries, as many intercontinental journeys also include a shorter flight from a hub like Amsterdam. For instance a flight from Canada to Lisbon goes via Amsterdam, meaning that the corresponding medium haul flight from Amsterdam to Lisbon will appear in the data with Canada as `Point of Origin`. The destination region of the flight are represented by the `Complex Group` and are aggregated into areas bigger than single countries like Middle East & Gulf or South East Asia.

While most characteristics about passengers and bookings are similar on medium haul flights compared to intercontinental flights, especially the flight details can have different values. There are for instance no medium haul flights over night, the aircraft types are different depending on the

flight distance and also the destinations can be anywhere for intercontinental flights while medium haul flights target countries within and close to Europe.

Variable	**Type**	**Range**	
		Intercontinental	**Medium haul**
`Age` (years)	Discrete	{0,. . . , 99}	
`Child`	Binary	{0,1}	
`Male`	Binary	{0,1}	
`Baggage Amount`	Discrete	{0,1,2,3}	
`Weekend Trip`	Nominal	{In No, In Yes, Out No, Out Yes, Single, Unk Yes, Unk No}	
`PNR Size`	Discrete	{1,. . . , 10+}	
`Business traveller`	Binary	{0,1}	
`Direct/Indirect - Online/Offline`	Nominal	{Direct Online, . . . , Indirect Offline}	
`Booking Lead Time` (days)	Discrete	{0,. . . , 90+}	
`Origin/Booking comp`	Binary	{0,1}	
`Point of Origin`	Nominal	{United States, Canada, Netherlands, . . . }	
`Complex Group`	Nominal	{Middle East & Gulf, East Africa, . . . }	{UK Ireland, Nordics, BeNeLux, East Europe, . . . }
`Part of Connection`	Binary	{0,1}	
`After Connection`	Binary	{0,1}	
`Connection Time (A)` (minutes)	Discrete	{0,. . . , 1440}	
`Connection Time` (minutes)	Discrete	{0,. . . , 1440}	
`Flight length` (minutes)	Discrete	{270, . . . , 1020}	{45, . . . , 225}
`Arrival Time` (quarters of day)	Discrete	{0, . . . , 95}	
`Departure Time` (quarters of day)	Discrete	{0, . . . , 95}	
`Overnight` I	Binary	{0,1}	
`Economy Comfort capacity`	Discrete	{27, 30, 36, 40, 42, 48}	{4, 6, 8, 12, 16, 18, 20, 24, 30, 36, 42}

Table 3.6: Features used in modelling

3.3 Price feature

In the business environment, the hypothesis is that the price has a great influence on the purchasing decision of the customers. As this project tries to optimize pricing, this feature is critical and will be explained in detail.

The data from the operational database contains the prices the passengers paid for their seats, specifically the prices of the EC seats are shown by the feature `Economy Comfort (Rev)`. In most

of the cases this value is zero, because the passenger did not buy an EC seat. This provides a challenge, as now only the accepted prices are recorded, the declined prices by the passengers are however unknown and the average price they saw most likely when checking the seating options has to be estimated. In theory, there exists a base price per destination and per month which holds for all passengers and is recorded in a table in the ATPCO pricing system. As already mentioned in the introduction, following simple business rules, the prices can be discounted up to 10% in the last days before departure depending on the load of the EC section. In some cases, a heavy discount is granted at check-in, when the other sections in the Economy cabin are overfilled, which would mean that people would have to be upgraded for free into the EC section to make everyone fit. On top of these discounts, there are certain discounts for the EC seats granted to members of the frequent flyer program at all times. Passengers with the status *Explorer* receive a 10% discount on EC seat prices, *Silver* members get a 25% discount and *Gold* members buy the seats at a discount rate of 50%.

3.3.1 Unknown price estimation

To use the price as a feature in the prediction model, the prices that were seen by all passengers who did not buy a seat have to be estimated. The data was extended by the base price as it was given for the respective flight and month in a separate table from the ATPCO pricing system. After reversing the discounts of *Explorer, Silver, Gold* members (by multiplying all prices of *Explorers* by 10/9, ...) the prices paid by the passengers still deviated significantly from the base price due to various different discounts. The idea was to look at the average discounts that the passengers received on a flight level and to draw the same discounts from a normal distribution for the price estimation. Therefore it was necessary to clean the prices from any outliers that might effect the discount distribution. At first the heavy discounts at check-in were detected as outliers and removed from the prices. They were recognized by the attribute *NOMP_disc* of the feature `Farelogix Seat Merchandizing (NOM)`. After removing these discounts, the higher discounts of the `FF Level` *Non-FB Program* are noticeable in Figure 3.1. *Non-FB Program* is short for *Non Flying Blue Program*, meaning that the passenger is not collecting any loyalty points. A reason for the higher discounts of these passenger is that some were booked via Delta Airlines over the joint venture with AFKL, in which case they might have a higher FF level with the corresponding discounts in the loyalty program of Delta Airlines, but appear as *Non-FB Program* in the AFKL system. At this point only about 4.5% of the passengers have this FF level, which is why these prices are removed as well. Lastly, all discounts that lay further away from the overall mean discount than 3 standard deviations where marked as outliers and removed. The average mean discount was approximately 0.0326 with a standard deviation of 0.0702. That means that discounts outside of $[-0.178, 0.2432]$

were dropped. In Figure 3.2 a comparison of the prices in the raw data and the cleaned prices without outliers is given over all flights of the intercontinental data set. While the average discount could be slightly reduced by cleaning the prices, the standard deviation decreased significantly.
After computing the mean and standard deviation of the discount on flight level, the unknown prices for all passengers can be estimated, by drawing the discount from the normal distribution of the respective flight of each passenger, and apply this to the base price of the passenger. Additionally, the frequent flyer discounts were applied, to obtain the effective price the passengers would have paid.

3.3.2 Price feature engineering

As the price feature is crucial for the price optimization succeeding the modelling process, multiple different price features were created which were tested and evaluated in experiments further explained in the methods (Chapter 4). The price features created mostly differ in two ways: 1. **Overwrite** captures whether the known prices were overwritten and has the distinctions *no overwrite, part overwrite, full overwrite*. *No overwrite* means that only the unknown prices were estimated but the already known prices were kept, including all outliers and heavy discounts. *Part overwrite* means that all prices that were not used in determining the base discounts, thus were handled as outliers (see previous Section 3.3.1), are overwritten with the same estimation method used for the unknown prices. *Full Overwrite* means that all prices were estimated, overwriting all previously known prices. 2. **Discounts** defines whether frequent flyer levels have their usual seat discount reflected in the price or not.
All the price features are binned into 5% equidistant bins and all are normalized by the base price. The lowest bin is larger and contains all prices between 0.0 and 0.5, as there are not many customers in this price segment who are mostly *Gold* members of the frequent flyer program. The highest price bin contains prices from 1.05 to 1.1. While the buckets are defined as intervals, they will be replaced by the interval's mean, converting it to a numerical feature. Because in the lowest bucket, most prices are only slightly lower than 0.5, this bucket will be represented by a price of 0.475. The prices are given in fractions of the base price, meaning that a price of 0.75 means 75% of the base price.
It was considered as well to normalize the seat prices by the ticket price, the flight length or not at all, but these methods achieved similar results so for a better comparability and easier optimization on the prices later it was decided to normalize all prices by the base price.
The final price feature `ATPCO` contains the fully overwritten prices with all discounts. `ATPCO nodisc` holds the same fully overwritten prices but the discounts of frequent flyers are corrected and `ATPCO`

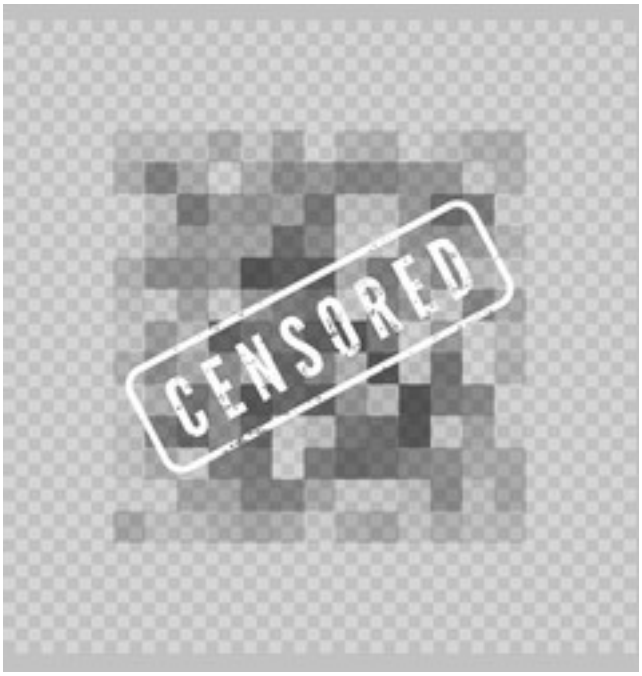

Figure 3.1: Discount distribution per FF Level

`partov` and `ATPCO noov` have all discounts included but the prices are only partly, or not at all overwritten respectively.

Price feature name	Overwrite	Discounts	Normalization
`ATPCO`			`ATPCO price`
`ATPCO nodisc`			`ATPCO price`
`ATPCO partov`			`ATPCO price`
`ATPCO noov`			`ATPCO price`

Table 3.7: Price features

3.4 Exploratory Data Analysis

In this section, various plots will be presented to gain a better understanding of the most important features which are summarized in Table 3.6. There are six figures with nine separate plots, each displaying data of a different feature. In figures 3.3 and 3.4 the frequencies of the values of all numerical features are shown for intercontinental and medium haul flights respectively. In tables 3.5 and 3.6 the corresponding trends with regards to the purchase probability are plotted for the same numerical features. Lastly, in figures 3.7 and 3.8 the trends with regards to the purchase probability are shown for nine selected categorical features.

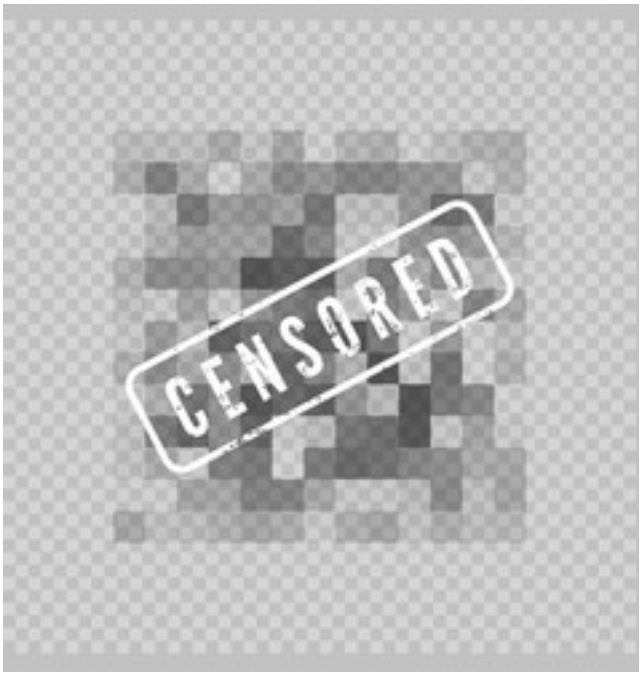

Figure 3.2: Comparison of the distribution of discounts on the flight AMS - DMM before and after outlier removal.

When the purchase probabilities in the groups are compared, a chi-square test was applied to the contingency table which holds the normalized count of passengers with label 0 and label 1 (buyers and non-buyers), comparing in turn each of the groups for statistical significance. Most trends are already likely to be significant by looking at the plots, but in this way a statistical measure is used to support the observation.

3.4.1 Numerical & categorical features

Starting with tables 3.3 and 3.4 a good overview of the frequencies of the numerical features can be established. Comparing intercontinental and medium haul flights, it can be observed that medium haul passengers book less in advance and travel often without larger baggage, whereas most intercontinental passengers travel with at least one suitcase. As there are no medium haul flights over night, there are in contrast to intercontinental flights no arrivals early in the morning or departures very late in the evening. Many medium haul flights are not part of a connection, which is why the connection time is lower than on intercontinental flights. The different aircraft types used for the two different flight types is indicated by the different Economy Comfort capacities, where bigger aircrafts used on longer flights have more capacity.

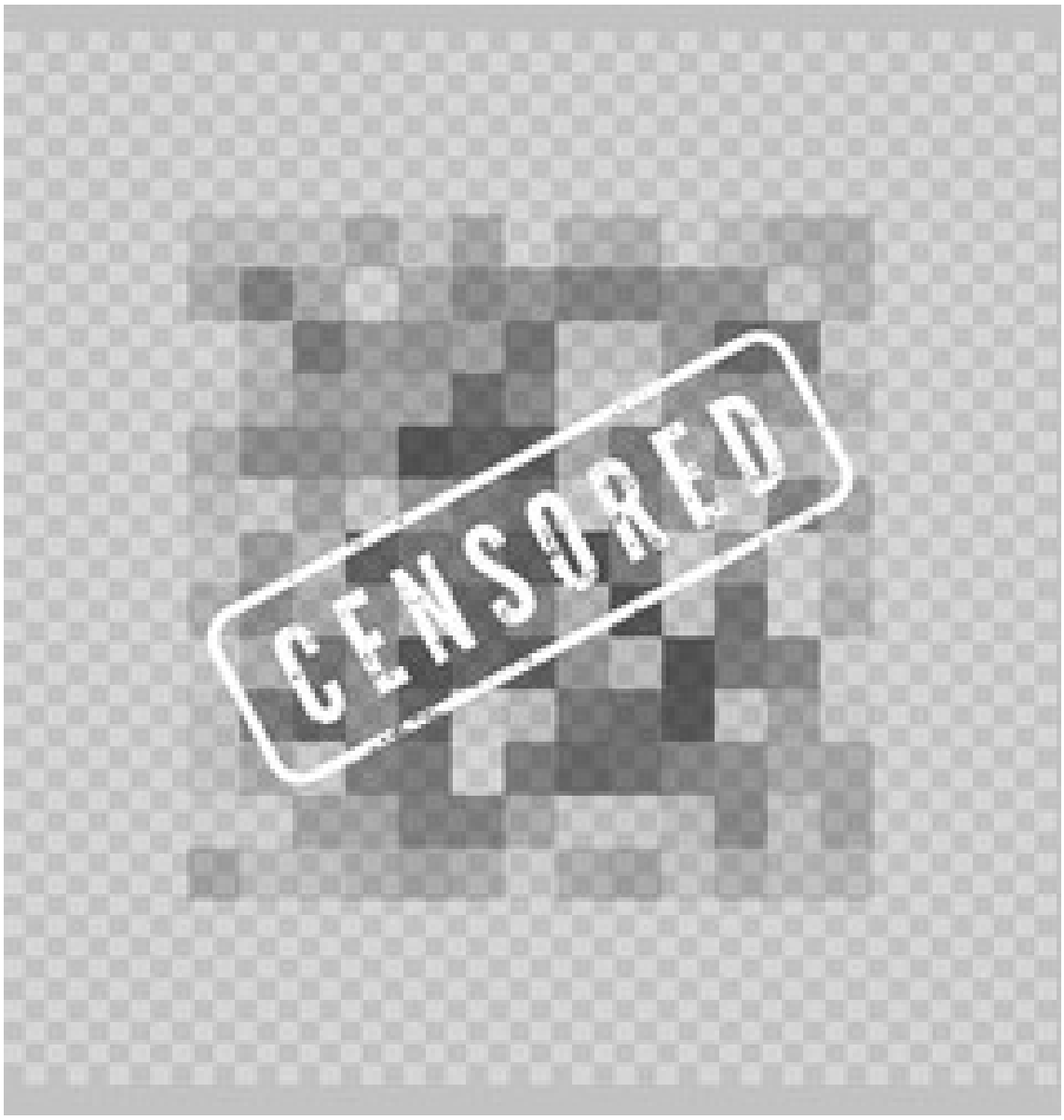

Figure 3.3: Numerical features frequencies intercontinental

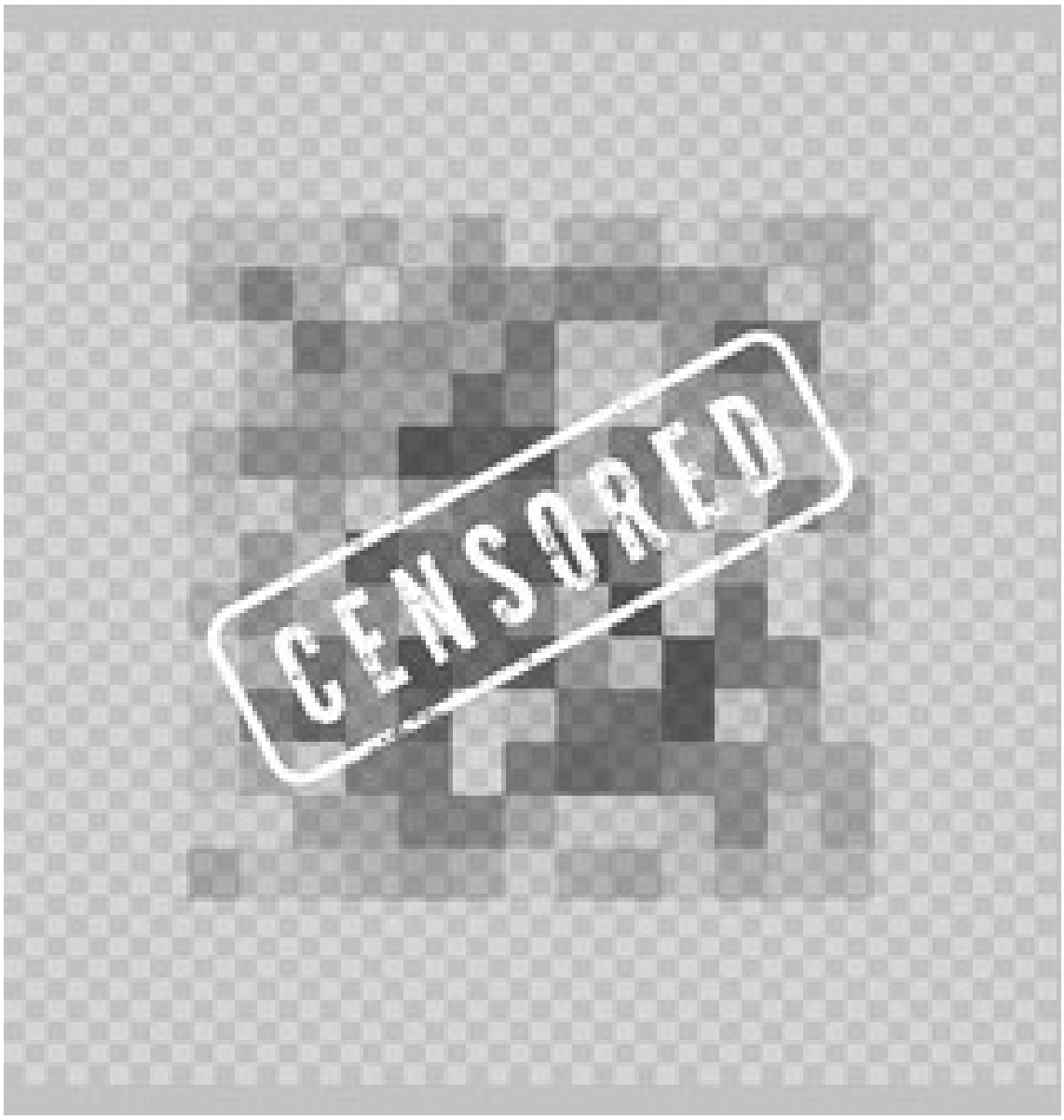

Figure 3.4: Numerical features frequencies medium haul

After looking at the frequencies of the numerical features, the influence of these features on the purchase behaviour can firstly be reviewed by binning the features and comparing the average normalized purchase probabilities within the bins with each other. Some features like `PNR Size`, `Baggage Amount` or `Economy Comfort capacity` did not need binning, as they are already comprised of only a few discrete values. Other features like `Arrival Time` and `Departure Time` were binned as explained in the previous feature engineering Subsection 3.2.3. The remaining features were binned into an amount of bins (usually 10) based on their quantiles, so that every bin

holds approximately the same number of passengers. Per bin, the count of buyers and non-buyers normalized by section capacity (see Subsection 3.2.3 paragraph *Normalization by capacity*) was utilized to compute the probability of purchase in that group. The results are displayed in figures 3.5 and 3.6.

On intercontinental flights, multiple trends can be observed. With more people in the booking (increasing `PNR Size`), the likelihood of an EC seat purchase clearly decreases. With more people in the booking, it is in general harder to find seats next to each other due to availability, especially in a quiet limited section like the EC section, which might influences this finding. The probability of purchase seems to increase as well with increasing age, which might be explained by higher income or higher need/willingness to pay for comfort. People travelling with more suitcases are less likely to also pay for an EC seat. Passengers without connection (indicated by a connection time of 0 to 10 minutes), buy significantly more often than passengers with connection, where an increase in connection time has a slightly negative effect on the purchase probability. Customers booking with more time in advance have a higher probability of purchase, this could be affected by the fact that people who buy further in advance, have more opportunities and time to still buy their EC seat than someone who books shorter before departure. One could also think that this trend is simply due to a reduced availability of EC seats closer to departure. For medium haul flights, the same trends are visible for connection time and booking size as on intercontinental flights. Moreover, the age seems to have the same positive correlation with the probability of purchase as on intercontinental flights, however less distinct. This might have to do with the shorter flight duration and the less difference in comfort compared to the respective standard seats. A smaller booking lead time does not lead to a decrease in purchased seats. That can be explained with the movable curtain (see Section 3.2.3 paragraph *Section capacity*), that the capacity can in the most cases be adjusted 3 days before departure, which is why there will most likely be a sufficient availability, sometimes maybe even only sufficient for the last three days before departure. On the longest medium haul flights, the purchase probability is significantly higher for the longest flights than for shorter flights. It can furthermore be observed that in aircrafts with less EC capacity more seats are bought.

All features have at least one pair of attributes that signals significant differences, except `Arrival Time` and `Departure Time` which have no significant influence on the purchase behaviour in neither medium haul nor intercontinental flights. For this reason these two features will be excluded from the input for the classification models.

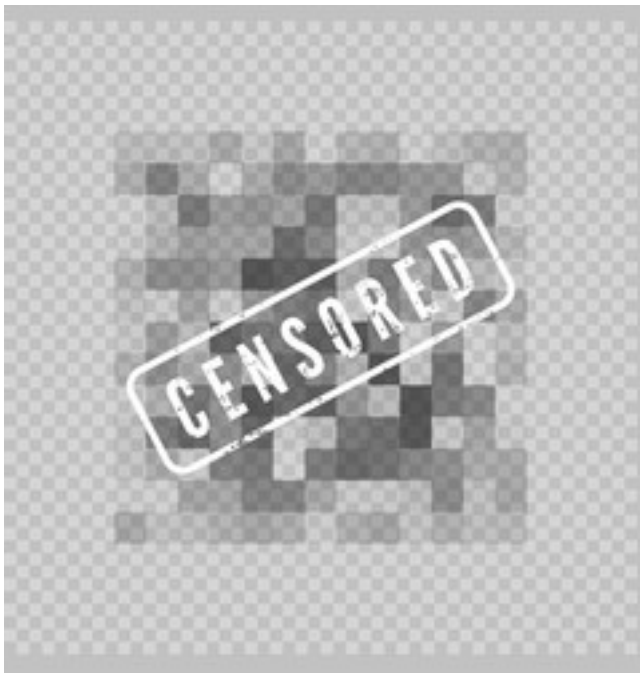

Figure 3.5: Numeric features purchase probabilities intercontinental

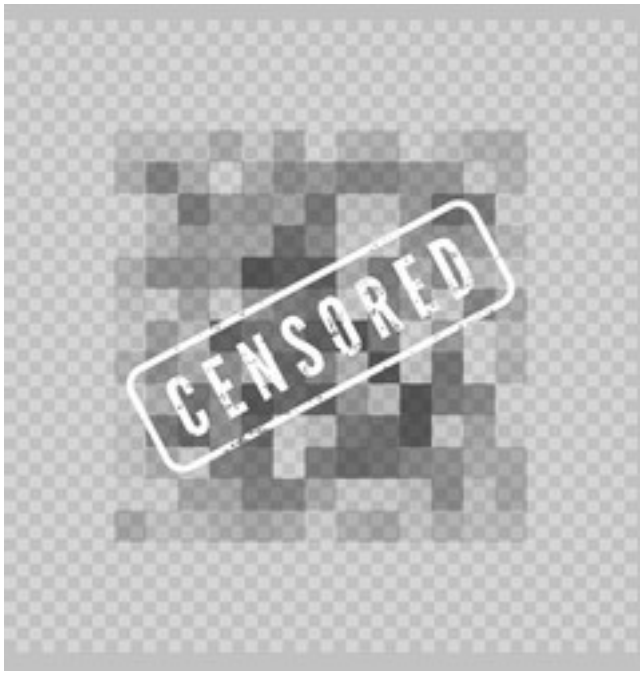

Figure 3.6: Numeric features purchase probabilities medium haul

Just like for the numerical features, also the categorical features show significant influences on the purchase decision of the customers. For intercontinental passenger the feature plots are displayed

in Figure 3.7. A clear observation is that passengers of a higher frequent flyer level buy significantly more seats than of a any lower level. Whether this is related to the price or to other attributes that characterize frequent flyers cannot be concluded here. Business travellers and customers not staying over the weekend buy more seats, than others. Many of these passengers are overlapping, leading to the same trend. As seen before, passengers without connection on their trip are more likely to purchase an EC seat.

Medium haul passengers show the same behaviour as intercontinental customers, although the trend for business travellers is less distinct. An additional influence that is very significant is the channel the ticket was bought. On medium haul flights, the percentage of seat purchases in the *Direct - Online* channel is significantly greater than in all other channels. Especially the direct offline channel sells a great amount of EC seats less on medium haul flights than on intercontinental flights.

All categorical variables seem to have an influence on the purchase behaviour, which is why all of them will be used as input for the classification model.

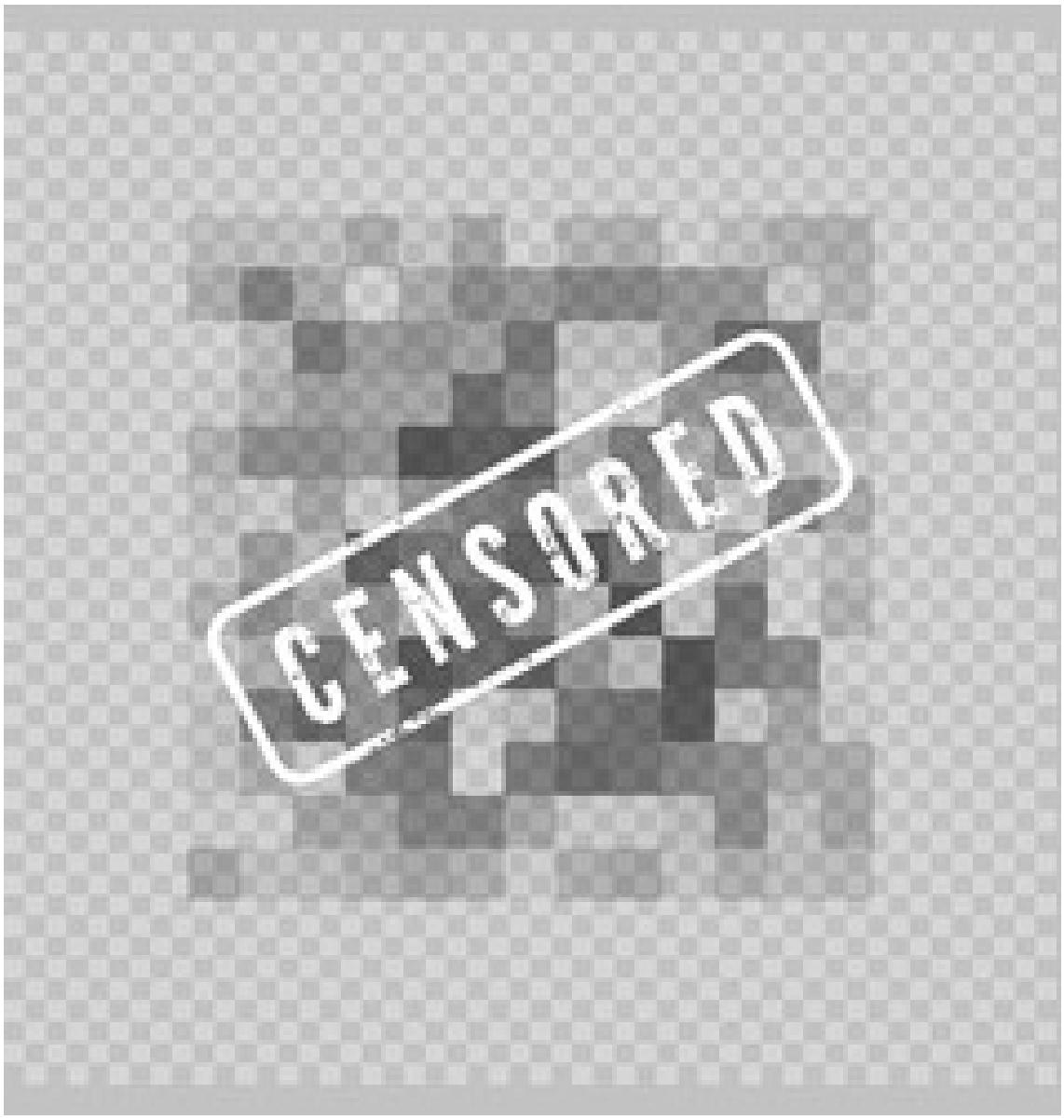

Figure 3.7: Categorical features purchase probabilities intercontinental

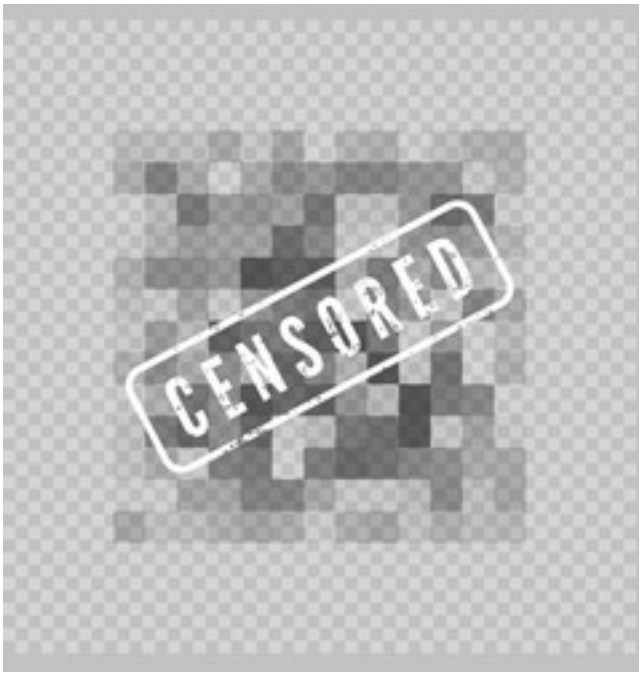

Figure 3.8: Categorical features purchase probabilities medium haul

3.4.2 Price feature

In figures 3.9 and 3.10 the four different price features and their effect on the probability of buying an EC seat are plotted. A negative price elasticity is essential to later optimize the prices. From the plots it can be seen, that taking out the discounts of frequent flyers like in the feature `ATPCO nodisc` is not good for the price elasticity of intercontinental and medium haul passengers, as it seems that a higher price does not change anything in the purchase behaviour. In the feature `ATPCO noov` where no prices are overwritten, all customers who receive a lower price than 0.775 are buyers, as the price estimation would usually not assign higher discounts than that from the normal distribution. When partly overwriting the prices as in `ATPCO partov`, there seems to be partly the same problem, as the price elasticity suddenly drops significantly when increasing the price. Lastly, the price feature `ATPCO` delivers a more or less smoothly decreasing purchase probability for an increasing price.

3.4.3 Feature correlation

As last part of the exploratory data analysis, the correlation between the selected features is shown in figures 3.11 and 3.12 for intercontinental and medium haul passengers respectively. The seat price feature is the price feature `ATPCO`, as this will be identified as the most suitable price feature in the following sections.

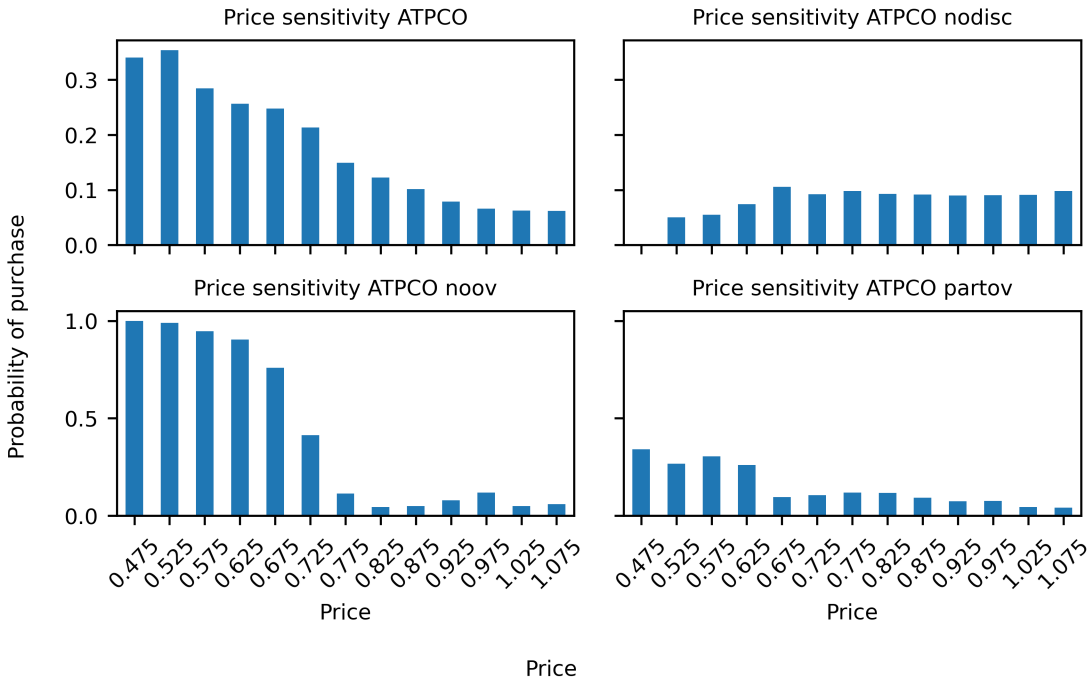

Figure 3.9: Price sensitivity before modelling intercontinental

One can see from the graphs that the highest correlations occur within the features describing the connection of the trip like `Connection Time, After Connection` and `Connection Time (A)`, which is not surprising, as these features are very similar. On intercontinental flights, the flight length is correlated with the complex group. This is due to the fact that most intercontinental flights start in a hub, which is either Paris or Amsterdam, and thus the flight length is usually very similar for the same destination regions. None of the features seems to be correlated significantly with the seat price, which is important for obtaining a clean price elasticity.

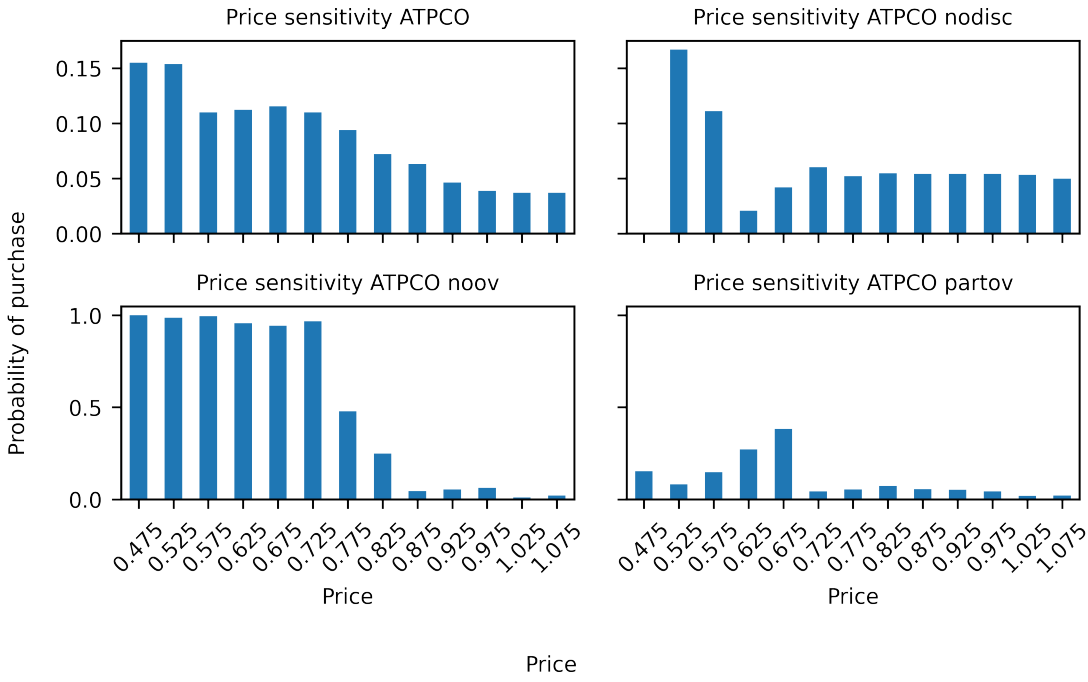

Figure 3.10: Price sensitivity before modelling medium haul

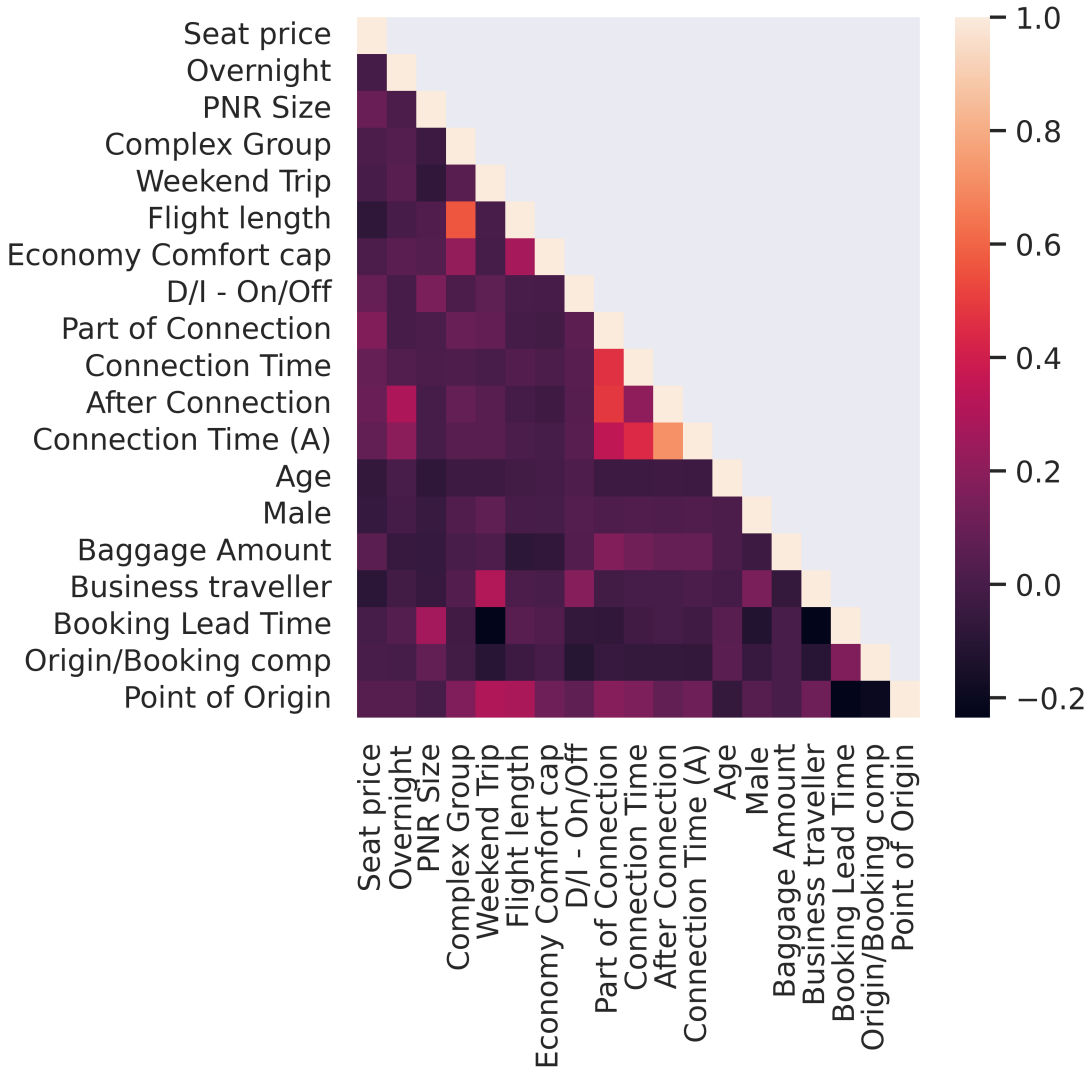

Figure 3.11: Correlation of features intercontinental

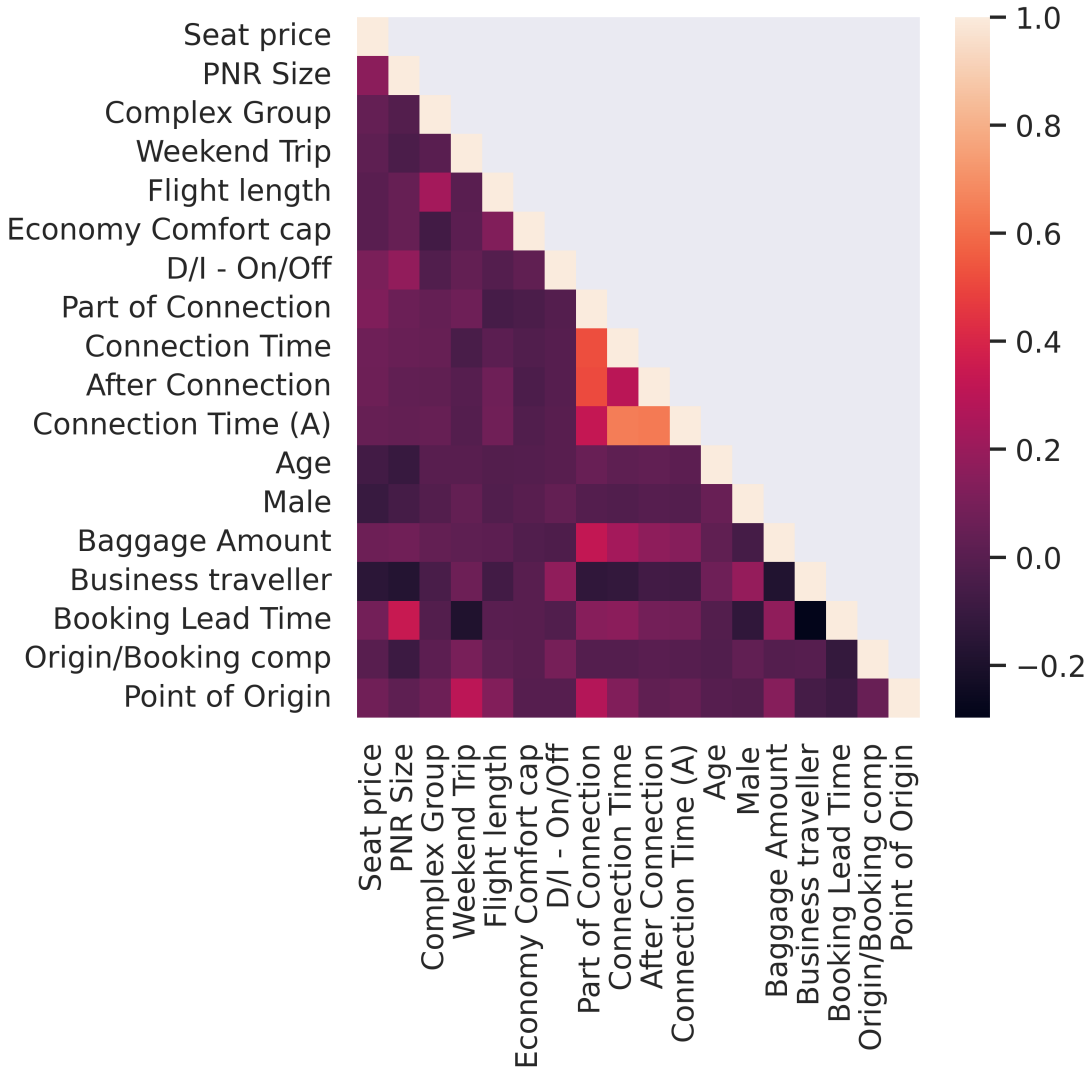

Figure 3.12: Correlation of features medium haul

Chapter 4

Methods

This chapter will start by introducing some important notation in Section 4.1. In Section 4.2 the classification algorithms are introduced as well as the model training, evaluation, tuning and the feature importance evaluation. Section 4.3 covers the experiment conducted with the different price features and finally the model optimization is described in Section 4.4.

4.1 Notation

The following sets are used throughout this project: $I = \{1, \ldots, N\}$ is the set of passengers in the data set with N being the total number of passengers in the data set, $P = \{0.475, 0.525, \ldots, 1.075\}$ is the set of all Prices and $F = \{DMM, EBB, LOS, \ldots\}$ is the set of flights in the data set. Note, that the set of flights F and the total number of passengers N in the data set changes depending on which data set is used, this does however not influence the notation. The true label of a passenger $i \in I$ is given by $y_i \in \{0, 1\}$, effective price (price that has to be paid) by $p_i \in P$ and f_i is the flight passenger $i \in I$ was on. The estimated probability of purchase by the models depends on the price and is given by $\hat{\pi}_{ip}$ with $i \in I$ and $p \in P$, but can also be given without the price as $\hat{\pi}_i$ in which case it is the probability of purchase given the true effective price of the customer.

4.2 Classification Modelling

This section provides an explanation of all procedures associated with classification models, beginning with the models themselves. Following this, three different metrics for model evaluation are introduced, followed by information about model training and hyper parameter tuning. The final section introduces two separate metrics that were employed as tuning objectives. This section concludes with a method for determining the relevance of features.

4.2.1 Classification Algorithms

The most popular classification models are tree based learners derived from **Gradient Boosting Decision Trees (GBDT)**. In order to understand the functionalities, as well as the specialties of the three learners Light Gradient Boosting Machines (LightGBM), Extreme Gradient Boosting (XGBoost) and Categorical Boosting (CATBoost) the general concepts used by all of these models are introduced first.

A **Decision Tree** used for prediction splits the predictor space into a number j of distinct and non-overlapping regions $R_1, \ldots, R_j$. Every observation falls into exactly one region R_i and gets as prediction the majority class in that region. The predictor space is defined as the space of all possible values the features can assume $X_1, \ldots, X_p$ where X_i are the different input features. The regions are defined by the leafs of a binary decision tree which can have varying depths. At each node in the tree, one feature is selected and its values divided at a split point to follow either the left or right branch. Ideally, features that can classify the data better are closer to the root. The decision tree is built starting at the root. Here, all features with various possible split points are evaluated. The goal is to have as little variance (also known as entropy) in the created leafs as possible. All options are compared based on their **information gain**, which is a suitable metric like the Gini impurity score and aims to assess the quality of the split. The split with the highest information gain is applied. Successively, the leafs are split further until certain stopping critera are reached.

Boosting is a powerful learning idea originally designed for classification models and independently introduced by Robert Schapire and Yoav Freud in various papers in the early 1990s. [14] The idea of boosting is to combine various weak learners trained in sequential boosting rounds on a modified training data set and finally create a strong predictor by majority voting [20]. A weak learner is defined as a model whose predictions are only slightly better than random guessing. **Gradient Boosting** is a special form of boosting, where in every boosting round, the new model is trained to fit the residual errors (also known as the negative gradients [16]) of the previous model with respect to some loss function. The goal is to create a strong ensemble of weak classifiers by minimizing the loss function iteratively with each new model trained. The idea of minimizing a loss function by using its gradients is called **gradient descent** although in this case of gradient boosting there are no weights updated in each iteration like in the **gradient descent algorithm** but rather a new model is trained in each iteration to minimize the errors of the previous one [14]. While the models are all based on the previously introduced principles, the main differences between them can be explained by the tree growth method, the handling of categorical features and

the sampling of data in each boosting round. Additionally, they have their own methods to improve computational performance.

4.2.1.1 Extreme Gradient Boosting (XGB/ XGBoost)

XGBoost has been around since 2014, but was officially introduced in the paper [8] by Chen and Guestrin in 2016. It was designed to deal with huge and complex data sets. As such, it makes gradient boosting highly scalable by significantly decreasing computation time making use of a novel algorithm for parallel tree learning. Compared to normal gradient boosting, extreme gradient boosting is also known as regularized gradient boosting. XGBoost introduces regularization terms to make the model less sensitive to single data points and to reduce the model complexity by pruning (dropping) certain branches and leaves of the tree away. XGBoost grows the tree depth-wise, which means that the trees are build level by level. Categorical features have to be encoded before feeding it to the algorithm, as XGBoost only takes numerical features. The sampling method in each boosting round can either be uniform or gradient based, which means that the selection probability for each training instance is proportional to the regularized absolute value of gradients [4]. This ensures, that high gradients are selected with a higher probability, allowing the next model to focus on the more problematic instances which are producing greater errors. More information about this popular algorithm can be found on the website [4], in the paper [8] or in various tutorials like the series on XGB by StatQuest [15].

4.2.1.2 Light Gradient Boosting machine (LightGBM)

A major set-back of gradient boosting machines is the unsatisfactory scalability, which is mainly because all instances have to be evaluated for all features to estimate the split point with the best information gain. LightGBM was firstly introduced in 2017 by Ke et al. [16] and suggests two additional concepts: **Gradient-based One-Side Sampling (GOSS)** and **Exclusive Feature Bundling (EFB)** which reduce this problem. Similarly to the gradient based sampling of XGBoost, GOSS gives more weight to high gradients by randomly dropping instances with low gradients in each boosting step. In this way, a significant proportion of the training data is excluded without affecting the accuracy. EFB bundles mutually exclusive features (features which rarely take zero values simultaneously) thus reducing the number of features. These two methods allow LightGBM to accelerate the training process by up to 20% while achieving the same accuracy as GBDT. For an in depth explanation of the two concepts, be referred to the paper by Ke et al. [16]. LightGBM grows trees leaf wise, so regardless of the depth, it will expand at the leaf that gives the highest information gain. Categorical features can be passed into LightGBM and will be binned

based on the accumulated objective values of each category. A more detailed explanation of this idea is given in the official documentation [3].

4.2.1.3 Categorical Boosting (CatBoost)

CatBoost is an algorithm for gradient boosting on decision trees and was introduced by Yandex researchers and engineers in their paper [20] in 2017. The key features of CatBoost compared to other gradient boosting decision trees is the growing of symmetric decision trees, the special handling of categorical features through **order based target encoding** and the concept of *ordered boosting*. A symmetrical tree has the same feature and same split point on every node of each level. This makes the tree a weaker learner, but considerably increases the time to make predictions. As trees are weak learners in general, and they are combined to form a strong predictor, the symmetrical structure of the trees does not result in a decrease in accuracy.

Order based target encoding is a form of target based encoding, where the data is treated as if it was given into the model sequentially. Usually, target based encoding would assign the average target value of every category to each data point of the respective category. However, now the target of every data point is influencing the encoding of itself, by simply being part of the average. This is another form of leakage, which leads to overfitting. By treating the data sequentially and adjusting the formula of the average slightly, the encoding is only based on all previous data point's target values and is therefore overcoming this problem of leakage. In boosting decision trees, a tree is build in each boosting round on all the training data and afterwards gives a leaf output (prediction of the residual) to all the training data based on that same tree. This results in another problem of leakage just like before, as every data point's target value is contributing to its respective leaf output. This leads to a shift from the estimated distribution of residuals to the true distribution which leads to over fitting and is known as **prediction shift**. To overcome this problem, **ordered boosting** is applied, which builds the trees just as explained in the beginning of this section, but treats the data sequentially ensuring that the residual of a data point, is not part of the leaf output of that same data point. For instance a row j in the training data is given into the tree, ends up in a leaf, and the residual of j, that was computed before building the tree based on an initial prediction, is assigned to that leaf. Now the leaf output for j are the average residuals in that same leaf, however this value is calculated only on the average of all the previous $j-1$ data points that ended up in the same leaf. Like this, the residual of j does not have an influence on its leaf output. The data is randomly shuffled in the beginning of each boosting round, as the order is now important and should not always be the same.

For more detailed information about the algorithm please be referred to the release paper of the Yandex researchers [20].

4.2.1.4 Logistic Regression (LogReg)

A simpler model that estimates a linear relation between some independent variables and a dependent variable with a probability as output is the **Logistic regression**. By learning coefficients for a linear expression combining the input features it works similar to linear regression, but as the output is not continuous but a probability, the logit function $logit(x) = 1/(1 + \exp\{-x\})$ is used to transform the output. In Equation (4.1), it is shown how the log odds are the target of the linear regression, where $\beta_0, \beta_1, \beta_2$ are the regression coefficients and X_1, X_2 are two features.

$$\ln\left(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right) = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 \tag{4.1}$$

The coefficients can now be estimated via different methods, but are commonly estimated with the maximum likelihood estimation (MLE).

4.2.2 Model evaluation

Classification models are commonly evaluated based on metrics that capture the missclassification error. An example is the **accuracy score**, which reflects the percentage of correct predictions, **recall** which represents the percentage of positives that were correctly classified as positives or **precision** which is the percentage of all positive predictions that were correctly predicted. To assess the class probabilities of a model, it is common to use a prediction bias plot, also known as **calibration plot** and the negative logarithm of the loss called **Logloss**. As for this project, the aim of the classification model is to give a probability of purchase to a passenger, rather than a simple binary label, the calibration plot will be used to evaluate the different models. It is furthermore important to look at the price sensitivity in order to use the purchase probabilities in optimization. Lastly, the impact of the probability prediction bias on the total predicted revenue is reflected in the **bias** of the revenue prediction.
The concept of *K-Fold Cross validation* will be used multiple times to show the robustness of the results. In this concept, the total data set is split into $K = 5$ different, equally large parts, and the model is trained K times, always using another one of the K parts as test set and the remaining K-1 parts as training set. The outcome on the K test sets is then averaged to get a general result.

4.2.2.1 Price elasticity plot

The price elasticity plot is showing the average probability of purchase for passengers grouped by their effective price offered plotted against their effective price. This function should be decreasing to ensure a negative price sensitivity meaning that a higher price decreases the probability of purchase. To show the robustness of this plot, 5-fold cross validation was performed, plotting the

average price elasticity for the five test sets. The price sensitivity in the plots was adjusted in such a way, that the slopes have a non-increasing gradient (smaller or equal to zero) as this adjustment will also be used in the optimization.

4.2.2.2 Calibration plot

The calibration plot groups the test data by the price and plots the predicted average probability of purchase against the true fraction of purchases for each bucket. The perfectly calibrated model follows closely the bisector, meaning that predicted and true probabilities align in each bucket. Like in the price sensitivity plot, 5-fold cross validation was used to additionally show the robustness of the calibration. In order to easier compare the models of the different plots, a **Calibration loss** was introduced, which is defined in Equation (4.2), where $I_p = \{i|p_i = p\}$ is the set of passengers who paid price $p \in P$.

$$calloss = \frac{1}{|P|} \sum_{p \in P} \left(\frac{1}{|I_p|} \sum_{i \in I_p} (y_i - \hat{\pi}_i) \right)^2 \tag{4.2}$$

It can be understood as the mean squared error (MSE) between the predicted and true probability of purchase over all the bins.

4.2.2.3 Bias in revenue prediction

Let $r = \sum_{i=0}^{N} y_i \cdot p_i$ be the total true revenue. The true revenue has to be estimated using the revenue estimator on passenger level $\hat{r} = \sum_{i=0}^{N} \hat{\pi}_i \cdot p_i$. The difference between $\hat{r}$ and r is the error of the revenue prediction. μ_{err} and σ_{err} are the mean and standard deviation of this error over multiple runs of the model, each time with a different training set. The formulas are given in equations (4.3) and (4.4) and the bias of the prediction is given by $Bias(\hat{r}) = \mu_{err}$.

$$\mu_{err} = \mathbb{E}(\hat{r} - r) = \mathbb{E}\left(\sum_{i=0}^{N} p_i(\hat{\pi}_i - y_i) \right) \tag{4.3}$$

$$\sigma_{err} = \sqrt{\mathbb{E}(\hat{r} - \mu_{err})^2} \tag{4.4}$$

Bootstrapping was used to generate 10 different training sets, keeping the test set the same. In Bootstrapping, a new training set is generated from the original one, by randomly selecting instances with replacement until the new training set contains the same number of data points as the original training set. The bias of the model is the average bias in the revenue prediction over the 10 rounds.

4.2.3 Model Training

The models were trained with 5-fold cross validation and all results averaged over the predictions on the five test sets. As a random seed was used, the same training and test sets were generated for all different models and tests.

In Section 4.2.2 it was discussed already that the model will be evaluated based on the price elasticity, the calibration and the prediction bias. Ideally, one or all of these metrics should be used in a combined loss function when training the model. To my knowledge, such metrics are not used directly. Especially the bias which appears to be a good loss indicator is difficult, as it is build around the idea that false positives and false negatives cancel out. Most existing loss functions try to reflect however the absolute error, regardless of the direction. So the prediction bias would suggest a loss function which takes the mean errors, not however the mean absolute errors (**MAE**). The loss function of a machine learning model has to be twice differentiable, as will be explained in the following:

In gradient boosting, each new tree is build on the errors of the previous model. The loss $\mathcal{L}^t = l\left(y, \hat{y}^{t-1} + f_t(X)\right) = l\left(y, \hat{y}^{t-1} + \delta\hat{y}\right)$ of tree t gives the error of the prediction as calculated by the loss function, where y is the vector of all the true labels, $\hat{y}^{t-1}$ is the prediction up until the previous tree and $f_t(X)$ is the current trees leave output for the data X. This loss function is approximated with Taylor expansion and minimized, to find the optimal $f_t(X)$ which can be understood as the leave score or leaf weight. The leaf score of each data point x_i is the same for all instances in the same leaf j. This score can also be expressed by $w_j = f_t(x_i)$, where instance i is assigned to leaf j by the tree. The optimal leaf score that minimizes the loss is dependent on the gradient G and the hessian H of all instances in the respective leaf and is given by $w_j = -\frac{G}{H+\lambda}$, where λ is the L2 regularization parameter (see Chapter 5 for information about the parameters). In order to use the gradient and the hessian, the function has to be twice differentiable.

The mean error (just like the MAE) would have a constant gradient and a hessian matrix of zeros. From the previous calculation it can be seen, that this loss function would result in all leaf scores always being the same (no matter the tree structure, features used and split points), which means there can be no improvement and all trees have exactly the same score.

As the model cannot use a simple metric that cancels out false positives and false negatives, the negative Logloss function was used as objective in the model training. This loss function averages the negative logarithm of the predicted probabilities of the respective true labels and penalizes low probabilities, which are equivalent to higher uncertainties. The negative Logloss function is shown in Equation (4.5).

$$Logloss = -\frac{1}{N}\sum_{i\in I}\big(y_i \ln \hat{\pi}_i + (1-y_i)(\ln(1-\hat{\pi}_i))\big) \tag{4.5}$$

4.2.4 Hyper-parameter tuning

Hyper-parameter optimization is the act of searching the space of possible configuration parameters for a training algorithm in order to find a set of variables that allows the algorithm to achieve more desirable results (Bergstra et al. [7]). Machine learning models in general have many hyper-parameters and for most data sets, only a few of these are important which are however different depending on the data sets (Bergstra and Bengio [6]). For the hyper-parameter tuning only the training set was used. For the *Hyperopt* method a validation set of 20% was split of the training data set, whereas for the Grid Search method 5-fold cross validation was used. In the following sub sections, the tuning methods are explained, as well as the different tuning metrics tested. For XGBoost and LightGBM the tuning metrics were employed as the objective functions of the Hyperopt search.

4.2.4.1 Hyperopt

Hyperopt-sklearn is a python library, that provides automatic algorithm configuration of the Scikit-learn machine learning library, which includes many of the methods used in this project and is the standard machine learning library in python (Komer et al. [17]). After choosing an optimization algorithm, any objective function can be constructed and a search domain for all relevant parameters defined. Hyperopt handles real-valued, discrete and conditional variables and is very efficient in the search of the parameter space by parallel evaluations. The parameter space is searched with the aim of minimizing the objective function. Hyperopt has improved best-known scores of many standard bench marking sets, making it an effective and practical tuning method (Bergstra et al. [7], Komer et al. [17]).

4.2.4.2 Grid Search with cross validation

The CATBoost algorithm was tuned with Grid Search, where all parameters that are to be tested with their discrete values are used as input, and Grid Search tries all possible combination in an exhaustive search. Additionally, all combinations were tested on their robustness with 5-fold cross validation. The parameters with their values are given in the following Section 5.

4.2.4.3 Different tuning metrics

As mentioned in the model training (Section 4.2.3) the training objective is the negative Logloss. This is the first metric that all gradient boosting models (XGB, CAT and LGB) were tuned on with the aforementioned tuning methods. The models using the resulting hyperparameters are identified by the ending **ll**, like for instance XGB ll. LGBoost and XGBoost where moreover tested after tuning with two different metrics besides the negative Logloss, to eventually tune the models

even better in the direction of the suggested evaluation methods. The first other metric is weighing the Logloss by the probability of purchase, to give more weight to errors around higher purchase probabilities. This metric is displayed in Equation (4.6) and will be called **llprob**.

$$llprob = -\frac{1}{N}\sum_{i \in I}\big(y_i \ln \hat{\pi}_i + (1 - y_i)(\ln(1 - \hat{\pi}_i))\big) \cdot \hat{\pi}_i \tag{4.6}$$

The second other metric is aiming to minimize the prediction bias within the price buckets. It gives the average difference between the estimated and the true probability of purchase over all price buckets, hence trying to improve the calibration of the model. This metric will be called **calloss** and was introduced in Equation (4.2) in Section 4.2.2.

4.2.5 Feature importance

The feature importance will be evaluated with the **permutation importance**. The permutation importance is computed by randomly shuffling each feature in turn and computing the loss in the model score. This indicates how important the feature is for the prediction and it can be compared with the other features. Each features is randomly shuffled 10 times and its score computed by the average loss in model score of the 10 rounds. The permutation score is computed on the test data, which avoids misleading results that can occur when the model is over fitted on the training data and the feature importance is calculated on that same data, how it is the case for other feature importances.

4.3 Price feature experiments

In order to use the price feature for revenue optimization, the price elasticity and how it is captured by the model plays an important role. In classification models it is however not possible to shape the price elasticity during the modelling process. This is why the idea is to ensure a negative price elasticity in the data while maintaining a low correlation between any feature and the price and evaluate how well the price elasticity is preserved after the modelling process. All price features created for testing were explained in Section 3.3 and are summarized in Table 3.7. For all the price feature experiments, the CATBoost model was trained on all features listed in Table 3.7 additionally to the respective price feature. For the price feature `ATPCO nodisc` which does not apply discounts, the `FF Level` feature was included as well, for all other experiments this feature was however left out, as it is otherwise highly correlated with the price and thus impacting the price sensitivity.

After the CATBoost model was trained on the selected features, probabilities of purchase were predicted for all different price bins. If a higher price led to a lower probability of purchase, the corresponding passenger has a correct price sensitivity between these prices. As a metric to compare the correct price sensitivity between the different experiments, the percentage of passengers that had a correct price sensitivity between the price buckets $(0.85, 0.9], (0.9, 0.95], (0.95, 1], (1, 1.05], (1.05, 1.1]$ was computed. Comparing this metric to the price elasticity curve, it was decided that a 2% of positive price elasticity could be accepted, so that the metric **PS correct** gives the percentage of customers, whose probability of purchase declined, or increased by less than 2% over the previously mentioned price buckets.

Moreover, the price elasticity curve was plotted by the average probability of purchase in each of the price buckets. This gives additional insights into the overall understanding of the price by the model. While the metric *PS correct* only considers a logical price sensitivity in between the higher prices, this plot also depicts the probability of purchase for lower prices down to 0.475, which will be the price for many passengers with frequent flyer discount.

In addition to the price sensitivity, the **Logloss** was used to roughly capture the model performance. As already explained in the model training Section 4.2.3, this loss is not ideal but sufficient to get an idea if the performance of the model differs greatly between the different experiments.

Lastly, the importance of the respective price feature was captured through the score and relative rank compared to the other variables given by the **PredictionValuesChange** method of CATBoost. The score represents how much on average the prediction changes if the feature value changes and is normalized so that the scores of all features sum up to 100. For more information on the exact calculation of this importance measure, be referred to the official CATBoost documentation [1].

4.4 Price Optimization

After the classification model has been trained and predicts probabilities of purchase for the customers following a reasonable price sensitivity, the expected revenue can be determined. By inserting the different prices into the model for all customers and estimating the probability of purchase, the expected revenue can be computed for each passenger for each price.

Optimization is now performed on three levels:

1. The highest level is giving the same price to all customers and will be termed **Simple opt**. IT uses the price sensitivity (average probability of purchase for a price p) and multiplies it by the price. The formula is given in Equation (4.7), where $I_p = \{i \in I | p_i = p\}$ is the set of all passenger

who paid price $p \in P$.

$$p^* = \arg\max_p \left\{ \frac{p}{|I_p|} \cdot \sum_{i \in I_p} \hat{\pi}_{ip} \right\} \tag{4.7}$$

2. The middle level is optimizing the prices for all flight, which is why the model will be termed **Flight opt**. The formula for this optimization is given in Equation (4.8), where $I_{pf} = \{i \in I | p_i = p, f_i = f\}$ is the set of all passengers $i \in I$ who were on flight $f \in F$ and paid price $p \in P$. Finally, p_f^* is the optimal price given to all passengers on flight f.

$$p_f^* = \arg\max_p \left\{ \frac{p}{|I_{pf}|} \cdot \sum_{i \in I_{pf}} \hat{\pi}_{ip} \right\} \tag{4.8}$$

3. The most granular optimization level is the passenger level, where the prices are optimized based on the respective customer. This model will be termed **Passenger opt** and the optimization formula is given in Equation (4.9).

$$p_i^* = \arg\max_p p \cdot \pi_{ip} \tag{4.9}$$

In theory, the models could select any price for the passengers, in practice however there are various constraints as to why this is not feasible. First of all, there is no capacity constraint taken into consideration, which means that it has to be ensured that all the seats sold are actually available. Secondly, the seats sold belong only to the economy comfort class, although there are other sections in the economy cabin like *Extra Legroom*, *Economy Front* or *Economy Standard*. The influence of a lower price of one section on any other section is not considered, but in order to maintain a reasonable price relation it can be assumed that lowering the price of the EC section too much, should also decrease all other seat prices which might again change the buying behaviour substantially. Lastly it is maybe not even desirable to give too high discounts to keep some extend of exclusiveness. For these reasons, it was decided to perform optimization only within the range of a minimal boundary and the upper boundary of 1.1. For the default optimization this minimal boundary was 0.85, which means that the effectively lowest price was $M = 0.875$. To apply the price boundary to all customers, a brief distinction between the *effective price* used until now and the *price seen* has to be given. Unlike everywhere else in the project, the boundary will be applied to the *prices seen*, which only makes a difference for the frequent flyer levels *Explorer, Silver* and *Gold*, where the effective price is lower by the respective discount than the price seen. This means for example, that for a boundary of (0.85, 1.1] the prices for *Silver* level members are selected between $0.875 * 0.75 = 0.6563$ and $1.1 * 0.75 = 0.825$. As the prices are not continuous, the closest true prices will be used as boundaries. As 0.6375 falls in the price bin $(0.6, 0.65]$, the lower bound

of the prices for *Silver* passengers will be 0.625.

The hypothesis of this thesis is that revenue can be increased best through passenger-level pricing. Consequently, the **Passenger opt** method will be the focus of this evaluation and the other two optimization methods will be used as comparison. The performance of the passenger model will first be compared to the true revenue. This evaluation is based on the mean revenue increase in percent (over the 5 folds of cross validation) and the expected range of revenue increase considering the respective model biases as computed for the evaluation of the classification model (Section 4.2), with respect to the true revenue. Let r^* be the mean predicted revenue optimized on passenger level, r the true revenue, $\mu_{err} = bias(\hat{r}), \sigma_{bias}$ the mean and standard deviation of the prediction error on passenger level as given by equations (4.3) and (4.4), then the lower and upper bound (lb, ub) of the expected revenue increase through passenger level optimization are given by:

$$lb = (r^* - bias(\hat{r}) - 2 * \sigma_{err})/r - 1 \tag{4.10}$$

$$ub = (r^* - bias(\hat{r}) + 2 * \sigma_{err})/r - 1 \tag{4.11}$$

Chapter 5

Implementation

5.1 Feature encoding

As seen in the model description (see section 2.2), models like XGBoost require all categorical variables to be encoded into numerical features. This was done by **One-hot encoding**, where every attribute of a categorical column becomes a new binary feature. The feature `Point of Origin` had however about 190 different countries, which would have created 190 new features with the one-hot encoding method. In order to maintain a lower number of features, only the top ten most frequent nations were used and all other nations left out. For this reason, CATBoost received slightly more information by including all nations than LightGBM and XGBoost which received the pre-processed versions of the data. categorical features one hot with some being left out

5.2 Model tuning and parameters

The three tree-boosting models XGBoost, CATBoost and LightGBM have many hyper-parameters that could be explored in the tuning phase, most of which ensure a balance between over- and under-fitting in affecting the structure and learning of the decision trees. The hyper-parameters are described and their final values given for all models separately in this section. All hyper-parameters not mentioned were not tuned and used with their default values. The DT and LogReg models were used without tuning and with their default values as they are set in the *Scikit-learn* package in python. For more information about the default values and parameters, be referred to the official package documentation [2].

XGBoost The parameters selected for tuning are explained in Table 5.1 and the resulting hyper-parameters that were used to train the respective XGB models are given in Table 5.2.

Parameter	Description
learning rate η	The stepsize shrinkage when updating the weights after one boosting round. The model becomes more robust and less likely to over-fit with a lower η.
reg_lambda λ	Is the regularization parameter used to reduce over-fitting. A higher λ makes individual observations count less towards the overall prediction and thus reduces it's sensitivity individual observations.
minimum split loss γ	Defines by how much the loss function has to decrease in order for a split of a node to be performed. More conservative and less over-fitting for larger γ.
maximal depth	The maximal depth of the tree, where a higher depth allows the model to learn specific relations and can lead to over-fitting.
minimum child weight	Minimum sum of weights of all observations required in a child.
colsample_bytree	Indicates the fraction of columns to be used when building each tree.

Table 5.1: XGBoost parameter description

Parameter	Search range	XGB	XGB ll	XGB llprob	XGB calloss
learning rate η	{0.001, 0.003, 0.005, 0.01, 0.03, 0.1}	0.3	0.1	0.03	0.1
reg_lambda	[0, 1]	1	0.2540	0.0668	0.5845
gamma	{0,. . . , 9}	0	7.1450	5.3826	6.8779
max_depth	{3,. . . , 18}	6	9	3	14
min_child_weight	{0,. . . , 10}	1	1	9	6
colsample_bytree	[0.5, 1]	1	0.9105	0.6791	0.9080

Table 5.2: XGB models parameters

LightGBM All parameters of the LightGBM model selected for tuning are explained in Table 5.3 and the resulting hyper-parameters that were used to train the respective LightGBM models are given in Table 5.4.

CATBoost All parameters of the CATBoost model selected for tuning are explained in Table 5.5 and the resulting hyper-parameters that were used to train the respective CATBoost models are given in Table 5.6.

Parameter	Description
learning rate η	The stepsize shrinkage when updating the weights after one boosting round. The model becomes more robust and less likely to over-fit with a lower η.
boosting_type	There were two different boosting types tested. *GBDT* is the traditional gradient boosting decision tree. *GOSS* is used to speed up training while maintaining high accuracy. As observations with higher gradients contribute more to the information gain, this method randomly drops instances with smaller gradients.
lambda_l1	Is the regularization parameter used to reduce over-fitting. A higher λ makes individual observations count less towards the leaf score and serves to reduce the depth of trees.
lambda_l2	An additional regularization parameter, that instead of aiming at sparsity like the *lambda_l1* (reducing the weights to 0), it encourages smaller weights of individual observations towards the leaf score and overall loss.
num_leaves	Controls the number of leaves a tree can have where in each leave the final output is predicted.
min_child_weight	Sets the minimum sum of weights necessary to perform a split at any node. Higher values limit the complexity of the tree and prevent over-fitting.
min_data_in_leaf	Minimum number of observations that fall in the decision criteria of each leaf.
feature_fraction	Defines the fraction of columns used for training and is used to speed up training as well as reduce over-fitting.
subsample_for_bin	Gives the number of observations to use when aggregating the continuous features into discrete bins for simplifying the model performance. Lower values can increase the training speed but can in some cases lead to a loss of information.

Table 5.3: LightGBM parameter description

Parameter	Search range	LGB	LGB ll	LGB llprob	LGB calloss
learning_rate	[0.01, 0.2]	0.1	0.1981	0.0139	0.1579
boosting_type	{gbdt, goss}	gbdt	goss	gbdt	goss
lambda_l1	[0, 7]	0.0	1.3639	0	0
lambda_l2	[0, 7]	0.0	0	0.2891	0
num_leaves	{30,. . . ,150}	31	140	59	75
min_child _weight	[0, 148]	0.001	15.68	141.50	5.19
min_data_in _leaf	{1,. . . , 400}	20	11	230	44
feature_fraction	[0.5, 1]	1	0.8046	0.9606	0.9682
subsample_for _bin	{20,000;. . . ; 300,000}	200,000	240,000	60,000	240,000

Table 5.4: LightGBM model parameters

Parameter	Description
learning rate η	The stepsize shrinkage when updating the weights after one boosting round. The model becomes more robust and less likely to over-fit with a lower η.
l2_leaf_reg	Is the regularization weight used to reduce over-fitting by controlling the model complexity.
max_depth	The maximal depth of the tree, where a higher depth allows the model to learn specific relations and can lead to over-fitting.

Table 5.5: CATBoost parameter description

Parameter	Search range	CAT	CAT ll
learning_rate	{0.03, 0.1, 0.15, 0.2}	0.03	0.03
l2_leaf_reg	{1,3,5,7,9}	3	3
max_depth	{1,. . . ,30}	6	11

Table 5.6: CATBoost model parameters

Chapter 6

Results

This chapter is broken into three sections that exhibit and explain the results achieved by the methods described in Chapter 4. The first Section 6.1 examines the experiments around the price features presented in Section 4.3. The feature importance including the selected price feature, is evaluated based on its permutation importance in Section 6.2. Utilizing these variables, the classification models are trained and evaluated in Section 6.3. The optimization based on the classification models described in Section 4.4 will be lastly assessed in Section 6.4.

6.1 Price feature experiments

The four different price features were all tested in turn combined with the other prepared features (see Table 3.6) as input for the CAT model. As mentioned in Section 4.3, Table 6.1 shows the correct price sensitivity across all customers in the test set, as well as the Logloss and relevance of the respective price feature via score and rank. Although `ATPCO nodisc` and `ATPCO partov` have acceptable percentages, the price feature `ATPCO` has the highest percentage of correct price sensitivities. The Logloss is approximately the same for all, indicating that there is no discernible performance variation due to the pricing features. For the price feature importance it seems that with more overwritten prices the feature looses importance. Where the price feature has a relatively high score and is ranked first for the `ATPCO noov` feature, after overwriting all prices and correcting all discounts the price only reaches rank 16 in feature `ATPCO nodisc`. This can be explained by the method to estimate the unknown prices of customers who did not buy an EC seat. As seen already in Section 3.3, the discounts in the original prices underlie great variance, which is not reflected in the estimated prices. It is simply not realistic to randomly assign a 70% discount to someone, and assume that person did not buy, as it is rather unlikely they ever saw such a discount. For this reason the model knows with a high certainty, that people who saw very different prices (outliers),

must always have bought a seat, because this price would not have been estimated for anyone else. That is why the price feature is so important, and the performance is slightly better for the feature `ATPCO noov`. The big disadvantage of keeping the original prices is however reflected in the low percentage of correct price sensitivities and also in the price elasticity curve in Figure 6.1. In the plot it can be observed, that the `ATPCO nodisc` feature follows a very constant price elasticity, whereas `ATPCO noov` assigns a high probability to the low price classes as these contain mainly buyers, before following a less extreme but highly variant course for higher prices. `ATPCO` delivers the best and only reasonable price elasticity.

While the features `ATPCO nodisc` and `ATPCO partov` have acceptable price elasticities for higher prices greater than 0.875 as shown by the metric PS correct, in the plot it becomes clear that especially for lower prices the probabilities of purchase do not follow the same smooth trend as the `ATPCO` feature, but are highly variant.

For these reasons, the price feature `ATPCO` was further used as model input in the following sections.

	ATPCO	**ATPCO nodisc**	**ATPCO noov**	**ATPCO partov**
PS correct	0.9897	0.9525	0.5474	0.9057
Logloss	0.2519	0.2490	0.2335	0.2469
Price feat score	8.1904	1.0471	18.2811	11.1084
Price feat rank	5	16	1	2

Table 6.1: Price feature experiments results

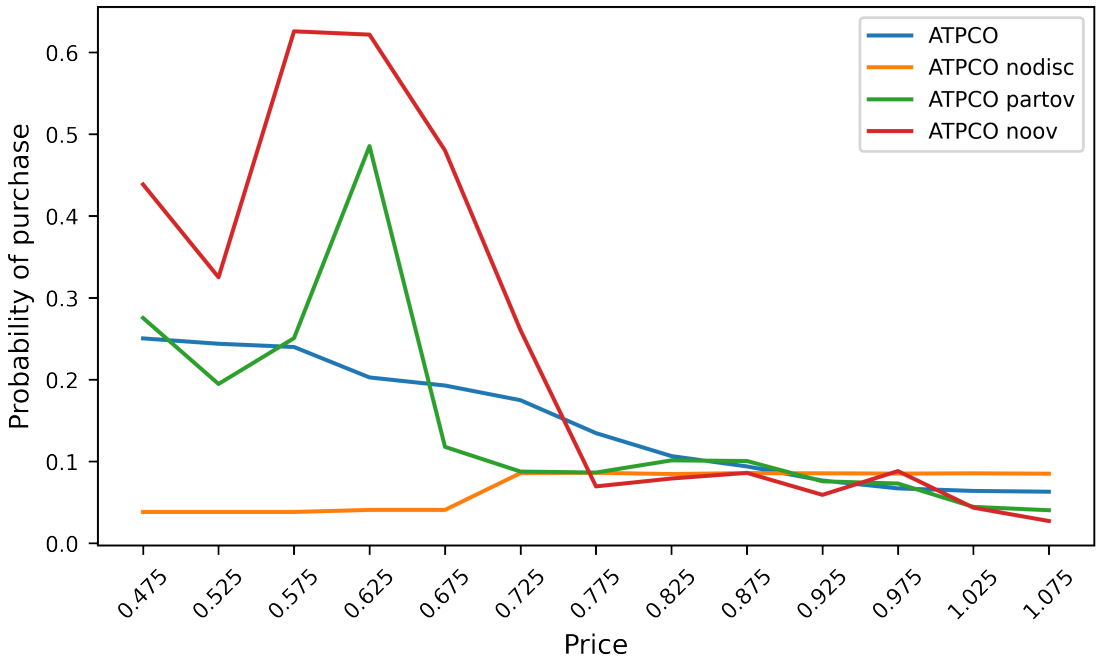

Figure 6.1: Price feature experiment price elasticity

6.2 Feature importance

In Figure 6.2 the features used as model inputs are ranked according to their permutation importance score for medium haul and intercontinental flights. It can be noted, that the origin and destination of the flights (`Complex Group, Point of Origin`), both seem to be among the most important features. A reason might be that they are the two features that are correlated with the absolute price paid by the passengers, which is otherwise not included in the features, as the `Seat Price` contains the relative price only. But the base prices are determined by the origin and destination of the flight, which makes these features good indicators for the absolute price. For medium haul passengers, the features `Male, Business traveller` and `Part of Connection` even have a negative permutation importance, meaning that the model loss was less when shuffling these features randomly. The connection features are relatively high correlated with each other, which is why it is not possible to assess how important the individual features are separate from the others. Interestingly, the `Business traveller` feature is relatively important on intercontinental flights in contrast to medium haul flights. The booking lead time is on the very top for medium haul flights, which is surprising, as no such effect was observed in the data analysis (see Section 3.4). As the normalized data was examined in the analysis section which accounted for the final section capacity of every flight, it could be that the increased capacity on medium haul flights in the last

days before departure gives more people the opportunity of buying an EC seat. Finally the price feature is also difficult to evaluate with permutation importance, as the prices were already shuffled to begin with. The only difference now is, that the lower prices of the different frequent flyer levels are now randomly assigned to passengers of possibly different levels.

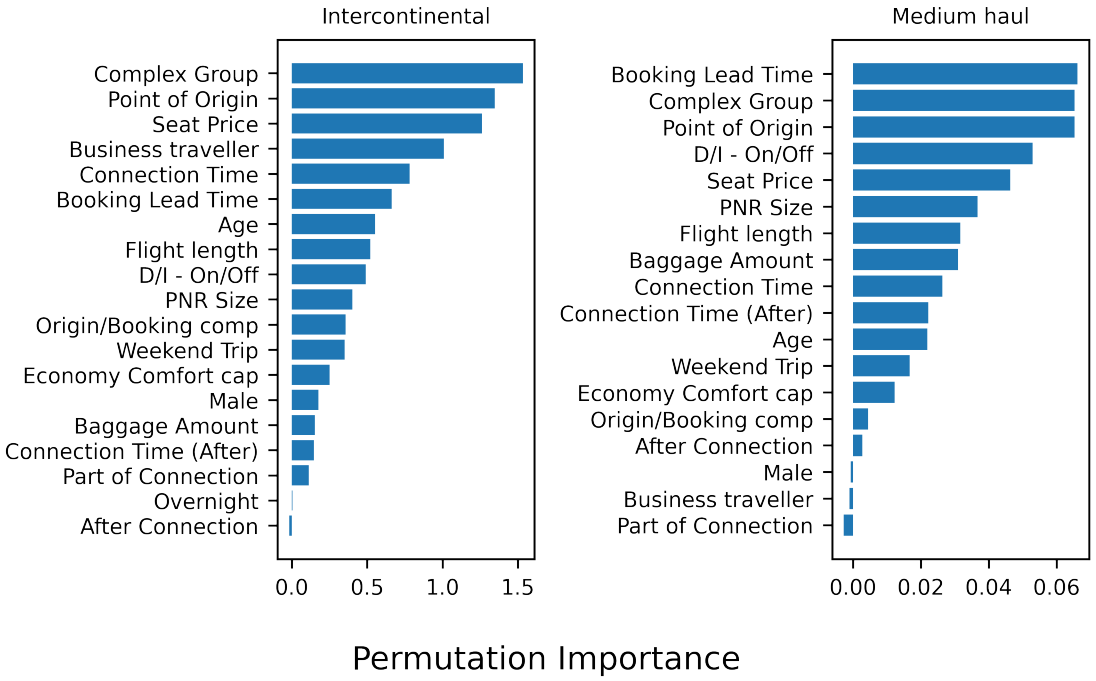

Figure 6.2: Permutation importance CATBoost

6.3 Model comparison

In tables 6.2 and 6.3 the bias of the twelve models is given in percent. The overall true revenue in the test set of the intercontinental data is [redacted]€. Most models' estimated revenues deviate from this amount by less than 1%, which is acceptable. On medium haul flights the true revenue is at [redacted]€ and again most models predict a revenue that is deviating from the truth by less than 1%.

The models XGB llprob and LGB ll have a significantly higher bias than the other models and will have a low prediction accuracy on both data sets.

Model	Error mean/Bias (%)	Error std (%)
DT	0.82	0.50
LogReg	-0.05	0.37
XGB	-0.62	0.26
XGB ll	-0.83	0.28
XGB llprob	26.37	0.30
XGB calloss	-2.56	0.28
CAT	-0.67	0.00
CAT ll	-2.31	0.39
LGB	0.06	0.28
LGB ll	351.51	0.75
LGB llprob	1.68	0.29
LGB calloss	-0.22	0.29

Table 6.2: Model results Intercontinental

Model	Bias mean (%)	Bias std (%)
DT	1.60	0.28
LogReg	1.02	0.27
XGB	0.35	0.27
XGB ll	-0.13	0.22
XGB llprob	45.62	0.23
XGB calloss	-2.30	0.26
CAT	0.95	0.00
CAT ll	-2.49	0.30
LGB	0.81	0.23
LGB ll	639.32	1.95
LGB llprob	2.81	0.24
LGB calloss	0.56	0.26

Table 6.3: Model results medium haul

After the model bias, the following graphs 6.3, 6.4, 6.5 and 6.6, 6.7, 6.8 show the price elasticity and the model calibration for all twelve models split in three graphs on Intercontinental ds and Medium haul ds respectively. From the plots it can be seen, that the LGB ll model is not well calibrated, estimating very high probabilities of purchase for the customers on both medium haul and intercontinental flights. It can be observed moreover that the DT model is also estimating high probabilities and is not perfectly calibrated on neither intercontinental nor medium haul flights compared to the other models. In contrast to the high probabilities of the DT model, the LogReg model predicts significantly lower probabilities than the other models on the intercontinental data

and is not very well calibrated either. The XGB llprob model is calibrated worse than the other models on both data sets (see figures 6.4 and 6.7), giving higher probabilities on the medium haul flights and both lower probabilities for lower prices and higher probabilities for higher prices on intercontinental flights compared to the other models.

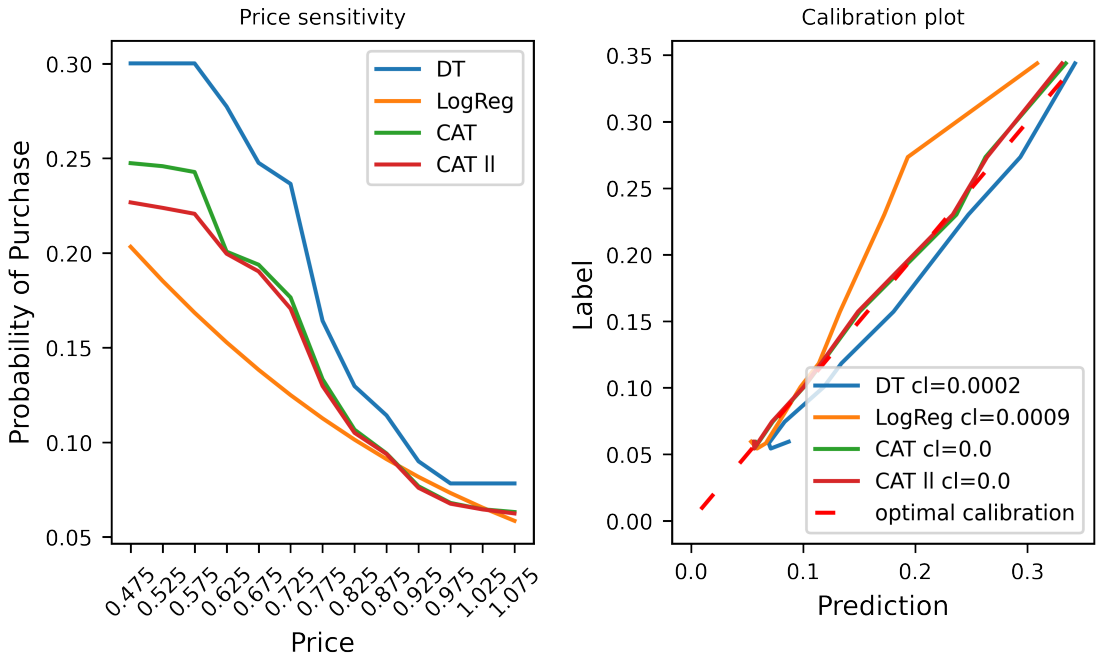

Figure 6.3: Price sensitivity & Calibration Intercontinental (1)

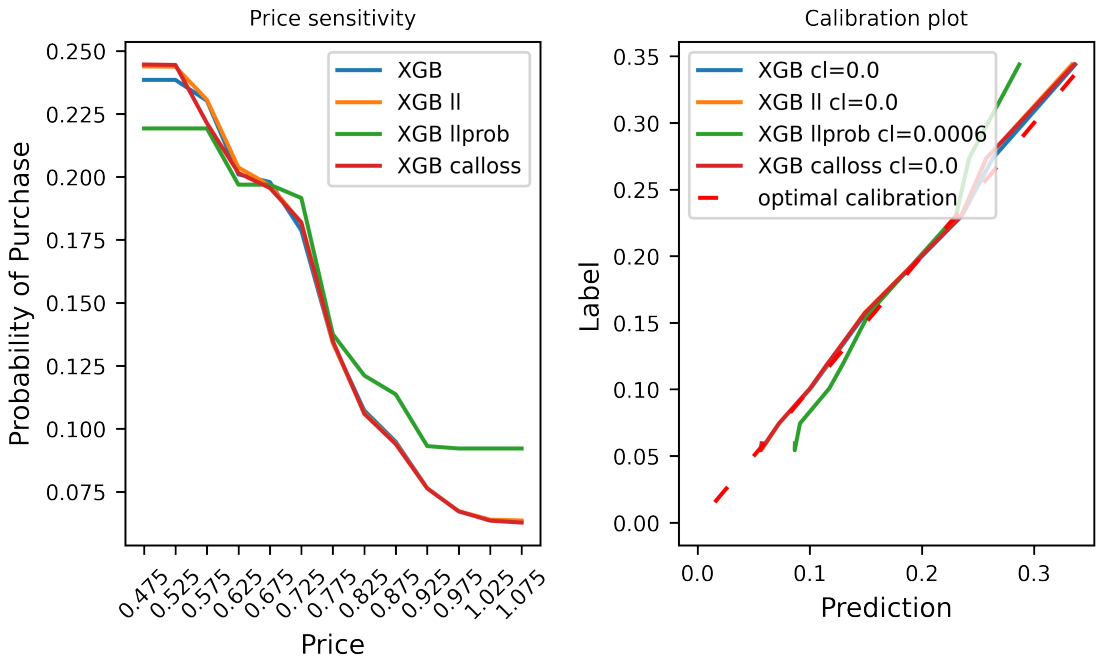

Figure 6.4: Price sensitivity & Calibration Intercontinental (2)

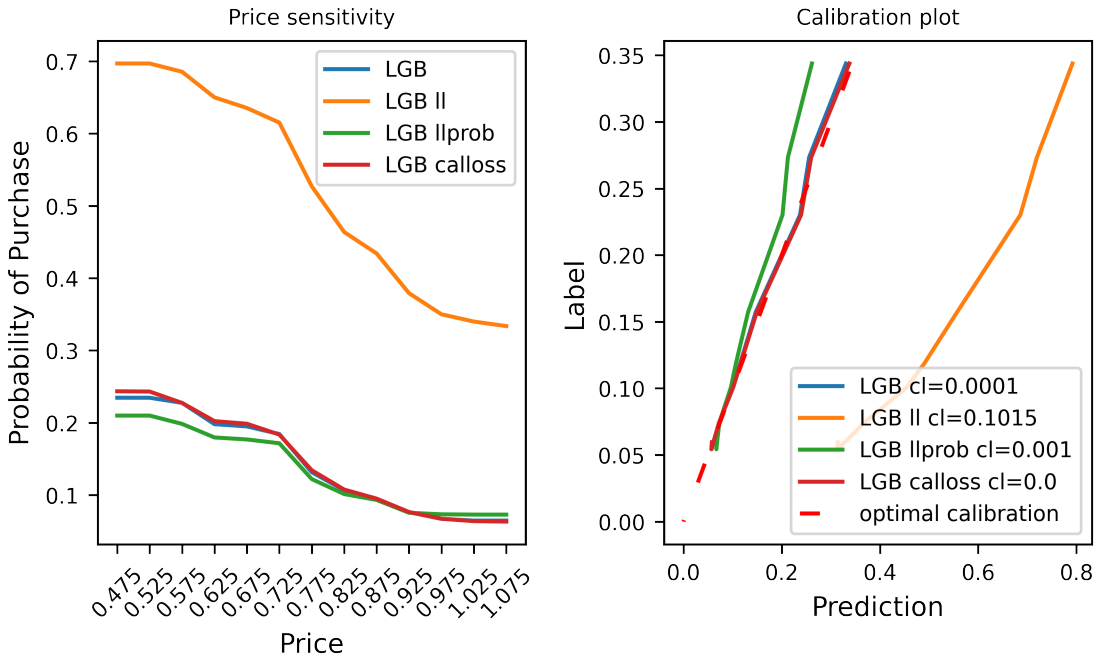

Figure 6.5: Price sensitivity & Calibration Intercontinental (3)

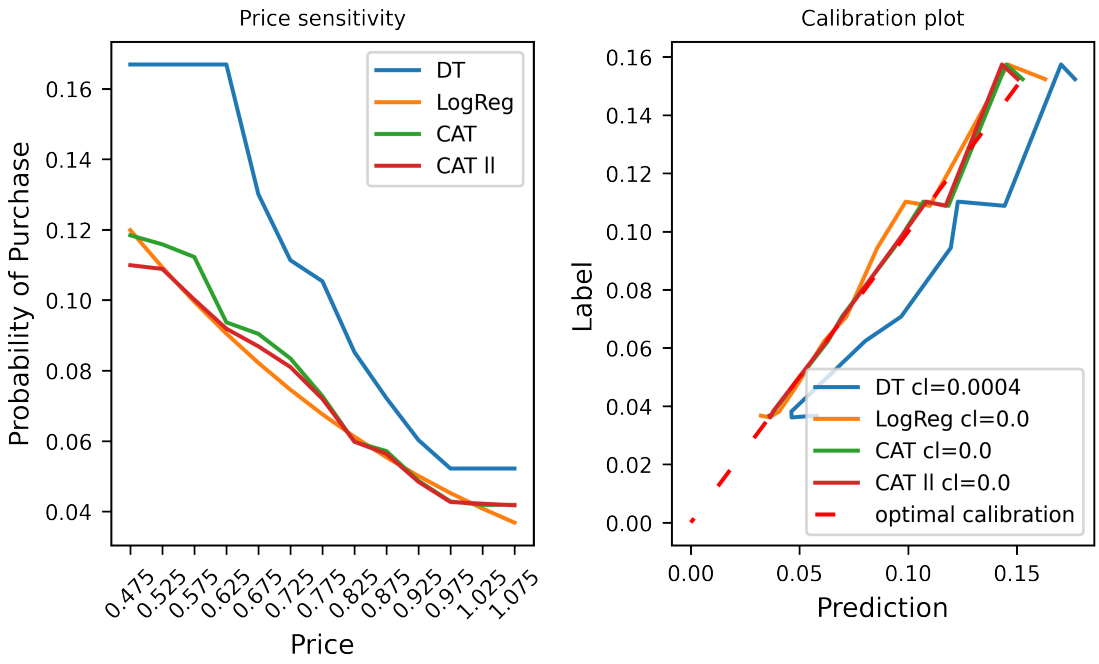

Figure 6.6: Price sensitivity & Calibration medium haul (1)

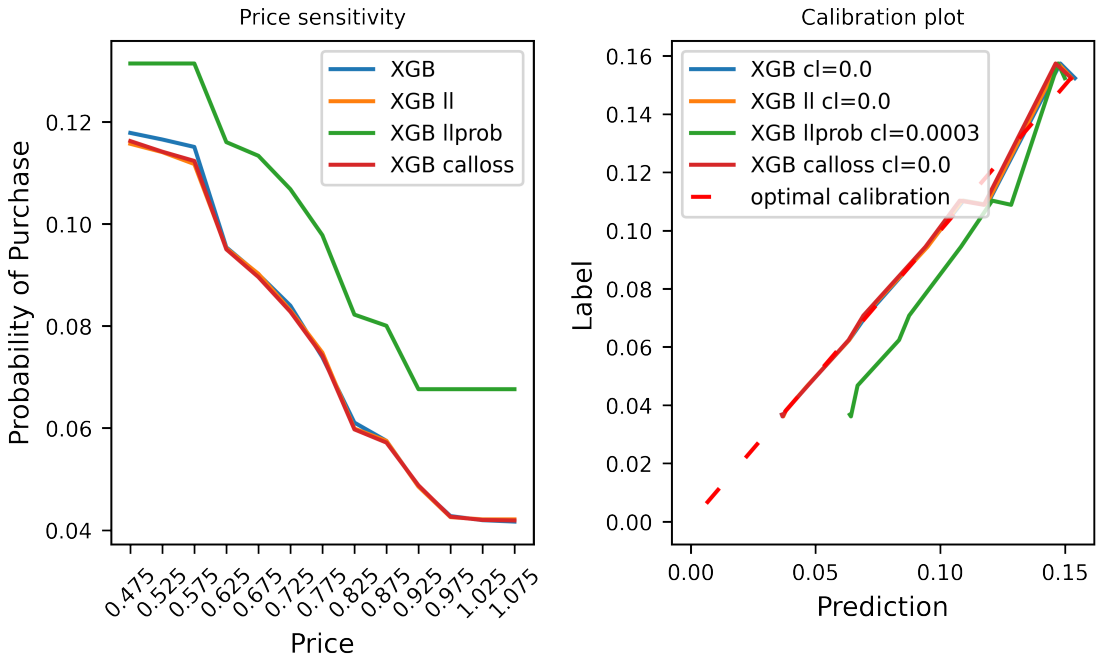

Figure 6.7: Price sensitivity & Calibration medium haul (2)

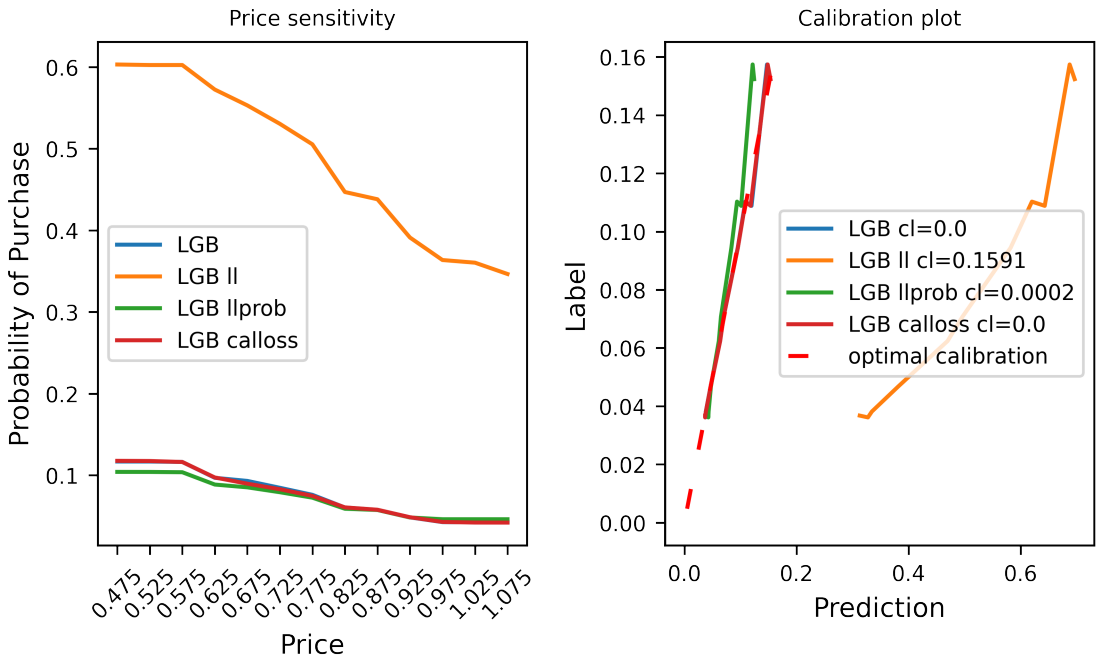

Figure 6.8: Price sensitivity & Calibration medium haul (3)

After the model results described previously, optimization does not need to be conducted with all twelve models on both data sets. The XGB llprob and LGB ll model are exluded from both data sets because of their high biases and poor calibration. The DT model is excluded as well, as it is calibrated worse than the other models on both data sets.

6.4 Optimization results

The remaining nine models were utilized for pricing optimization with the aim to maximize total revenue. The outcomes are assessed in two steps: First, the passenger-level optimization (Passenger opt) is compared to the true revenue in Section 6.4.1. Second, in Section 6.4.2, Passenger opt will be compared to Simple and Flight opt.

6.4.1 Passenger opt and true revenue

The results of the revenue change of Passenger opt with respect to the true revenue are displayed in tables 6.4 and 6.5 for intercontinental and medium haul data respectively. The minimal and maximal revenue increase expected do often not include the predicted revenue increase, because this interval is shifted by the bias. The interval includes two standard deviations of the error, meaning that with a probability of 95%, the actual revenue increase will be in the corresponding

interval.

The logistic regression model is on both data sets clearly outperformed by the boosting models. The XGB calloss model reaches the highest upper bound for the expected revenue of 23.63% on intercontinental and 19.52% on medium haul flights, but has one of the highest biases and greatest deviations from the predicted revenue increase.

All boosting models are giving similar results for intercontinental passengers, excluding the LGB llprob model, which is performing worse than the others with a comparably high bias. The average expected revenue increase is between 18.95% and 20.5%.

On medium haul flights the CATBoost models achieve less revenue increase than XGBoost and LightGBM except the LGB llprob model, which is performing similar to the CAT model. The average expected revenue increase of the boosting models is lower than on intercontinental flights and lies between 14.03% and 15.37%.

Model	**Revenue increase predicted (%)**	**Min revenue increase expected(%)**	**Max revenue increase expected(%)**
LogReg	9.66	8.60	10.86
XGB	19.66	19.82	21.39
XGB ll	19.77	20.18	21.87
XGB calloss	18.87	21.91	23.63
CAT	17.54	18.55	18.55
CAT ll	17.49	19.82	22.17
LGB	19.66	18.73	20.42
LGB llprob	16.63	13.21	14.95
LGB calloss	19.94	19.38	21.17
AVG (excl. LogReg)	18.69	18.95	20.5

Table 6.4: Passenger opt results (0.85, 1.1] Intercontinental

Model	Revenue increase (%)	Min revenue increase (%)	Max revenue increase (%)
LogReg	8.75	6.40	8.02
XGB	15.56	14.21	15.85
XGB ll	16.21	15.75	17.06
XGB calloss	15.27	17.96	19.52
CAT	11.68	10.26	10.26
CAT ll	11.11	13.95	15.79
LGB	16.68	14.75	16.15
LGB llprob	15.33	10.37	11.81
LGB calloss	16.66	15.04	16.58
AVG (excl. LogReg)	14.81	14.03	15.37

Table 6.5: Passenger opt results (0.85, 1.1] medium haul

After looking at the revenue increase that is expected with the optimization on passenger level, in figures 6.9 and 6.10 the frequency of the prices assigned to the customers is displayed for the two data sets. It becomes evident from the graphs, that all models grant the highest possible discount (of 12.5%) to a great majority of the customers.

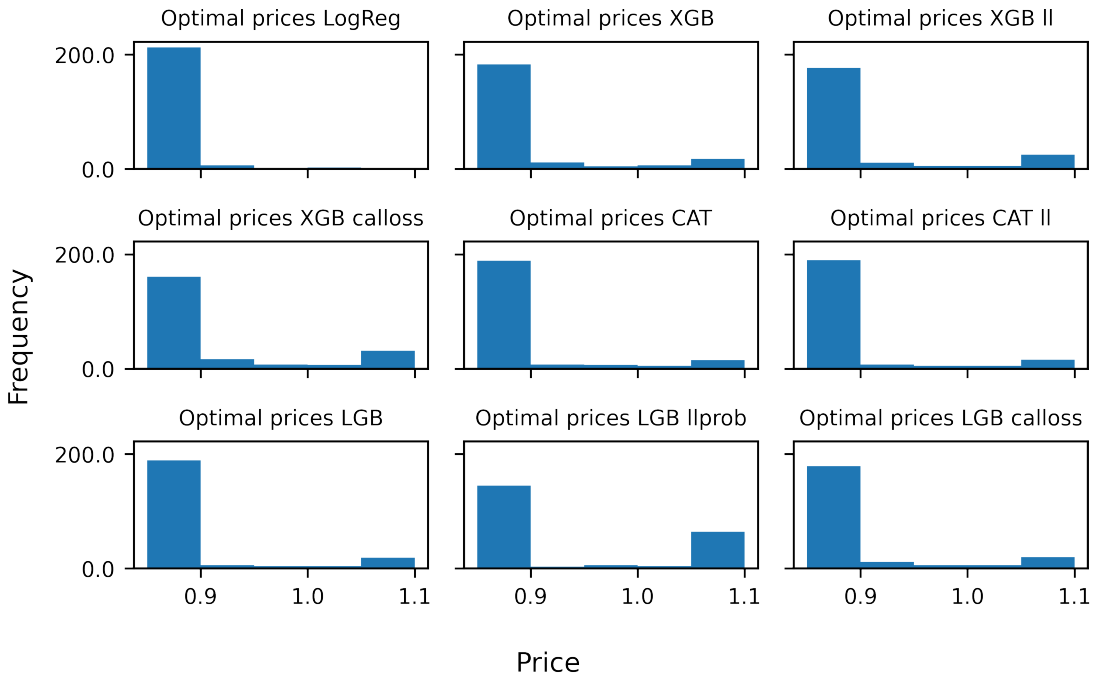

Figure 6.9: Optimal prices Intercontinental

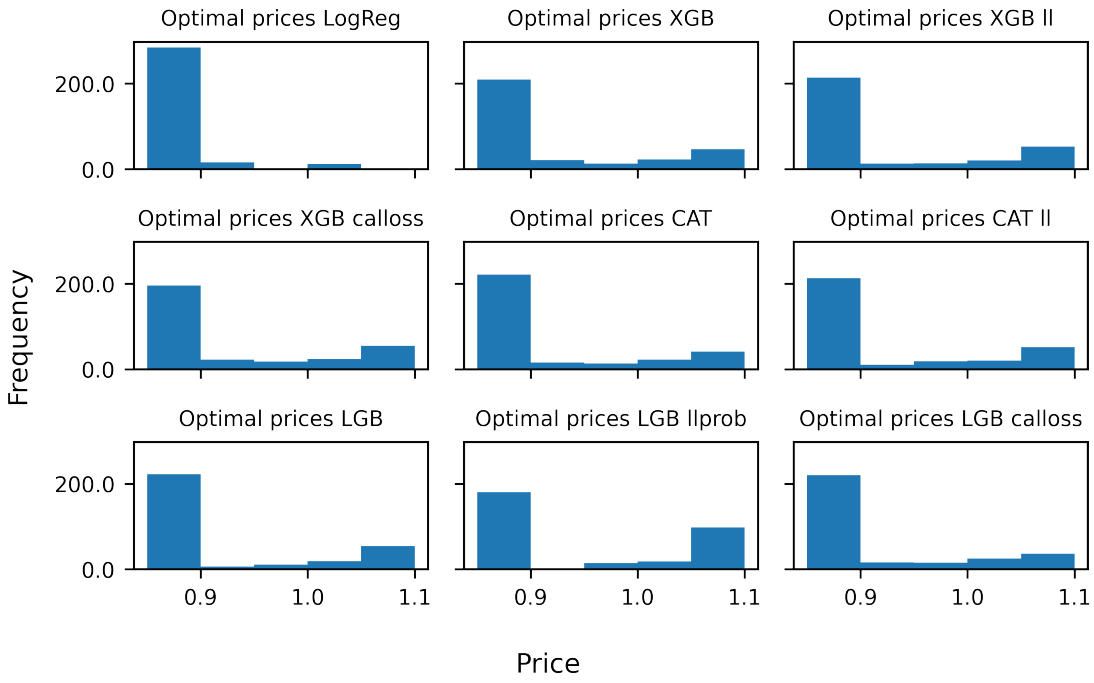

Figure 6.10: Optimal prices medium haul

6.4.2 Different optimization levels

The different optimization levels are compared with regards to their total revenue using the models XGB ll and LGB. The results are displayed in figures 6.11 and 6.12. As seen previously, the Passenger opt method already achieves the true revenue from the moment it can give prices as low as 0.975 to intercontinental passengers and 1.025 to medium haul passengers. The Passenger opt method outperforms both, Simple opt and Flight opt for all minimal prices on both data sets, although the gap seems to be biggest for high minimal prices. The Simple opt method always obtains the lowest revenue, except for the minimal price of 0.925, where it achieves a higher revenue than Flight opt on intercontinental data and with the XGB ll model also on medium haul data. For minimal prices lower than 0.925 also the Simple opt and Flight opt methods are able to generate more revenue than the true revenue.

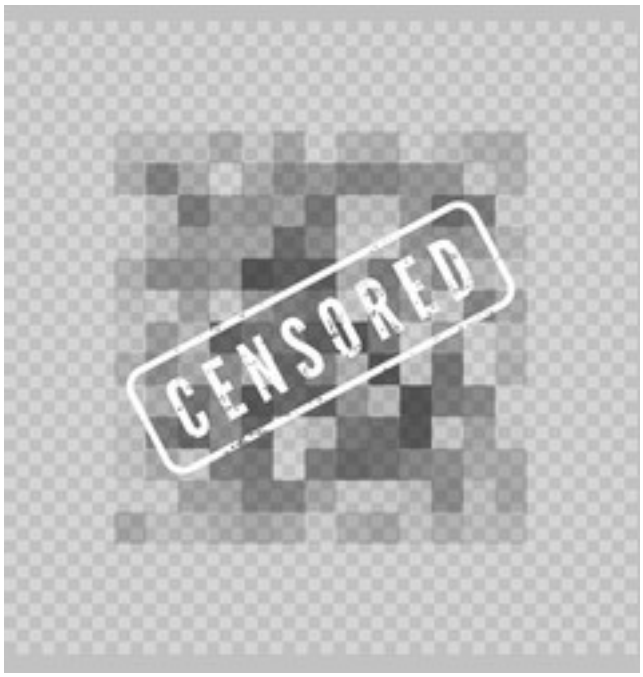

Figure 6.11: Comparison true revenue with different optimizations for different minimal prices intercontinental (Revenue)

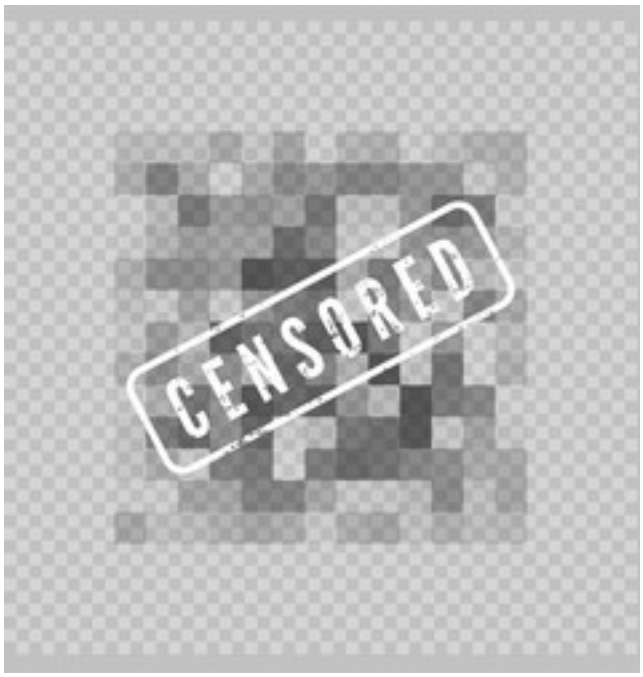

Figure 6.12: Comparison true revenue with different optimizations for different minimal prices medium haul (Revenue)

In tables 6.6 and 6.7 the difference in revenue between the Passenger opt and the Flight opt method is given in percent for all models and different minimal prices. As already seen in the plot of XGB ll and LGB (figures 6.11, 6.12), the optimization on passenger level always yields a higher revenue than the optimization on flight level. Interestingly, the percentages are very different depending on the minimal price and the model used. Most models estimate the highest performance difference of the two methods for a minimal price of 0.975, where Passenger opt method reaches on average 7.9% more revenue than the Flight opt method for intercontinental passengers, and about 6.71% more revenue for medium haul passengers. For the minimal price of 0.875 which was used in the previous section when comparing Passenger opt to the true revenue, the difference between the two optimization methods is smallest on both data sets, with 1.76% for intercontinental and 3.09% for medium haul customers. The XGB calloss model estimates on average the highest difference over all the minimal prices of 4.24% and 5.57% on intercontinental and medium haul respectively. Passenger opt and Flight opt achieve very close total revenues when using the price elasticity of the LogReg model, with a difference of only 0.11% and 0.07%.

Model	**Minimal price M (%)**							**AVG**
	0.725	**0.775**	**0.825**	**0.875**	**0.925**	**0.975**	**1.025**	
LogReg	0.22	0.16	0.11	0.09	0.09	0.08	0.00	0.11
XGB	2.03	2.06	1.73	1.29	3.28	7.95	3.95	3.18
XGB ll	2.16	2.33	1.92	1.69	4.19	8.62	3.70	3.52
XGB calloss	2.89	3.21	2.96	2.77	4.91	8.69	4.24	4.24
CAT	2.98	2.69	1.74	1.24	2.17	8.00	3.38	3.17
CAT ll	2.23	2.24	1.53	1.13	2.87	8.30	3.22	3.07
LGB	1.46	1.45	1.36	1.06	4.19	9.03	3.40	3.14
LGB llprob	1.57	2.12	5.42	3.33	5.59	4.86	0.74	3.38
LGB calloss	1.92	2.17	1.86	1.62	3.38	8.10	3.89	3.28
AVG (excl. LogReg)	2.15	2.28	2.31	1.76	3.82	7.94	3.31	3.37

Table 6.6: Difference (%) Passenger opt and Flight opt intercontinental

Model	Minimal price M (%)							AVG
	0.725	**0.775**	**0.825**	**0.875**	**0.925**	**0.975**	**1.025**	
LogReg	0.10	0.08	0.08	0.08	0.09	0.09	0.00	0.07
XGB	4.29	3.28	4.74	2.71	4.87	7.20	3.76	4.41
XGB ll	4.16	3.34	4.82	2.96	6.34	7.32	3.89	4.69
XGB calloss	5.77	4.70	6.11	3.95	5.64	8.26	4.56	5.57
CAT	6.02	4.88	4.88	2.80	3.82	6.50	3.62	4.65
CAT ll	5.52	4.89	4.49	3.05	4.92	5.90	3.05	4.55
LGB	3.03	2.42	3.97	2.32	7.49	6.76	3.53	4.22
LGB llprob	3.69	3.13	5.06	4.06	4.03	3.89	1.30	3.59
LGB calloss	4.26	3.26	4.68	2.92	4.78	7.91	4.24	4.58
AVG (excl. LogReg)	4.59	3.73	4.84	3.09	5.23	6.71	3.49	4.53

Table 6.7: Difference (%) Passenger opt and Flight opt medium haul

After comparing the revenue differences, it is interesting to see the expected number of seats that are sold with the optimization methods and for the different minimal prices. As there is no capacity constraint, the number of sold seats could theoretically exceed the available seats. In figures 6.13 and 6.14 the optimization methods are compared using the XGB ll and LGB model again, but comparing the sold seats this time. The Simple opt method always sells the most seats, whereas all three optimization methods sell more seats than the true number of seats sold from a minimal price lower than 0.925. Especially for higher minimal prices, the difference between the amount of seats sold is greater, as the Flight opt method sells less significantly less seats than the other two optimization methods for both data sets and with both models.

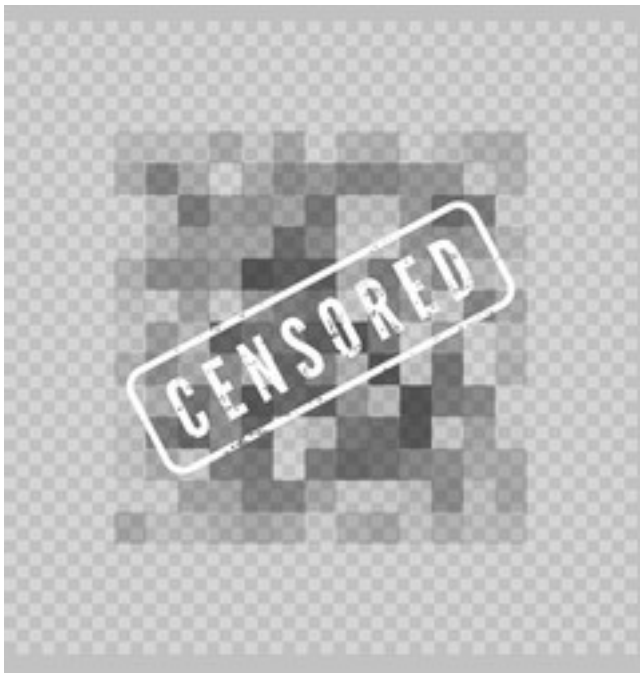

Figure 6.13: Comparison true revenue with different optimizations for different minimal prices intercontinental (Seats sold)

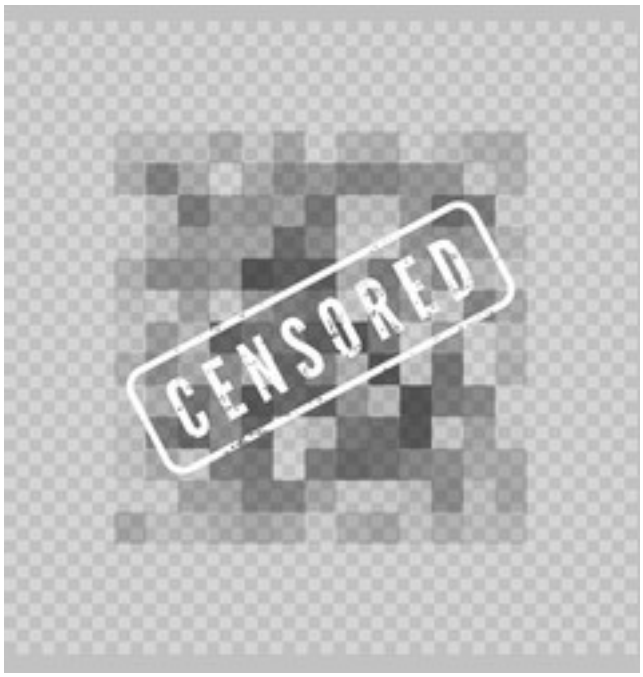

Figure 6.14: Comparison true revenue with different optimizations for different minimal prices medium haul(Seats sold)

Chapter 7

Discussion & limitations

This chapter will present a summary of the project's most important findings and examine them in relation to their limitations. It will begin by reviewing the most important features for the classification task (Section 7.1) and will proceed with a brief comparison of the models and their results (Section 7.2). The discussion will then shift to the increase in revenue and sold seats obtained through the Passenger opt approach (Section 7.3) and critically evaluate its added value in comparison to the Flight opt method (Section 7.4). Finally, in Section 7.5, the chapter will compare and contrast intercontinental and medium haul flights.

7.1 Feature importance

The feature importance was analyzed in two ways: First, before modelling in the exploratory data analysis (Section 3.4) and later with the permutation importance within the CATBoost model (Section 6.2).

In the EDA, most features had significant influence on the purchase decision on both data sets, where the most distinct trends where observed for the features `PNR Size, Age, Connection Time, Part of Connection, Business traveller, Weekend Trip` and `Point of Origin` for intercontinental passengers, and the features `PNR Size, Economy Comfort capacity,Connection Time, Part of Connection, Weekend Trip` and `Point of Origin` for medium haul passengers. Note, that most of these variables describe details of the booking and not of the travelers or flights. This is a first indication, that passenger level pricing can increase the revenue.

The permutation importance reflects the different importance of the `Business traveller` for intercontinental and medium haul flights as it was observed in the EDA. The significant relevance of the `Complex Group` in the permutation importance might partly be explained by the correlation with the absolute seat price, as discussed previously in the results (Section 6). In the EDA this

may not be so obvious, because only the seven most occurring regions were presented for both, origin and destination of the flight. In the permutation importance of CATBoost, all countries were used however, as CATBoost naturally deals with categorical data and no pre-processing is needed. As a result, the impact of less frequently visited countries is considered in this importance measure as well, which seems to make the feature more important. The ranking given by the permutation importance is however not necessarily the whole truth, as it can change fundamentally when adding or dropping single features, and also gives different results than other importance measures. So while it is a good indicator of features that have influence and features that do not (assuming a low correlation between all variables), it should be seen together with the EDA and one might even consider looking at different importance measures.

7.2 Model Comparison

Apart from the DT, LGB ll and XGB llprob models, all other nine models were found to have a reasonably low prediction bias, as well as a good model calibration and logical price elasticity. While XGB and XGB ll had the lowest bias on medium haul data of 0.35% and -0.13%, the models LogReg, LGB and LGB calloss were the most accurate for intercontinental passengers with biases of only -0.05%, 0.06% and -0.22%. However, with an average absolute bias of 1% for intercontinental and 1.26% for medium haul flights, all models have very similar biases with similar standard deviations of the error. The highest potential revenue increase compared to the true revenue is predicted by the XGB calloss model, with an upper bound of 23.63% on intercontinental and 19.52% on medium haul data.

To better understand the model differences in price elasticity predictions on passenger level, the added value of the passenger information in the models was compared in tables 6.6 and 6.7. Looking at the average added value of passenger level prediction versus flight level over the different minimal prices, the XGB calloss model shows a greater difference than the other models, which could mean that this model is specifically good in understanding the customer willingness to pay.

7.3 Passenger optimization and true revenue

This section will discuss the change in revenue and sold seats of the Passenger opt approach separately.

Revenue The expected revenue increases by the Passenger opt method are very promising on first sight, with an expected increase of 19-20.5% on intercontinental and 14-15.4% on medium haul flights. This is under the assumption, that prices can only be selected between 0.85 and 1.1. Based

on the known revenue generated in 2022, this approach would recover 61.7% and 13.4% of the rough maximum additional revenue estimate that was computed in Section 1.2 on intercontinental and medium haul flights, respectively.

When looking at the suggested prices (figures 6.9, 6.10), it becomes clear that the passenger-level optimization of each of the models gives the lowest possible price to a great majority of the customers. This suggests that the prices are simply too high, as lowering the base price already increases the overall revenue significantly. The question that poses itself now, is why that has not happened yet, if it seems to be so evident? One answer is that there is more to it than just the EC price. As already mentioned when explaining the price restrictions for the optimization in Section 4.4, the seat prices for the EC section have to be seen in relation to all other seat categories, which is not done here in the model. Another answer can be found by examining a limitation of this approach. The overall expected revenue gain is determined and constructed on the price elasticity curve as predicted by the model over all passengers. The revenue increase is substantially greater for the steeper curves of the boosting models than for the more moderate curve of the logistic regression model, for instance. Looking back at the way the price feature was created in Section 3.3, all prices were estimated and even the known prices were overwritten. So, while the price elasticity follows a logical course, it is unclear what the true curve looks like. The steepness of the true curve can have a significant impact on the outcomes presented in this framework.

To see this relation of probability of purchase versus price better, in Figure 7.1 the prices were multiplied by its probability of purchases for all approximately 230,000 customers of the intercontinental data test set for the XGB ll model and the resulting expected revenue fraction for all customers aggregated in box plots for all the prices. It becomes clear, that the average expected revenue does in fact increase for lower prices. This means that the price elasticity curve is steep enough, that the revenue loss from a price decrease is more than balanced out by an increase in purchase probability.

Looking into the price elasticity more closely, the only reason why it follows this desirable trend after overwriting all prices before modelling (see figures 3.9, 3.10) are the frequent flyers. As already seen in the EDA (Section 3.4), with increasing frequent flyer level the probability of purchase increases. When all prices are overwritten, the frequent flyer discounts are applied to the estimated prices afterwards, meaning that most *Gold* members will be around the price of 0.5, *Silver* members around 0.75 and *Explorers* around 0.9. This makes the price elasticity decrease automatically with increasing price. The problem with frequent flyers is however that it is not possible to say whether they buy the seats because of the lower price or because they are flying often, hence perhaps have a higher income, need/want more comfort, etc. This is a great limitation

of the price feature, which will automatically reflect on the predictions of the models.
If the true prices would have been used, the result would be a clear indication to lower the prices if possible. However, in this framework with estimated prices, the influence of a possibly different estimated price elasticity could completely change this result in both directions. For this reason it can also not be said for sure that the revenue would increase by these great percentages, if a boosting model with price optimization on passenger level was deployed. The true prices could give a more certain answer.

Sold seats In 2022, the paid load factor of the Economy Comfort section was [redacted]% and [redacted]% on intercontinental and medium haul flights respectively. The following calculation will compute the new load factor and seat capacity bounds based on the XGB ll model whose sold seats for various minimal prices are plotted for all optimization levels in figures 6.13 and 6.14. With [redacted] sold seats in the intercontinental test set and [redacted] sold seats in the medium haul test set, the target paid load factor of 95% would allow for [redacted] and [redacted] sold seats, respectively. According to the figures, the minimal prices that get closest to this objective number of sold seats is 0.875, which sells [redacted] seats on intercontinental flights and a minimal price of 0.775, for which [redacted] seats are sold on medium haul flights. This would results in a paid load factor of 90% and 92% on intercontinental and medium haul flights, respectively.
This relatively high paid load factor is accomplished without any capacity constraints on flight level. Consequently, for a higher load factor the probability increases that on specific flights more seats were sold than the capacity allows. Taking this capacity on flight level in to account is an interesting addition for any possible framework in the future.

7.4 Comparison optimization levels

The aim of this section is to get an idea of the value added by the passenger level optimization compared to higher levels like flight level or simply one price for all passengers.
The results of this comparison were plotted in figures 6.11 and 6.12. In this comparison, the passenger level optimization was always outperforming the other optimization methods, although the revenue difference was significantly less impressive than in the previous section. Depending on the minimal price allowed, Passenger opt was on average 3.37% higher than Flight opt based on the boosting models for intercontinental, and 4.53% higher for medium haul passengers.
Especially for $M = 0.975$, Flight opt yields a significantly lower revenue than Passenger opt. The reason for that are most likely the frequent flyers discounts. While Passenger opt already knows

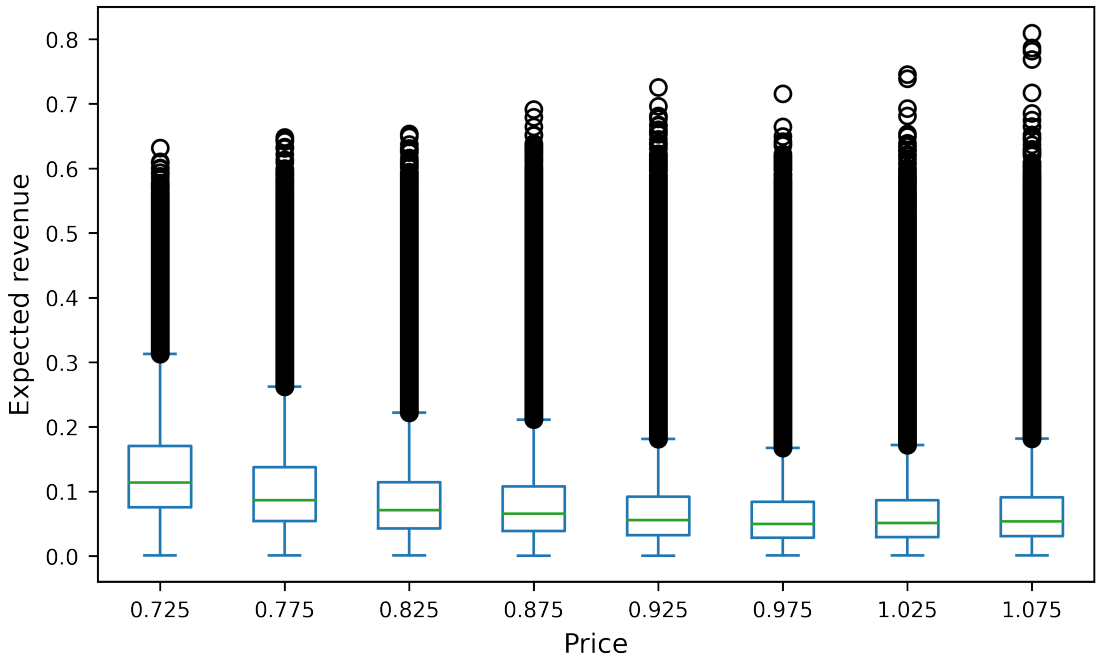

Figure 7.1: Expected revenue fraction XGB ll model intercontinental

the discounts on passenger level when optimizing the prices, Flight opt is simply looking at the whole flight and thus can only select from the prices the passengers see, which are all restricted by the minimal price M. The passenger-level optimization is using the same range for all customers without frequent flyer discount, but is optimizing in adjusted price ranges for all other passengers. Why this makes a bigger difference for a minimal price of 0.975 can be seen in the expected revenue fraction in Figure 7.1. Because the estimated revenue for all passengers is larger for higher rates (when selecting prices between 0.975 and 1.075), the flight optimization algorithm will most likely choose higher pricing for more passengers aggregated by flights. While this may be an excellent option for normal passengers, the optimization is completed within the adjusted interval for persons with discounts. For example, the best *price paid* for *Silver* members is between 0.725 and 0.825, which equates to a *price seen* equivalent to that of a passenger without frequent flyer level (between 0.975 and 1.075). From the revenue fraction figure for all prices (Figure 7.1) it can be seen that in most circumstances, the optimization will choose the lowest price in the adjusted price interval because it will generate the maximum expected revenue. However, only the passenger optimization technique can account for such changes, and as a result, it outperforms the flight optimization method greatly in this scenario.

Because the same price elasticity is employed in both cases, the distinction between Passenger opt and Flight opt can be stated with greater certainty. A limitation of this comparison is however,

that for the Flight opt, the probabilities of purchase were first estimated on passenger level and then aggregated on flight level, rather than using solely flight data for the price elasticity prediction. While this limitation could affect the results in either direction, it is more likely that predicting price sensitivity on passenger level and then aggregating gives a better understanding of the customers' willingness to pay on each flight than using only flight data and aggregating all passengers from the start. As a consequence, how Flight opt is defined here is thought to provide a more accurate prediction of price sensitivity than the prediction results based only on flight data.
Although the percentages may differ for real data, the comparison revealed that learning and predicting the price elasticity on passenger level can contribute an extra 3-5% to the flight-level estimation. Passenger level pricing optimization is achieved using feature engineering and gradient boosting models that successfully categorize clients enough to increase revenue.

7.5 Intercontinental and Medium haul

The results have shown that the added value of passenger-level pricing over flight-level pricing is greater for medium haul than for intercontinental passengers. In general, if customers are less price sensitive, the added value of passenger-level pricing should be higher because the pricing decision depends on the passengers' features rather than the price. When customers are extremely price sensitive, however, passenger characteristics should be less important because the price would be the primary driver of the pricing decision, resulting in a lower added value of passenger-level optimization. Consequently, people on medium haul flights may be less price sensitive than those on intercontinental trips. This observation is further supported by the price elasticity curves over all passengers in the results Section 6, where probabilities of purchase decrease on intercontinental flights from approximately 24% to 7% with increasing prices while medium haul purchase probabilities decrease from around 12% to 5%.
In the previous discussion, it was stated that the Passenger opt method applied to intercontinental flights is expected to recover 61.7% of the maximum additional revenue whereas only 13.4% can be recovered on medium haul flights. This difference could be the result of the different price sensitivities of the passengers. Firstly, the assumption in the computation of the maximum additional revenue is, that all seats up to a paid load factor of 95% can be sold at a 50% discount on the total average price, which is more realistic for the price-driven passengers of intercontinental flights than for medium haul passengers. This upper bound for the additional revenue is on medium haul flights 110% higher than the total revenue of 2022, while the upper bound on intercontinental flights would increase this revenue by only 38.2%. This already suggests that for medium haul flights this bound is extremely optimistic. Secondly, the lower purchase probabilities of medium

haul passengers in general lead to lower expected revenues, regardless of the price. This also shows, that the EC seat on medium haul flights is simply 'worth' less than on intercontinental flights. It stands to reason, that a comfortable seat is valued more on longer flights than on shorter flights. Summarizing this findings, optimization in general appears to increase the revenue on intercontinental flights more, while specifically the traveler details add more value on medium haul flights.

Chapter 8

Conclusion & future work

This project was divided into three parts. Initially, the data was studied, features were created through the combination and extraction of the raw data and their influence on customer purchase decisions was analyzed. To provide a suitable price elasticity, the pricing feature was generated from the partially existing prices that were later overwritten. Following this, the classification models Logistic regression, Decision Tree, XGBoost, LightGBM and CATBoost were selected, trained and evaluated in terms of revenue prediction bias, model calibration and estimated price elasticity. The three boosting methods were additionally tuned on different metrics and the resulting models were included in the comparison. Lastly, the estimated price elasticity per passenger was used to optimize prices on various levels and the consequent expected revenues were compared to the true revenue. Furthermore, the optimization on passenger level was compared to the flight level and global optimization methods in order to comprehend the added value of detailed passenger information for the EC seat price revenue.

Although not every detail about the passengers is recorded, various interesting tendencies were discovered, especially among booking-related data, that KLM might investigate. For instance, it is significant how people with a connecting flight buy less seats than people without. Perhaps, the booking flow might be modified to make it more appealing for customers to purchase seats for their entire trip. Another point to consider is why the booking channel *Direct Offline* sells significantly fewer seats on medium haul flights than on intercontinental flights.

All boosting models outperformed the basic logistic regression in terms of revenue increase while maintaining an acceptable prediction bias. Although none of the eight boosting models selected for optimization is strictly outperforming the others, XGB calloss predicts the highest revenue increase and is capable of increasing the revenue most from passenger- to flight-level prediction.

The absolute expected revenue increase from deploying any of the boosting models and making pricing decisions on passenger level is difficult to predict with certainty, as many of the true prices

were not recorded (everyone who did not buy). If KLM recorded all the seat prices a passenger was offered at different stages (e.g. ticket purchase, online check-in, last minute upgrade options), the prices would not have to be estimated prior to modelling, and the outcome in terms of total revenue increase would be more reliable. The revenue increase would therefore be determined by the freedom of the model in optimizing the price, i.e. how the minimal price M is set.

Following the comparison of the passenger level pricing versus flight level pricing, it was discovered that passenger level optimization is expected to increase the revenue by approximately 3.3% on intercontinental flights and by around 4.5% on medium haul flights, depending on the model's minimal price.

Overall, this research has demonstrated how existing historical data may be leveraged to increase revenue. With the true prices as input, this project concludes that it would be possible to use the historical data to predict the willingness to pay for the Economy Comfort seat class and optimize prices on passenger level in the presented framework. Next to this conclusion, additional insights were given into the characteristics of the passengers and their influence on their decision to purchase Economy Comfort seats.

A careful tracking of all offered seat prices at various phases, as well as some altering of the basic rates, would be required to improve the outcomes. With the recorded true prices, it might be helpful to reduce the size of the price bins, which were set to 5% in this study. A more granular price selection could benefit the optimization even further. In addition, it would be interesting to integrate other features and possibly investigate how the feature space could be reduced with the use of auto-encoders in order to successfully use a bigger number of features. However, it cannot be guaranteed that the results presented would improve significantly with a greater amount of features.

While this project treated all Economy Comfort seats equally, they have multiple characteristics in reality describing the location of the seat in the aircraft, i.e. Window, Middle, Aisle, Front, Center, Back. These characteristics could be integrated into the framework, by pricing these seats differently for the passengers. A booking with two people for instance, values the middle seat significantly more than a single traveler.

For future work, it would be important to understand how the prices of different seat products influence each other and how the preference for specific seats shifts with price changes in one seat category. With this understanding, it would be possible to predict customer preferences for different seat products and all prices could eventually be optimized simultaneously.

It could also be interesting, to add the section capacity in the optimization framework. This is in general a different optimization problem, which gives the optimal cabin sizes and seat distribution given the expected demand for a flight, but in this project, it is not taken into consideration that

it might be beneficial to save some capacity for passengers with a higher willingness to pay who might book their ticket later. This project as well as other research has shown, that there is still tremendous potential and unexplored possibilities in the domain of airline seat pricing.

Bibliography

[1] Catboost feature importance. URL `https://catboost.ai/en/docs/concepts/fstr#regular-feature-importance`.

[2] Machine learning in python scikit-learn package documentation. URL `https://scikit-learn.org/stable/index.html`.

[3] Lightgbm documentation. URL `https://lightgbm.readthedocs.io/en/latest/index.html`.

[4] Xgboost parameter documentation. URL `https://xgboost.readthedocs.io/en/stable/parameter.html`.

[5] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54:1937–1967, 2021.

[6] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[7] James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D Cox. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1):014008, 2015.

[8] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[9] Robert G Cross. *Revenue management: Hard-core tactics for market domination*. Currency, 2011.

[10] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[11] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[12] Guillermo Gallego and Masoud Talebian. Demand learning and dynamic pricing for multiversion products. *Journal of Revenue and Pricing Management*, 11:303–318, 2012.

[13] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[14] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[15] Josh Stammer. XGBoost. YouTube video, December 2019. URL `{https://www.youtube.com/watch?v=OtD8wVaFm6E}`.

[16] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[17] Brent Komer, James Bergstra, and Chris Eliasmith. Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. In *ICML workshop on AutoML*, volume 9, page 50. Citeseer Austin, TX, 2014.

[18] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21, 2013.

[19] Suranga Perera and David Tan. In search of the "right price" for air travel: First steps towards estimating granular price-demand elasticity. *Transportation Research Part A: Policy and Practice*, 130:557–569, 2019.

[20] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[21] Masoud Talebian, Natashia Boland, and Martin Savelsbergh. Pricing to accelerate demand learning in dynamic assortment planning for perishable products. *European Journal of Operational Research*, 237(2):555–565, 2014.

[22] Saravanan Thirumuruganathan, Noora Al Emadi, Soon-gyo Jung, Joni Salminen, Dianne Ramirez Robillos, and Bernard J Jansen. Will they take this offer? a machine learning price elasticity model for predicting upselling acceptance of premium airline seating. *Information & Management*, 60(3):103759, 2023.

[23] Natalya Vinokurova. Reshaping demand landscapes: How firms change customer preferences to better fit their products. *Strategic Management Journal*, 40(13):2107–2137, 2019.

[24] Yang Yang, Wan-Ling Chu, and Cheng-Hung Wu. Learning customer preferences and dynamic pricing for perishable products. *Computers & Industrial Engineering*, 171:108440, 2022.