Vrije Universiteit Amsterdam

Master Project Business Analytics

Forecasting E-commerce Key Performance Indicators

Author:
Chi Chun Wan

November 13, 2017

Supervisor:
Mark Hoogendoorn

Supervisor:
Vesa Muhonen

VU UNIVERSITY AMSTERDAM

FASTERINTEL

Contents

Preface **3**

Abstract **4**

1 Introduction **5**

2 Background **7**

2.1 Company Description . . . 7

2.2 KPIs and Channels . . . 7

2.2.1 KPIs . . . 7

2.2.2 Channels . . . 8

2.3 Current Model . . . 9

3 Literature Review **11**

4 Data **13**

4.1 Google Analytics Data . . . 13

4.2 External Data . . . 15

4.3 Data pre-processing . . . 16

4.4 Data Analysis . . . 16

5 Methods and Models **21**

5.1 Models . . . 21

5.1.1 Ratio-based model . . . 21

5.1.2 ARIMA . . . 23

5.1.3 Random Forest . . . 25

5.1.4 Artificial Neural Networks . . . 26

5.2 Multi-step Forecasting Strategies . . . 32

5.3 Evaluation . . . 33

6 Experimental Setup **35**

6.1 Training and Testing . . . 35

6.2 Experiments . . . 35

6.2.1 Experiment A . . . 35

6.2.2 Experiment B . . . 36

6.3 Hyperparameters Optimization . . . 37

7 Results **39**
7.1 Experiment A 39
7.1.1 Hyperparameters 39
7.1.2 Model Performance 45
7.1.3 Feature Importance 45
7.1.4 Model Comparison 47
7.2 Experiment B 51
7.2.1 Hyperparameters 51
7.2.2 Model Performance 51
7.2.3 Feature Importance 52
7.3 Summary 54

8 Discussion and Conclusion **55**

References **57**

Appendices **60**

A Additional Results Data Analysis **60**
A.1 The KPIs over monthly periods 60
A.2 The KPIs over weekdays 60

B Model Performance Experiment A **60**
B.1 Visits 60
B.2 Transactions 60
B.3 Revenue 60
B.4 Media Spend 60

Preface

This internship report was written as the final part of the Master's program in Business Analytics at the Vrije Universiteit Amsterdam. The goal of the Master's program in Business Analytics is to improve business performance by applying a combination of methods that draw from mathematics, computer science and business management. The internship was performed at FasterIntel. During this internship, I have focused on the forecast of e-commerce KPIs with the goal to improve this forecast. The present thesis reports the results of my internship.

I would like to thank FasterIntel for giving me the opportunity to complete my thesis. I would also like to thank Eduard Belitser for being my second reader. Finally, I want to give special thanks to my internal supervisor Vesa Muhonen and VU supervisor Mark Hoogendoorn. They have provided me great guidance throughout the internship and useful feedback on the process and the report and I am very grateful for that.

Chi Chun Wan
Amsterdam, November 2017

Abstract

Key performance indicators, or KPIs, are important metrics used in organizations to indicate their progress towards a defined business goal. In the competitive e-commerce world, it is important to obtain actionable insights as soon as possible. Forecasting KPIs can help these e-commerce companies to have foresight to act on with their actions to ensure business goals will be achieved. The main objective of this thesis was to improve the prediction of KPIs using historical data. Moreover, the prediction will be implemented in the software platform of FasterIntel and will be extended for multiple e-commerce companies. Therefore, only data that is available in this software platform and data that is generic enough to be available from all companies will be used. These data are web tracking data from Google Analytics and publicly available data.

In this thesis, we investigated eight different KPIs where four of them are predicted and the remaining four are derived from the predictions. The four predicted KPIs are the visits, transactions, revenue and media spend. Moreover, the forecast is based on 7 days ahead forecast. Linear and non-linear models were implemented to produce the forecast. The linear models are the Ratio-based model and ARIMA and the non-linear models are Random Forest, Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM).

We conducted two experiments. In the first experiment, the past history of the KPIs and time derived data from Google Analytics were used and in the second experiment we extended this with external data and other Google Analytics features. The results showed that this external data and these Google Analytics features did not improve the forecast. The best performance achieved for visits and transactions was with Random Forest and for revenue and media spend with MLP. The improvement compared to the current model, which is the simple linear regression, was 41.75%, 17.04%, 17.02% and 56.00% for visits, transactions, revenue and media spend, respectively. However, the forecast error was still large. The reason is the limited historical data available and the generic data. More specific data might be useful to improve the performance further.

1 Introduction

With the evolution of the internet from a basic tool of communications into an interactive market of products and services [1], many enterprises across the world attempt to embrace e-commerce. E-commerce, also known as electronic commerce, consists of electronic business transactions to purchase and deliver goods and services. E-commerce sales worldwide has grown from 1.37 trillion US dollars to 1.86 trillion US dollar in 2016 and it is expected to grow to 4.48 trillion US dollars in 2021 [2].

To aim for better performance in e-commerce, measuring the key performance indicators of e-commerce websites are important and cannot be overemphasized. Key performance indicators, or KPIs, are metrics that organizations track to measure their progress towards a defined business goal. By tracking and measuring these indicators, recommendation for operational improvement can be made based on actual data.

There are many KPIs for an e-commerce website, some of which are website traffic, conversion rate, sales and revenue. In the field of KPIs, a lot of research has been done on finding factors that affect KPIs such as factors that will have impact on sales [3, 4, 5]. However, for forecasting KPIs, most research has only focused on sales forecast [6, 7, 8]. For example, Taylor [6] has focused on the forecast of daily supermarket sales to be applied in inventory control systems. Lenort and Besta [7] have focused on the forecast of apparel sales to improve the effectiveness of the retailer's sourcing strategy. In this research we will focus on the forecast of more KPIs besides sales. The forecast of KPIs will help e-commerce marketers to retrieve insights as soon as possible which is important in the competitive and crowded e-commerce world. They will have foresight to act on with their actions to ensure business goals will be achieved.

The scope of this thesis is the forecast of KPIs for e-commerce brands which is offered in the software product of FasterIntel. This software product is a plug and play marketing platform for e-commerce brands and it currently displays the daily predictive performance of KPIs over different channel groups until the end of a week. However, the simple linear regression which is the current model that is used for the prediction of these KPIs is not accurate enough. When e-commerce marketers rely on these predictions to make marketing decisions, these predictions should be as accurate as possible.

In this research, the goal is to improve the prediction of the KPIs using historical data. However, since the prediction will be implemented in the software product of FasterIntel and it must be scalable for multiple clients, we will encounter restrictions that makes it challenging. One of these restrictions is that only data that is available in the software product can be used. This data consists of web-tracking data from Google Analytics. Moreover, if web-tracking data alone is insufficient to make accurate predictions, then we have to find external data that is still generic enough to be used for multiple clients. This restriction leads to the following research question:

With how much can the forecast of KPIs be improved compared to the current model using only data that is generic enough to be available from all clients?

The structure of the thesis consists of eight chapters. The first chapter is the current introduction. Chapter 2 provides background information about FasterIntel, the KPIs, the channels and the current model used in the software product. Next, chapter 3 gives a literature review about the common models used for time series forecasting and their application in forecasting KPIs. Chapter 4 describes the data used followed by their pre-processing and analysis. Chapter 5 describes the methods and models that have been applied for the forecast. Chapter 6 describes the experimental setup and chapter 7 presents its results. Lastly, chapter 8 reports the discussion about the results, recommendation for future research and the conclusion.

2 Background

2.1 Company Description

FasterIntel is a digital marketing company focusing on developing a plug and play marketing platform for e-commerce brands. The features in the platform are designed to save time and to boost revenue. Features such as decision insights that enable you to focus on making the right decisions and conversion insights that provide insights and conversion rate trends. FasterIntel was separated from its parent company ConversionMob in 2017. ConversionMob, founded in 2011, is an online marketing agency that provides digital and advertising consultancy services.

2.2 KPIs and Channels

2.2.1 KPIs

FasterIntel displays the following eight most important e-commerce KPIs:

1. Visits: The total number of sessions, where a session is a continuous user interaction with your website. The session expires after 30 minutes of inactivity and at midnight.
2. Transactions: The total number of completed purchases on the website.
3. Revenue: The total revenue from web e-commerce or in-app transactions.
4. Media Spend[1]: The total amount paid for advertising within Google.
5. Conversion Rate: The proportion of visits that resulted in a transaction.

$$Conversion\ Rate = \frac{Transactions}{Visits}$$

[1] Media Spend is not considered as a KPI but it is needed to compute ROAS and CPA

6. ROAS (Return On Ad Spend): The profit earned on the advertising expenses.

$$ROAS = \frac{Revenue}{Media\ Spend} * 100\%$$

7. CPA (Cost Per Action): The average amount of advertising expenditure that is needed to generate an action.

$$CPA = \frac{Media\ Spend}{Transactions}$$

8. Order Value: The average value of a transaction.

$$Order\ Value = \frac{Revenue}{Transactions}$$

In this thesis, we predict four out of these eight KPIs because the rest can be derived. These four KPIs are the visits, transactions, revenue and media spend.

2.2.2 Channels

The KPIs can be filtered on different channel groups. Channel grouping are rule-based groupings of the traffic sources. It gives the ability to understand exactly how customers arrived at your website. Moreover, the performance of each of the traffic channels can be obtained. This is useful as it gives insight about which channel is performing best and which channel may require some more work. FasterIntel has defined the following 7 channels:

1. Direct: Going directly to the website
2. Organic: Going via Google search
3. Paid: Going via Google ads
4. Referral: Going via other websites/pages
5. Social: Going via social media
6. E-mail: Going via an e-mail such as newsletter
7. Other: None of these 6 above

In this thesis, we focus on the paid channel because during analysis we found out that this channel gathers the most traffic. Moreover, this channel is affected by advertising which makes it harder to predict than the other channels.

2.3 Current Model

The simple linear regression is the current model that is used in the software product of FasterIntel to forecast the KPIs. This linear regression model contains only one independent (explanatory) variable. It is used to model statistical relationship between the independent (explanatory) variable and the dependent (response) variable [9]. The explanatory variables are variables that have an effect on the response variable. The response variable expresses the observations of the data. The model assumes a linear relationship between the response variable and the explanatory variable and finds the best-fitting line (regression line) that describes this relationship. The model is expressed as

$$Y_i = a + bX_i + e_i, \tag{2.1}$$

where the regression parameter a is the intercept and the regression parameter b is the slope of the regression line. Y_i is the response variable at the i^{th} observation, X_i is the explanatory variable at the i^{th} observation and e_i is the residual at the i^{th} observation. The residual is the difference between the observed response variable Y_i and the predicted response variable $\hat{Y}_i$. The predicted response variable $\hat{Y}_i$ is given by

$$\hat{Y}_i = a + bX_i \tag{2.2}$$

The best-fitting line is found using the least-square method. This method estimates the intercept a and the slope b such that the sum of squares error (SSE) is minimized. The SSE is given by:

$$SSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{2.3}$$

Filling equation 2.2 into equation 2.3 and applying algebraically manipulation, we will obtain the least squares estimates of the intercept $\hat{a}$ and the slope $\hat{b}$:

$$\begin{aligned} \hat{b} &= \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2} \\ \hat{a} &= \bar{Y} - \hat{b}\bar{X} \end{aligned} \tag{2.4}$$

Here $\bar{X}$ is the mean of the explanatory variable and $\bar{Y}$ is the mean of the response variable.

The current model always uses up to a maximum of 12 weeks of daily historical data to fit the regression model. The fitted regression model is then used for multi-step ahead prediction. Daily predictions are made upon the end of the week (Sunday). Thus, the number of days ahead forecast depends on the current day and this maximum number will never exceed 7 days ahead. The regression model is refitted daily using new historical data. This process will continue for each day and is visualized in Figure 2.1.

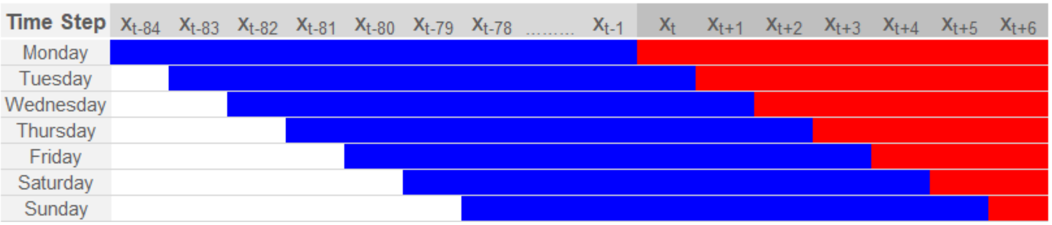


Figure 2.1: The training and forecasting of the current model over the days in each week. Here, the blue observations form the training sets and the red observations are the forecast time steps.

3 Literature Review

The daily prediction of the KPIs using their past observations is known as time series modeling. A time series is a sequential set of data points measured over successive times. The aim in time series modeling is to collect and analyze the past observations of a time series to develop an appropriate model that describes the structure of the time series. This model is then used to forecast future values for the time series. Thus, time series forecasting refers to the process of predicting future events based on past observations using mathematical models. A popular time series model is the Autoregressive Integrated Moving Average (ARIMA) [10]. It is popular because of its flexibility and simplicity to represent different varieties of time series. However, a major limitation is the assumption that the time series is linear and follows a particular known statistical distribution. This approximation does not hold in many complex real-world problems. Therefore, non-linear models are proposed such as artificial neural networks (ANNs) [11]. The major benefit of ANNs is their capability of flexible nonlinear modeling. It is not needed to specify a particular model form because the model is adaptively formed based on the features presented from the data.

These models have been applied in forecasting sales and demand. Cho [12] has applied exponential smoothing, ARIMA and neural networks to predict travel demand. The analysis has shown that neural networks perform best for forecasting visitor arrivals, especially in those series without obvious pattern. The models were trained on the tourist arrival statistics in Hong Kong from 1974 to 1998. Thiesing and Vornberger [13] have forecast the weekly demand on items in a supermarket using neural networks. The neural network has shown to outperform the naive method and the statistical method. The neural network was 17% more accurate than the statistical method and 32% more accurate than the naive method. The naive method is to use the last known value of the times series of sales as the forecast value for the next time step. The statistical method is the moving average of sales of past weeks. Data that was used are the historical sales, season and supermarket information such as the price and discount for every item, holiday information and opening hours of the store. Alon et al. [8] compared artificial neural networks and traditional methods for forecasting aggregate retail sales. The results have shown that ANN performed the best followed by ARIMA and

Winters exponential smoothing. Moreover, ANN was able to capture the dynamic nonlinear trend and seasonal patterns and their interactions.

Features that are relevant in forecasting sales are information about the products that are sold. However, these features cannot be used in this research because they are not generic enough. Reijden and Koppius [14] have shown that measures of online product buzz variables increase the accuracy of sales forecasting with 28%. These online product buzz refers to online expression of interest in a product, such as online product review, blog post and search trends. Moreover, they found that a random forest technique is very suitable to incorporate these online product buzz variables. Ferreira et al. [15] found that the features with largest variable importance are the discount, price, popularity and brand of the product. They used machine learning techniques to forecast demand of products for an online retailer and found that regression trees with bagging outperformed other regression model. Meulstee and Pechenizkiy [16] improved the food sales prediction for each product by constructing new groups of predictive features from publicly available data about the weather and holidays and data from related products. Moreover, Murray et al. [17] provided empirical evidence to explain how the weather affects consumer spending. They found that when exposure to sunlight increases, negative affect decreases and consumer spending tends to increase.

4 Data

This section describes the data and its pre-processing. Moreover, data analysis will be performed.

4.1 Google Analytics Data

Google Analytics (GA) is a web analytics service offered by Google that tracks and reports website traffic [18]. It sets a first party cookie as tracking code on each visitor's device. Thus, users are just cookies and not individuals. For this research, we used GA data from one of the clients of FasterIntel. Every report in GA is made up of dimensions and metrics. Dimensions are attributes of your data and metrics are quantitative measurements. In our dataset the dimensions are the date, source and medium which are combined with different metrics. The dataset along with its description is shown in Table 4.1.

Data	**Description**
Dimensions	
Date	The date of the active date range
Source	The place users were before seeing your content
Medium	Description of how users arrived at your content
Metrics	
Time	
Year	The year of the date
Month	The month of the date
Week	The week of the date
Weekday	The weekday of the date
Season	The season of the date according to the four-season calendar reckoning
Holiday	Variable indicating whether the date is a holiday in the Netherlands
KPIs	
Sessions	The total number of sessions (visits)
Transactions	The total number of completed purchases on the website
Revenue	The total revenue from web e-commerce or in-app transactions
Ad Cost	The total amount paid for advertising (media spend)
Advertising	
Impression	The number of times your ad is shown on a search result page or other site on the Google network
Ad Clicks	The number of times someone clicks on your ad
Active campaigns	The number of active campaigns
Sessions	
Bounce Rate	The percentage of all sessions on your site in which users viewed only a single page and triggered only a single request to the Analytics server
Percentage New Sessions	The percentage of sessions by users who had never visited the property before
Average Session Duration	The average duration (in seconds) of users' sessions
Users	
Users	The total number of users for the requested time period
New Users	The number of sessions marked as a user's first sessions
Site Speed	
Average Page Load Time	The average time (in seconds) pages from the sample set take to load, from initiation of the pageview to load completion in the browser
Average Redirection Time	The average time (in seconds) spent in redirects before fetching this page

Table 4.1: Google Analytics data and its description per day

The dataset consists of metrics for each day from 1 July 2014 to 31 May 2017 for each corresponding source and medium. The metrics can be divided into categories. The time category consists of data that are derived from the date. The rest of the categories consists of data that are self explanatory.

4.2 External Data

Besides the GA data, external data is used. This external data consists of the weather, inflation and the unemployment in the Netherlands. The weather data is obtained from Koninklijk Nederlands Meteorologisch Instituut (KNMI) [19] and the others are obtained from Centraal Bureau voor de Statistiek (CBS) [20]. The external data along with its description can be found in Table 4.2. This external data could have an effect on the KPIs. For instance, weather has an effect on consumer spending as shown in the literature review.

Data	**Description**
Weather	
Average Temperature	The average daily temperature (in 0.1 degree Celsius)
Minimum Temperature	The daily minimum temperature (in 0.1 degree Celsius)
Maximum Temperature	The daily maximum temperature (in 0.1 degree Celsius)
Rainfall Amount	The daily amount of rainfall (in 0.1 mm)
Rainfall Duration	The daily rainfall duration (in 0.1 hour)
Sunshine Duration	The daily sunshine duration (in 0.1 hours) calculated from the global radiation
Inflation	
Inflation Rate	The monthly inflation rate (consumer price index) in percentage
Unemployment	
Unemployment	The percentage of unemployed (monthly)

Table 4.2: External data and their description

4.3 Data pre-processing

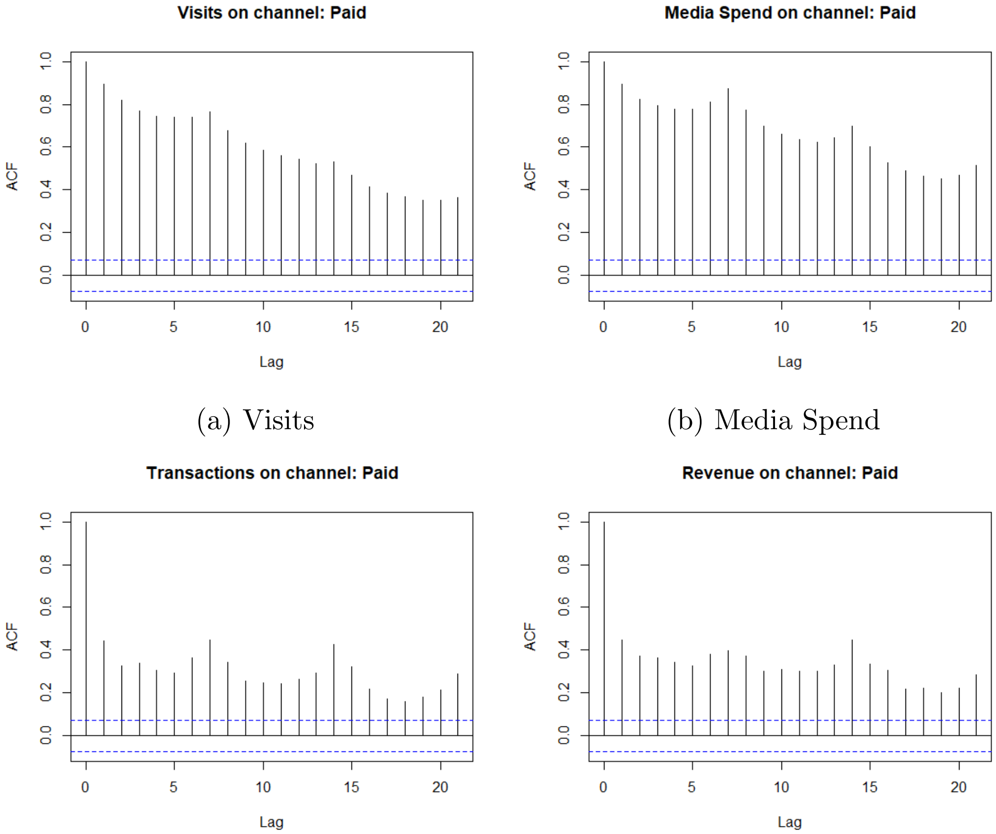
4.4 Data Analysis

In order to obtain insights from the data, several analyses have been performed.

From these analyses an overview of the time effect on the KPIs is obtained.

The next analysis is the correlation between the KPI values and their own lags. This is analyzed using the autocorrelation function (ACF) and partial autocorrelation function (PACF). Both functions measure the correlation between the current observation and an observation k time step ago, but the partial autocorrelation also takes into account the observations at intermediate lags (i.e. at lags $< k$) [21]. They can be used to determine the magnitude of the past values related to future values [10].

The ACF and PACF plots of the KPIs on the paid channel are shown in Figures 4.1 and 4.2.



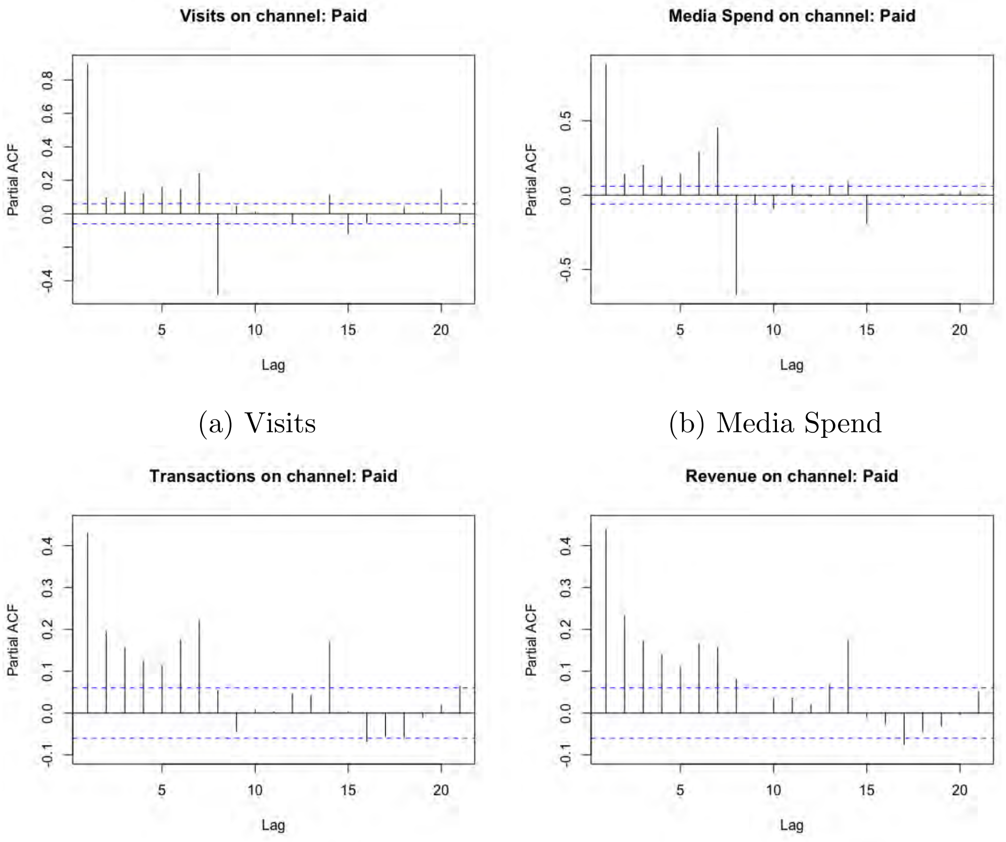
(a) Visits

(b) Media Spend

(c) Transactions

(d) Revenue

Figure 4.1: ACF plot of the KPIs on the paid channel. The blue dashed lines indicate the point of statistical significance.



(a) Visits

(b) Media Spend

(c) Transactions

(d) Revenue

Figure 4.2: PACF plot of the KPIs on the paid channel. The blue dashed lines indicate the point of statistical significance.

We observe that the ACF plot shows a slight peak at every 7^{th} lag for all KPIs. This indicates a weekly pattern. The PACF plot shows a significant spike at lag 1 for all KPIs. This indicates that the KPI value of previous day is very strong correlated with the value of current day. For visits and media spend, the lags 1 to 7 is significantly positive correlated. Moreover, lag 8 is negative correlated and lags beyond 8 become less significant. For transactions and revenue, we observe significance until lag 7 followed by a spike at lag 14 that is caused by the weekly pattern.

The final analysis is the exploration of the statistical relationship between the KPIs and the other metrics in the data. The Pearson correlation between the different variables in the data is computed. Since we use past observations to predict the KPIs, the correlation is between the current day KPIs value and the metrics value of previous day except for weather, inflation and unemployment. The values for weather are at the same day because weather forecast can be derived from other resources. This forecast can then be used as predictor for the KPIs. The value for inflation and unemployment is from the previous month because daily observations are not available. Even though this is the case it can still help in daily forecast as inflation and unemployment usually have a long term effect. The correlation matrix for the paid channel is shown in a correlogram in Figure 4.3.

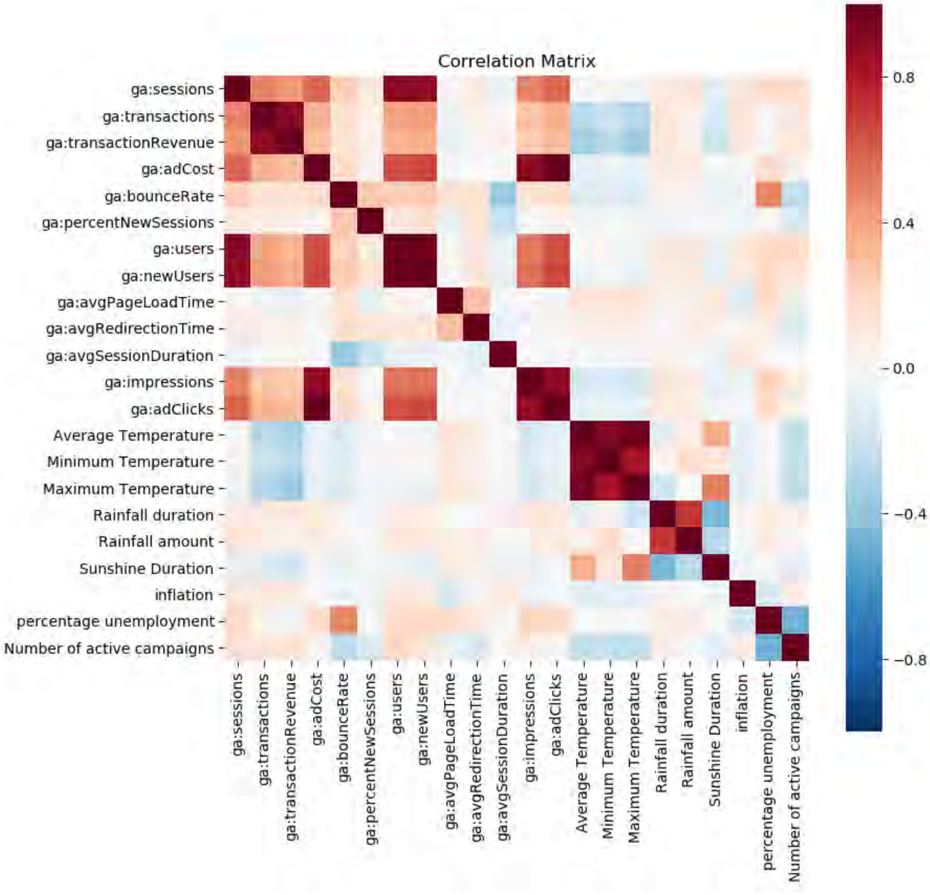


Figure 4.3: Pearson correlation matrix of the data on the paid channel

We observe correlations between KPIs themselves. Visits (ga:sessions) is positive correlated with media spend (ga:adCost) which indicates that an increase in media spend of the previous day results in an increase in next day visits. Transactions is strong positive correlated with revenue (ga:transactionRevenue) which is obvious. The current day visits is strong correlated with previous day users and new users. This can be explained by the fact that every single user generates at least one session. Moreover, a positive correlation is observed between visits and impressions and adClicks which shows the positive relation between advertising and visits.

For media spend, we observe a positive correlation with impressions, adclicks and number of active campaigns which is natural.

With the obtained insights from the data analysis, we have a good interpretation about which data can be used as features in the models. This will be further described in the experiments in section 6.2.

5 Methods and Models

This section describes the models that are used for the forecast, the multistep forecasting method and the evaluation method.

5.1 Models

5.1.1 Ratio-based model

The Ratio-based model is used as a benchmark model. This model takes the historical data of a certain number of weeks before and computes the change in ratios over the same day for every week. Next, these ratios are averaged into a single ratio and this single ratio is used as predictor for the next time step.

If we define X_t as the KPI at time step t and we take k weeks as historical data, then the ratio change for X_t is given by:

$$Ratio_t = \frac{1}{k}\sum_{i=1}^{k} \frac{X_{t+1-7i} - X_{t-7i}}{X_{t-7i}} \tag{5.1}$$

However, problem occurs when the historical KPIs are zero, i.e. $X_{t-7i} = 0$ for $i = 1, .., k$. This will result in a division of zero which is not possible. Therefore, in this case, we take the difference of the KPIs instead of the ratio.

$$Difference_t = \frac{1}{k}\sum_{i=1}^{k} X_{t+1-7i} - X_{t-7i} \tag{5.2}$$

Moreover, the difference forecast will be applied when X_t is zero as the ratio forecast will always result in a zero. Thus, the forecast on X_{t+1} is given by:

$$X_{t+1} = \begin{cases} \mathrm{X}_t * (1 + Ratio_t) & \text{if } \mathrm{X}_t > 0 \text{ or } Ratio_t \text{ is defined} \\ \mathrm{X}_t + Difference_t & \text{if } \mathrm{X}_t = 0 \text{ or } Ratio_t \text{ is undefined} \end{cases} \tag{5.3}$$

This equation expresses the single-step forecast. The mathematical expression for the recursive multi-step forecast is then:

$$X_{t+n} = \begin{cases} \prod_{i=0}^{n-1}(1 + \mathit{Ratio}_{t+i}) * X_t & \text{if } \mathrm{X}_t > 0 \text{ or } \mathit{Ratio}_{t+i} \text{ is defined} \\ \sum_{i=0}^{n-1} \mathit{Difference}_{t+i} + X_t & \text{if } \mathrm{X}_t = 0 \text{ or } \mathit{Ratio}_{t+i} \text{ is undefined} \end{cases} \tag{5.4}$$

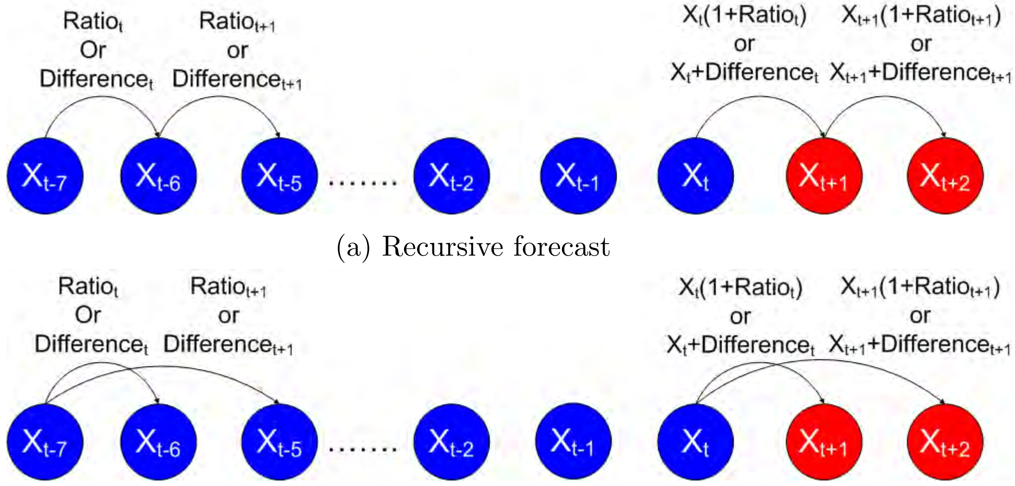
Notice that the ratio and the difference are always taken on the next time step i.e. the ratio and difference between X_t and X_{t+1}. However, we can also take the ratio and difference between multiple time steps i.e. between X_t and X_{t+n}. This will avoid the recursive strategy that usually has accumulated error. The ratio and the difference are then expressed as:

$$\begin{aligned} \mathit{Ratio}_{t+n} &= \frac{1}{k}\sum_{i=1}^{k}\frac{X_{t+n-7i} - X_{t-7i}}{X_{t-7i}} \\ \mathit{Difference}_{t+n} &= \frac{1}{k}\sum_{i=1}^{k} X_{t+n-7i} - X_{t-7i} \end{aligned} \tag{5.5}$$

Next, the non-recursive multi-step ahead forecast is given by:

$$X_{t+n} = \begin{cases} \mathrm{X}_t * (1 + \mathit{Ratio}_{t+n}) & \text{if } \mathrm{X}_t > 0 \text{ or } \mathit{Ratio}_{t+n} \text{ is defined} \\ \mathrm{X}_t + \mathit{Difference}_{t+n} & \text{if } \mathrm{X}_t = 0 \text{ or } \mathit{Ratio}_{t+n} \text{ is undefined} \end{cases} \tag{5.6}$$

The Ratio-based model for both the recursive and non-recursive multi-step ahead is illustrated in Figure 5.1.


Ratio$_t$ Or Difference$_t$
Ratio$_{t+1}$ or Difference$_{t+1}$
X$_t$(1+Ratio$_t$) or X$_t$+Difference$_t$
X$_{t+1}$(1+Ratio$_{t+1}$) or X$_{t+1}$+Difference$_{t+1}$
X$_{t-7}$
X$_{t-6}$
X$_{t-5}$
X$_{t-2}$
X$_{t-1}$
X$_t$
X$_{t+1}$
X$_{t+2}$

(a) Recursive forecast

(b) Non-recursive forecast

Figure 5.1: The Ratio-based model: recursive and non-recursive forecast using one week historical data. The blue observations are the historical observations and the red observations are the forecast time steps.

The advantage of this model is its simplicity and flexibility. The model can be easily implemented for all the KPIs and it is able to produce a forecast even when little data is available. The disadvantage is the reliability of the model as large variation in KPIs over the weeks can easily result in inaccurate forecast.

5.1.2 ARIMA

The ARIMA model is a stochastic time series model that combines three processes: the Autoregressive process (AR), the strip off of the integrated (I) time series by differencing and the Moving Averages (MA) [10]. In an AR process the future value of a variable is assumed to be a linear combination of p past observations plus a random error and a constant. In a MA process, the past observations are the q past errors instead of the past observations. Thus, it is a linear combination of q past errors plus a random error and the mean of the series. These two processes can be combined together to form a more effective time series model. This model is known as ARMA(p,q) [22] and is given by:

$$X_t = \alpha_1 X_{t-1} + ... + \alpha_p X_{t-p} + \epsilon_t + \beta_1 \epsilon_{t-1} + ... + \beta_q \epsilon_{t-q}, \tag{5.7}$$

where X_t is the observation at time t and ϵ_t is the random error at time t. This equation can be written implicitly as:

$$\Big(1 - \sum_{k=1}^{p} \alpha_k L^k\Big) X_t = \Big(1 + \sum_{k=1}^{q} \beta_k L^k\Big) \epsilon_t, \tag{5.8}$$

where p and q are the order of the autoregressive and moving average parts of the model respectively and L is the lag operator.

However, ARMA can only be used for stationary time series i.e. the mean and variance of the time series do not depend upon time. This is not the case in many time series such as in economic time series that contain trends and seasonal patterns. Therefore, the integrated term is introduced to include the cases of non-stationarity by applying differencing. The difference operator takes a differencing of order d and it is expressed as follows:

$$\Delta^d X_t = (1 - L)^d X_t, \tag{5.9}$$

where X_t is the data point at time t, L is the lag operator and d is the order. Thus, the three processes, AR(p), I(d) and MA(q) are combined, interacted and recomposed into the ARIMA(p, d, q) model.

The general method to identify the order p in an AR process and q in a MA process is to use the autocorrelation function (ACF) and the partial autocorrelation function (PACF) [21]. For a pure AR(p) process, the PACF cuts down to zero after lag p. For a pure MA(q) process, the ACF cuts down to zero after lag q. For ARMA(p, q) and ARIMA(p, d, q) processes, it is more complicated to estimate the parameters. A method to do this is to perform a grid search over these parameters using an evaluation metric for model selection.

5.1.3 Random Forest

Random Forest is a machine learning model for classification and regression [23] and it is illustrated in Figure 5.2.

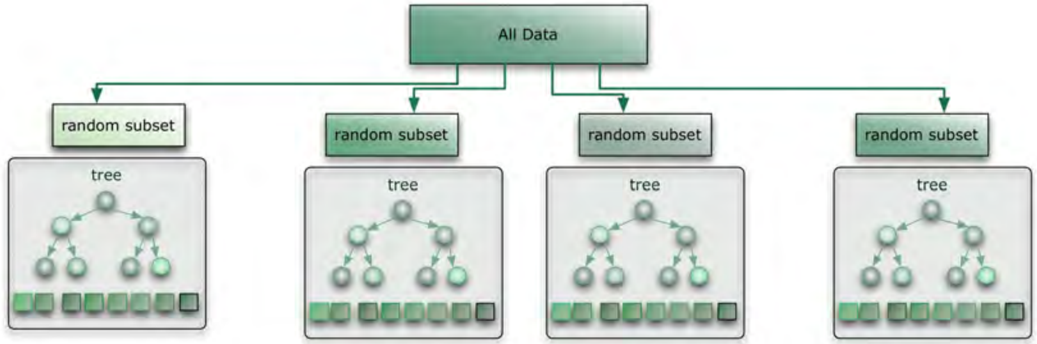


Figure 5.2: Random Forest [24]

Random Forest is an ensemble learning method that combines a group of weak learners to form a more powerful model. These weak learners are decision trees and they can either be used for classification or regression. In our case, the focus is on regression.

A decision tree is made up of decisions nodes, terminal nodes (leaves) and branches. These branches represent the possible paths from one node to another. Decision trees are recursive partitioning algorithm that split the dataset in smaller subsets using a top-down greedy approach. It starts at the root node (top decision node) where a certain feature variable is used to split the data into subsets. Because the dataset can be split in numerous ways, the feature variable and its split point that yields the largest reduction in the residuals sum of squares (RSS) is chosen.

$$\sum_{n=1}^{N} \sum_{i \in R_n} (y_i - \hat{y}_n)^2, \tag{5.10}$$

where y_i is the training observation i and $\hat{y}_n$ is the mean of the response values for the training observation that fall into the subset n.

Next, the decision tree recursively splits the dataset from the root node onwards at each decision node until the terminal nodes. These terminal nodes are the endpoints of the decision tree and represent the final result of a combination of decisions and events. They are reached when all records after the split belong to the same target variable or until a stopping criterion is met.

In Random Forest a large number of these decision trees are constructed and each tree makes a random selection of independent variables. This results in variance in predictions between various decision trees which reduce overfitting and variance. Then the prediction is produced by taking the mean prediction of the individual trees.

Besides the number of trees in the forest, it has other parameters that can be varied such as the maximum depth of the tree, the minimum number of samples required to split a decision node, the minimum number of samples required to be at a leaf node and the number of features to consider when looking for the best split.

5.1.4 Artificial Neural Networks

ANNs are inspired by biological systems and the human brain and they have shown powerful pattern classification and recognition capabilities [11]. ANNs learn from experience by recognizing patterns in the input data. Next, they make predictions based on their prior knowledge. ANNs are applied in many different fields of industry, business and science [25]. There are different types of ANNs, we have applied the feed-forward neural network (FNN) and the recurrent neural network (RNN) [26, 27]. The FNN is the Multi-Layer Perceptron and the RNN is the Long Short-Term Memory.

5.1.4.1 Multi-Layer Perceptron

The Multi-Layer Perceptron (MLP) is a class of FNN. Figure 5.3 illustrates the architecture of a MLP.

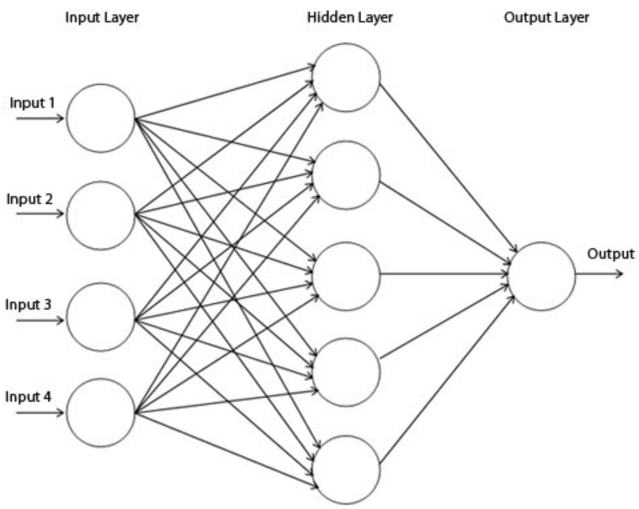


Figure 5.3: Architecture of a MLP with 3 layers (input, hidden and output) with 4 input neurons, 5 hidden neurons and 1 output neuron [28]. Notice that MLP can have multiple hidden layers.

The network consists of processing elements, also known as neurons that are organized in layers. Every neuron in a layer is connected with all the neurons in the previous layers by acyclic links, each with an associated weight. These weights encode the knowledge of a network. The neurons in the layers are used for transformation from input to output. The output of each neuron y in the hidden layer(s) and output layer is a (non) linear function over the dot product of the weights of the connections with the outputs of the neurons in the previous layer.

$$y = \phi\left(\sum_i \omega_i x_i + b\right) \tag{5.11}$$

Here x_i are the inputs of the neuron, ω_i are the weights of the neuron and b is the bias. ϕ is the activation function that determines the activation level of the neuron.

During training of the network, the weights are modified such that the difference between the network output and the actual output are minimized.

Thus, the estimation of the weights are based on the minimization of the cost function. In our model, this cost function is the mean squared error. In order for the network to know how to adjust the weights, it uses learning algorithms. The traditional learning algorithm is backpropagation [29] and it works as follows:

1. Compute the feed-forward signals from the input to the output.
2. Compute the error E of each neuron of the output layer using the cost function.
3. Backpropagate the error by the gradients of the associated activation functions and weighting it by the weights in previous layers.
4. Compute the gradients for the set of parameters based on the feed-forward signals and the backpropagated error.
5. Update the parameters using the computed gradients.

$$\Delta\omega_{ij} = \alpha\delta_j y_i, \text{ where} \tag{5.12}$$

$$\delta_j = \begin{cases} \phi'(v_j)(t_j - y_j) & \text{if } j \text{ is an output node} \\ \phi'(v_j)\sum_{k \text{ of next layer}} \delta_k\omega_{jk} & \text{if } j \text{ is an hidden node} \end{cases}$$

Here ω_{ij} is the weight between node i and node j, α is the learning rate, y_j is the output of node j, v_j is the activation of node j and t_j is the target output of node j.

Even though backpropagation is used to solve many practical problems, it shows problems such as slow convergence and getting stuck at local minima [29]. Therefore, other learning algorithms have been developed to improve the performance of the backpropagation algorithm. One of these learning algorithms is the Adaptive Moment Estimation (Adam) [30]. Adam is a stochastic gradient descent algorithm based on estimation of first and second order moments of the gradients. The algorithm works as follows:

1. Compute the gradient using the current parameters
2. Update biased first order and second order moments estimate.

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1)g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2)g_t^2 \end{aligned} \tag{5.13}$$

Here m_t is first order moment at time step t, v_t is the second order moment at time step t, g_t is the gradient at time step t and g_t^2 indicates the element-wise square of g_t. β_1 and β_2 are the exponential decay rates between 0 and 1.

3. Compute the bias-corrected first and second order moments.

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t} \end{aligned} \tag{5.14}$$

4. Compute weight update and update parameters.

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{(\sqrt{\hat{v}_t} + \epsilon)} \tag{5.15}$$

Here θ_t is the parameter vector at time step t and α is the learning rate.

Adam computes adaptive learning rates for each parameter [31] that speeds up the learning phase. This leads to faster convergence. Furthermore, little manual tuning of hyperparameters are required because they have intuitive interpretation. All these advantages are beneficial for all the models that need to be constructed. Therefore, Adam is used as learning algorithm.

5.1.4.2 Long Short-Term Memory

In FNNs information is only propagated in one direction. This is in contrast to recurrent neural networks (RNNs) where information is propagated in bi-directions using feedback loops. These feedback loops transfer internal and output signals back to the input level, which allows information to persist. Thus, a RNN has an internal memory that captures information about what has been calculated so far. This becomes clear if we unfold the RNN through time as shown in Figure 5.4.

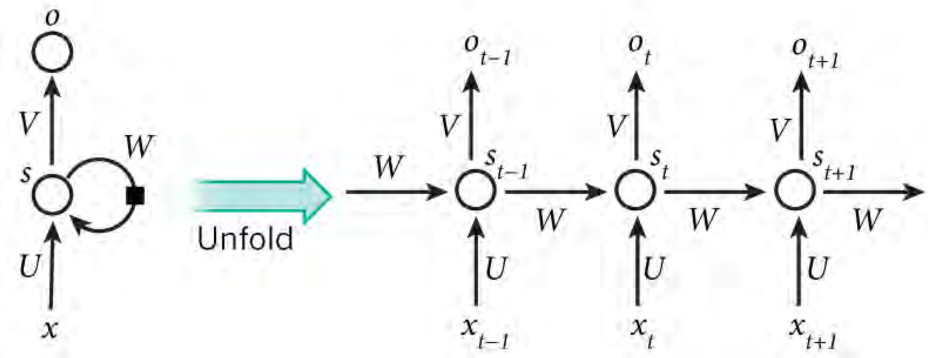


Figure 5.4: A RNN unfolded in time [32]. Here U, V, W represent the set of weights. x_t is the input at time step t, s_t is the hidden state at time step t and o_t is the output at time step t.

Therefore, RNNs can model temporal dependencies of unspecified duration between the inputs and the associated desired outputs [33]. Because of this RNNs are suitable for time series predictions. However, when the gap between the relevant information and the prediction becomes larger, regular RNNs become unable to learn to connect the information because of the vanishing gradients (i.e. the gradient gets so small that learning either becomes very slow or stops). The problem of vanishing gradients can be avoided using Long Short-Term Memory (LSTM).

LSTM is a specific RNN that is capable of learning long-term dependencies and it was introduced by Hochreiter and Schmidhuber [34]. The network takes three inputs. The input X_t of the current time step t, the output from the previous LSTM unit h_{t-1} and the memory of the previous unit C_{t-1}.

The LSTM unit contains special units called memory blocks in the recurrent hidden layer. An individual memory block is shown in Figure 5.5. Each memory block contains memory cells and three gated cells that is shared by all the memory cells. These three gates are the forget, input and output gate and control the information flow to the cells. The forget gate decides what information will be removed in the memory. The input gate decides what information will be updated. The output gate defines what information inside the cells will be exposed to the external network. These gates are controlled by a simple one layer neural network using the sigmoid activation function. Thus, the network outputs a value between 0 and 1, where 0 represents closed gates and 1 represents open gates.

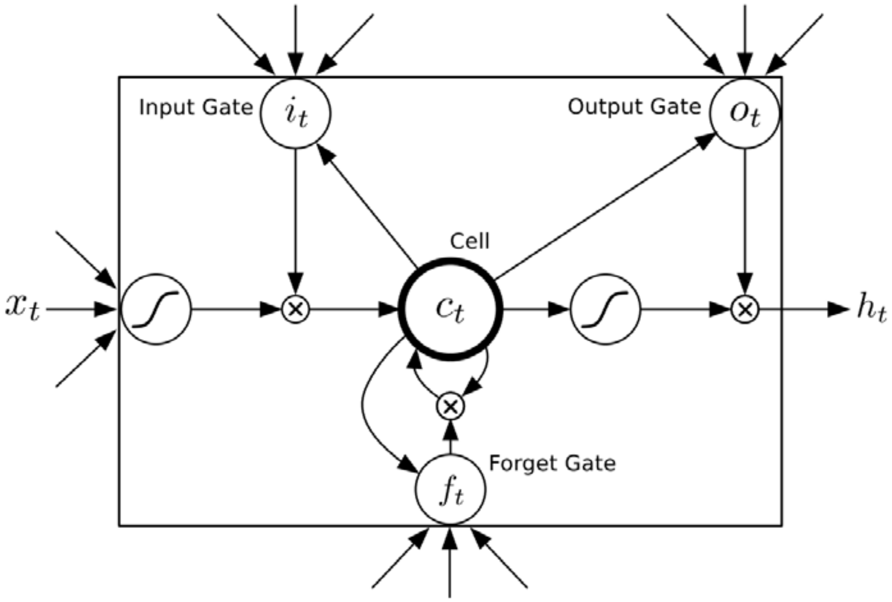


Figure 5.5: A single LSTM memory block [35]

The forget and input gate use the three LSTM network inputs X_t, h_{t-1} and C_{t-1} and a bias vector as input. The output gate uses the h_{t-1}, X_t, the new memory C_t and a bias vector as input. Thus, the forget gate f_t, input gate i_t and output gate o_t of the current time step t is given by the following equations:

$$\begin{aligned} f_t &= \sigma(b_f + W_{xf}X_t + W_{hf}h_{t-1} + W_{cf}C_{t-1}) \\ i_t &= \sigma(b_i + W_{xi}X_t + W_{hi}h_{t-1} + W_{ci}C_{t-1}) \\ o_t &= \sigma(b_o + W_{xo}X_t + W_{ho}h_{t-1} + W_{co}C_t) \end{aligned} \tag{5.16}$$

Here, σ is the sigmoid function, b is the bias vector and W represents the set of weights.

The new memory C_t is also generated by a simple neural network, but it uses the hyperbolic tangent as activation function instead of the sigmoid function. The new memory C_t is then generated by element-wise multiplication of the input gate i_t and by adding the old memory C_{t-1}.

$$C_t = f_t C_{t-1} + i_t \tanh(b_c + W_{xc} X_t + W_{hc} h_{t-1}) \tag{5.17}$$

Here, tanh is the hyperbolic tangent function, b is the bias vector and W represents the set of weights.

The output h_t for the LSTM unit is then generated using the new memory C_t that is controlled by the output gate o_t.

$$h_t = o_t \tanh(C_t) \tag{5.18}$$

Again the weights in the network need to be learned. The learning algorithm Adam is used for this purpose because it can handle the complex training dynamics of RNNs better than plain gradient descent [30]. Moreover, since the network uses the hyperbolic tangent function as activation function, which is bounded between -1 and 1, the data is normalized between -1 and 1.

5.2 Multi-step Forecasting Strategies

Since FasterIntel always shows predictions of the KPIs up to Sunday (end of the week) for each week, it means that it requires a maximum of 7 days ahead forecast. Therefore, we decided to forecast the KPIs always 7 time steps ahead on each day. Predicting multiple time steps into the future is also known as multi-step time series forecasting. Different strategies are used for multi-step time series forecasting [36]. The most common strategies are the direct multi-step forecast and the recursive multi-step forecast. The direct multi-step forecast uses a separate model for each forecast time step while the recursive multi-step forecast uses a one-step model multiple times. It takes the prediction for the prior time step as an input for making a prediction on the following time step. A disadvantage of the recursive strategy is the accumulation of the prediction errors as the forecast horizon increases while this does not hold in the direct strategy. However, this does not mean that the direct strategy is always better than the recursive strategy. It all depends on the model specification. Another disadvantage of the recursive strategy are the unknown values for other time series when they are used as predictors. This means that each time series need to be predicted. The disadvantage of the direct strategy is that it involves a heavier computational load than the recursive strategy. Moreover, since the direct strategy forecasts each time step in isolation of the other time steps, it can produce completely unrelated forecasts over the whole forecast horizon. This can lead to unrealistic discontinuities whereas time series have some aspect of continuous behavior in reality [37]. Therefore, the recursive strategy is applied in this research.

5.3 Evaluation

The performance of the models is evaluated using the time series cross validation. This process is illustrated in Figure 5.6.

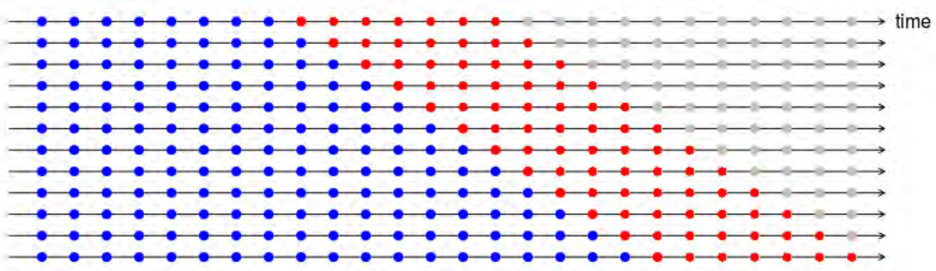


Figure 5.6: The series of training and test sets in time series cross validation. The blue observations form the training sets and the red observations form the test sets [38].

This validation has a series of training and test sets. Each of the test sets consists of n observations that corresponds with the n time step ahead forecast. The corresponding training set consists of features that occurred prior to the first observation of the test set. This mimics the real-world scenario where new observations become available for each time step t. Next, this validation method walks each series of the test sets one at a time and makes n predictions for each time step.

Next, the real observations and the predictions for each time step are compared with each other using an error measurement. The error measurement that is used is the Root Mean Squared Error (RMSE) and it is given by:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2} \tag{5.19}$$

Here $\hat{y}_i$ is the predicted value for observation i, y_i is the actual value for observation i and m is the total number of observations.

The RMSE penalizes extreme errors as it is much affected by large individual errors. Thus, we obtain n RMSE values where each of these values represents the error for a single time step $t, t+1, ..., t+n$.

6 Experimental Setup

This section describes the training and testing of the models, the two experiments that are conducted and the optimization of the model hyperparameters.

6.1 Training and Testing

Since the KPIs have seasonality within a year as shown in the analysis in section 4.4, it is important that the models are evaluated on at least one year test data to capture their overall performance. Therefore, we chose the last 365 days of the dataset as test set starting from 25 May 2016. Notice that it does not start from 31 May 2016 because observations after 31 May 2017 are not available in the dataset when performing 7 days ahead forecast.

The models are retrained after each 30 series. This corresponds with retraining the model every month. The decision to not retrain the models every day is because retraining on one new observation in the training set will hardly make any difference in terms of performance.

6.2 Experiments

In this research two experiments have been conducted. These experiments are categorized into experiment A and experiment B.

6.2.1 Experiment A

In experiment A, the data used for each KPI forecast are the corresponding historical KPI data along with time related data. The time related data are the month, week, weekday, season and holiday. The choice of performing this experiment is because the current forecast model only uses the corresponding historical KPI data. Thus, it is interesting for FasterIntel to know about the predictive performance of each model using only this data. The time related data is used to capture temporal information from the time series for Random Forest and MLP because they are non-temporal models. This is in contrast with ARIMA and LSTM which are temporal models that should be able to catch temporal information automatically from the time series. Thus, the input data for ARIMA and LSTM are only the lagged values of the KPI under investigation whereas Random Forest and MLP added additional time related data and the weekly moving average of the KPI as input to spot the weekly trend.

The total number of lags to consider in each model are determined in different ways depending on the model. In ARIMA, the number of lags is the order p in the AR part and q in the MA part. This order is determined using a grid search method on the validation set that will be described in section 6.3. In LSTM, a lag of one is used because information of the lags are passed through each recurrent state to the next state. In Random Forest and MLP, the number of lags are based on the PACF analysis in section 4.4. For visits and media spend, the first 8 lags are used and for transactions and revenue the first 7 lags and the 14^{th} lag are used.

Random Forest and ANNs are run for multiple times and the average performance was taken as final performance for the models. This is because these models are stochastic and therefore they return different results on different runs of the same algorithm on the same data.

6.2.2 Experiment B

In experiment B, all the data are used in the KPIs forecast. With this experiment, we are able to show the difference in performance compared to experiment A when more data is added. This could be interesting for FasterIntel as they can set up new businesses with their clients e.g. offering better KPIs predictive performance for only premium clients. However, we only forecast one step ahead instead of multi-step ahead because it becomes computationally too intensive when additional time series features are added. The reason is that the recursive strategy requires predictions for every additional time series feature and developing models to do so was out of scope in this thesis. Notice that ARIMA is excluded in this experiment as it is a univariate time series model.

In order to ensure that the difference in performance is caused by adding additional features, the same features in experiment A plus the additional features are used. These additional features are chosen based on the analysis of the Pearson correlation matrix as shown in Figure 4.3. The additional features for the KPIs are shown in Table 6.1.

KPI	**Additional Features**
Visits	"ad cost", "users", "new users", "impression", "inflation", "unemployment"
Transactions	"ad cost", "users", "new users", "impression", "average temperature", "maximum temperature", "inflation", "unemployment"
Revenue	"ad cost", "users", "new users", "impression", "average temperature", "maximum temperature", "inflation", "unemployment"
Media Spend	"impression", "campaigns", "ad clicks"

Table 6.1: KPI and its additional features

Again the number of lags for each feature had to be determined in the input data as in experiment A. Zero lag is chosen for the temperature data as your current behaviour can be influenced by the current weather condition. A lag of one is chosen for the other additional features because it is the most representative for the current time step. Moreover, the moving average of 7 days for all Google Analytics features and the weather is included which give information about the weekly trend of each time series feature.

6.3 Hyperparameters Optimization

ARIMA, Random Forest, MLP and LSTM all have hyperparameters that need to be optimized to achieve good performance. The hyperparameters in ARIMA are the auto regressive order p, the integration order d and the moving average order q. For Random Forest, MLP and LSTM not all hyperparameters are considered. The ones that are considered in Random Forest are the number of trees in the forest, the number of features to consider when looking for the best split and the minimum sample leaf size. The hyperparameters that are considered in MLP are the number of hidden layers and neurons in each layer and the activation functions. The hyperparameters that are considered in LSTM are the number of hidden layers and LSTM units in each layer.

All the combinations of the hyperparameters were tested on the validation set. This validation set is a separation of the first training set in the time series cross validation and contains one month of data starting from 25 April 2016 to 24 May 2016. Thus, the models were trained on the data before 25 April 2016. Next, each model was trained again on the whole first training set using the optimal hyperparameters and evaluated using time series cross validation as described in section 5.3.

The hyperparameter optimization for ARIMA was performed using a grid search method. This method tries every hyperparameter setting over a specified range of values. The search range for the autoregressive order p and moving average order q was set based on the PACF and ACF analysis in section 4.4, respectively. For the autoregressive order p it was set from 0 to 8 for visits and media spend and from 0 to 14 for transactions and revenue. For the moving average order q it was set from 0 to 7 for all the KPIs. The search range for the integration order d was set from 0 to 1. The choice for a maximum of first order differencing is because second order differencing is rarely needed to achieve stationarity [10].

In Random Forest, the optimal number of trees was determined to be in the range from 10 to 400 and the optimal minimum sample leaf size was determined to be in the range from 1 to 50.

In MLP, the optimal hidden neurons in each hidden layer was determined in the range 10 to 100. The number of hidden layers was set to a maximum of two layers. The rectified linear unit (ReLU) activation function was used in the hidden layers because it was found to be the most efficient when applied to the forecasting of non-stationary and noisy time series. Next, the linear activation function was used in the output layer to output the forecast value of the time series.

In LSTM, the number of hidden layers was set to one and the number of LSTM units was determined in the range 10 to 40. The exponential decay rates β_1 and β_2 were set to default values of 0.9 and 0.999, respectively. These default values are proposed in the original paper [34].

7 Results

In this section the results of the two experiments are presented. First, section 7.1 presents the results for experiment A and it contains results with regards to model hyperparameters, model performance, model comparison and feature importance. Next, section 7.2 presents the results for experiment B and it contains results with regards to model hyperparameters, model performance and feature importance. Finally, section 7.3 summarizes the results of both experiments.

7.1 Experiment A

7.1.1 Hyperparameters

The results for the hyperparameters for ARIMA, Random Forest and MLP are shown below.

ARIMA

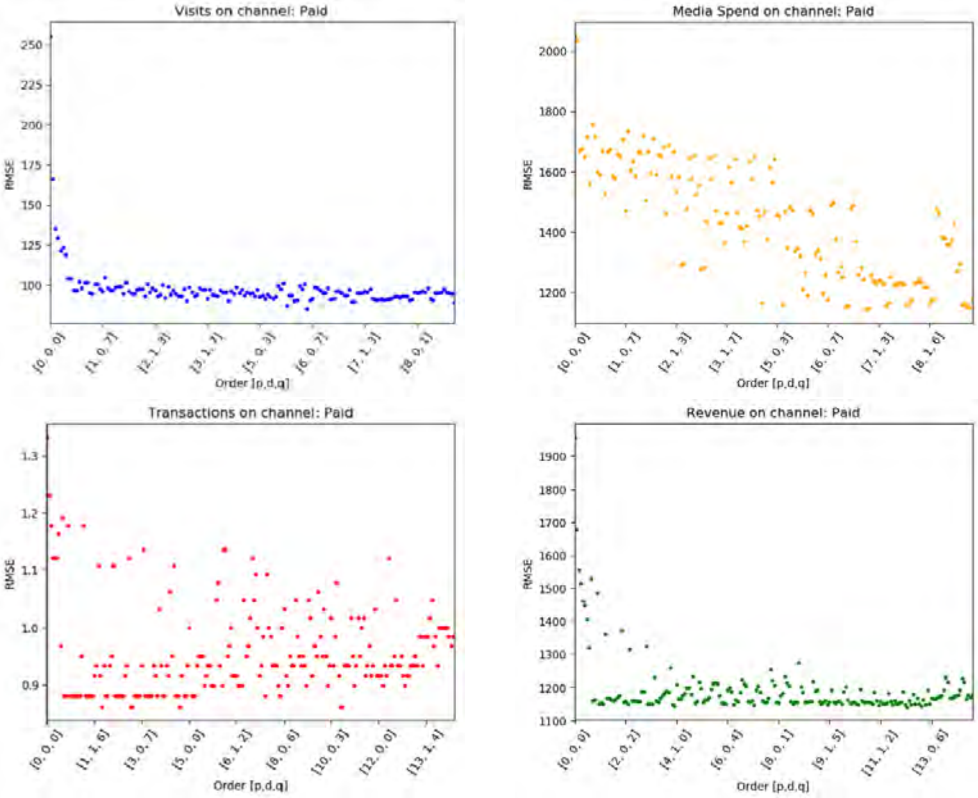


Figure 7.1: ARIMA Order(p, d, q)[2]

Figure 7.1 shows the RMSE for different combinations of order(p, d, q). We observe many fluctuations in errors for the different combinations. The optimal order achieved for visits, transactions, revenue and media spend is (6,0,5), (2,0,2), (4,0,2) and (7,0,6), respectively.

[2] The order(p, d, q) starts at (0,0,0) and is first incremented for q, next for d and as last for p

Random Forest

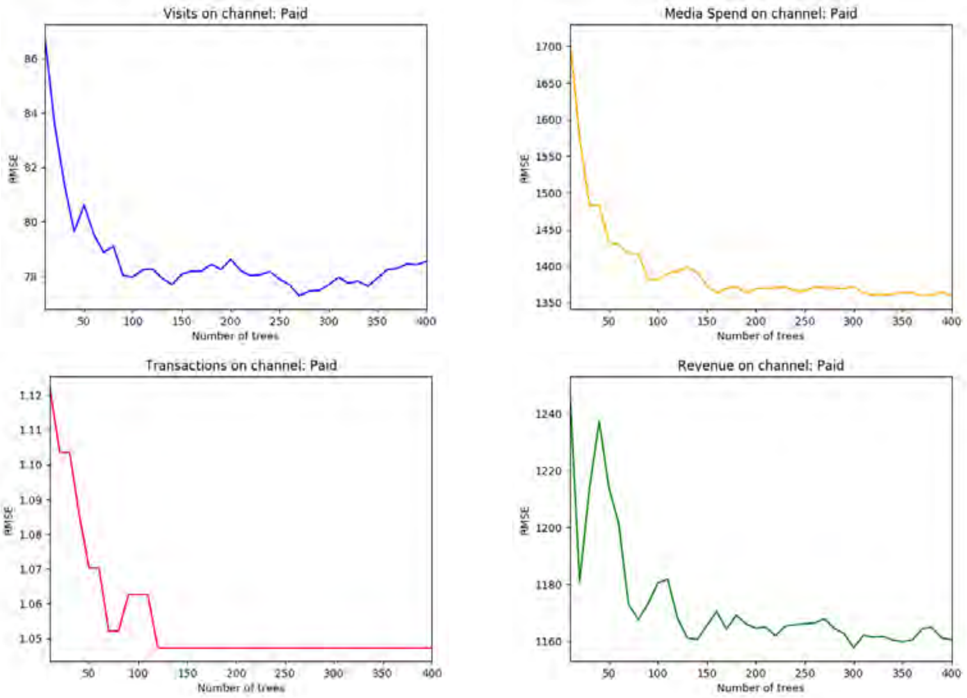


Figure 7.2: Random Forest Number Of Trees

Figure 7.2 shows the RMSE for different number of trees. We observe that the optimal number of trees for visits is 270 as increasing the number of trees will not significantly decrease the RMSE anymore. For transactions, the RMSE becomes stable after 120 trees. The number of trees that is chosen for revenue and media spend is 300 and 370, respectively. It is not necessary to increase the number of trees further as the performance stays roughly the same while the computational time will increase.

The optimal maximum number of features to look for the best split was found to be the square root of the number of input features. The optimal minimum sample leaf size in experiment A was 7, 2, 3 and 2 for visits, transactions, revenue and media spend, respectively. In experiment B, the optimal minimum sample leaf size was 4, 1, 4 and 1 for visits, transactions, revenue and media spend, respectively.

MLP

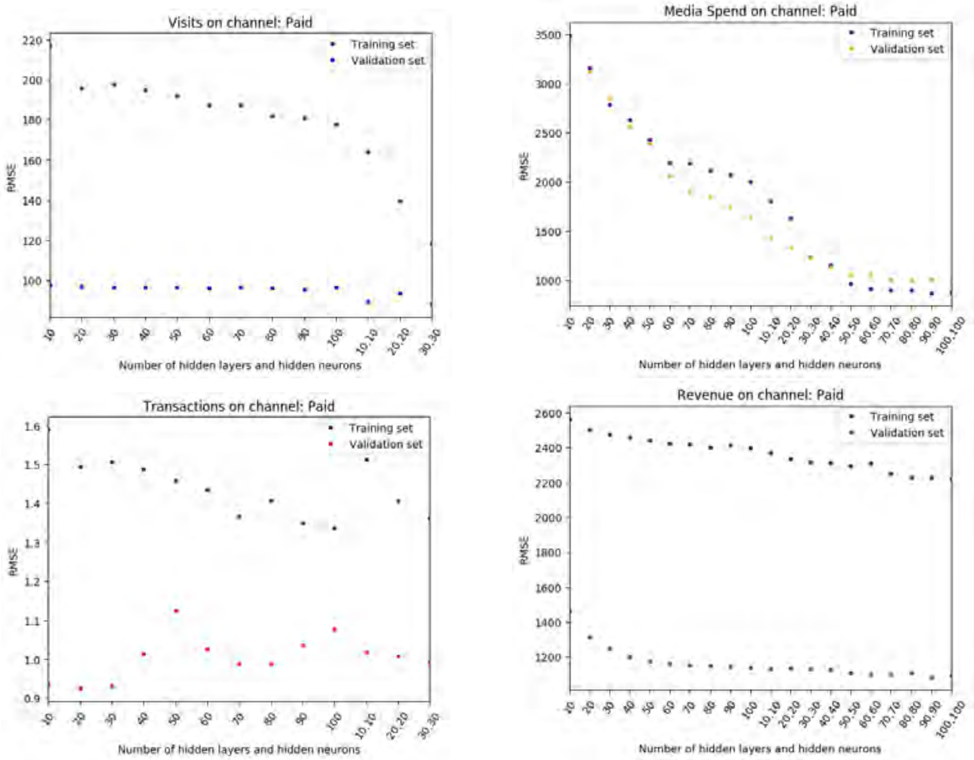


Figure 7.3: MLP number of hidden layers and hidden neurons

Figure 7.3 shows the RMSE for different number of hidden layers and hidden neurons in the training and validation set. We observe that the error of the training set in visits, transactions and revenue is larger than the error in the validation set. This is caused by the different cases in both sets. The time series in the training set contains data over multiple months whereas the time series in the validation set contains only data over one month. The seasonality over the different months causes that some months are harder to learn than other months.

We observe a linear decrease in the error of the training set when two hidden layers are applied for visits. However, this is not observed in the validation set. The errors are rather stable over the different number of hidden layers and neurons. The lowest error is achieved using two hidden layers with 10 neurons in each layer. For media spend and revenue the use of two hidden layers is also sufficient, both errors in the training set and validation set decrease when the number of neurons is increased. The error becomes almost constant after 70 neurons in both layers for media spend and after 60 neurons in both layers for revenue. The lowest error in the validation set is achieved using two hidden layers with 100 neurons in each layer for media spend and two hidden layers with 90 neurons in each layer for revenue. For transactions, one hidden layer with 20 neurons gives the smallest error in the validation set.

LSTM

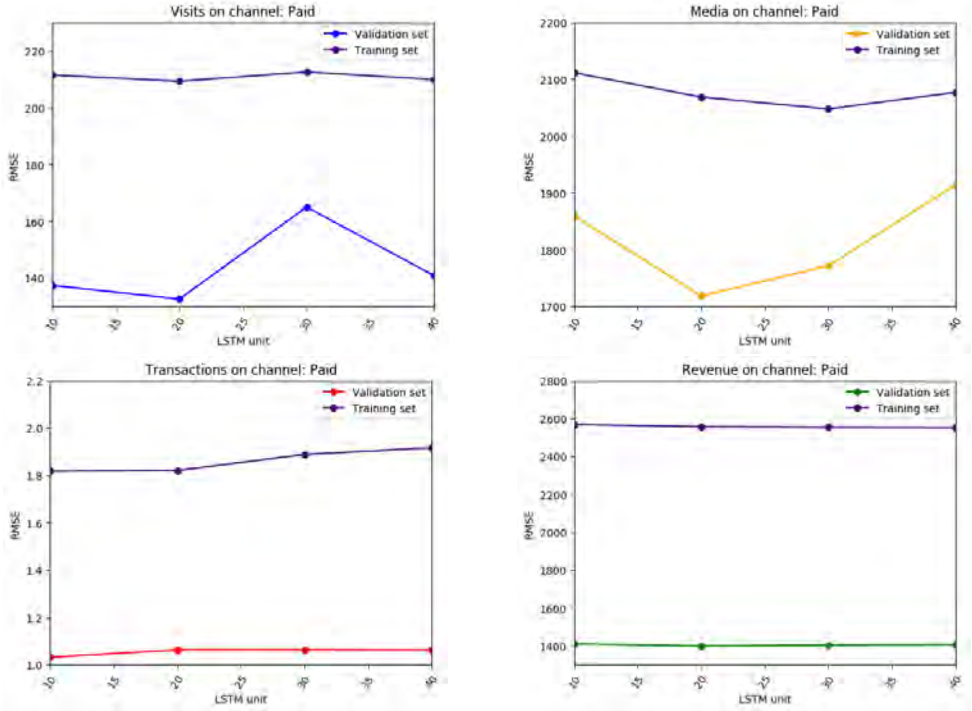


Figure 7.4: LSTM number of units

Figure 7.4 shows the RMSE for different number of LSTM units on the training and validation set. We observe again that the training set error is larger than the validation error. Moreover, increasing the number of LSTM units does not really improve the training set error whereas the validation error for some KPIs increases. The validation error increases after 20 LSTM unit for visits and media spend. The validation error for transactions and revenue are relatively stable for different number of LSTM units. The lowest error is achieved using 10 LSTM unit for transactions and 20 LSTM unit for revenue.

7.1.2 Model Performance

The performance of each model using the time series cross validation on the paid channel over 10 runs is shown in Figure 7.5.

Figure 7.5: Model performance on the paid channel

The Ratio-based model is excluded in transactions and revenue because the error is off the scale that made the other models not comparable anymore. The exact values of the RMSE is also shown in tables in Appendix B.

We observe that ARIMA, MLP and Random Forest outperform the linear regression model (current model) for the four KPIs. ARIMA and Random Forest both perform well for visits. However, MLP starts to perform better than ARIMA and Random Forest after time step $t+4$. For media spend, we observed the opposite where MLP performs better until time step $t+3$ and gets bypassed by Random Forest after time step $t+3$. Random Forest shows the best performance for transactions and MLP shows the best performance for revenue. The Ratio-based model has the worst performance of all the models and LSTM performed the worst when comparing to ARIMA, MLP and Random Forest. These results will be discussed in the discussion in section 8.

7.1.3 Feature Importance

After training the Random Forest, we can obtain their feature importance and this is shown in Figure 7.6 for the paid channel.

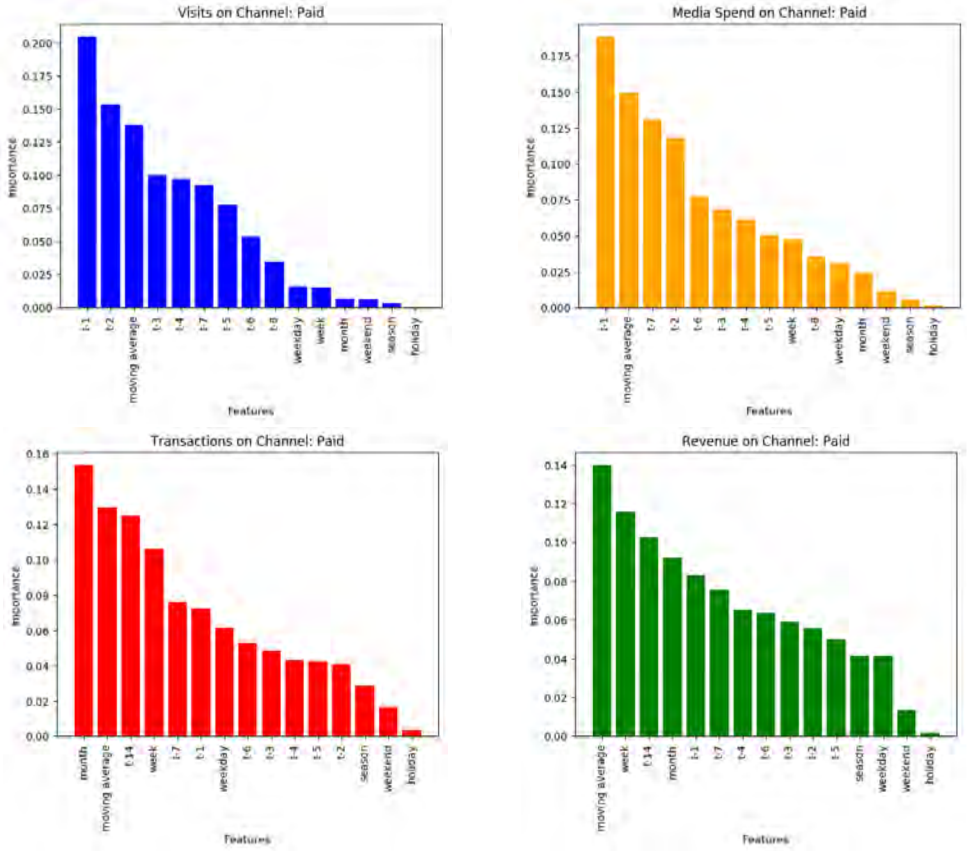


Figure 7.6: Feature Importance of Random Forest on the paid channel

We observe that for visits and media spend the most important feature is $t-1$ (lag 1). This means that previous day value is important in predicting the value on the next day. Moreover, the moving average shows importance in all the KPIs that give information about trends or cycles. Interesting to observe is that lag 8 ($t-8$) shows little importance comparing to the other lags while it shows a strong correlation in the PACF plot in Figure 4.2. Furthermore, the model performance is barely affected when lag 8 is removed from the model.

7.1.4 Model Comparison

Based on the model performance in Figure 7.5, we can already compare the models with each other. However, to ensure that these model performance results are statistically significant, the squared errors of the models are tested using the two-sided Kolmogorov-Smirnov test (K-S test). The two-sided K-S test is used to test whether two independent samples are drawn from the same continuous distribution. In our case we are testing whether the squared errors per instance of two different models are drawn from the same continuous distribution. This is also known as the KS Predictive Accuracy (KSPA) test [39]. Thus, we test the null hypothesis

$$H_0 : \text{The squared error of model A and model B} \\ \text{come from the same continuous distribution}$$

against the alternative hypothesis

$$H_1 : \text{The squared error of model A and model B} \\ \text{do not come from the same continuous distribution}$$

The test is performed for all the forecast horizon. However, the result for time step t is only shown here for clarity. For visits, we obtained the following p-values between the different models as shown in Table 7.1.

p-value	Linear	Ratio Rec	Ratio NonRec	ARIMA	Random Forest	MLP	LSTM
Linear	1	1.57e-24	1.57e-24	1.49e-21	1.25e-26	1.23e-15	4.54e-19
Ratio Rec	1.57e-24	1	1	2.16e-05	3.26e-05	3.07e-05	1.32e-09
Ratio NonRec	1.57e-24	1	1	2.16e-05	3.26e-05	3.07e-05	1.32e-09
ARIMA	1.49e-21	2.16e-05	2.16e-05	1	1.86e-03	0.19	3.21e-03
Random Forest	1.25e-26	3.26e-05	3.26e-05	1.86e-03	1	5.43e-03	1.40e-08
MLP	1.23e-15	3.07e-05	3.07e-05	0.19	5.43e-03	1	5.43e-03
LSTM	4.54e-19	1.32e-09	1.32e-09	3.21e-03	1.40e-08	5.43e-03	1

Table 7.1: The p-values from the KSPA test between the models for the KPI visits for time step t

If we assume a significance level of 95% (p-value less than 0.05), we observe that most of the p-values are significant. The only insignificant one is the distribution of the squared errors between ARIMA and MLP with a p-value of 0.19. This means that we cannot reject the null hypothesis and the squared errors are likely to come from the same distribution. This is unexpected as the RMSE for ARIMA is much lower than MLP as observed in Figure 7.5. Further analysis, we found that the empirical distribution of the squared errors of ARIMA and MLP look almost the same but that MLP contains a few much larger errors than ARIMA. This caused the larger RMSE for MLP. Next, the same analysis is performed for transactions and the result is shown in Table 7.2.

p-value	Linear	ARIMA	Random Forest	MLP	LSTM
Linear	1	0.69	0.50	0.45	0.13
ARIMA	0.69	1	0.99	0.99	0.98
Random Forest	0.50	0.99	1	1	0.99
MLP	0.45	0.99	1	1	0.98
LSTM	0.13	0.98	0.99	0.98	1

Table 7.2: The p-values from the KSPA test between the models for the KPI transactions for time step t

We observe that none of the p-values are significant. This indicates that the errors for all the models come from the same distribution. From the RMSE as shown in Figure 7.5, we already observe that the difference between these models are very small. This difference is mostly caused by the several large transactions in the testset where Random Forest predicts these large transactions the closest.

The p-values for revenue are also insignificant for all the models as shown in Table 7.3. The difference in RSME is also caused by the outliers in the testset where MLP predicts these the closest.

p-value	Linear	ARIMA	Random Forest	MLP	LSTM
Linear	1	0.56	0.22	0.22	0.22
ARIMA	0.56	1	0.30	0.16	0.45
Random Forest	0.22	0.30	1	0.63	0.91
MLP	0.22	0.16	0.63	1	0.30
LSTM	0.22	0.45	0.91	0.30	1

Table 7.3: The p-values from the KSPA test between the models for the KPI revenue for time step t

Table 7.4 shows the p-values between the models for the media spend. We observe that most of them are significant and the insignificant ones are the errors between Random Forest and MLP, Random Forest and ARIMA and MLP and ARIMA with p-values 0.63, 0.63 and 0.45, respectively. Moreover, the difference in RMSE between these models is small as shown in Figure 7.5. Thus, it is expected that these squared errors come from the same distribution.

p-value	Linear	Ratio Rec	Ratio NonRec	ARIMA	Random Forest	MLP	LSTM
Linear	1	1.25e-26	1.25e-26	3.50e-29	6.20e-30	1.19e-34	6.58e-16
Ratio Rec	1.25e-26	1	1	0.01	0.04	0.01	5.56e-09
Ratio NonRec	1.25e-26	1	1	0.01	0.04	0.01	5.56e-09
ARIMA	3.50e-29	0.01	0.01	1	0.63	0.45	1.29e-07
Random Forest	6.20e-30	0.04	0.04	0.63	1	0.63	3.47e-09
MLP	1.19e-34	0.01	0.01	0.45	0.63	1	1.98e-07
LSTM	6.58e-16	5.56e-09	5.56e-09	1.29e-07	3.47e-09	1.98e-07	1

Table 7.4: The p-values from the KSPA test between the models for the KPI media spend for time step t

When the KSPA test was applied for the other forecast horizons. We found for visits that the forecast generated by ARIMA and MLP are not statistically different from each other for all time steps. Moreover, the squared error between Random Forest and MLP and the squared error between Random Forest and ARIMA become insignificant after time step t with p-value of 0.10 and 0.6, respectively. This mean that there is no significant difference between these three models after time step t.

For transactions, we found that neither of the models are statistically significant different from each other for all the forecast horizon. This means that all models are suitable for forecasting transactions. The reason is that the variance of transactions is relatively small which makes the forecast error relatively similar for all models.

For revenue, we also found that neither of the models are statistically significant different from each other for all the forecast horizon. This while we observe a significant difference in RMSE between some models over the forecast horizon. The difference is caused by large revenues that is better captured by some models.

For media spend, we found that the forecast error between ARIMA and Random Forest becomes statistically significant after time step $t+4$ (p-value of 0.04). Random Forest is then performing statistically better than ARIMA. Furthermore, the forecast error of MLP and Random Forest are insignificant for all time steps which means that they are both performing the same.

7.2 Experiment B

7.2.1 Hyperparameters

The hyperparameters need to be optimized as in experiment A. The same number of trees was used as in experiment A because the same behaviour was observed in experiment B. However, this same behaviour was not observed for the number of neurons in the neural networks. In this experiment, more neurons were needed in the neural networks because of the increased number of inputs. The change of the number of neurons in MLP is shown in Table 7.5 and the change of number of LSTM units is shown in Table 7.6.

MLP	Experiment A	Experiment B
Visits	10,10	20,20
Transactions	20	40
Revenue	90,90	100,100
Media Spend	100,100	100,100

Table 7.5: The optimal neurons in the hidden layers for experiment A and B

LSTM	Experiment A	Experiment B
Visits	20	20
Transactions	10	40
Revenue	10	30
Media Spend	20	40

Table 7.6: The optimal LSTM units for experiment A and B

7.2.2 Model Performance

The performance of the models are obtained over 10 runs and the result along with the result from experiment A are presented. Table 7.7 shows the RMSE for Random Forest in both experiments and its p-value of the KSPA test between the squared error of the model in the two experiments.

Table 7.7: The RMSE of time step *t* on the paid channel using Random Forest of experiment A and B and its p-value of the KSPA test

We observe that the RMSE in experiment B does not differ significantly from the RMSE in experiment A for all KPIs. This is also confirmed by the KSPA test as non of the p-values are significant. Thus, the additional features are not contributing in improving the performance. The reason is that these additional features are largely irrelevant that will be shown in the feature importance in Figure 7.7.

Next, the result for MLP is shown in Table 7.8.

Table 7.8: The RMSE of time step *t* on the paid channel using MLP of experiment A and B and its p-value of the KSPA test

The same result as in Random Forest is observed here. The performance is not improving and it is almost the same as experiment A. This behaviour is also observed in LSTM as shown in Table 7.9. From the results, it becomes clear that it has no benefit to add the additional features as the performance is not improving.

Table 7.9: The RMSE of time step *t* on the paid channel using LSTM of experiment A and B and its p-value of the KSPA test

7.2.3 Feature Importance

The feature importance for Random Forest can again be derived to obtain the importance of the additional features. The result is shown in Figure 7.7.

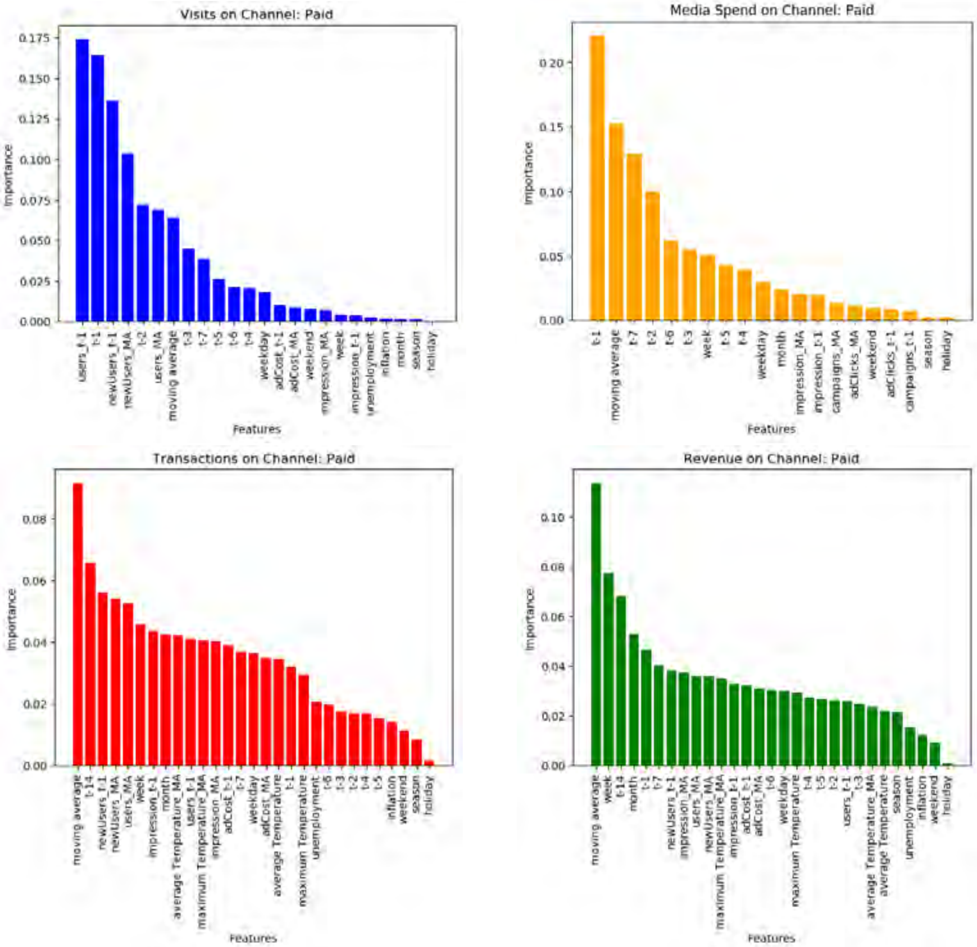


Figure 7.7: Feature Importance of Random Forest including additional features on the paid channel

We observe for visits that the features "users" and "new users" are quite important because they have a very strong correlation with visits as shown in Figure 4.3. "Ad cost" and "impression" on the contrary shows little importance.

The most important features in media spend are still the same features in experiment A. The additional features "impression", "ad clicks" and "campaigns" show little importance.

The moving average is still the top important feature in transactions and revenue. The additional features seem to have an importance but they are all below 0.08 which indicates a very low importance.

7.3 Summary

From the results of both experiments, we obtained that adding the additional features in experiment B did not result in improvement. Moreover, when we statistically compared the performance of the models using the KSPA test, some of the models were insignificant while there was a significant difference in RMSE. Further analysis, we found out that this difference was caused by some outliers in the testset. Some of the models predicted these outliers better than others. The model with the lowest error for each KPI performed better than the current model. The best model for each KPI along with its average improvement over the time steps on the paid channel is shown in Table 7.10.

KPI	Best Model	Improvement
Visits	Random Forest	41.75%
Transactions	Random Forest	17.04%
Revenue	MLP	17.02%
Media Spend	MLP	56.00%

Table 7.10: The best models for each KPI on the paid channel and their improvement with respect to the current model

Transactions and revenue have the smallest improvement because these are the hardest to predict. The media spend has the greatest improvement. However, the RMSE of these best models are still large when they are compared to the magnitude of the values in the test set.

8 Discussion and Conclusion

The implemented models except the Ratio-based model were able to improve the forecast of the current model. The Ratio-based model resulted in huge errors as the variability between the same days over the weeks are large. Even though improvement is made, the performance is still not outstanding. The problem is not caused by the implemented models but the limited amount of historical data and features available. We only have data for three years where one year was already used for testing. Thus, training on two years of historical data is just too small for the models to learn and understand the data. This especially holds for complex models like neural networks where more training data are needed for them to generalize. This may explain the bad performance of LSTM whereas ARIMA and Random Forest require less data to generalize. Moreover, the validation set consists of only one month data instead of one year that could give the wrong optimal hyperparameters. However, if one year of data is taken as validation set, then only one year of data remains for the training set. This is inadequate for training a good model as the yearly patterns cannot be captured. Besides the limited data, the recursive strategy can play a role in the performance of the multi-step ahead forecast. From the results, we have observed that the error increases along with the time horizon. This error might be reduced when using the direct strategy.

From experiment A, we obtained that it is hard to make accurate predictions of the KPIs based on only their own past values. Therefore, we included external data that is still applicable for all clients and other Google Analytics features in experiment B. However, they have shown less predictive value. Moreover, the external data was coarse grained except for weather which makes it less valuable for daily forecast. Based on the feature importance, the moving average has shown to be always on the top 3 most important feature because it captures the overall trend of the KPIs over the time steps. However, neither of the most important feature exceeded an importance value of 0.25. This indicates that the data have low predictive value and other data are needed.

Data that could improve the performance is internal company data. However, these data cannot be derived from Google Analytics. Data such as information about the products they offer. Relevant information such as the price, rating, reviews and discounts. This information could help in the prediction of transactions and revenue as shown in the literature review. Moreover, information about their upcoming media strategy can be relevant [40]. This will all have an impact on the performance of the KPIs. Besides online marketing, offline marketing is also used to increase awareness such as informative pamphlets, flyers, television and radio advertising and magazines [41]. In conclusion, internal company data might be important as they describe the behaviour of KPIs more explicitly. However, this suggests that we have to acquire these data that is not generic anymore and does not match with the product point of view. Thus, there is a trade off between performance and scalability. High performance can be acquired when more attention is paid on each client [42]. However, this will result in higher cost and more time from the business side.

Based on this research, we would recommend to be cautious with using only Google Analytics data to forecast the KPIs because limited knowledge can be gained from this. Moreover, it is dangerous when marketeers rely on these bad predictions to take actions, because this will lead to wrong actions. For future research, it would be interesting to acquire company internal data and more history if available. This data together with Google Analytics data can then be applied and might improve the KPIs prediction further. Furthermore, we have only focused on the paid channel in this research. It might be also interesting for future research to forecast the KPIs on the other channels. Channels that are less affected by advertising such as the direct channel might be easier to predict.

The formulated research question was: *"With how much can the forecast of KPIs be improved compared to the current model using only data that is generic enough to be available from all clients?"*. The findings in this research show that the forecast of visits improved with 42%, transactions and revenue with 17% and media spend with 56%. However, the prediction error is still large for all KPIs. The reason is that generic data can only capture generic information on the KPIs whereas many other specific factors can affect these KPIs. Therefore, it is really difficult to make accurate predictions using only generic information.

References

[1] Shuojia Guo and Bingjia Shao. “Quantitative evaluation of e-commercial Web sites of foreign trade enterprises in Chongqing”. In: 1 (June 2005), 780–785 Vol. 1.

[2] statista. *Retail e-commerce sales worldwide from 2014 to 2021 (in billion U.S. dollars)*. `https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/`.

[3] Alanah J. Davis and Deepak Khazanchi. “An Empirical Study of Online Word of Mouth as a Predictor for Multi-product Category e-Commerce Sales”. In: 18 (May 2008), pp. 130–141.

[4] C. Ranganathan and Elizabeth Grandon. “An Exploratory Examination of Factors Affecting Online Sales”. In: *Journal of Computer Information Systems* 42.3 (2002), pp. 87–93.

[5] Qiang Ye, Rob Law, and Bin Gu. “The Impact of Online User Reviews on Hotel Room Sales”. In: 28 (Mar. 2009), pp. 180–182.

[6] James W. Taylor. “Forecasting daily supermarket sales using exponentially weighted quantile regression”. In: *European Journal of Operational Research* 178.1 (2007), pp. 154–167.

[7] Besta P. Lenort R. “Hierarchical Sales Forecasting System for Apparel Companies and Supply Chains”. In: 21 (2013), pp. 7–11.

[8] Ilan Alon, Min Qi, and Robert J. Sadowski. “Forecasting aggregate retail sales:: a comparison of artificial neural networks and traditional methods”. In: *Journal of Retailing and Consumer Services* 8.3 (2001), pp. 147–156.

[9] Gareth James et al. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[10] By George E. P. Box et al. *Time Series Analysis: Forecasting and Control*. Vol. 68. Jan. 2016. ISBN: 978-1-118-67502-1.

[11] GP.Zhang. “Neural Networks for Time-Series Forecasting”. In: *Handbook of Natural Computing* (2012), pp. 461–477.

[12] Vincent Cho. “A comparison of three different approaches to tourist arrival forecasting”. In: *Tourism Management* 24.3 (2003), pp. 323–330.

[13] F. M. Thiesing and O. Vornberger. “Sales forecasting using neural networks”. In: *Neural Networks,1997., International Conference on* 4 (1997), 2125–2128 vol.4.

[14] Peter v.d. Reijden and Otto Koppius. “THE VALUE OF ONLINE PRODUCT BUZZ IN SALES FORECASTING”. In: *International Conference on Information Systems* (2010).

[15] Kris Johnson Ferreira, Bin Hong Alex Lee, and David Simchi-Levi. “Analytics for an Online Retailer: Demand Forecasting and Price Optimization”. In: *Manufacturing & Service Operations Management* 18.1 (2016), pp. 69–88.

[16] P. Meulstee and M. Pechenizkiy. “Food Sales Prediction: ”If Only It Knew What We Know””. In: *2008 IEEE International Conference on Data Mining Workshops* (2008), pp. 134–143.

[17] Kyle B. Murray et al. “The effect of weather on consumer spending”. In: *Journal of Retailing and Consumer Services* 17.6 (2010), pp. 512–520.

[18] *Google Analytics Solutions*. `https://www.google.com/analytics/analytics/#?modal_active=none`.

[19] *Klimatologie:Daggegevens van het weer in Nederland*. `http://projects.knmi.nl/klimatologie/daggegevens/selectie.cgi`.

[20] *Centraal Bureau voor de Statistiek: StatLine*. `http://statline.cbs.nl/Statweb/?LA=nl`.

[21] Ratnadip Adhikari and R K. Agrawal. *An Introductory Study on Time series Modeling and Forecasting*. Jan. 2013. ISBN: 978-3-659-33508-2.

[22] B.Choi. *ARMA Model Identification*. Springer Series in Statistics, 1992. ISBN: 978-1-4613-9745-8.

[23] T.K. Ho. “Random decision forests”. In: *In: Document Analysis and Recognition* (1995), pp. 278–282.

[24] Dan Benyamin. *A Gentle Introduction to Random Forests, Ensembles, and Performance Metrics in a Commercial System*. `http://blog.citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics`.

[25] M.A. Lehr B.Widrow D.E.Rumelhart. “Neural networks: Applications in industry, business and science”. In: *Communications of the ACM* (1994), pp. 93–105.

[26] Z. Tang and P.A. Fishwick. “Feedforward Neural Nets as Models for Time Series Forecasting”. In: *ORSA Journal on Computing* (1993), pp. 374–385.

[27] G. Dorffner. “Neural Networks for Time Series Processing”. In: 6 (1996).

[28] J. Cazala. *Perceptron*. `https://github.com/cazala/synaptic/wiki/Architect`.

[29] R. Hecht-Nielsen. “Theory of the backpropagation neural network”. In: (1989), 593–605 vol.1.

[30] P.D.Kingma and J.L.Ba. “Adam: a method for stochastic optimization”. In: *ICLR* (2015).

[31] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: (Sept. 2016).

[32] D. Britz. *Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs.* `http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/`.

[33] M. Assaad and R. Boné. “A new boosting algorithm for improved time-series forecasting with recurrent neural networks”. In: *Information Fusion* 9.1 (2008), pp. 41–55.

[34] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-term Memory”. In: 9 (Dec. 1997), pp. 1735–80.

[35] colah. *Understanding LSTM Networks.* `http://blog.otoro.net/2015/05/14/long-short-term-memory/`.

[36] S.B. Taieb and R.J. Hyndman. “Recursive and direct multi-step forecasting: the best of both worlds”. In: (2012).

[37] S. B. Taieb and A. F. Atiya. “A Bias and Variance Analysis for Multistep-Ahead Time Series Forecasting”. In: *IEEE Transactions on Neural Networks and Learning Systems* 27.1 (Jan. 2016), pp. 62–76.

[38] Rob J Hyndman. *Cross-validation for time series.* `https://robjhyndman.com/hyndsight/tscv/`.

[39] Hossein Hassani and Emmanuel Sirimal Silva. “A Kolmogorov-Smirnov Based Test for Comparing the Predictive Accuracy of Two Sets of Forecasts”. In: *Econometrics* 3.3 (2015), pp. 590–609.

[40] Brian Smith. “The effectiveness of marketing strategy making processes: A critical literature review and a research agenda”. In: *Journal of Targeting, Measurement and Analysis for Marketing* 11.3 (Jan. 2003), pp. 273–290.

[41] Jura Liaukonyte, Thales Teixeira, and Kenneth C. Wilbur. “Television Advertising and Online Shopping”. In: *Marketing Science* 34.3 (2015), pp. 311–330.

[42] Bob Edmundson, Michael Lawrence, and Marcus O’Connor. “The use of non-time series information in sales forecasting: A case study”. In: *Journal of Forecasting* 7.3 (1988), pp. 201–211.

Appendices

A Additional Results Data Analysis

A.1 The KPIs over monthly periods

A.2 The KPIs over weekdays

B Model Performance Experiment A

B.1 Visits

B.2 Transactions

B.3 Revenue

B.4 Media Spend