FACULTEIT ECONOMIE EN BEDRIJFSKUNDE

Predicting car sales with search trends data and Instagram data

A case study in the Netherlands

Word count: 18,539

Jasper Verdonck

Student number: 01508185

Supervisor: Prof. dr. Dirk Van den Poel

Co-supervisor: Lisa Schetgen

A dissertation submitted to Ghent University in partial fulfilment of the requirements for the degree of Master of Business Engineering: Data Analytics

Academic year: 2019 - 2020

UNIVERSITEIT GENT

PERMISSION

I declare that the content of this Master's Dissertation may be consulted and/or reproduced, provided that the source is referenced.

Name student: Jasper Verdonck

Signature:

Jasper Verdonck

Foreword

I am delighted to end my five years of education with this master thesis. Furthermore, I would like to thank the following people for steering me through my master dissertation.

First of all, I would like to thank prof. dr. Van den Poel who provided me with the freedom to choose my own thesis subject, who scraped the Instagram data necessary for my research and who gave advice throughout this dissertation. Because of prof. dr. Van den Poel, I was able to combine my passion for data analytics with my affection for cars. Moreover, I am grateful for the initial guidance of Sarah Carron. Misses Carron found me a new mentor, Lisa Schetgen. I would like to express my gratitude towards misses Schetgen for giving me useful feedback and bringing new insights when I was facing difficulties. Last but not least, I would like to thank my family and friends for their unconditional support, for motivating me to get the best out of myself and for proofreading my master dissertation.

Impact of the Covid-19 crisis

I hereby declare that the research conducted for my master dissertation did not suffer from the Covid-19 crisis. Due to the flexibility of my promotor prof. dr. Van den Poel and my mentor Lisa Schetgen, I was able to contact them via e-mail and online appointments when I faced difficulties.

However, it is possible that the actual car sales in the year 2020 of the car models used in this thesis are influenced by the Covid-19 crisis. As it is almost impossible to foresee the impact of this crisis on the car sales in the Netherlands, this was not captured by the forecasting algorithms. Consequently, the performance of these algorithms could be slightly altered due to the impact of the Covid-19 crisis.

Table of contents

List of abbreviations

ANNs	Artificial neural networks
API	Application programming interface
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive–moving-average
ARSA	Autoregressive Sentiment-Aware
ARSQA	Autoregressive Sentiment and Quality Aware
ATM	Automated teller machines
DJIA	Dow Jones Industrial Average
i.i.d.	Independent and identically distributed
LSTM	Long Short-Term Memory
MA	Moving average
MAE	Mean absolute error
MAPE	Mean Absolute Percentage Error
MSE	Mean squared error
OLS	Ordinary least squares
ReLu	Rectified Linear Unit
RMSE	Root Mean Squared Error
RNNs	Recurrent neural networks
SAM	Sequence-Alignment Method
tanh	Hyperbolic tangent
URL	Uniform Resource Locator
VADER	Valence Aware Dictionary and sEntiment Reasoner
VIF	Variance inflation factor
WOM	Word-of-mouth

List of tables

List of figures

1 Introduction

The car industry is characterized by high costs and uncertainties (Fantazzini & Toktamysova, 2015). In particular, it comes with extremely large development costs and intense competition (Dodourova & Bevis, 2014). On top of that, customers' expectations are high as they require more for the same price (Ili, Albers, & Miller, 2010). Consequently, it is of vital importance that car manufacturers are aware of the number of cars that will be sold in the future, as a shortage or an oversupply of vehicles may harm the firms' reputation as well as its financial health (Fantazzini & Toktamysova, 2015). Conducted research concerning the prediction of car sales concludes that traditional forecasting models, which are solely based on historical sales, give unreliable forecasts (Ahn & Spangler, 2014).

A potential explanation is that these models do not incorporate the influence of recent events on future sales. Such events can for instance exist of word-of-mouth (*WOM*) (Ahn & Spangler, 2014). *WOM* can be seen as the exchange of product information between customers (Chu & Kim, 2011). It plays a vital part in influencing the behavior of consumers towards goods and services. On top of that, *WOM* is considered more reliable than brand marketing. Hence, customers' attitude towards products are influenced by *WOM*. As a result of the rise of the internet, *WOM* gained popularity under the term electronic *WOM*, for which social media platforms such as Facebook, Twitter and Instagram serve as the ideal instrument (Chu & Kim, 2011). Consequently, information obtained from social media can be a valuable source to predict sales.

Instagram is a social media platform that has gained a lot of popularity during the past years. With more than one billion users, it is one of the most used social media networks (Clement, 2020). Nonetheless, as far as my knowledge extends, no research has studied the predictive power of Instagram features in terms of predicting sales.

Another interesting source of information that emerged since the rise of the internet is the search behavior of internet users. The prediction of forecasting sales by means of search queries data proved to be successful in the past, especially in regions where the amount of internet users is significantly large (Ginsberg et al., 2009). A possible reason why search queries data seems to be a better source of information than traditional macro-economic data, is that the former may provide a more rational picture about the things customers are interested in (Wu & Brynjolfsson, 2009).

Given the relevance of car manufacturers predicting future car sales as accurate as possible, the goal of this master thesis is to forecast the monthly car sales in the year 2020 of the most sold car models in the Netherlands. Considering the potential of Google Trends data and Instagram data to predict sales, I make use of three different datasets, on top of the historical car sales data, to forecast the aforementioned monthly car sales. Firstly, the predictive power of Google Trends data is examined. Secondly, the predictive power of Instagram-related features is investigated. Thirdly, the combination of Google Trends data and Instagram data is explored.

This master thesis is structured as follows. Chapter 2 summarizes the relevant literature about the prediction of trends based on social media and search trends data. It also covers relevant literature concerning natural language processing, or in particular, sentiment analysis. Chapter 3 consists of the proposed research questions, which are based on the literature in chapter 2. Chapter 4 describes the relevant data as well as the data collection. In chapter 5, the applied methodology is explained in five parts. The first part focuses on a clustering algorithm that places car models with similar sales patterns in the same group. The second part elaborates on the forecasting approach. More specifically, it comprises the forecasting algorithms utilized in this master dissertation, the data preparation prior to the implementation of the forecasting algorithms and the actual implementation of these algorithms. The third part of chapter five deals with the applied performance measures to determine the predictive power of the utilized forecasting algorithms. Part four discusses a cross-validation technique in the context of time series. The statistical tests used to evaluate a potential significant difference between the performance of the forecasting techniques are mentioned in the final part of chapter 5. Chapter 6 discusses the results of the applied methodology. Finally, chapter 7 presents the conclusion of this master dissertation and deals with its limitations and suggestions for future research.

2 Literature review

In this chapter, an overview of the relevant literature is provided. Firstly, the prediction of social and economic trends is discussed, which is based on data derived from social media, search data and a combination of both. Additionally, an overview of the used literature is visualized in table 1. Secondly, a specific part of natural language processing, namely sentiment analysis, is described.

2.1 Prediction of social and economic trends

2.1.1 Prediction based on social media

Since the rise of social media, several researchers have utilized data obtained from social media in order to predict the future. Predicting future social and economic trends by means of social media has been researched in a variety of domains, such as the movie sector (Asur & Huberman, 2010; Yu, Liu, Huang, & An, 2010), the financial sector (Bollen, Mao, & Zeng, 2011), the mobile industry (Lassen, Madsen, & Vatrapu, 2014), the car industry (Ahn & Spangler, 2014; Pai & Liu, 2018), etcetera.

Asur and Huberman (2010) made use of tweets to predict the box-office revenues gained by movies. The authors implemented a linear regression model based on the number of posted tweets concerning movie-related topics (Assur & Huberman, 2010). In addition, Yu et al. (2010) forecasted the box-office revenues of movies by deploying sentiment analysis on *IMDB* reviews. An Autoregressive Sentiment-Aware (*ARSA*) model was compared with Autoregressive (*AR*) models which do not include sentiment data. On top of that, a quality indicator was added to the *ARSA* model to specify the quality of the reviews, resulting in an Autoregressive Sentiment and Quality Aware (*ARSQA*) model (Yu et al., 2010). The abovementioned researches conclude that the predictive power is significantly improved by including sentiment analysis into the model (Asur & Huberman, 2010; Yu et al., 2010). Furthermore, Asur and Huberman (2010) observed that their model scored better in terms of accuracy in contrast to the Hollywood Stock Exchange.

Lassen et al. (2014) were able to forecast iPhone sales by analyzing tweets using a linear regression model. However, sentiment features only slightly enhance the performance of the model (Lassen et al., 2014). Furthermore, Ahn and Spangler (2014) studied the influence of data from multiple social media sources on car sales of two automobile brands. To realize this, an Autoregressive Integrated Moving Average *(ARIMA)* model was fit three times, each time on a different dataset. First, the model was solely fit on historical sales data. Second, the model was trained on historical sales data as well as sentiment data. The final fit was applied on historical sales data, sentiment data and topical keyword frequency. Overall, the incorporation of social media data significantly improves the predictions of car sales (Ahn & Sprangler, 2014).

In order to predict car sales in the United States, Pai and Liu (2018) utilized stock market values and sentiment scores of tweets. The conclusion of the research is twofold. Firstly, both sentiment scores and stock market values improve the forecasting accuracy. Secondly, removing the seasonality from the explanatory variables as well as from the response variable seems to expand the forecasting power of the model (Pai & Liu, 2018).

Bollen et al. (2011) examined the usefulness of sentiment features obtained from Twitter data to estimate the Dow Jones Industrial Average (*DJIA*) by training a Self-Organizing Fuzzy Neural Network. More specifically, positive and negative sentiments along with six dimensions of mood (e.g. calm, alert, sure, vital, kind and happy) were extracted from tweets. In contrast to the previously mentioned studies, the researchers pointed out that sentiment as well as five out of six mood dimensions were not predictive of the *DJIA*, leaving the calmness of the public as the only significant variable (Bollen et al., 2011).

The literature mentioned above focuses on extracting features from text-based social media platforms, such as Twitter (Hu, Manikonda, & Kambhampati, 2014). However, image-based social media platforms, such as Instagram, also contain promising information (Colliander & Marder, 2018; Hu et al., 2014; Pittman & Reich, 2016).

Colliander and Marder (2018) examined the effects of the setting in which a picture is taken by a brand in terms of the perceived image of that brand by customers and the recommendation of that brand to others. The authors conclude that in case a brand posts pictures in a less professional setting, followers are more inclined to like the post of the brand, recommend it to their social environment and believe the brand to be more credible in contrast to professional pictures posted by the brand (Colliander & Marder, 2018). Moreover, Highfield and Leaver (2015) suggested to analyze Instagram data based on research concerning Twitter data, as both types of data make use of hashtags (Highfield & Leaver, 2015).

In order to perform supervised learning techniques on pictures, these should be annotated (LeCun, Bengio, & Hinton, 2015). As manually labeling thousands, let alone millions of pictures, can take quite some effort, automatically labeling images may be a faster and less labor-intensive alternative (Giannoulakis & Tsapatsoulis, 2015, 2016a). Giannoulakis and Tsapatsoulis (2015, 2016a) conducted several studies about hashtags associated with images posted on Instagram. In particular, the researchers investigated whether these hashtags could be used to label the related image. The results state that, on average, only 25 percent of the hashtags are directly related to the content of the image, whereas the remaining 75 percent are depicted as stophashtags (Giannoulakis & Tsapatsoulis, 2015, 2016a). Stophashtags include all hashtags that are not directly related to the image itself but are rather utilized to enhance the searchability of the image or as a form of metacommunication (Giannoulakis & Tsapatsoulis, 2016b).

In summary, the incorporation of sentiment features extracted from text-based social media generally significantly improves the model's predictive power in terms of forecasting vehicle sales. Although the extraction of useful information such as images and videos from an image-based social media platform such as Instagram seems promising (Colliander & Marder, 2018; Hu et al., 2014; Pittman & Reich, 2016), no literature has been written about the usage of Instagram data for the prediction of car sales, let alone the prediction of sales in general, as far as my knowledge extends.

2.1.2 Prediction based on search trends data

The importance of web search behavior to predict social and economic trends has been studied extensively over the years, varying from predicting flu epidemics (Ginsberg et al., 2009; Polgreen, Chen, Pennock, Nelson, & Weinstein, 2008; Santillana, Nsoesie, Mekaru, Scales, & Brownstein, 2014; Santillana et al., 2015) to forecasting car sales (Barreira, Godinho, & Melo, 2013; Choi & Varian, 2009, 2012).

Ginsberg et al. (2009) mentioned that online search queries may offer a faster way to predict flu outbreaks in a certain region, especially when a large part of that region uses the internet, instead of the traditional approximations. A possible explanation is that these traditional approximations are typically published later in contrast to trends data (Santillana et al., 2015; Santillana et al., 2014).

As far as my knowledge extends, Choi and Varian (2009) were the first researchers that investigated the importance of Google Trends search data when predicting automotive sales. The authors conclude that non-complex *AR* models along with fixed effects models including Google Trends predictors appear to surpass models without these predictors in terms of performance (Choi & Varian, 2009).

Moreover, Seebach, Pahlke and Beck (2011) forecasted new car sales of Germany's two main car producers by using Google Trends data. The findings of the authors indicate that search trend-based models perform better than well-recognized benchmark models. Over the following years, other researchers have investigated the predictive power of Google Trends data in the context of German car sales as well (Fantazzini & Toktamysova, 2015). In particular, Fantazzini and Toktamysova (2015) executed multivariate models making use of economic variables and variables extracted from Google Trends in order to predict monthly car sales of ten German car brands, for different forecast horizons up to two years. The results show that, for most car brands and forecast horizons, models including data from Google's search engine significantly perform better than other models. This is especially the case in terms of forecast horizons above twelve months (Fantazzini & Toktamysova, 2015).

Carrière-Swallow and Labbé (2013) made use of Google Trends data to nowcast the car sales in Chile. More specifically, simple nowcasting models incorporating a self-made automotive index, which were implemented since Google did not provide search categories for Chile at that time, tend to outperform well-accepted benchmark models (Carrière-Swallow & Labbé, 2013).

In contrast to the previously mentioned studies, research of Barreira et al. (2013) reveals that the inclusion of search queries does not always improve the forecasting power of models to predict car sales. By means of an autoregressive–moving-average *(ARMA)* model, the authors studied the impact of search data on the accuracy of nowcasting car sales for France, Portugal, Spain and Italy. The incorporation of Google Trends data only seemed to significantly improve the forecasting power in case of Portugal (Barreira et al., 2013). In a similar study, Tomczyk and Doligalski (2015) applied linear regression to Google Trends search data and a macroeconomic index to predict new car registrations in Poland. The findings of this study reveal that Google search data and the macroeconomic index statistically influence the vehicle registrations of five major car brands, at least for a one-month timeframe (Tomczyk & Doligalski, 2015).

More recently, Nymand-Andersen and Pantelidis (2018) examined the predictive power of Google search data on new car registrations in Europe (i.e. Belgium, Germany, Ireland, Spain, France, Italy, the Netherlands, Austria, Portugal and Slovenia). The *AR* models including search data seem to statistically outperform the baseline model as well as most of the equivalent *AR* models without Google search data (Nymand-Andersen & Pantelidis, 2018).

In summary, the inclusion of online search data significantly improves the forecasting of car sales. A possible explanation is that online internet searches indicate a more realistic view of customers' interests compared to traditional macro-economic variables (Wu & Brynjolfsson, 2009).

2.1.3 Prediction based on both social media and search trends data

Based on the previously mentioned literature, I can conclude that search engine data as well as features gathered from social media provide meaningful information in terms of the purchasing behavior of consumers. In this section, researchers that compared these two types of data and investigated the relevance of combining them, is discussed.

One of the first researches that implemented search queries data along with social media data to predict car sales is conducted by Geva, Oestreicher-Singer, Efron and Shimshoni (2013, 2015). The conclusion of the research is validated on both a linear regressor and a nonlinear neural network model. Firstly, the combination of forum data and Google Trends search data provides better results than utilizing only one of them. Secondly, search data is more informative than data extracted from forums (Geva et al., 2013, 2015).

Benthaus and Skodda (2015) further elaborated on the work of their colleagues Seebach et al. (2011) (see subsection 2.1.2), by also including blog data from Twitter. Linear regression models using both social media data and Google Trends data are better at predicting car sales than the same models using only one of these types of data (Benthaus & Skodda, 2015). Moreover, internet trends data and social media data complement each other. Consequently, combining these two types of data provides a better result (Santillana et al., 2015; Geva et al., 2013, 2015).

To conclude, the combination of search trends data and social media data leads to a more informative data source. A possible explanation for this conclusion is given by Geva et al. (2013). When taking the nature of the two types of data into account, the authors observed a clear contrast between search trends data and social media data. More specifically, the first tends to capture the actual products customers are interested in without affecting the opinions of their social environment, while the latter plays an important role in the way others perceive products (Geva et al., 2013).

Table 1. Literature review

Study	Google Trends	Twitter	Other	Content
Ahn & Spangler, 2014			X	Estimating monthly car sales of two car brands based on sentiment data and topical key words extracted from social media websites
Asur & Huberman, 2010		X		Forecasting box-office revenues for movies using Twitter data
Barreira, Godinho, & Melo, 2013	X			Studying the impact of Google search data on the predictive power of nowcasting unemployment rates and car sales for France, Italy, Portugal and Spain
Benthaus & Skodda, 2015	X	X		Nowcasting vehicle sales of two of the main German automobile manufacturers using Google Trends data and Twitter data
Bollen, Mao, & Zeng, 2011		X		Using Twitter mood to estimate the Dow Jones Industrial Average (DJIA)
Carrière-Swallow & Labbé, 2013	X			Nowcasting car sales in Chile utilizing macroeconomic variables and Google search data
Choi & Varian, 2009	X			Nowcasting automotive sales, travel destinations and house sales by means of Google Trends data
Fantazzini & Toktamysova, 2015	X			Predicting car sales in Germany by means of Google Trends data and economic variables
Geva, Oestreicher-Singer, Efron, & Shimshoni, 2013	X		X	Analyzing the influence of sentiment derived from forum data as well as the impact of Google Trends data on the prediction of sales of new cars and light trucks in America.
Geva, Oestreicher-Singer, Efron, & Shimshoni, 2015	X		X	Analyzing the influence of sentiment derived from forum data as well as the impact of Google Trends data on the prediction of sales of new cars and light trucks in America.
Ginsberg et al., 2009	X			Predicting influenza epidemics based on Google Trends data
Lassen, Madsen, & Vatrapu, 2014		X		Forecasting quarterly iPhone sales by making use of Twitter

Nymand-Andersen & Pantelidis, 2018	X			Approximating new car registrations in Europe by means of Google Trends data
Pai & Liu, 2018		X		Utilizing stock market values and sentiment scores of Twitter data to forecast car sales in the USA on a monthly basis
Santillana, Nsoesie, Mekaru, Scales, & Brownstein, 2014	X			Detecting influenza epidemics through Google Trends data.
Santillana et al., 2015	X	X	X	Predicting influenza epidemics via Google Trends data, Twitter data, Google Flu Trends data and macroeconomic variables
Seebach, Pahlke, & Beck, 2011	X			Predicting new car sales of Germany's two largest car producing companies utilizing Google search data
Tomczyk & Doligalski, 2015	X			Forecasting new vehicle registrations via a macroeconomic index and Google Trends index
Wu & Brynjolfsson, 2009	X			Forecasting house sales, the house price index and the demand for home appliances in the US by means of Google Trends data
Yu & Liu, 2012			X	Forecasting box-office revenues for movies by making use of sentiment extracted from IMDB-reviews

2.2 Natural language processing

2.2.1 Sentiment analysis

As most of the abovementioned literature conclude that the incorporation of sentiments extracted from social media improves the predictive power, further research concerning sentiment analysis is discussed below. Two main approaches to tackle sentiment analysis exist, namely lexicon-based techniques and machine learning methods (Dhaoui, Webster, & Tan, 2017; Gezici, Dehkharghani, Yanikoglu, Tapucu, & Saygin, 2013; Meire, Ballings, & Van den Poel, 2016; Mudinas, Zhang, & Levene, 2012; Ortigosa, Martín, & Carro, 2014; Zhang, Ghosh, Dekhil, Hsu, & Liu, 2011).

In case of the lexicon-based approach, a pretrained sentiment lexicon is utilized in order to give each word in the analyzed text a sentiment score (Ding, Liu, & Yu, 2008; Taboada, Brooke, Tofiloski, Voll & Stede, 2011). The given scores of all these words serve as input for a function that calculates the sentiment score of that text (Turney, 2002). A sentiment lexicon is a dictionary of words, in which each word is assigned a positive and a negative score (Bravo-Marquez, Mendoza, & Poblete, 2014). Alternatively, sentiment analysis can be approached by making use of machine learning methods, also called statistical methods

(Taboada et al., 2011). In this approach, models are trained on a classified training set. Since this approach is supervised, the training data has to be labeled. The input of these classifiers implemented to predict the sentiment of text (Taboada et al., 2011) consists of features extracted from text, such as unigrams, bigrams, part-of-speech, etc. (Pang, Lee, & Vaithyanathan, 2002).

The lexicon-based approach requires less time than the machine learning approach for the following reasons. Firstly, lexicon-based methods have the advantage of not needing an annotated training set (Ortigosa et al., 2014; Tan, Wang, & Cheng, 2008), whereas the training set of machine learning models is mostly labeled manually (Dhaoui et al., 2017). Manually labeling the training set is often time-consuming as it needs to be large enough to ensure a good classification accuracy (Dhaoui et al., 2017). Secondly, machine learning classifiers have to be trained before usage. Consequently, a significant amount of time is consumed (Chaovalit & Zhou, 2005).

In general, the machine learning approach provides better results in contrast to lexicon-based techniques in terms of accuracy (Chaovalit & Zhou, 2005). However, in less domain specific contexts, lexicon-based techniques seem to outperform machine learning techniques (Ortigosa et al., 2014). Furthermore, a possible alternative for manually labelling the training set in order to use supervised machine learning techniques, is to automatically label the data based on sentiment lexicons (Zhang et al., 2011; Tan et al., 2008). In recent research performed by Dhaoui et al. (2017), this hybrid approach outperformed both the lexical-based approach and the machine learning approach using manually labeled input data.

As far as my knowledge extends, researches that made use of sentiment analysis to predict car sales, implemented a lexicon-based approach.

3 Research questions

Social media already proved its usefulness in forecasting social and economic trends, at least in case of text-based social media (Hu et al., 2014). Sentiment-based features are prominent in predicting car sales by means of text-based social media. To the best of my knowledge, lexicon-based approaches are the only type of sentiment analysis methods utilized in the context of car sales, even though machine learning techniques generally seem to give more accurate results compared to lexicon-based techniques (Chaovalit & Zhou, 2005), especially in domain specific contexts (e.g. car industry) (Ortigosa et al., 2014). A potential explanation for this trend is the time efficiency of lexicon-based approaches, as it is not necessary to label the training sets (Ortigosa et al., 2014; Tan et al., 2008) and no machine learning classifiers are required to be trained prior to usage (Chaovalit & Zhou, 2005).

Image-based social media platforms such as Instagram might also contain useful information (Colliander & Marder, 2018; Hu et al., 2014; Pittman & Reich, 2016). Nonetheless, to the best of my knowledge, no research has investigated the predictive power of Instagram data to predict sales. Consequently, it may be interesting to investigate whether features extracted from the image-based social media platform Instagram serve as a useful input to forecast car sales. Hence, this leads to the following research question:

- Is it possible to predict car sales making use of Instagram features?

The literature review of this master dissertation (see section 2.1.2) concludes that, in general, search trends data is a valuable source of information in addition to traditional macro-economic variables to predict vehicle sales. The previous may be clarified by the following reasons. Firstly, search trends data is typically published earlier than traditional approximations, resulting in search queries data to be more recent (Santillana et al., 2015; Santillana et al., 2014). Secondly, online search data may provide a more rational view of the interests of customers than traditional economic variables (Wu & Brynjolfsson, 2009). However, the research of Barreira et al. (2013) illustrates that this is not always the case.

To determine whether the inclusion of search trends data leads to better forecasts of car sales, I would like to provide an answer to the following research question:

- Is it possible to predict car sales making use of Google Trends search data?

It can be derived from the literature review of this master dissertation (see chapter 2) that both search trends data and features extracted from text-based social media such as Twitter contain valuable information to predict sales. Consequently, it is not surprising that some researchers investigated the combination of these two types of data sources. Benthaus and Skodda (2015) verified that linear regressors using both Google Trends search data and features from Twitter as input, managed to forecast car sales more accurately compared to linear regressors having only one of these types of data as input.

Geva et al. (2013, 2015) came to the same conclusion concerning both a linear model and a nonlinear neural network model. Additionally, in the context of forecasting car sales, the authors found that search data and forum data incorporate a significantly equivalent predictive power. In summary, both Benthaus and Skodda (2015) and Geva et al. (2013, 2015) conclude that search queries data and social media data complement one another. This conclusion can be declared by the difference in nature of the two types of data (Geva et al., 2013).

Analogous to Benthaus and Skodda (2015) and Geva et al. (2013, 2015), I would like to investigate whether Instagram data and Google Trends data complement one another. As a consequence, the following research question is investigated:

- Is a higher predictive performance present when combining Instagram features and Google Trends data to predict car sales?

In order to provide an answer to the aforementioned research questions, I will predict the car sales of the most popular car models in the Netherlands in the months January to July of the year 2020.

4 Data

This chapter covers the three types of data that were utilized in this master dissertation to predict the monthly sales of car models in the Netherlands of the year 2020. The first section provides information about the car sales dataset, i.e. the dependent variable, that is used in this master dissertation. The second section gives an overview of the data scraped from Instagram along with its relevant variables. In the third section, the search trends data of Google, i.e. Google Trends data, is discussed. The fourth section is dedicated to the imputation methodology used to handle missing values.

4.1 Car sales

The first dataset consists of the dependent variable, i.e. the monthly car sales of the 200 most sold car models in the Netherlands of the year 2019. From my own experience, Belgian car sales are solely available at the car brand level and not at the car model level, as is the case for Dutch car sales. Consequently, I decided to make use of Dutch car sales instead of Belgian car sales. The monthly car sales of 2019 are used as training set to forecast the monthly car sales of 2020. In figure 1, the training set is depicted as the 'independent period', whereas the 'dependent period' covers the months that will be predicted based on the training set.

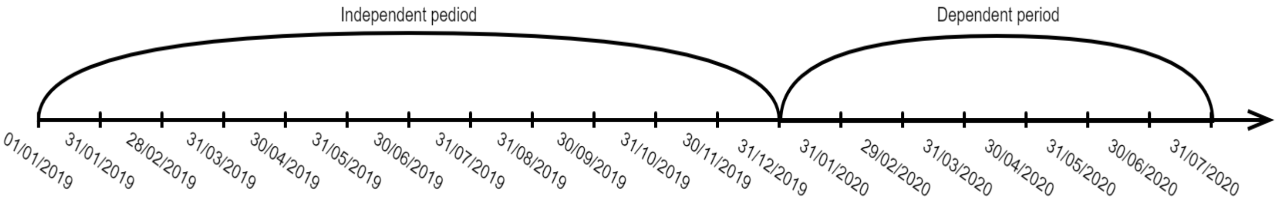


Figure 1. Time window

The car sales data is gathered from https://www.autoweek.nl/verkoopcijfers/, which provides the monthly sales of new vehicles in the Netherlands. An overview of the 200 car models along with their total sales of 2019 is provided in appendix 1.

4.2 Instagram data

As suggested by Highfield and Leaver (2015), the usage of hashtags on Twitter is the main approach to find tweets related to a certain topic. The authors also advise to make use of hashtags to collect Instagram data associated to a particular topic. However, the work of Giannoulakis and Tsapatsoulis (2015, 2016a) state that, on average, only a quarter of the hashtags are directly linked to the content of the image. Consequently, to ensure that all images are directly associated to the desired hashtag or topic, one can manually filter out the so called stophashtags (Giannoulakis & Tsapatsoulis, 2015, 2016a). As this solution is too time intensive, I decided to apply the following methodology, based on the proposition of Highfield and Leaver (2015).

For each car model in appendix 1, a hashtag is used to scrape public Instagram posts related to that car model. The selection procedure of the hashtags was performed by manually choosing the hashtags that seemed to be representing a car model the most. If multiple candidate hashtags were found, the hashtag containing most public posts was chosen. For instance, in case of Volkswagen Arteon, I found two candidate hashtags, i.e. '#arteon' and '#volkswagenarteon'. As depicted on figure 2, the hashtag '#arteon' contained 102.218 public posts at the moment of choosing the hashtags, while the hashtag '#volkswagenarteon' contained 13.381 public posts at that moment (see figure 3). Hence, the public Instagram posts containing the hashtag '#arteon' were selected to represent the public posts of the car model Volkswagen Arteon, as this hashtag was clearly more popular. Both hashtags, '#arteon' and '#volkswagenarteon', can be accessed respectively by the following Uniform Resource Locators *(URLs)*: https://www.instagram.com/explore/tags/volkswagenarteon/ and https://www.instagram.com/explore/tags/arteon/.

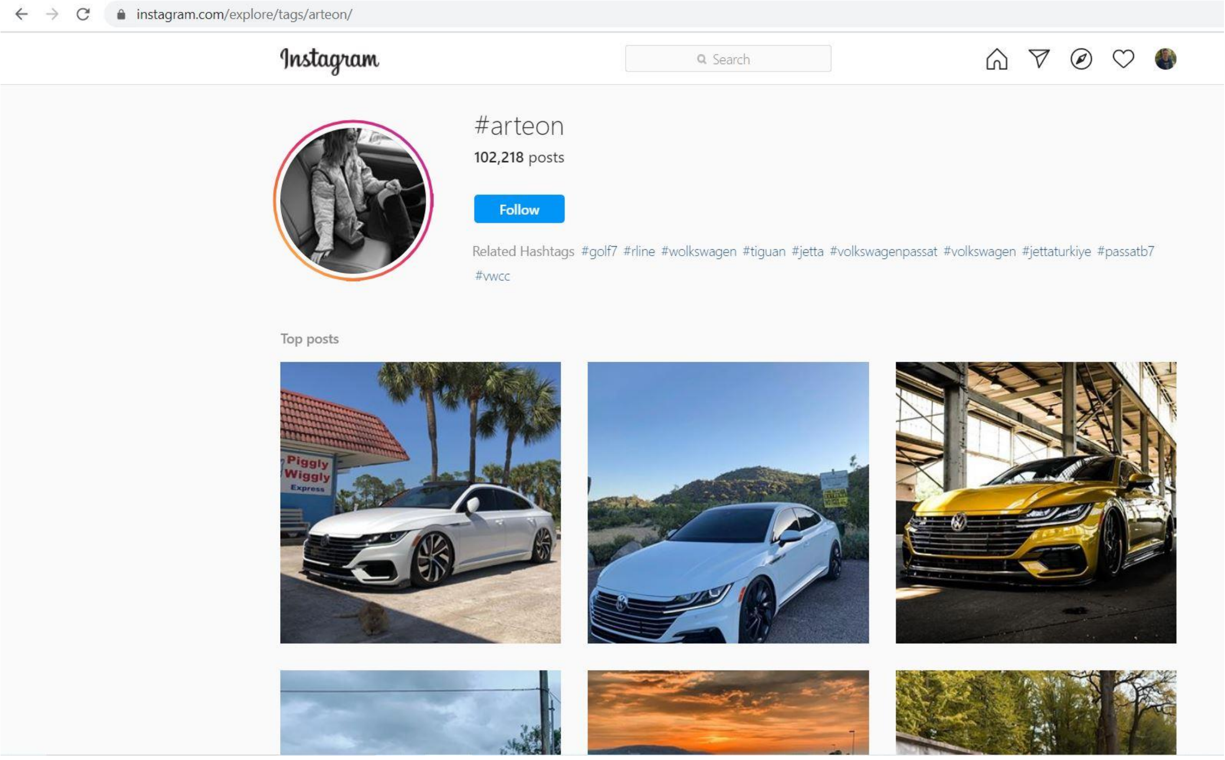

instagram.com/explore/tags/arteon/
Instagram
Search
#arteon
102,218 posts
Follow
Related Hashtags #golf7 #rline #wolkswagen #tiguan #jetta #volkswagenpassat #volkswagen #jettaturkiye #passatb7 #vwcc
Top posts
Piggly Wiggly Express

Figure 2. Public posts containing the hashtag '#arteon'

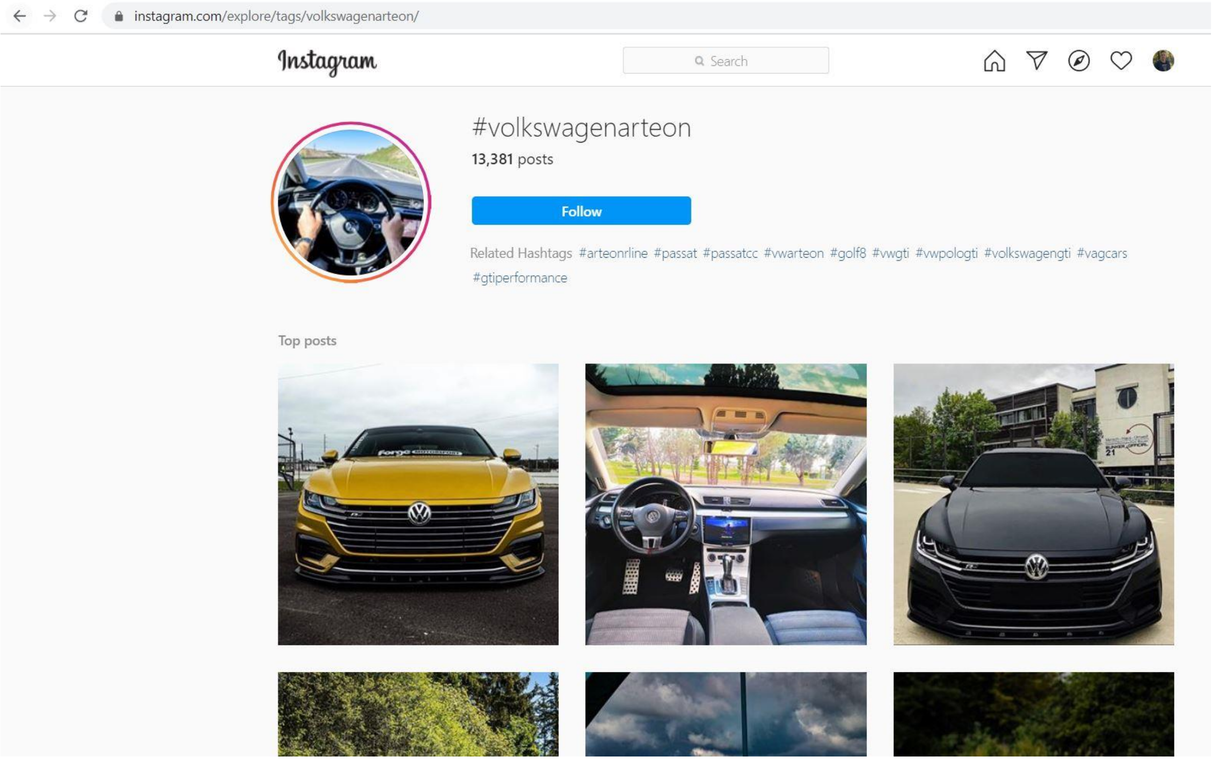

instagram.com/explore/tags/volkswagenarteon/
Instagram
Search
#volkswagenarteon
13,381 posts
Follow
Related Hashtags #arteonrline #passat #passatcc #vwarteon #golf8 #vwgti #vwpologti #volkswagengti #vagcars #gtiperformance
Top posts

Figure 3. Public posts containing the hashtag '#volkswagenarteon'

A table of the car models along with the hashtags used in this master dissertation to scrape the car related hashtags can be found in appendix 2.

The Instagram data was scraped by my promotor, Prof. dr. Dirk Van den Poel, who made use of the UGent Instagram application programming interface (*API)*. It is important to note that solely posts created in 2019 are retained from the scraped Instagram data to ensure that the training set only contains data from the year 2019.

4.2.1 Relevant variables

For each hashtag, the following features were extracted from the scraped Instagram data and aggregated on a monthly level: (1) the total number of likes, (2) the total number of comments, (3) the total number of posts, (4) the total number of videos, (5) the total amount of views of these videos and (6) the average valence of the posts' captions.

The choice of features is based on the research of Hoffman and Fodor (2010). Firstly, the number of likes on a post about a brand is an indicator of the *WOM* concerning that brand. Secondly, the number of comments on a post about a brand contributes towards the brand engagement. Thirdly, the total number of posts, the total number of videos, the total amount of views on these videos and the valence of posts' captions of a brand are features that help measure the brand awareness (Hoffman and Fodor, 2010).

In subsection 4.2.1.1, the principle of ex-ante forecasts as well as its relevance to features (1), (2) and (5) are explained. The tool used to compute the valence of posts' captions, i.e. Valence Aware Dictionary and sEntiment Reasoner (*VADER*) (Hutto & Gilbert, 2014), is described in subsection 4.2.1.2. Finally, the dataset of the Instagram features of the hashtag '#8series' is provided as an illustration in subsection 4.2.1.3.

4.2.1.1 Ex-ante forecasting

For each hashtag, I want to compute the abovementioned variables on a monthly basis. However, the point in time at which the Instagram posts are scraped has an influence on the number of likes, comments and views (in case the post contains a video). For instance, a post that is placed on January 2019 can have a different number of comments in the month January 2019 than in December 2019, i.e. the moment this post is scraped. In particular, the variables (1), (2) and (5) of posts created in the months previous to the month of scraping may contain information that was not yet available at the moment the posts were placed. Hence, forecasting car sales of February 2019 by means of the information related to a post placed in January 2019, which was scraped in December 2019, is called ex-post forecasting (Hyndman & Athanasopoulos, 2018). Consequently, to prevent this type of data leakage, only the information of a post available in the same month that the post was created needs to be retained as input to predict the future car sales (i.e. ex-ante forecasting) (Hyndman & Athanasopoulos, 2018).

In practice, I was only able to perform ex-ante forecasting for the number of comments. The reason behind this is that solely the date a comment was placed is available, while the date of a given like or a video view is unknown. In what follows, an example of this methodology is provided.

Table 2 represents a part of the information about the comments placed on a certain Instagram post that contains the hashtag '#arteon'. Each row represents a comment placed on the relevant Instagram post. Each comment or row has a unique *id* (i.e. column '*id*'). The column 'created_at' contains the timestamp a comment is placed, whereas the column 'text' stands for the content of the comment. The amount of likes a comment has gained at the time of scraping is represented by the column 'likes_count'. Finally, the column 'answers' represents the content of the comments that are placed as a reply on the relevant comment. As can be seen in table 2, none of the comments was answered.

The timestamp of the post itself equals 2019-05-14 22:51:54, indicating that the post is created at May 2019. Only the comments that are placed starting from the creation of the post until the end of May 2019 are considered to be known for the month May of the year 2019. Consequently, all comments of table 2 are retained except for the comment created at timestamp 2019-06-29 00:54:25. Note that this information is only available in case a post contains at least one comment.

Table 2. Comments placed on an Instagram post

id	created_at	text	likes_count	answers
17906182294326122	2019-06-29 00:54:25	your snapshot is really GREAT :)	1	[]
18032331400160622	2019-05-30 03:29:19	👍👍	1	[]
18043555216188961	2019-05-27 05:57:43	What an amazing shot! 😍 I think you might also like mine. 😉	1	[]
17847324658451259	2019-05-24 20:41:38	Absolutky Amazing 🔥🔥🔥🔥	1	[]
18046871983127800	2019-05-23 04:52:49	Hey Great Picture! 🔥👍	1	[]
17866744912385419	2019-05-23 04:08:18	Amazing!	1	[]
17891621935330411	2019-05-22 22:02:35	Jawdropping	1	[]
17865958162379067	2019-05-19 11:06:27	Love the content	1	[]
17850744163430639	2019-05-15 05:56:44	Nice!	2	[]

4.2.1.2 Valence

To be able to calculate the valence of a caption of an Instagram post, the lexicon-based and rule-based sentiment analysis tool *VADER* is utilized (Hutto & Gilbert, 2014). The lexicon consists of social media related features along with their valence or intensity, including a full list of Western-style emoticons, slang and acronyms. On top of that, *VADER* incorporates the following five heuristics that may influence the lexicon-based valence score. A first heuristic takes the presence of punctuations into account. For example, the inclusion of exclamation marks increases the absolute intensity value without changing the semantic orientation. Secondly, words in uppercase receive a higher absolute intensity value, with the semantic orientation remaining intact. Thirdly, if a sentence contains the conjunction ‘but’, it will receive the valence score of the words located after this conjunction. For instance, the sentence ‘This car’s acceleration is mind-blowing, but its fuel efficiency is bad.’ receives a negative valence score (i.e. the valence score of the sentence part ‘its fuel efficiency is bad.’). Fourthly, the presence of degree adverbs such as ‘very’ and ‘marginal’ are taken into account. The inclusion of the degree adverb ‘very’ in a sentence increases the absolute intensity value of the sentence, whereas the adverb ‘marginal’ decreases the sentence’s absolute intensity value. Fifthly, *VADER* analyzes each trigram (i.e. sequence of three words) prior to a sentiment-laden lexicon feature (i.e. word with a valence score different from zero). Consequently, *VADER* is able to detect 90 percent of the cases in which a negation changes the semantic orientation of a sentence (Hutto & Gilbert, 2014).

Practically, a *VADER* sentiment analysis Python module (Hutto & Gilbert, 2014) was used to calculate the compound score. The compound score is formed by taking the sum of the valence scores of all words and normalizing this sum into a number within a continuum ranging from –1 (i.e. extremely negative valence) to +1 (i.e. extremely positive valence). A sentence containing a compound score of zero means that the sentence does not have any valence. A positive compound score indicates a positive valence, while a negative compound score stands for a negative valence (Hutto & Gilbert, 2014). For example, the sentence ‘The car's comfort is really good.’, has a compound score of 0,69. When a negation is added to this sentence, for example ‘The car's comfort isn't really good.’, a compound score of -0,03 is assigned. In this case, *VADER* was able to detect the negation after the trigram ‘The car’s comfort’, which changes the semantic orientation of the sentence.

4.2.1.3 Illustration

Table 3 is an illustration of a dataset regarding the Instagram features extracted from the hashtag ‘#8series’. Hence, this dataset includes the Instagram features of the car model ‘BMW 8-serie’. As already mentioned in section 4.2.1, the features are aggregated on a monthly level. The variables ‘nr_likes’, ‘nr_comments’, ‘nr_posts’, ‘nr_videos’ and ‘video_view_count’ are aggregated by taking the monthly sum, whereas ‘polarity’ is aggregated by taking the monthly average.

Table 3. Instagram features of the hashtag ‘#8series’

date	nr_likes	nr_comments	nr_posts	nr_videos	video_view_count	polarity
2019-01-31	5.770.010	29.035	3.447	690	6.218.536	0,23178181
2019-02-28	6.912.360	31.791	3.726	676	7.304.917	0,230466989
2019-03-31	3.183.087	17.048	1.997	220	4.675.447	0,225629094
2019-04-30	3.250.977	15.582	1.673	163	2.137.230	0,214542558
2019-05-31	2.886.680	15.328	1.926	228	3.106.338	0,183578453
2019-06-30	4.803.574	22.182	2.110	206	3.579.117	0,229742228
2019-07-31	3.696.567	16.499	1.495	187	3.381.570	0,208475987
2019-08-31	4.087.685	16.220	1.569	200	2.685.594	0,21575443
2019-09-30	3.462.327	14.113	1.713	223	860.378	0,168648103
2019-10-31	3.426.845	15.467	1.807	268	1.993.449	0,227137576
2019-11-30	3.619.948	14.161	1.591	281	3.045.701	0,209352168
2019-12-31	3.433.225	12.837	1.698	407	3.164.254	0,247837868

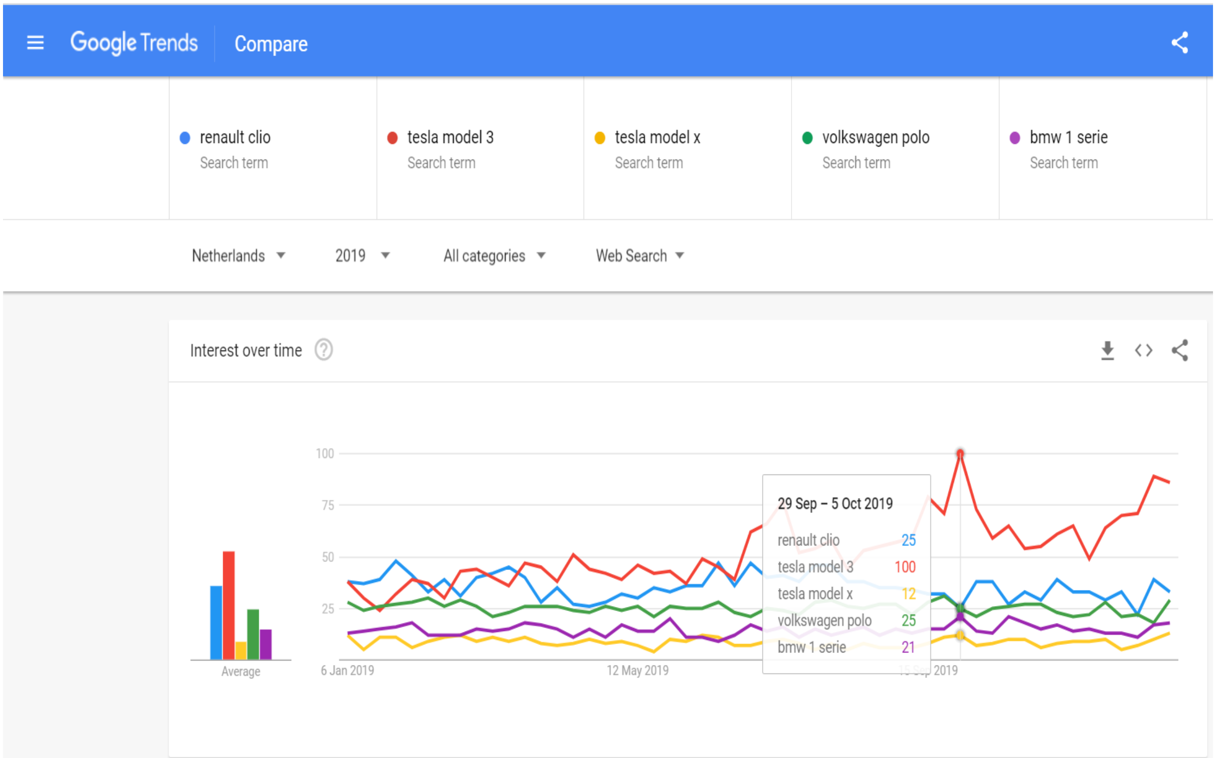
4.3 Search trends data

The majority of the literature in section 2.1 made use of data obtained from Google’s search engine. Furthermore, Google is the most popular search engine in the Netherlands, having a market share of almost 96 percent (de Best, 2020). Additionally, an *API* named ‘pytrends’ (General Mills, 2016) is available, making it possible to automatically extract the necessary Trends data (see subsection 4.3.2). Due to the beforementioned reasons, Google Trends data is an appropriate source of information and is therefore utilized in this master thesis.

4.3.1 Google Trends data

Google Trends data represents people's search interest in search queries over time (Rogers, 2016). Google makes it possible to easily extract this data for any period and desired region, starting from 2004. Moreover, search terms can be filtered based on different search categories such as 'Autos & Vehicles', 'Finance', 'Health', etcetera (Rogers, 2016).

To limit computational efforts, Google draws an unbiased sample of the full Google search dataset (Rogers, 2016). The yearly increase of search engine users makes it impossible to compare absolute search volumes at different points in time. On top of that, the absolute volumes also depend on the geographical region, which can be problematic when comparing searches in terms of location. To tackle these problems, Google normalizes the search volumes. More specifically, for a certain topic or search term, the absolute number of searches for that topic is divided by the total amount of searches on all topics in a specific region and time. In addition, Google scales the normalized data by giving each datapoint a number ranging from zero to 100. The datapoint with the largest search interest for the predefined time period and location is labeled as 100. Based on this datapoint, the other points receive a number in proportion to their relative search interest (Rogers, 2016).



Google Trends Compare

● renault clio — Search term
● tesla model 3 — Search term
● tesla model x — Search term
● volkswagen polo — Search term
● bmw 1 serie — Search term

Netherlands ▾ 2019 ▾ All categories ▾ Web Search ▾

Interest over time

29 Sep – 5 Oct 2019

renault clio	25
tesla model 3	100
tesla model x	12
volkswagen polo	25
bmw 1 serie	21

Figure 4. Google Trends time series

To clarify the interpretation of Google Trends data, figure 4 is shown, which can be consulted through the following *URL*: https://trends.google.com/trends/explore?date=2019-01-01%202019-12-31&geo=NL&q=renault%20clio,tesla%20model%203,tesla%20model%20x,volkswagen%20polo,bmw%20

1%20serie. As depicted on figure 4, the Google Trends data is provided on a weekly basis for the following five search terms: ‘renault clio’, ‘tesla model 3’, ‘tesla model x’, ‘volkswagen polo’ and ‘bmw 1 serie’. The time period is set between 01-01-2019 and 31-12-2019, whereas the region is set to the Netherlands. The search term ‘tesla model 3’ is increasing in popularity on average and reaches its popularity peak at the week of September 29th to October fifth in the year 2019. As this is the most popular search term out of the five terms for the mentioned period and region, it is indexed at a value of 100. During that week, the search terms ‘renault clio’ and ‘volkswagen polo’ each had an index value of 25, indicating that their popularity is only one fourth of the popularity of ‘tesla model 3’. After this peak, the index value of ‘tesla model 3’ decreases. This decline reflects a reduction in popularity regarding other search terms but does not necessarily indicate an absolute decrease in the search volume of the respective term (Google News Initiative, n.d.).

4.3.2 Data collection

I was able to automatically collect the Google Trends data for all car models mentioned in appendix 1 by means of an *API* named ‘pytrends’ (General Mills, 2016). The region is limited to the Netherlands and the period is set between 01-01-2019 and 31-12-2019.

The number of search terms that can be compared simultaneously for a given time period and region, is limited to five (Briggs, 2017). As only one search term is used for each car model, it is necessary to extract data for 200 terms. To deal with the limitation of only being able to compare five terms, Google Trends data needs to be scraped multiple times, for which each time different search terms are used. However, all extracted datasets need to have one search term in common to be able to contrast the different datasets. The common term can then be used as a reference to transform all datasets to the same scale (Briggs, 2017).

The selection of the search terms is based on the work of Mavragani and Ochoa (2019). Even though Google Trends is not case sensitive, it is sensitive to accent marks and spelling errors. Consequently, it is almost impossible to cover all searches related to a certain topic. To partly solve this problem, Mavragani and Ochoa (2019) suggest adding multiple search terms to one query by using the ‘+’ sign. For example, the car model ‘Renault Mégane’ can be searched under the search term ‘Renault Mégane’ as well as under the term ‘Renault Megane’, i.e. without accent mark. To obtain the search interest of both search terms, the search query ‘Renault Mégane+Renault Megane’ is utilized. Considering the previous, each search term consists of the following structure. On the one hand, if the car brand does not contain accent marks, the car brand is followed by the Dutch model name with each word separated by a single space. Additionally, each word is put in lowercase. On the other hand, if a car brand does contain accent marks, the same structure is applied for both the car model written with accent marks and the car model written without accent marks. The search query is then formed by including both search terms separated by a ‘+’ sign. For example, the search query for the car model ‘Renault Mégane’ is annotated as ‘renault mégane+renault megane’, whereas the search query for the car model ‘Volkswagen Arteon’ is represented by ‘volkswagen arteon’. A list of the car models with their corresponding search terms can be consulted in appendix 3.

Barreira et al. (2013) state that the use of queries without category constraints usually led to the best results in explaining and nowcasting car sales. However, to avoid ambiguity in the search terms, it is advised to limit the searches to a certain category (Mavragani & Ochoa, 2019). For instance, the word ‘jaguar’ can refer to the car brand ‘Jaguar’, or it can refer to an animal. Consequently, for this master dissertation, the search terms are filtered by the category ‘Autos & Vehicles’ (Wachter, Widmer, & Klein, 2019).

As mentioned in subsection 4.3.1, Google only provides a sample of the search volume population (Rogers, 2016). Consequently, downloading data from Google Trends multiple times results in the occurrence of a small variation in the index values of the extracted data for each download (Barreira et al., 2013; Carrière-Swallow & Labbé, 2013; Fantazzini & Toktamysova, 2015). Analogous to the aforementioned researchers, I took the average of the same data extracted at different moments in time to reduce this variance (Barreira et al., 2013; Carrière-Swallow & Labbé, 2013; Fantazzini & Toktamysova, 2015).

5 Methodology

In this chapter, the methodology applied in this master thesis is discussed. Firstly, the implementation of the clustering algorithm based on the car sales is described. Secondly, the different forecasting algorithms are explained, followed by the necessary data preparation and the execution of these algorithms. Thirdly, an overview of the utilized performance measures is provided. Fourthly, cross-validation applied in the context of time series is explained in more detail. Lastly, the statistical tests that have the purpose of determining significant differences in model performance are covered.

5.1 Clustering

I decided to cluster the car models by making use of the Taylor-Butina algorithm since this results in a decrease of computational complexity (Butina, 1999; Venkatesh, Ravi, Prinzie and Van den Poel, 2014). In addition, I preferred to utilize the Sequence-Alignment Method (*SAM*) as a distance measure, which serves as input for the Taylor-Butina algorithm. The reason behind this is the ability of *SAM* to deal with sequential information (Levenshtein, 1966). The approach applied to cluster the car models is based on the research of Venkatesh et al. (2014).

After clustering automated teller machines (*ATM*s) based on their similarity in day-of-the-week cash withdrawal patterns by implementing the Taylor-Butina algorithm, Venkatesh et al. (2014) forecasted the cash demand of *ATM* centers based on these clusters. Analogous to the work of Venkatesh et al. (2014), I grouped car models that have similar monthly sales patterns. A possible advantage of clustering the car models is the decrease in computational complexity, which in turn leads to a higher forecasting accuracy (Venkatesh et al., 2014).

In what follows, the subsequent aspects will be explained: (1) the steps executed prior to the implementation of the *SAM* (Levenshtein, 1966), which is utilized to determine the distance or similarity between the car models (see subsection 5.1.1), (2) *SAM* as well as its implementation (see subsection 5.1.2) and (3) the Taylor-Butina clustering algorithm, for which the *SAM* distances between the car models serve as input (Butina, 1999) (see subsection 5.1.3).

5.1.1 Sequence of twelve integer seasonality parameters

Firstly, for each car model, a multiplicative time series model $Y = T * S * C * I$ is fit, where *T* stands for the trend, *S* reflects the seasonality, *C* represents the cyclic movement and *I* serves as the time series' irregular part of the model (Venkatesh et al., 2014). I made the assumption that the data contains neither cycles nor irregular components. To evaluate whether the previous assumption holds true, the car sales data of the formed cluster centroids are plotted in time. All plots can be consulted in appendix 4. Based on these plots, it is clear that solely a fixed frequency is present in the fluctuations of the car sales data. As a result, only a seasonality is included in the data (Hyndman & Athanasopoulos, 2018). Hence, my assumption regarding the absence of cycles and irregular components in the data can be verified for all cluster centroids. Furthermore, an ordinary least squares (*OLS*) regression model is fit on the time series data to estimate the trend (Venkatesh et al., 2014). Afterwards, the seasonality part is acquired by dividing the time series data by its trend (Venkatesh et al., 2014).

In the second step, multiple years of car sales data are needed. As a consequence, I made use of data that comprises monthly car sales in the period from 01-01-2016 to 31-12-2019 (i.e. the sales of each car model are represented by a sequence of 48 monthly observations). However, in case of 54 out of the 200 car models, the sales data starting from the year 2016 was absent. The reason behind this is that either the car models were not yet available for sale in 2016 or that none of the car models were bought yet in the Netherlands. As a result, I decided to predict the car sales for the 146 car models that do have available data starting from 01-01-2016. Similar to Venkatesh et al. (2014), the 48 monthly observations for each car model are substituted by a sequence of twelve “month of the year” seasonality parameters. This is accomplished by averaging the seasonal values of each month, e.g. for the month January, the average seasonality of observation one, thirteen, twenty-five and thirty-seven is calculated. Next, each car model’s sequence of twelve continuous seasonality parameters is discretized to ease the clustering process. For each car model, the quartiles of the seasonality values are computed for each month of the year. Subsequently, each “month of the year” seasonality parameter is replaced by the number of the quartile it belongs to (Venkatesh et al., 2014). An example of discretizing the twelve continuous seasonality parameters of the car model Audi A1 is given in table 4.

Table 4. Discretizing the monthly seasonality parameters of car model Audi A1

Month	1st quartile	2nd quartile	3rd quartile	4th quartile	continuous seasonality	discrete seasonality
January	<1,51	1,51-1,54	1,54-1,61	>1,61	1,58	3
February	<0,82	0,82-0,83	0,83-0,94	>0,94	0,93	3
March	<0,99	0,99-1,11	1,11-1,17	>1,17	1,05	2
April	<0,84	0,84-0,91	0,91-0,99	>0,99	0,93	3
May	<0,75	0,75-0,87	0,87-1,03	>1,03	0,90	3
June	<1,02	1,02-1,07	1,07-1,14	>1,14	1,09	3
July	<1,09	1,09-1,13	1,13-1,18	>1,18	1,14	3
August	<0,92	0,92-1,17	1,17-1,44	>1,44	1,18	3
September	<0,59	0,59-0,67	0,67-0,79	>0,79	0,71	3

October	<0,91	0,91-1,11	1,11-1,26	>1,26	1,06	2
November	<0,90	0,90-1,02	1,02-1,26	>1,26	1,15	3
December	<0,18	0,18-0,20	0,20-0,31	>0,31	0,28	3

5.1.2 *SAM*

In the third step, Venkatesh et al. (2014) determined the similarity or distance between each pair of *ATM*s by means of the *SAM*, which was introduced by Levenshtein (1966). The original *SAM* determines the distance between two strings (i.e. the source string and the target string) as the minimum amount of operations (i.e. insertions, deletions and replacements) needed to transform the source string into the target string (Venkatesh et al., 2014). A benefit of *SAM* is its ability to deal with sequences of different lengths as well as to consider sequential information. Hence, the similarity between each pair of car models is illustrated by the minimum number of operations required to align their “month of the year” sequences with one another (Venkatesh et al., 2014). This implies that the original *SAM* can only take positive integer values. To further illustrate, the *SAM* distance between the car models Audi A1 and Alfa Romeo MiTo is provided in table 5.

Table 5. Discretized sequences of seasonality parameters

Car model	Discretized sequence of seasonality parameters
Audi A1 (Target)	3 3 2 3 3 3 3 3 3 2 3 3
Alfa Romeo MiTo (Source)	2 2 2 3 3 3 2 3 2 2 2 2

As shown in table 5, the distance between Audi A1 and Alfa Romeo MiTo amounts six since the six underlined numbers need to be replaced to align the sequences with one another. This distance is calculated for each pair of car models. All pairwise *SAM* distances of the car models are represented by a distance or dissimilarity matrix. In practice, the distance is calculated by means of the distance module of the Levenshtein package (Necas, 2014).

5.1.3 Taylor-Butina algorithm

Lastly, similar to Venkatesh et al. (2014), I utilized the Taylor-Butina clustering algorithm (Butina, 1999). In what follows, the algorithm is explained. Firstly, a threshold based nearest-neighbor table is created by means of the previously formed distance matrix (see subsection 5.1.2). This table or matrix contains one row for each car model. Each row includes the neighbors of the car model represented by that row. Two car models are considered neighbors of one another when their relative distance is below the predefined distance threshold. Secondly, all empty rows are withheld from the table as these rows represent the car models without any neighbor, based on the imposed distance threshold. These car models are labeled as true singletons since they form a cluster on their own. Thirdly, the car model containing most neighbors for the predefined threshold is considered as a cluster centroid. As a result, this datapoint and its neighbors form a cluster. Fourthly, this row along with the datapoints belonging to this cluster are erased from all rows of the nearest-neighbor table. The third and fourth step are repeated on the updated nearest-neighbor table until the table consists of solely empty rows. Any leftover rows are labeled as false singletons. Even though false singletons have neighbors at the current distance threshold, all neighbors have been removed earlier by other datapoints having a larger number of neighbors. Hence, false singletons join the cluster of its nearest neighbor (Butina, 1999; Venkatesh et al., 2014).

The statistical quality of this clustering algorithm can be derived as the number of cases the standard deviation for the monthly seasonality parameters within the clusters is lower than the standard deviation for the respective monthly seasonality parameters of the full dataset (see subsection 5.1.1) (Venkatesh et al., 2014).

Practically, I made use of the Butina module from the rdkit.ML.Cluster package (Landrum, 2020). The hyperparameter to be tuned is the distance threshold of the threshold based nearest-neighbor table. I decided to determine the optimal distance threshold based on the aforementioned statistical quality of the Taylor-Butina algorithm. More specifically, I ran the clustering algorithm multiple times, for which each time a different distance threshold is used. In this research, the optimal distance threshold should be an integer value between one and five because of the following reasons. As mentioned in subsection 5.1.2, the distance used in this research (i.e. the original *SAM* distance) can only take positive integer values. In addition, a distance threshold of zero causes each car model to be a true singleton since no car model seems to have the same sequence of seasonality parameters. On top of that, the maximum distance that occurred in the distance matrix is equal to five. Hence, in case the distance threshold lies above five, every car model would be a neighbor of one another, resulting into one cluster.

A distance threshold of five leads to three clusters. For two of these clusters, the standard deviations of all twelve seasonality parameters are higher than for the full sample of car models. The remaining cluster has a lower standard deviation for all twelve seasonality parameters compared to the full sample. Consequently, for twelve out of 36 months or in 33,33 percent of the cases, the standard deviation within the clusters is lower than the standard deviation of the full dataset. The process of calculating the statistical quality is executed for all candidate distance thresholds. An overview of the thresholds along with their previously described statistical quality (denoted as ‘Percentage True’) is depicted in figure 5. The optimal distance threshold equals three, having a ‘Percentage True’ or statistical quality of 77,78 percent. Based on this distance threshold, twelve clusters are formed. The distribution of the formed clusters can be found in appendix 5. In appendix 5, the first element of each cluster is the cluster's centroid.

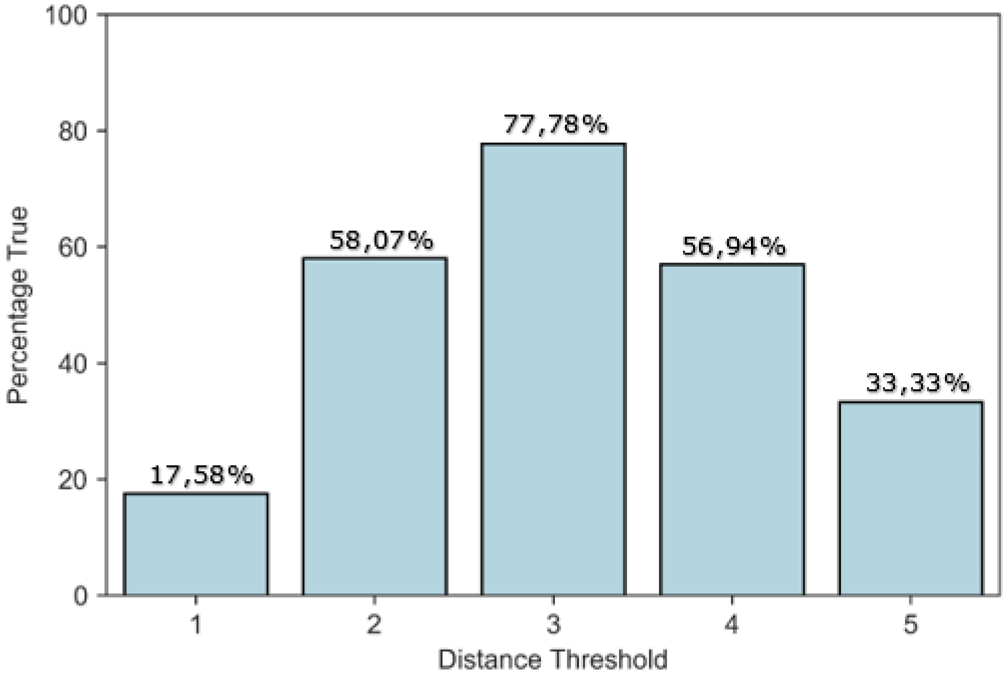


Figure 5. Optimal distance threshold

5.2 Forecasting

After clustering the car models in section 5.1, the car sales of these car models are forecasted making use of two popular forecasting algorithms, i.e. *ARIMA* (Box & Jenkins, 1970) and Long Short-Term Memory (*LSTM*) (Hochreiter & Schmidhuber, 1997) (see subsection 5.2.1). To be able to provide an answer to the research questions (see chapter 3), the forecasting algorithms need to predict the monthly car sales of 2020 in the Netherlands multiple times, each time having a different dataset as input. More precisely, the four datasets in table 6 are utilized to predict the car sales of 2020. Dataset 1 only contains the car sales of 2019, whereas dataset 2 comprises the car sales of 2019 as well as the search trends of 2019. In addition, dataset 3 consists of the car sales of 2019 and the Instagram features (i.e. the total number of likes, the total number of comments, the total number of posts, the total number of videos, the total amount of views of these videos and the average valence of the posts' captions). Lastly, dataset 4 incorporates the car sales of 2019, the search trends of 2019 and the Instagram features of 2019.

Table 6. Datasets used to predict car sales of 2020

Dataset\Feature	car sales 2019	search trends 2019	Instagram features 2019
1	x		
2	x	x	
3	x		x
4	x	x	x

To determine which forecasting algorithm to use, the car sales of 2020 will be predicted by an *ARIMA* model as well as an *LSTM* model with both dataset 4 as input. The model with the best predictive power in terms of root mean squared error (*RMSE*) and mean absolute error (*MAE*) (see section 5.3) is then utilized to forecast the car sales of 2020 three more times, each time with one of the three remaining datasets (i.e. dataset 1, dataset 2 and dataset 3) as input. Note that the algorithms are trained to forecast the car sales one month ahead.

Due to time limitations and limitations in computational power, the models are not trained on each of the 146 car models but are fit on each of the twelve cluster centroids. Next, for each car model, the car sales of 2020 are predicted by using the model trained on the cluster centroid that represents the cluster to which the car model belongs. To summarize, five different models will be used to forecast the car sales of the Netherlands of 2020 (see table 7).

Table 7. Overview models to forecast

Forecasting algorithm	**Dataset**
ARIMA	Dataset 4
LSTM	Dataset 4
ARIMA or *LSTM*	Dataset 1
ARIMA or *LSTM*	Dataset 2
ARIMA or *LSTM*	Dataset 3

In what follows, subsection 5.2.1 covers the concepts of the *ARIMA* forecasting algorithm and the *LSTM* forecasting algorithm. Subsection 5.2.2 is devoted to the steps needed to prepare dataset 2, 3 and 4 of each car model to make forecasts. In subsection 5.2.3 and 5.2.4, the methodology applied to implement the *ARIMA* model and *the LSTM* model is explained respectively.

5.2.1 Forecasting algorithms

In this subsection, two regression models are discussed. The first model, *ARIMA* (Box & Jenkins, 1970), represents a classical forecasting model. The second algorithm is denoted as *LSTM* (Hochreiter & Schmidhuber, 1997), which has been utilized in a wide range of domains such as speech recognition, natural language processing and time series (Cao, Li, & Li, 2019; Graves, Mohamed, & Hinton, 2013).

5.2.1.1 ARIMA

ARIMA (Box & Jenkins, 1970) models are one of the most popular and most used models to forecast time series (Siami-Namini & Namin, 2018). These types of models are *AR* models, meaning that the forecasts of a variable are based on the past values of that variable. In addition, financial and economic time series are typically not stationary (i.e. their statistical properties are not constant over time) (Ogasawara et al., 2010). To make time series stationary, the sequential observations are subtracted from each other, which is called differencing (Hyndman & Athanasopoulos, 2018). On top of that, an *ARIMA* model also takes past error terms of the model into account by including a Moving Average (*MA)* model.

A commonly used notation is *ARIMA(p,d,q)*, where *p* stands for the number of lags considered by the *AR* part of the model, *d* implies the order of differencing and *q* represents the order of the *MA* part (Helmini, Jihan, Jayasinghe, & Perera, 2019; Siami-Namini & Namin, 2018). This model assumes that the time series data is non-seasonal. To handle seasonal time series data, a seasonal *ARIMA(p,d,q)x(P,D,Q)S* is preferred. The lowercase parameters *p*, *d* and *q* have the same meaning as the parameters of the non-seasonal *ARIMA* model and are therefore called non-seasonal parameters. The seasonal parameters *P*, *D* and *Q* are equivalent to their non-seasonal counterparts, except for the lags being used. More precisely, the lags for the seasonal parameters are multiples of the seasonality length or period *S* (Siami-Namini & Namin, 2018).

Despite its popularity, *ARIMA* has difficulties to capture non-linear relationships between variables as the algorithm is based on linear regression (Siami-Namini & Namin, 2018).

5.2.1.2 LSTM

LSTM was first proposed by Hochreiter and Schmidhuber (1997) and updated over the years by several authors (Bayer, Wierstra, Togelius, & Schmidhuber, 2009; Graves, 2013; Schmidhuber, Wierstra, Gagliolo, & Gomez, 2007). In order to clarify the concept of *LSTM*, artificial neural networks (*ANN*s) as well as recurrent neural networks (*RNN*s) are first explained respectively.

*ANN*s are based on the biological neural network situated in our brain (Kalogirou, 2000). This network consists of interconnected biological neurons that transmit information to and receive information from one another. Analogous to the biological neural network, *ANN*s consist of artificial neurons that are connected to each other. A typical *ANN*, i.e. a multilayer feed-forward neural network, is illustrated in figure 6. The neurons are visualized as nodes and the arrows serve as an information flow between the neurons. The network consists of three types of layers, namely the input layer, the hidden layer and the output layer, with each layer defined as a collection of neurons. The input features of an observation are fed into the neurons of the input layer, which in turn extract information that seems relevant and transfer these features to the neurons of the next layer (i.e. the first hidden layer). In other words, the output of neurons belonging to a particular layer acts as input of the neurons of the subsequent layer. All layers situated between the input layer and the output layer are called hidden layers. The neurons of the output layer represent the predictions of the neural network (Kalogirou, 2000).

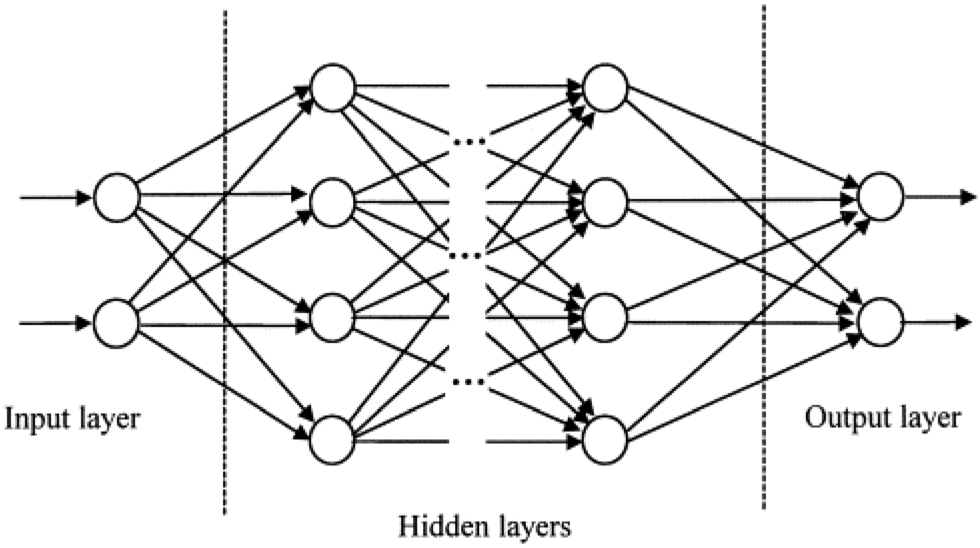


Figure 6. Multilayer feed-forward neural network

Note. Reprinted from "Applications of artificial neural-networks for energy systems", by Kalogirou, S.A., 2000, *Applied Energy, 67,* 21. Copyright 2000 by Elsevier Inc.

For each neuron, the importance of each input feature *X_t* is determined by the weights *W_{ij}* assigned to that feature (Kalogirou, 2000). In particular, as can be seen in figure 7, the following steps take place. First, each incoming feature *X_j* is multiplied by a weight *W_{ij}*, where features that are considered to be more important are assigned a higher weight. Next, the sum of these weighted values is computed. This weighted sum serves as input of an activation function, whose result determines the neuron's output (Kalogirou, 2000).

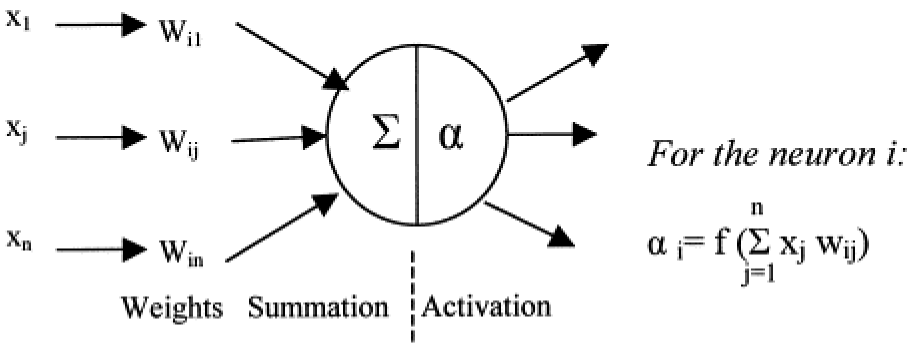


Figure 7. Artificial neuron

Note. Reprinted from "Applications of artificial neural-networks for energy systems", by Kalogirou, S.A., 2000, *Applied Energy, 67,* 21. Copyright 2000 by Elsevier Inc.

The learning phase of *ANN*s starts by initializing the weights (Kelleher, 2019). Next, for each observation of the training set, the features are fed into the network and outputs are generated. These outputs are contrasted to the actual values or labels to determine the error of the model. Based on the error, the weights are updated to minimize a certain cost function. A popular algorithm to update the weights is backpropagation (Rumelhart, Hinton, & Williams, 1986). The backpropagation algorithm works as follows. First, the gradient of the cost function (i.e. the local steepest slope) is calculated. Next, the weights of the output layer are updated in the opposite direction of the gradient (LeCun et al., 2015). The magnitude of the weights' update depends on the learning rate (Murphy, 2012). After adjusting the weights of the output layer, the weights of the previous layer are adjusted by making use of the chain rule for derivatives. This process is iterated for the previous layers of the neural network, all the way back to the input layer (LeCun et al., 2015). Once all weights of all layers are updated, the features are inserted again into the model and a new error is calculated based on the generated outputs for each observation. This whole learning process is repeated until a certain condition, such as reaching a minimum in the cost function, is met (Kelleher, 2019). The moment the weights are updated can vary (Wilson & Martinez, 2003). During batch training, the weight changes take place after the model has seen the whole training set (i.e. after one epoch). In stochastic gradient descent, the weights are adapted after each training example (i.e. after one instance). Alternatively, the weights are adapted after a set of instances. This process is called mini-batch training (Wilson & Martinez, 2003). When the neural network model is trained, the final weights are used to test the model's performance on unseen data (Kalogirou, 2000). This very popular learning algorithm for *ANN*s is called gradient descent (Nielsen, 2015).

*RNN*s are neural networks specialized in handling sequential data (Kelleher, 2019). An *RNN* processes the sequential data one by one, taking into account information from previous observations of the sequence. In other words, RNNs are capable of memorizing information of previously seen datapoints. This is realized by using the output of the previous observation along with the features of the current observation, as input of the current observation (Kelleher, 2019). However, in general, *RNN*s struggle to remember long sequences of data (Siami-Namini & Namin, 2018) due to the problem of vanishing gradients (Bengio, Simard, & Frasconi, 1994). This problem states that the deeper the neural network, the closer the gradients of the loss function approximate zero. Consequently, training the network becomes less straightforward. The problem of vanishing gradients can be explained by activation functions, such as the sigmoid function, that transform the input space into a value between zero and one, making its derivative small as well (Wang, 2019).

Due to this problem, Hochreiter and Schmidhuber (1997) proposed *LSTM* as a solution. An *LSTM* is a special type of *RNN* that outperforms other *RNN* models in terms of dealing with long range dependencies (Graves, 2013). Instead of sequentially remembering all information of previous observations, an *LSTM* is able to select which data from the past to convey and which to forget (Siami-Namini & Namin, 2018). This is accomplished by means of three types of gates, namely the forget gate, the memory gate and the output gate. The forget gate decides which information of the cell state (i.e. representation of all the remembered information of past observations and the current input) will be thrown away. This gate makes use of a sigmoid function that outputs values between zero and one, with zero meaning that the forget gate will fully forget the cell state and one causing the gate to remember the whole state of the cell. The second gate, the memory gate, decides which new information will be stored in the cell state. Finally, the output gate determines which part of the cell state will serve as output (Siami-Namini & Namin, 2018). The output of the *LSTM* model is also called the hidden state (Phi, 2018). In terms of predicting time series, *LSTM* is robust against outliers and change points (Guo et al., 2016). On top of that, in contrast to *ARIMA*, deep learning models such as *LSTM* models are able to capture non-linear patterns as well as other complex relationships in the data (Siami-Namini & Namin, 2018). In general, *LSTM* seems an appropriate algorithm for the prediction of time series (Cao et al., 2019).

5.2.2 Data preparation

5.2.2.1 Missing values

The Instagram dataset as well as the car sales dataset contain missing values for some car models. In case of the car sales dataset, a missing value indicates that zero new vehicles of that car model are sold during that month. Consequently, these missing values are imputed by zero. Similarly, the missing values of the Instagram dataset are imputed by zero as well, since this implies that there are no posts placed during that month for a certain hashtag.

5.2.2.2 Multicollinearity

An important assumption of regression is the absence of multicollinearity (Daoud, 2017). Multicollinearity indicates a linear relation between at least two independent variables (Alin, 2010). Violating this assumption results in an unreliable regression model (Daoud, 2017). In order to identify multicollinearity, multiple techniques can be applied. First of all, the pairwise correlation between predictors may give an indication of multicollinearity (Alin, 2010; Daoud, 2017). A large correlation between predictors implies multicollinearity. However, multicollinearity does not always mean a high correlation, since multicollinearity can also exist in case of a low correlation (Alin, 2010). On top of that, a common threshold to indicate whether a correlation is considered as small or high does not exist (Daoud, 2017). Because of these drawbacks, a more popular approach, i.e. variance inflation factor (*VIF*), is utilized. The *VIF* of a predictor *i* is formulated as follows (Alin, 2010; Daoud, 2017):

$$\mathrm{VIF_i} = \frac{1}{1 - \mathrm{R_i^2}} \text{ for } \mathrm{i} = 1,2,\dots,\mathrm{k}$$

Where R_i^2 stands for the coefficient of multiple determination of predictor *i* on the other *k-1* predictors and *k* represents the number of predictors.

A *VIF* of a predictor is considered large if it is above ten (Alin, 2010). Moreover, the average *VIF* also indicates whether multicollinearity is present or not. If the average *VIF* is above one, there is a high probability of multicollinearity (Alin, 2010).

In this master thesis, I decided to analyze whether multicollinearity is present in the Instagram features by looking at the correlation as well as the *VIF* of the twelve cluster centroids.

The *VIF* of the twelve car models can be consulted in appendix 6. Based on the average *VIF*, I observed that for each cluster centroid, a high probability of multicollinearity is present between the Instagram-related variables.

Based on the correlation matrices of all twelve cluster centroids in appendix 7, I concluded that for ten centroids all Instagram-related variables, with an exception of polarity, are highly correlated to one another (i.e. a correlation of one or close to one). In case of the cluster centroid 'Land Rover Range Rover', there is a high correlation between the variable 'nr_comments' and the variable 'polarity' and a rather low correlation between the variable 'nr_comments' and the other variables. In case of the remaining cluster centroid 'Land Rover Range Rover Evoque', the variable 'polarity' is highly correlated with the other Instagram-related variables. Nonetheless, the correlation between all variables, except for 'polarity', is still higher.

As in most cases all variables, except for polarity, are highly correlated, I decided to retain the variable polarity. On top of that, as proposed by Daoud (2017), multicollinearity is resolved by only keeping one of the highly correlated variables. Consequently, the variable 'nr_comments' is retained as this solves the problem of multicollinearity as well as the problem of ex-ante forecasting mentioned in subsection 4.2.1.1.

5.2.2.3 *Prediction of the external regressors*

The observations from January 2019 to December 2019 are utilized to train the *ARIMA* model and the *LSTM* model. However, in order to predict the monthly car sales of 2020 by means of dataset 2, dataset 3, and dataset 4 (see table 6), the observations of the external regressors (i.e. the Google Trends data and the data extracted from the Instagram hashtags) of 2020 are also needed as input. As the purpose of this master thesis is to solely use data of 2019, the external regressors for the desired months of 2020 need to be predicted. More precisely, for each external regressor, the observations of the months January, February, March, April, May, June and July of 2020 are forecasted by an *ARIMA* model, using the observations from 2019 of the relevant external regressor as input.

An overview of this dataset with the three types of data is represented in table 8. The first twelve observations of table 8, i.e. from t=1 until t =12, serve as training set. The test set is represented by the seven last rows of table 8, i.e. from t=13 until t=19.

Table 8. Dataset with the three types of data

Date	**car sales t**	**Google Trends t**	**Instagram features t**
31/01/2019 (t=1)	car sales 1	Google Trends 1	IG features 1
28/02/2019 (t=2)	car sales 2	Google Trends 2	IG features 2
31/03/2019 (t=3)	car sales 3	Google Trends 3	IG features 3
30/04/2019 (t=4)	car sales 4	Google Trends 4	IG features 4
31/05/2019 (t=5)	car sales 5	Google Trends 5	IG features 5
30/06/2019 (t=6)	car sales 6	Google Trends 6	IG features 6
31/07/2019 (t=7)	car sales 7	Google Trends 7	IG features 7
31/08/2019 (t=8)	car sales 8	Google Trends 8	IG features 8
30/09/2019 (t=9)	car sales 9	Google Trends 9	IG features 9
31/10/2019 (t=10)	car sales 10	Google Trends 10	IG features 10
30/11/2019 (t=11)	car sales 11	Google Trends 11	IG features 11
31/12/2019 (t=12)	car sales 12	Google Trends 12	IG features 12
31/01/2020 (t=13)	Predicted car sales 13	Predicted Google Trends 13	Predicted IG features 13

29/02/2020 (t=14)	Predicted car sales 14	Predicted Google Trends 14	Predicted IG features 14
31/03/2020 (t=15)	Predicted car sales 15	Predicted Google Trends 15	Predicted IG features 15
30/04/2020 (t=16)	Predicted car sales 16	Predicted Google Trends 16	Predicted IG features 16
31/05/2020 (t=17)	Predicted car sales 17	Predicted Google Trends 17	Predicted IG features 17
30/06/2020 (t=18)	Predicted car sales 18	Predicted Google Trends 18	Predicted IG features 18
31/07/2020 (t=19)	Predicted car sales 19	Predicted Google Trends 19	Predicted IG features 19

5.2.2.4 Scaling variables

It is advised to put the values of the different input variables at the same scale when training *LSTM* models, especially when the magnitude of the variables significantly differ from one another (Helmini et al., 2019). On top of that, transforming the input variables at the same scale forces the forecasting algorithms to treat each input variable with equal importance (Bollen et al., 2011). Consequently, the same scaled input features that serve as input for the *LSTM* model are also utilized as input for the *ARIMA* model.

The features used in this work differ in scale, since the valence ranges between minus one and plus one while the monthly number of comments on the post of a hashtag can take values ranging from zero to values having an order of a magnitude of five. Hence, the values of the input features are scaled between zero and one by applying the following function on each input feature:

$$y_t = \frac{(x_t - min)}{(max - min)}$$

where y_t represents the rescaled value of an input feature at time t, x_t represents the original value of an input feature at time t, and min and max represent the lower bound and the upper bound of the input variable respectively.

On the one hand, the bounds of the search trends index and the polarity are fixed in time, i.e. the search trends index ranges between zero and 100 whereas polarity can take values between minus one and plus one (see chapter 4). On the other hand, the remaining input features only have a lower bound of zero that is known a priori. For these features, the upper bound is set equal to its maximum value present in the training set.

The scaling is realized by making use of the scikit-learn object MinMaxScaler (“sklearn.preprocessing.MinMaxScaler”, 2018).

5.2.3 Implementation *LSTM*

In practice, the *LSTM* algorithm was employed by making use of the Keras (Chollet et al., 2015) library with TensorFlow (Abadi et al., 2016) as backend engine. In this section, the architecture of the *LSTM* model that is used to predict car sales will be discussed first (see subsection 5.2.3.1). Next, the *LSTM* model is optimized by tuning its hyperparameters (see subsection 5.2.3.2).

5.2.3.1 LSTM network architecture

The *LSTM* model utilized in this master thesis is a Vanilla *LSTM* (Brownlee, 2018). This consists of an input layer, an *LSTM* layer and an output layer which outputs the predicted number of cars sold in a certain month (Brownlee, 2018).

5.2.3.2 Hyperparameter tuning

To optimize the *LSTM* model, its hyperparameters need to be tuned. This can be done manually, but analogous to the work of Helmini et al. (2019), I decided to automate the finetuning of the model since an *LSTM* model has a lot of hyperparameters to be tuned. More specifically, the automation of the hyperparameter tuning is performed by means of a grid search algorithm. The validation of this algorithm is implemented by a time series cross-validation with ten splits (see section 5.4). The optimal values of the hyperparameters are determined by minimizing the mean squared error (*MSE*) on the validation set. After the optimal values are found, the model is trained on the full training set using these optimal values. The tuned hyperparameters along with its search space can be found in in table 9. In the next subsections (see subsection 5.2.3.2.1 – subsection 5.2.3.2.5), some important hyperparameters of *LSTM* are discussed.

Table 9. Hyperparameters and search space *LSTM*

Hyperparameter	Search space
Number of epochs	{5}
Learning rate	{0.0001,0.001, 0.01, 0.1, 0.2, 0.3}
Batch size	{1,2,3,4,5,6,7,8,9,10,11,12}
LSTM size	{8,16,32,64,128}
activation function of *LSTM*	{ReLU}
Optimization algorithm	{adam}

The optimal values of the hyperparameters for each cluster centroid using dataset 1, 2, 3 and 4 can be found in appendix 8 to appendix 11 respectively.

5.2.3.2.1 Learning rate

As already mentioned in subsection 5.2.1.2, the learning rate determines the extent in which the weights of the loss function are updated (Murphy, 2012). The size of the learning rate has an influence on the convergence towards the minimum of the loss function. Suppose the learning rate is a constant. If the learning rate is too small, a slow learning process takes place. If the learning rate is too high, the learning algorithm will take too large steps, which causes the algorithm to 'jump' over the minimum of the loss function. Consequently, the algorithm will never converge towards the minimum (Murphy, 2012). The effect of the learning rate on the convergence towards the minimum is depicted by figure 8. The horizontal axis represents the value of weight θ whereas J(θ) stands for the loss function.

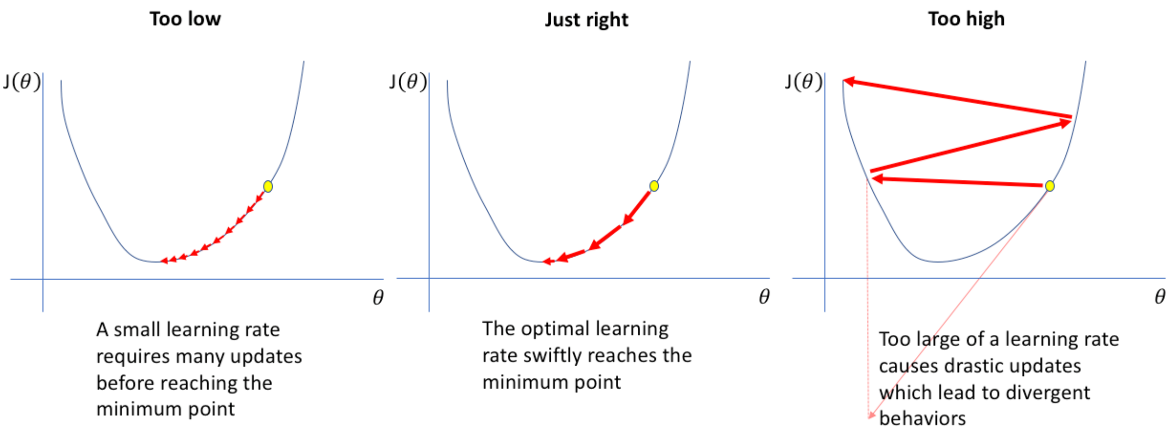

Too low
Just right
Too high
J(θ)
J(θ)
J(θ)
θ
θ
θ
A small learning rate requires many updates before reaching the minimum point
The optimal learning rate swiftly reaches the minimum point
Too large of a learning rate causes drastic updates which lead to divergent behaviors

Figure 8. Learning rate

Note. Adapted from "Setting the learning rate of your neural network.", by Jordan, J., (2018, March 1). Retrieved from https://www.jeremyjordan.me/nn-learning-rate/

5.2.3.2.2 Number of epochs and batch size

As already mentioned in subsection 5.2.1.2, an epoch represents the event where the model went through all training examples of the training set (Wilson & Martinez, 2003). The batch size indicates the amount of training examples the model needs to see before the weights of the loss function are updated (Wilson & Martinez, 2003).

5.2.3.2.3 *LSTM* size

The *LSTM* size of an *LSTM* layer equals the amount of *LSTM* cells the layer exists of. It is the number of hidden states an *LSTM* layer contains. In other words, this determines how many features the *LSTM* layer can remember (Phi, 2018).

5.2.3.2.4 Activation function

Currently, the most popular activation function is the rectangular linear unit (*ReLu*) function (Bingham, Macke, & Miikulainen, 2020). Other well-known activation functions such as the sigmoid function and the hyperbolic tangent (*tanh*) function cannot be utilized in deep neural networks due to the problem of vanishing gradients (see subsection 5.2.1.2). *ReLu* bypasses this problem because its derivative is either zero or one. Consequently, the model learns quicker and gives more accurate results (Brownlee, 2019). For this master dissertation, the *ReLu* activation function is applied because of its advantages.

5.2.3.2.5 Optimization algorithm

The method used in this thesis to optimize the model is Adam optimizer (Kingma & Ba, 2014). It combines the advantages of two well-known optimizers: AdaGrad (Duchi, Hazan, & Singer, 2011) and RMSProp (Tieleman & Hinton, 2012). The algorithm is computationally efficient and can handle problems that have a large number of parameters. On top of that, Adam seems to better converge compared to other optimizers such as RMSProp and the previously described stochastic gradient descent (see subsection 5.2.1.2) (Kingma & Ba, 2014).

5.2.4 Implementation ARIMA

To implement the *ARIMA* model, the auto.arima function from the R-package forecast (Hyndman & Khandakar, 2007) is utilized. The hyperparameter tuning of the *ARIMA* model is discussed in subsection 5.2.4.1.

5.2.4.1 Hyperparameter tuning

As already mentioned in subsection 5.2.1.1, *p* represents the number of lags considered by the *AR* part of the model, *d* stands for the order of differencing and *q* serves as the order of the *MA* part (Helmini et al., 2019; Siami-Namini & Namin, 2018). Based on the assumption that the data on which the *ARIMA* algorithm is trained contains seasonality (see subsection 5.1.1), the seasonal *ARIMA* algorithm is applied. Hence, the seasonal parameters *P*, *D*, *Q* and *S* are also activated. The minimum seasonality length *S* for seasonal data equals one, whereas the maximum seasonality length *S* in this case is twelve as the algorithm is trained on a dataset with twelve observations. Table 10 gives an overview of the tuned hyperparameters along with their search space.

Table 10. Hyperparameters and search space ARIMA

Hyperparameter	Search space
p	{0,1,2,3,4,5}
d	{0,1,2,3,4,5}
q	{0,1,2,3,4,5}
P	{0,1,2,3,4,5}
D	{0,1,2,3,4,5}
Q	{0,1,2,3,4,5}
S	{1,2,3,4,5,6,7,8,9,10,11,12}

The optimal values of the hyperparameters of each cluster centroid can be found in appendix 12.

5.3 Performance evaluation

To evaluate the different models and algorithms, the following performance measures are utilized, namely *RMSE* and *MAE*.

5.3.1 RMSE

The most commonly used performance measure to forecast car sales is the Mean Absolute Percentage Error (*MAPE*) (Choi & Varian, 2009; Geva et al., 2013, 2015; Nymand-Andersen & Pantelidis, 2018; Pai & Liu, 2018; Tomczyk & Doligalski, 2015). However, if time series contain zero values, *MAPE* becomes infinite (Kreinovich, Nguyen, & Ouncharoen, 2014; Venkatesh et al., 2014). For this reason, the second most popular performance measure to forecast car sales is utilized in this dissertation, namely *RMSE,* which is defined as follows (Ahn & Spangler, 2014; Benthaus & Skodda, 2015; Nymand-Andersen & Pantelidis, 2018; Seebach et al., 2011):

$$RMSE = \sqrt{\frac{1}{\mathrm{n}}\sum_{\mathrm{t}=1}^{\mathrm{n}}(\mathrm{y_t} - \widehat{\mathrm{y}_\mathrm{t}})^2}$$

where y_t stands for the actual value at time *t*, $\hat{y}_t$ stands for the forecasted value at time *t* and *n* represents the number of samples.

RMSE measures the difference between the actual and the predicted values. The outcomes range from zero to infinite. An advantage of the measure is that it is represented in the same unit as the forecasts (Siami-Namini & Namin, 2018).

5.3.2 MAE

Another frequently used evaluation metric to forecast car sales is *MAE* (Benthaus & Skodda, 2015; Choi & Varian, 2009; Nymand-Andersen & Pantelidis, 2018; Seebach et al., 2011). *MAE* is calculated as below:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \widehat{y_t}|^2$$

where y_t and $\hat{y}_t$ serve as the actual observation and the predicted observation at time *t* respectively and *n* represents the number of samples.

Similar to *RMSE*, *MAE* is a metric used to measure the difference between the true and the forecasted observations. Consequently, *MAE* can take values between zero and infinity as well. The major difference between *RMSE* and *MAE* is that the former is more sensitive to outliers (Benthaus & Skodda, 2015).

5.4 Cross-validation

The goal of regression is to fit a model that performs well on unseen data, or in other words, a model that generalizes well (Bergmeir & Benítez, 2012). This performance is validated on a validation set, which is usually excluded from the data on which the model is trained to ensure that the data of the validation set is unseen. Consequently, not all available training data is utilized to fit the model. This can be problematic, especially when dealing with a small dataset. On top of that, only one performance output is generated, which makes the output sensitive to biases based on the chosen train/validate split. To overcome these problems, *k*-fold cross-validation can be applied. This algorithm randomly splits the training set into *k* parts or folds. Next, the model is trained on *k*-1 folds and validated on the remaining fold. This process is repeated *k*-1 times, at which each time another fold serves as validation set and the remaining *k*-1 folds as training set. In other words, each fold acts only once as validation set and *k*-1 times as training set. Hence, the full training set is utilized to train and validate. In addition, *k* validation scores are obtained. The final performance output is determined by averaging these *k* validation scores, resulting in a more robust measure (Bergmeir & Benítez, 2012).

However, cross-validation comes with some complications when making use of time series data. First of all, cross-validation requires the data to be independent and identically distributed (*i.i.d.*) (Bergmeir & Benítez, 2012). This is often not the case for time series (Bergmeir & Benítez, 2012; Ogasawara et al., 2010). As already mentioned in section 5.2.1, most financial and economic time series are not stationary, implying that the condition of being identically distributed is violated in this case (Ogasawara et al., 2010). On top of that, sequential data is often correlated in time (e.g. autoregression), indicating that data from subsequent time steps are not independent (Bergmeir & Benítez, 2012). Second, time series data cannot be shuffled, otherwise data leakage may occur (Cochrane, 2018). In other words, the time series data must retain its chronological order, meaning that the training set has to contain observations that occurred before the observations contained by the validation set (Cochrane, 2018).

To reap the benefits of cross-validation without violating its fundamental assumptions (i.e. *i.i.d.*), forward-chaining (Cochrane, 2018), also called rolling-origin evaluation (Tashman, 2000) or rolling-origin-recalibration (Bergmeir & Benítez, 2012), is applied in this master thesis. This method works as follows.

By utilizing the forward-chaining method on a training set consisting of twelve observations (i.e. twelve months of the year 2019) with a forecasting horizon of one month, eleven iterations take place (see figure 9). In the first iteration (denoted in figure 9 as ‘CV iteration 1’), the first month serves as training set and the second month as validation set (Cochrane, 2018). Once the model has been trained and validated, the next iteration takes place. For this iteration, the validation set and the training set of the previous iteration now serves as training set, while the observation subsequent to the most recent training observation now acts as validation set. This step is repeated until there are no months left that can serve as validation set (in my case after eleven iterations). Finally, the output measure is gained by averaging the eleven performance outputs of each iteration (Cochrane, 2018).

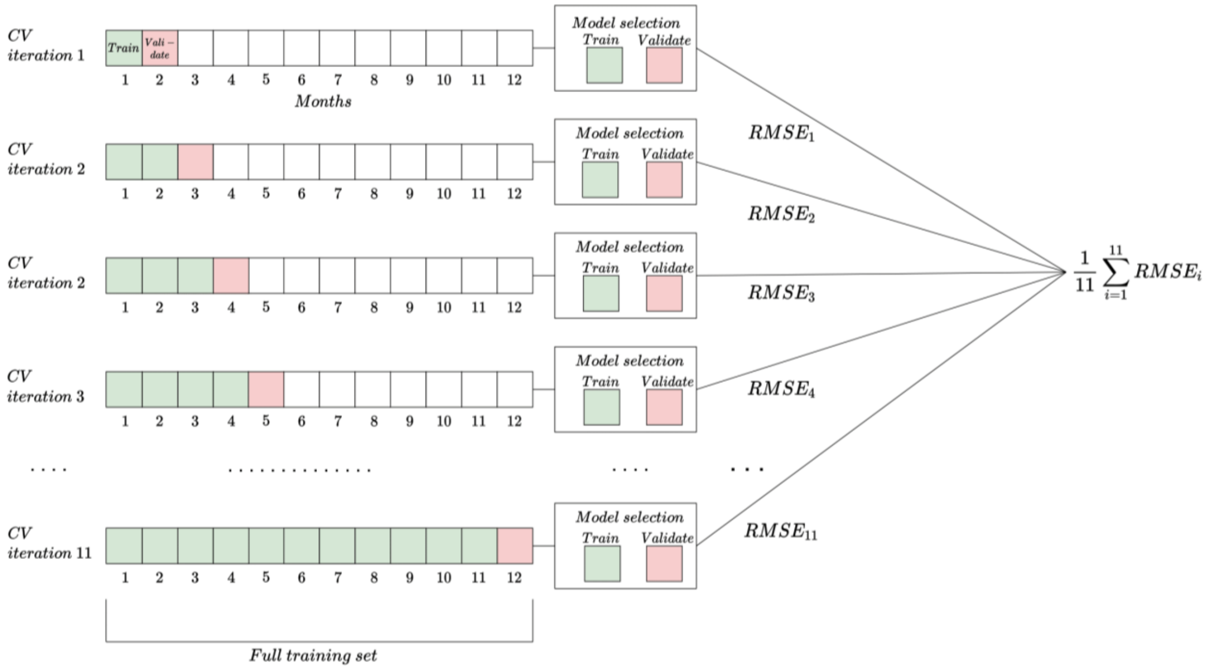


Figure 9. Forward-chaining with fixed horizon of one month

Note. Adapted from “Time Series Nested Cross-Validation”, by Cochrane, C., (2018, May 19). Retrieved from https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9

In practice, the time series cross-validation in case of the *LSTM* algorithm is implemented by making use of the scikit-learn object TimeSeriesSplit (“sklearn.model_selection.TimeSeriesSplit”, 2020). In case of the *ARIMA* algorithm, the forecasting function tsCV from the R-package forecast (Hyndman, 2020) is applied.

5.5 Statistical testing

To verify whether a significant difference between the performance of the *ARIMA* algorithm and the *LSTM* algorithm is present, the pairwise statistical test, i.e. Wilcoxon signed-rank test, is applied (Wilcoxon, 1946). The Wilcoxon signed-rank test is a non-parametric counterpart of the parametric paired Student’s t-test (Trawiński, Smętek, Telec, & Lasota, 2012; Wilcoxon, 1946).

Based on previous research, an *LSTM* regressor seems to outperform an *ARIMA* regressor in terms of predicting time-series (Siami-Namini & Namin, 2018; Weytjens, Lohmann, & Kleinsteuber, 2019). Consequently, I believe that in case of this master thesis, the *LSTM* model also performed better than the *ARIMA* model. Hence, a one-tailed Wilcoxon signed-rank test with the following null hypothesis H_0 and alternative hypothesis H_1 is tested (Trawiński et al., 2012; Wilcoxon, 1946):

> *H_0: The performance of the ARIMA algorithm is significantly better than the performance of the LSTM algorithm*

H_1: The performance of the ARIMA algorithm is significantly worse than the performance of the LSTM algorithm

As I utilized two performance measures, namely *MAE* and *RMSE*, the Wilcoxon signed-rank test is applied on both performance metrics.

Once the best regressor for the dataset consisting of car sales, Google Trends and Instagram data is known, the car sales of 2020 are also forecasted by this regressor for the remaining three datasets (i.e. one model based on car sales, one model based on car sales and Google Trends data, and one model based on car sales and Instagram data). To detect significant differences between the four different models (i.e. one model based on car sales, one model based on car sales and Google Trends data, one model based on car sales and Instagram data, and one model based on car sales, Google Trends and Instagram data), the non-parametric equivalent of the repeated measures ANOVA test, i.e. the Friedman test (Friedman, 1940), is applied. The following null hypothesis H_0 is tested:

H_0: There is no significant difference in the performance of the four models

The alternative hypothesis H_1 states as follows:

H_1: There is a significant difference in the performance between at least two of the four models

This test is conducted once using *MAE* as performance measure and once utilizing *RMSE* as performance metric. If the performance of at least two out of four models significantly differ, the null hypothesis is rejected. To determine which pairs of models significantly differ with regard to their performance, the Bonferroni-Dunn post-hoc test is applied (Dunn, 1961).

6 Results

In chapter 6, the results of the research conducted for this master dissertation are examined. In particular, section 6.1 discusses the formed clusters in more detail and evaluates whether the utilized clustering algorithm captured similar seasonality patterns in the same cluster. Next, section 6.2 assesses the performance of the *ARIMA* algorithm and the *LSTM* algorithm in terms of *RMSE* and *MAE*. More specifically, a comparison between the actual car sales and the forecasted car sales is made (see subsection 6.2.1), followed by the explanation of the outcomes of the applied statistical tests (see subsection 6.2.2). Lastly, all obtained results are discussed in section 6.3.

6.1 Clustering

The cluster distribution of the car models is given in appendix 5. The first car model of each cluster represents the cluster's centroid. The cluster centroids of clusters one to twelve are 'Land Rover Range Rover', 'Nissan Micra', 'Seat Leon', 'Peugeot 208', 'BMW X1', 'Mazda MX-5', 'Toyota RAV4', 'Land Rover Range Rover Evoque', 'Renault Captur', 'Volvo XC60', 'Toyota Auris' and 'Subaru Forester' respectively. The first cluster contains 47 car models, the second cluster consists of 30 car models, the third cluster has seventeen car models, cluster four comprises fourteen car models, twelve car models belong to cluster five, cluster six includes seven car models, cluster seven and eight both incorporate five car models, cluster nine exists of three car models, and clusters ten, eleven and twelve each contain two car models.

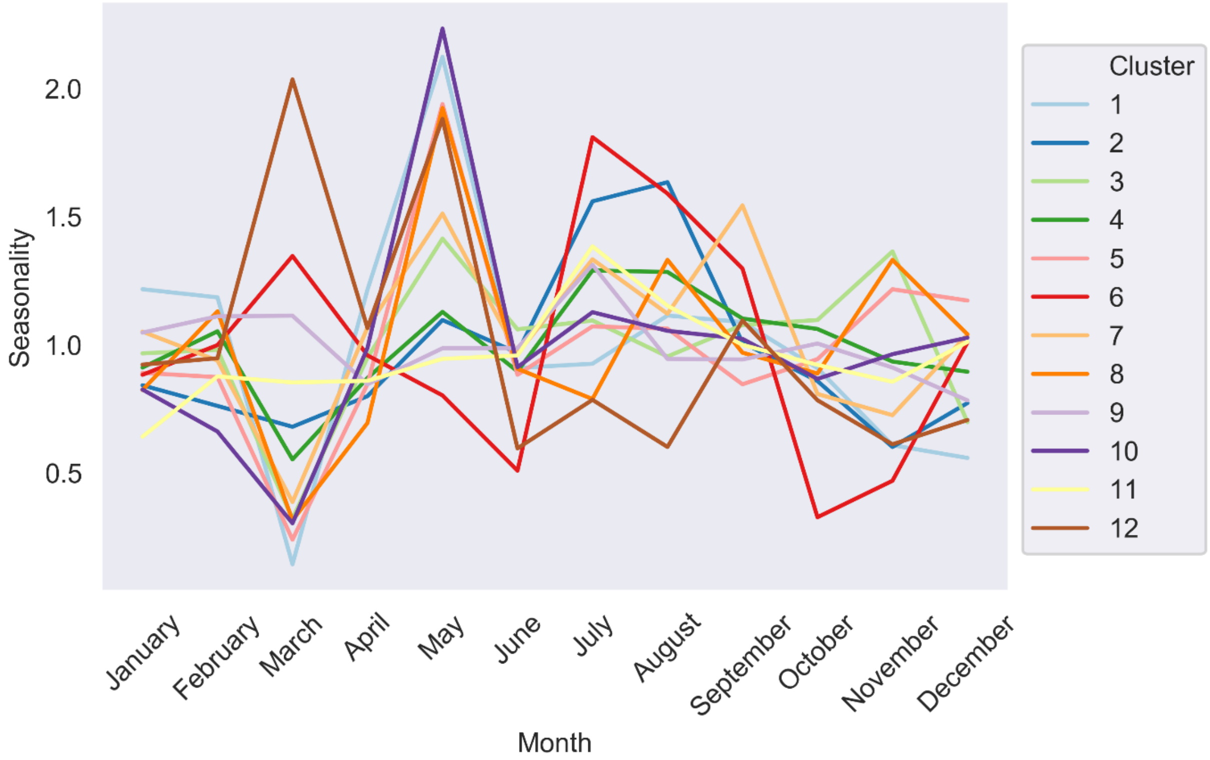


Figure 10. Monthly seasonality values of the cluster centroids

To verify whether there is a certain logic behind the distribution of the car models across the formed clusters, a plot of the monthly seasonality values of the twelve cluster centroids is illustrated in figure 10. As already mentioned in subsection 5.1.1, for each cluster centroid, the seasonality of a certain month is the average seasonality of the car sales in that month starting from 01-01-2016 to 31-12-2019. During the months January to June, all cluster centroids (with an exception of cluster six, cluster nine and cluster twelve) seem to follow a similar pattern in terms of seasonality. However, starting from the month July, all cluster centroids seem to show a different seasonality pattern.

On top of that, for each cluster, I took a closer look at the seasonality of all car models belonging to that cluster. As a result, I was able to observe that the seasonality of the car models within the same cluster are similar. Figures 11 to 22 represent the monthly seasonality of the cars belonging to cluster one to twelve respectively. The x-axis shows each month starting from 01-2016 to 12-2019. The labels on the x-axis are integers ranging from one to 48, with number one representing the first month of 2016 and number 48 the final month of 2019. The y-axis displays the monthly seasonality of all car models belonging to that cluster.

Based on these insights I can derive that the car sales of the cluster centroids appear to have a different seasonality pattern.

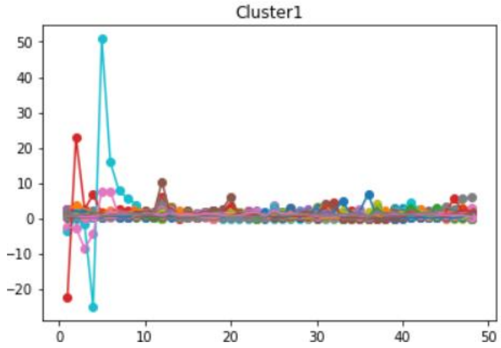


Figure 11. Seasonality of cluster 1

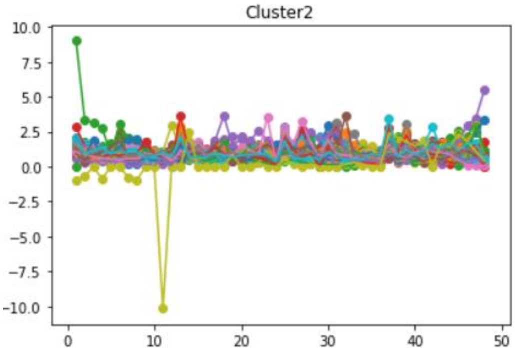


Figure 12. Seasonality of cluster 2

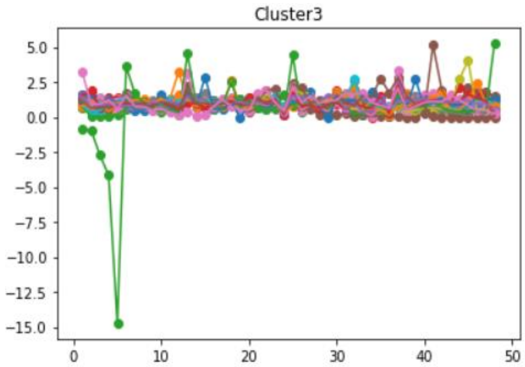


Figure 13. Seasonality of cluster 3

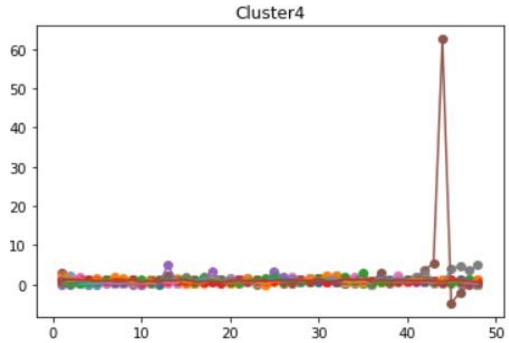


Figure 14. Seasonality of cluster 4

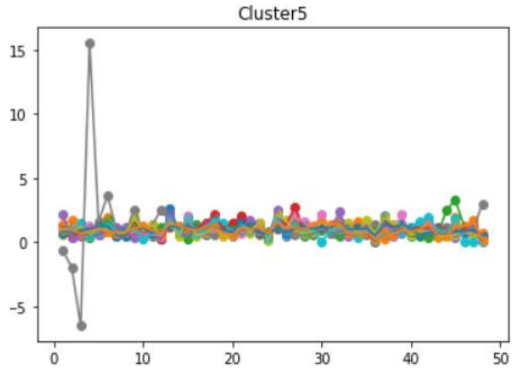


Figure 15. Seasonality of cluster 5

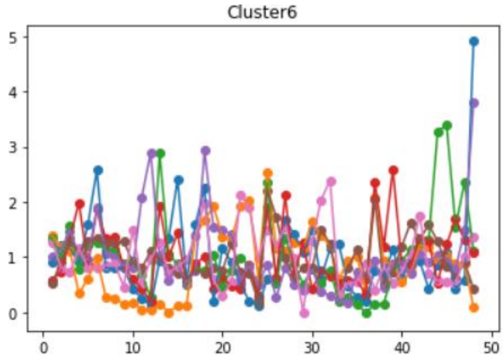


Figure 16. Seasonality of cluster 6

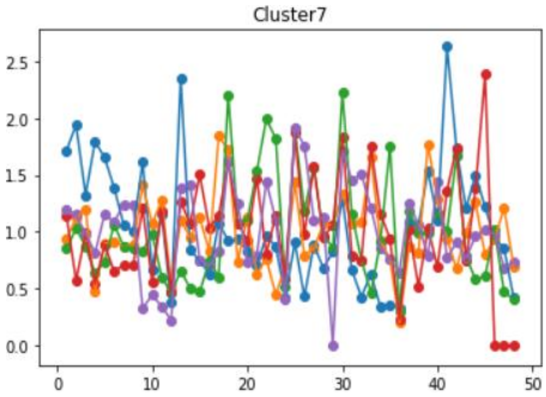


Figure 17. Seasonality of cluster 7

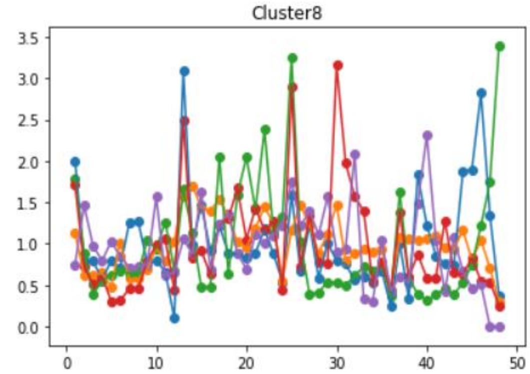


Figure 18. Seasonality of cluster 8

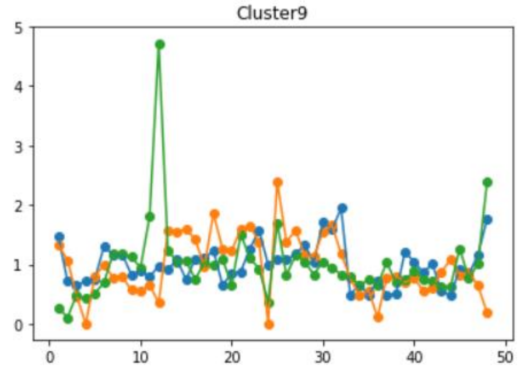


Figure 19. Seasonality of cluster 9

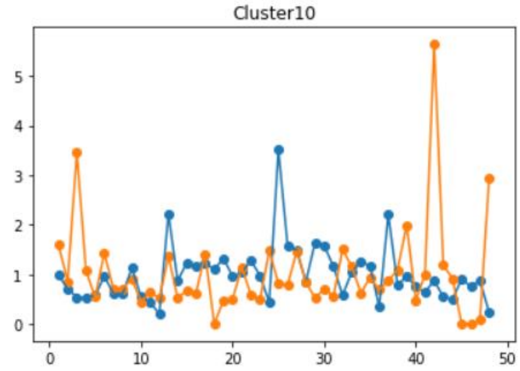


Figure 20. Seasonality of cluster 10

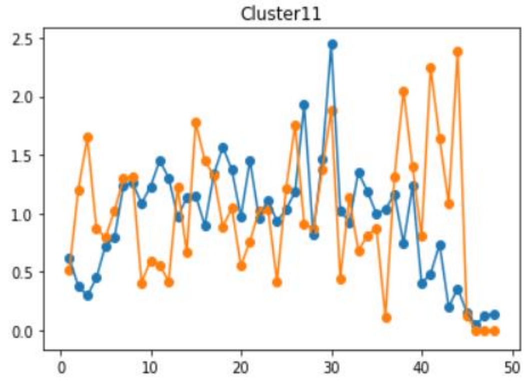


Figure 21. Seasonality of cluster 11

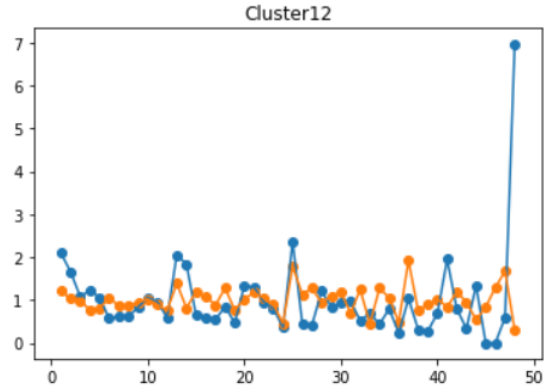


Figure 22. Seasonality of cluster 12

6.2 Forecasting algorithms

6.2.1 Predictions

The outcomes of the performance of the *ARIMA* model based on dataset 4 in terms of *MAE* and *RMSE* is shown in appendix 13. In addition, the performance output in terms of *MAE* and *RMSE* of the four *LSTM* models, i.e. one model with dataset 4 as input, one model that used dataset 1 as input, one regressor that utilized dataset 2 as input and one model that made use of dataset 3 as input, can be consulted at appendix 14, 15, 16 and 17 respectively. Subsection 6.2.2 explains the reason behind the implementation of the *LSTM* algorithm based on dataset 1, dataset 2 and dataset 3 instead of the implementation of the *ARIMA* algorithm based on these datasets.

Figures 23 to 70 depict the actual car sales (represented by a full blue line) and the predicted car sales (represented by a dotted orange line) of the twelve cluster centroids in the year 2020, which were forecasted by an *LSTM* model that either used dataset 1, dataset 2, dataset 3 or dataset 4 as input. Additionally, I noticed that some predictions of the sales of certain car models are negative, which should not possible. The previous is also reflected in the predictions of the sales of 'Toyota Yaris' by the *LSTM* model using dataset 2 (see figure 45), of 'Nissan Micra' by the model using dataset 3 (see figure 48) and of 'Mazda MX-5' by the *LSTM* model using dataset 4 (see figure 64).

The predicted car sales of the *LSTM* model based on dataset 1 (i.e. only car sales) are shown in figures 23 to 34. The graphs imply that the model did not grasp any trends nor seasonality of the car sales for all cluster centroids, with an exception of the car models 'Land Rover Range Rover' (see figure 23), 'Seat Leon' (see figure 25), 'BMW X1' (see figure 27), and 'Volvo XC60' (see figure 32). The slightly better performance of these four cluster centroids is also represented in their performance in terms of *MAE* and *RMSE* (see appendix 15), as these measures are rather low considering the number of actual sales in 2020.

Figures 35 to 46 represent the car sales of the twelve cluster centroids in 2020 that are predicted by the *LSTM* model using dataset 2 (i.e. car sales data and Google Trends data) as input. The interpretation of the forecasted sales is similar to the interpretation of the *LSTM* model based on dataset 1. More specifically, the *LSTM* model based on dataset 2 predicted approximately a constant in time for all cluster centroids except for 'Nissan Micra' (see figure 36), 'Seat Leon' (see figure 37), ' BMW X1' (see figure 39) and 'Volvo XC60' (see figure 44). This insight seems to be reflected in the performance of 'Seat Leon', 'BMW X1' and 'Volvo XC60' in terms of *MAE* and *RMSE* (see appendix 16). However, in case of 'Nissan Micra', this is not represented since the output of the *MAE* and *RMSE* are rather high taking into account the actual car sales.

The actual monthly car sales and the car sales for each cluster centroid in 2020 predicted by the *LSTM* model based on dataset 3 (i.e. car sales data and Instagram data) are visualized in figures 47 to 58. Most of the predictions appear to be approximately a constant, with an exception of 'BMW X1' (see figure 51), 'Land Rover Range Rover Evoque' (see figure 54) and 'Toyota Auris' (see figure 57). The predicted sales of 'BMW X1' and 'Land Rover Range Rover Evoque' seem to capture most of the increases and decreases of the actual car sales. The previous is also present in the performance in terms of *RMSE* and *MAE*, since both seem to perform approximately well related to the actual car sales (see appendix 17). Unfortunately, the predictions of the sales of 'Toyota Auris' are characterized by a number of decreases and increases while the actual car sales are close to zero, which is reflected in the rather low performance of the *LSTM* model in terms of *RSME* and *MAE*.

An overview of the car sales of 2020 predicted by the *LSTM* model that utilized dataset 4 (i.e. car sales, Google Trends data and Instagram data) as input and the actual car sales of the twelve cluster centroids is given by figures 59 to 70. The *LSTM* model seemed able to capture the trends or seasonality of the sales of the car models 'Nissan Micra' (see figure 60), 'BMW X1' (see figure 63), 'Toyota RAV4' (see figure 65) and 'Renault Captur' (see figure 67). Again, the predictions of the sales of 'Toyota Auris' (see figure 69) do not seem to reflect the actual car sales close to zero. Both of these insights also seem to be present in the performance in terms of *MAE* and *RMSE* of the *LSTM* model based on dataset 4 considering the actual car sales (see appendix 14). The predictions of the car sales of the other cluster centroids are approximately a constant.

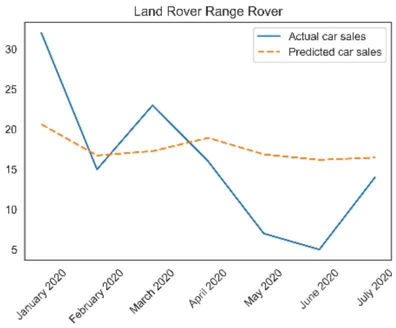


Figure 23. Predictions using dataset 1 for centroid 1

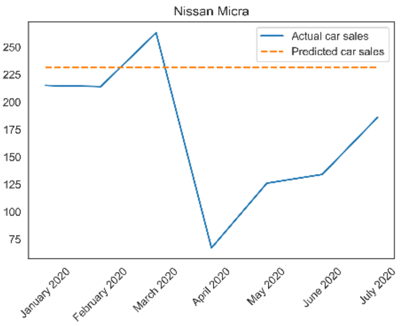


Figure 24. Predictions using dataset 1 for centroid 2

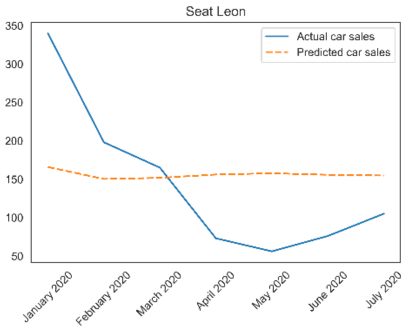


Figure 25. Predictions using dataset 1 for centroid 3

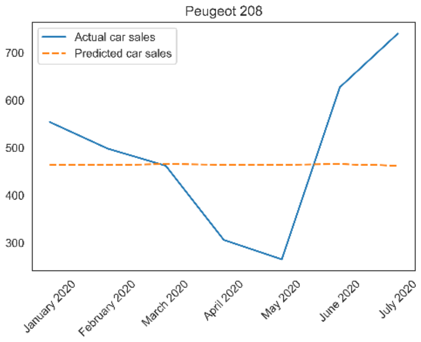


Figure 26. Predictions using dataset 1 for centroid 4

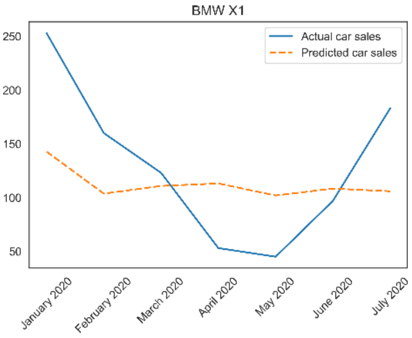


Figure 27. Predictions using dataset 1 for centroid 5

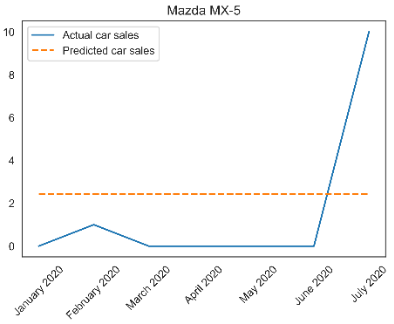


Figure 28. Predictions using dataset 1 for centroid 6

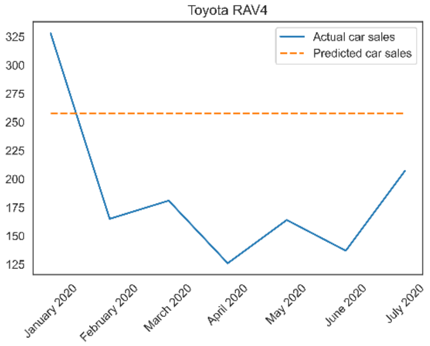


Figure 29. Predictions using dataset 1 for centroid 7

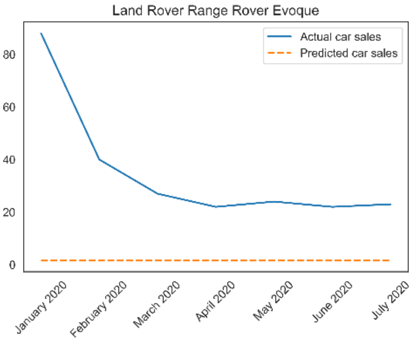


Figure 30. Predictions using dataset 1 for centroid 8

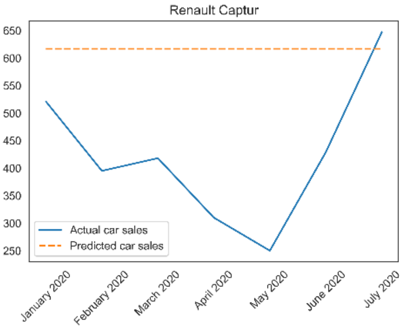


Figure 31. Predictions using dataset 1 for centroid 9

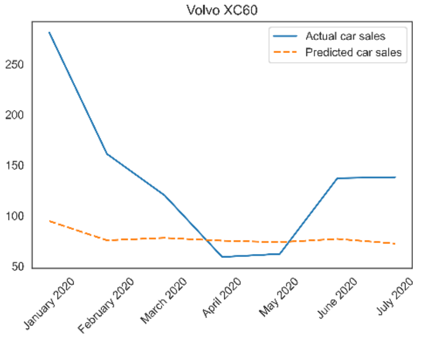


Figure 32. Predictions using dataset 1 for centroid 10

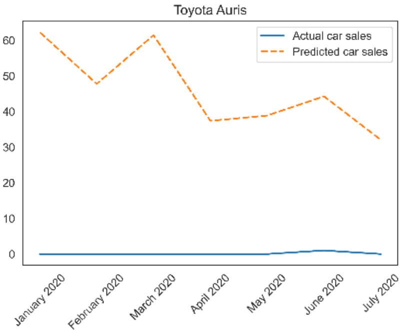


Figure 33. Predictions using dataset 1 for centroid 11

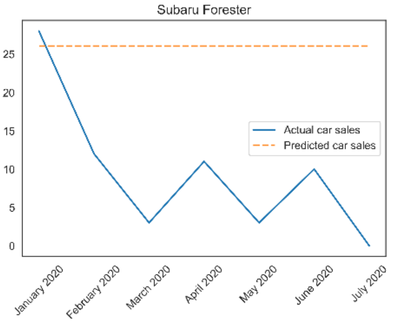


Figure 34. Predictions using dataset 1 for centroid 12

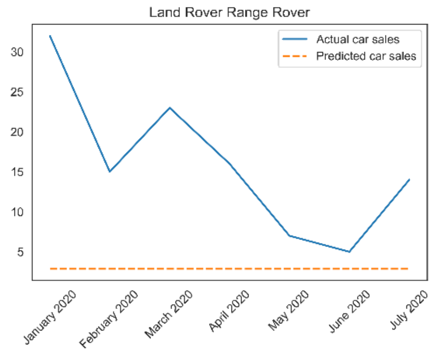


Figure 35. Predictions using dataset 2 for centroid 1

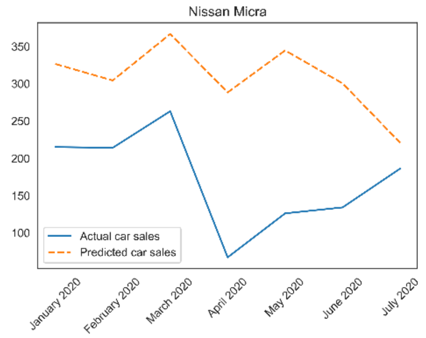


Figure 36. Predictions using dataset 2 for centroid 2

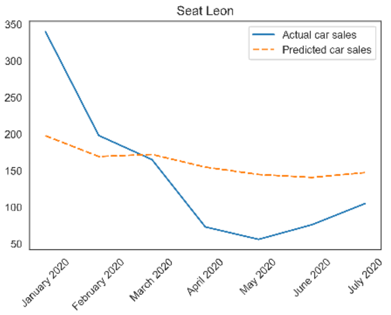


Figure 37. Predictions using dataset 2 for centroid 3

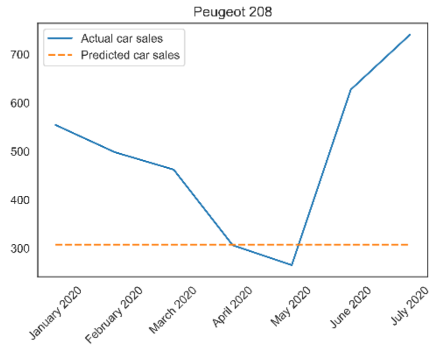


Figure 38. Predictions using dataset 2 for centroid 4

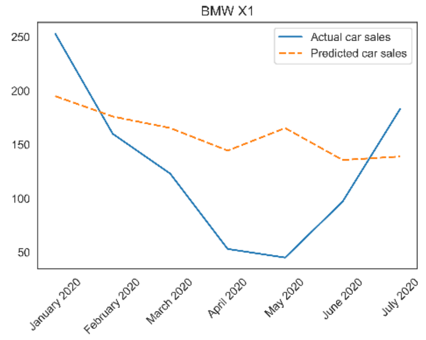


Figure 39. Predictions using dataset 2 for centroid 5

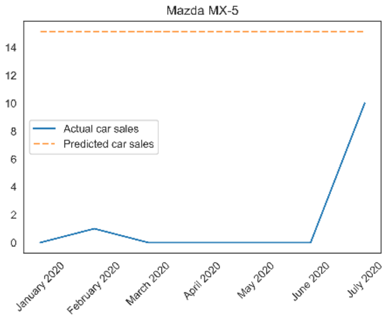


Figure 40. Predictions using dataset 2 for centroid 6

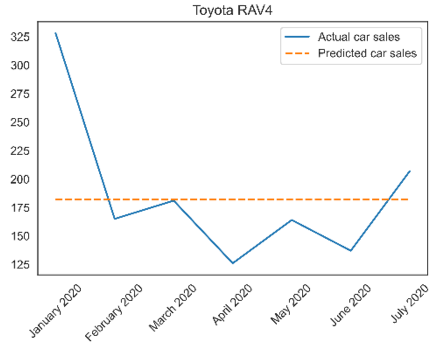


Figure 41. Predictions using dataset 2 for centroid 7

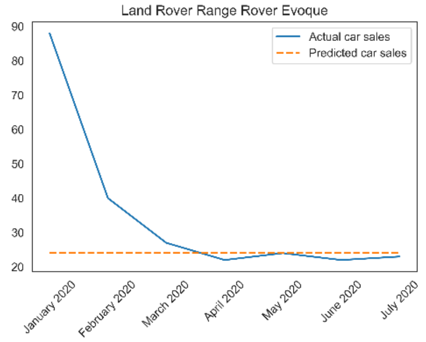


Figure 42. Predictions using dataset 2 for centroid 8

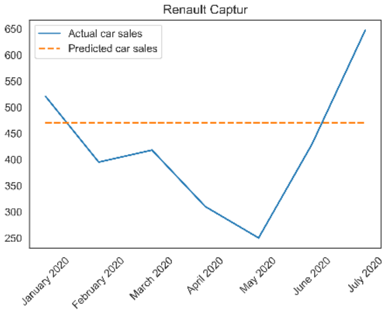


Figure 43. Predictions using dataset 2 for centroid 9

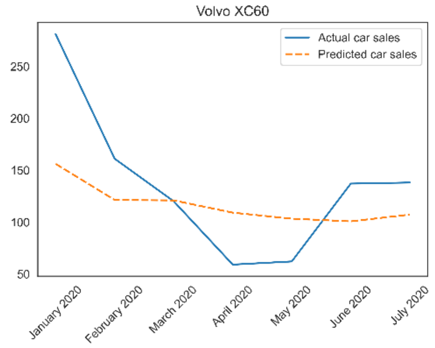


Figure 44. Predictions using dataset 2 for centroid 10

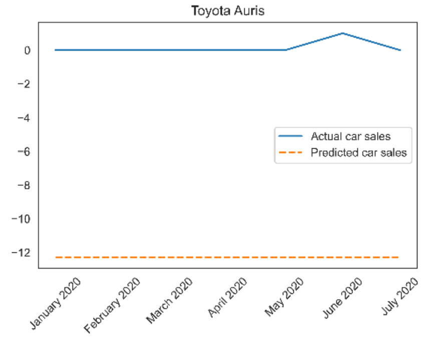


Figure 45. Predictions using dataset 2 for centroid 11

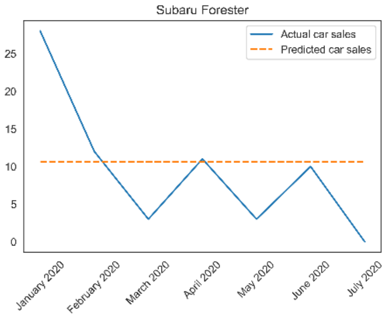


Figure 46. Predictions using dataset 2 for centroid 12

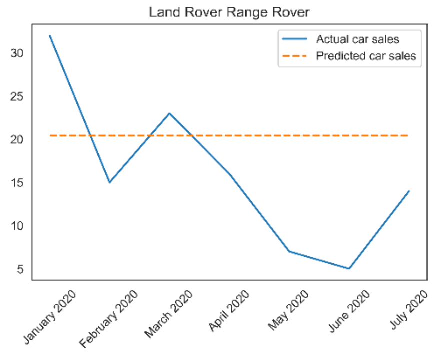


Figure 47. Predictions using dataset 3 for centroid 1

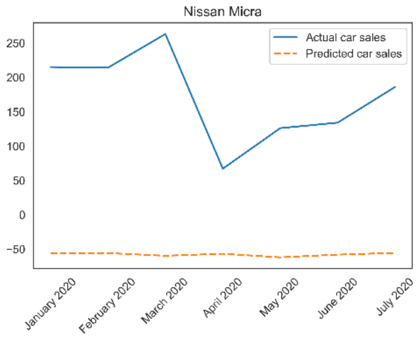


Figure 48. Predictions using dataset 3 for centroid 2

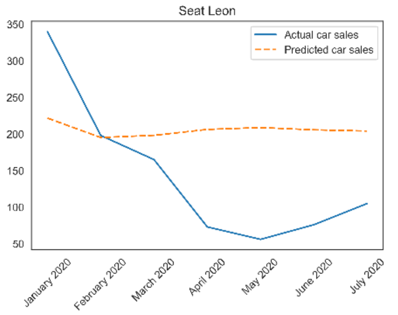


Figure 49. Predictions using dataset 3 for centroid 3

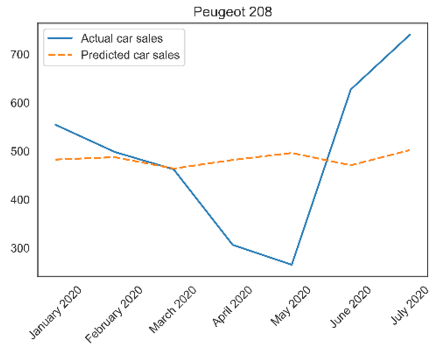


Figure 50. Predictions using dataset 3 for centroid 4

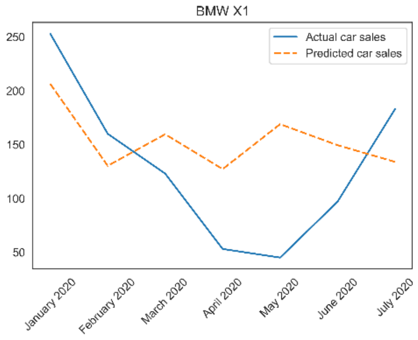


Figure 51. Predictions using dataset 3 for centroid 5

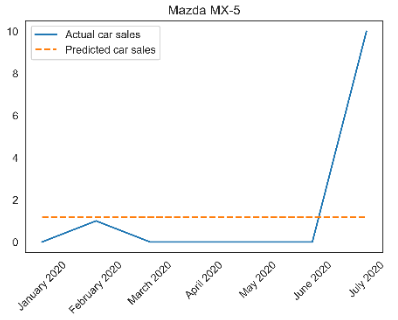


Figure 52. Predictions using dataset 3 for centroid 6

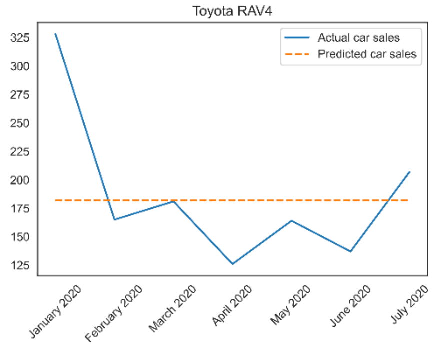


Figure 53. Predictions using dataset 3 for centroid 7

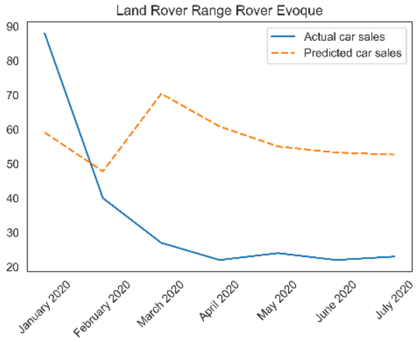


Figure 54. Predictions using dataset 3 for centroid 8

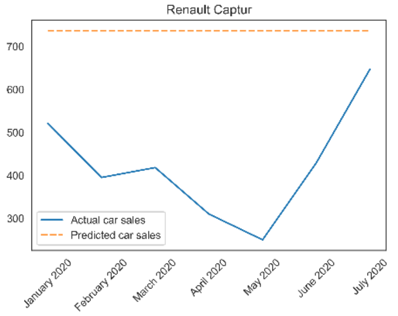


Figure 55. Predictions using dataset 3 for centroid 9

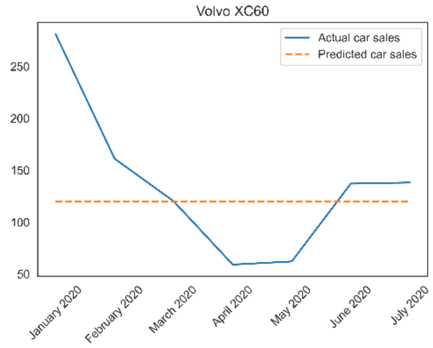


Figure 56. Predictions using dataset 3 for centroid 10

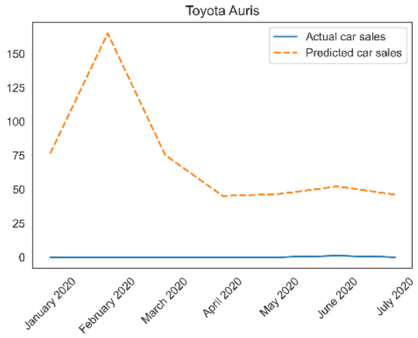


Figure 57. Predictions using dataset 3 for centroid 11

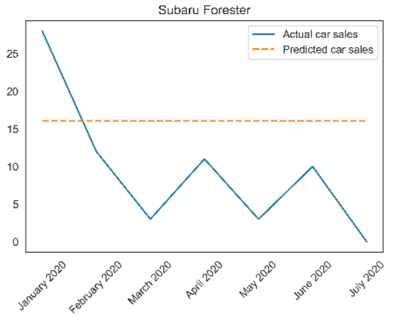


Figure 58. Predictions using dataset 3 for centroid 12

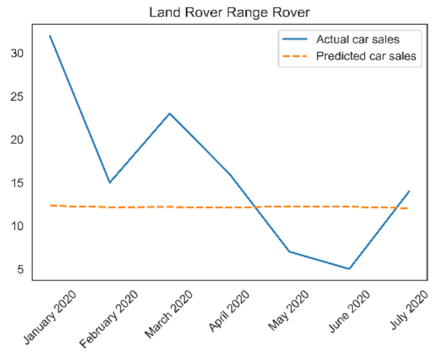


Figure 59. Predictions using dataset 4 for centroid 1

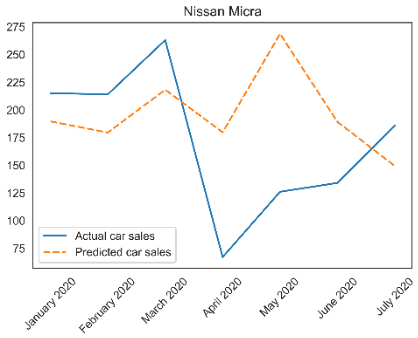


Figure 60. Predictions using dataset 4 for centroid 2

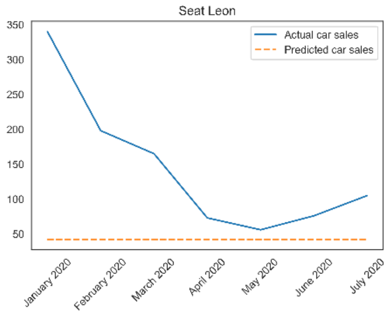


Figure 61. Predictions using dataset 4 for centroid 3

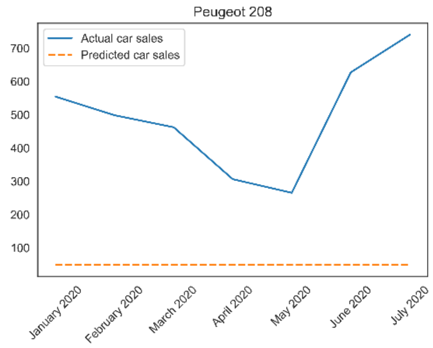


Figure 62. Predictions using dataset 4 for centroid 4

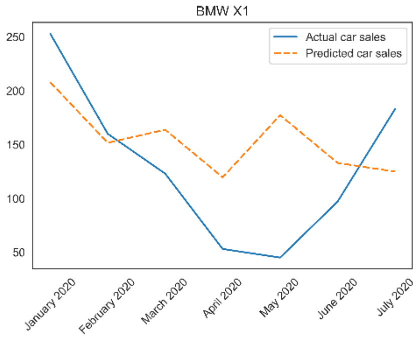


Figure 63. Predictions using dataset 4 for centroid 5

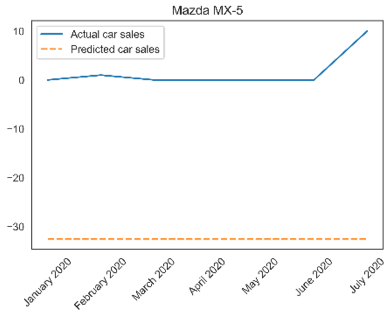


Figure 64. Predictions using dataset 4 for centroid 6

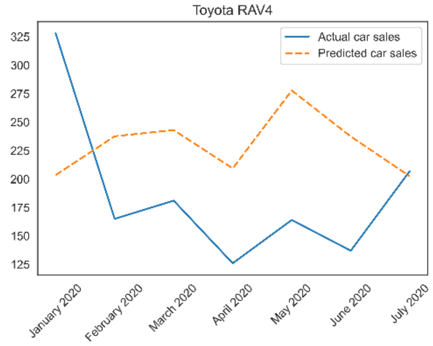


Figure 65. Predictions using dataset 4 for centroid 7

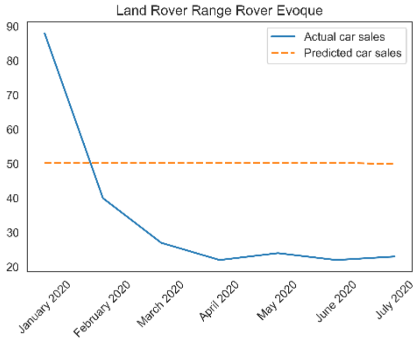


Figure 66. Predictions using dataset 4 for centroid 8

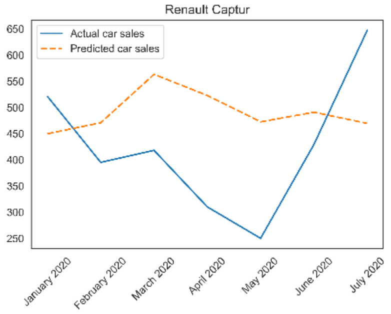


Figure 67. Predictions using dataset 4 for centroid 9

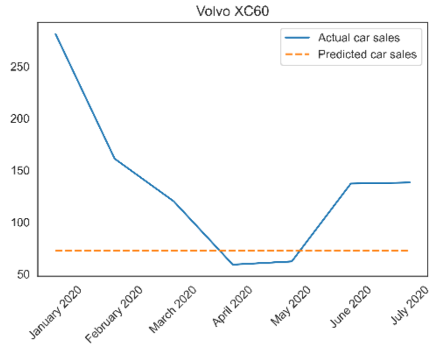


Figure 68. Predictions using dataset 4 for centroid 10

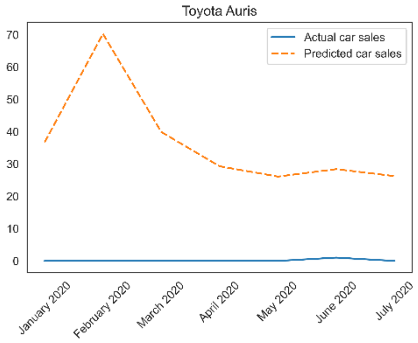


Figure 69. Predictions using dataset 4 for centroid 11

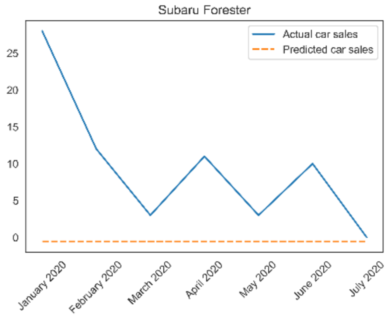


Figure 70. Predictions using dataset 4 for centroid 12

6.2.2 Statistical testing

To statistically determine which of the two regressors (i.e. *ARIMA* and *LSTM*) performed best in terms of predicting the car sales of 2020 utilizing dataset 4 as input, two one-tailed Wilcoxon signed-rank tests were applied (Wilcoxon, 1946). The first test is used to determine the best regressor in terms of *RMSE,* while the second test is executed to find the best regressor in terms of *MAE*. The outcomes of the performance of the *ARIMA* model and the *LSTM* model utilizing dataset 4 as input can be consulted in appendix 13 and appendix 14 respectively.

The first test has a p-value of 0,0329. Consequently, it can be rejected that the performance of the *ARIMA* algorithm in terms of *RMSE* is significantly better than the performance of the *LSTM* algorithm at the five percent level of significance. The null hypothesis of the second test is also rejected at the five percent level of significance since the p-value equals 0,0150. As a result, it can be rejected that the performance of the *ARIMA* algorithm in terms of *MAE* is significantly better than the performance of the *LSTM* algorithm at the five percent level of significance.

In summary, the *LSTM* algorithm statistically outperforms the *ARIMA* algorithm in predicting car sales based on the combination of car sales data, Google Trends data and Instagram data. As a consequence, the *LSTM* algorithm is implemented to predict the monthly car sales in the year 2020 based on solely the car sales (i.e. dataset 1), based on car sales and Google Trends data (i.e. dataset 2) and based on car sales and Instagram data (i.e. dataset 3). The performance outcomes of the *LSTM* model in terms of *RMSE* and *MAE* are represented in appendix 15, 16 and 17 respectively.

Next, the outcomes of the performance of the *LSTM* model based on dataset 1, dataset 2, dataset 3 and dataset 4 are statistically compared to each other. More specifically, as already mentioned in section 5.5, the Friedman test is utilized to statistically detect whether there are significant differences in the four *LSTM* models in terms of *MAE* and *RMSE* (Friedman, 1940). Based on the *MAE* output of the four models, the p-value of the Friedman test is equal to $1{,}1043*10^{-10}$. This value implies that the null hypothesis 'there are no significant differences between the performance of any of the four models in terms of *MAE*' can be rejected at the five percent level of significance. When comparing the models' performance in terms of *RMSE*, the p-value equals $8{,}8785*10^{-12}$. This indicates that the hypothesis 'there are no significant differences between the performance of any of the four models in terms of *RMSE*' can be rejected at the five percent level of significance. In order to determine which models significantly differ from one another, the Bonferroni-Dunn post-hoc test is applied (Dunn, 1961). Table 11 summarizes the p-values of this post-hoc test utilizing *MAE* as performance measure, whereas table 12 gives an overview of the p-values where *RMSE* serves as performance measure. Both tables indicate that the models using dataset 1, dataset 2 and dataset 3 as input do not significantly differ from one another at the five percent level of significance, whereas the model that utilized dataset 4 as input significantly differs from the other three models at the five percent level of significance, as its p-values are smaller than 0,05.

Table 11. p-values Bonferroni-Dun post-hoc test using *MAE* as performance measure

	Dataset 1	**Dataset 2**	**Dataset 3**	**Dataset 4**
Dataset 1	1,00	1,00	0.705	0,657*10^{-5}
Dataset 2	1,00	1,00	0.194	0,305*10^{-6}
Dataset 3	0,705	0,194	1,00	0,563*10^{-2}
Dataset 4	0,657*10^{-5}	0,305*10^{-6}	0,563*10^{-2}	1,00

Table 12. p-values Bonferroni-Dun post-hoc test using *RMSE* as performance measure

	Dataset 1	**Dataset 2**	**Dataset 3**	**Dataset 4**
Dataset 1	1,00	1,00	0.811	0,672*10^{-5}
Dataset 2	1,00	1,00	0.351	0,834*10^{-6}
Dataset 3	0,811	0,351	1,00	0,443*10^{-2}
Dataset 4	0,672*10^{-5}	0,834*10^{-6}	0. 443*10^{-2}	1,00

Considering appendices 14, 15, 16 and 17, I determined the percentage of the predictions of the car sales for which the *LSTM* model that utilized dataset 4 as input outperforms the other *LSTM* models. It seems that, on average, in 38,58 percent of the cases the *LSTM* model trained on dataset 4 performs better than the other three *LSTM* models in terms of *MAE*. More specifically, in 36,30 percent; 35,61 percent and 43,84 percent of the cases, the *LSTM* model fit on dataset 4 performed better than the *LSTM* model fit on dataset 1, the *LSTM* model fit on dataset 2 and the *LSTM* model trained on dataset 3 respectively. In terms of *RMSE*, the *LSTM* model trained on dataset 4 outperformed the *LSTM* model fit on dataset 1, the *LSTM* model fit on dataset 2 and *LSTM* the model fit on dataset 3 in 30,82 percent; 33,56 percent and 44,52 percent of the cases respectively. Hence, on average, the *LSTM* model fit on dataset 4 outperformed the other three LSTM models in 36,30 percent of the cases in terms of *RMSE*.

6.3 Discussion

This section provides some possible insights for the results obtained in section 6.2.

First of all, the better performance of the *LTSM* algorithm in comparison to the *ARIMA* model based on dataset 4 could be explained by the ability of the *LSTM* algorithm to find non-linear patterns and other complex relationships between the variables (i.e. the car sales, Google Trends, number of comments and polarity), that the *ARIMA* model was not capable to capture (Siami-Namini & Namin, 2018).

Secondly, no significant difference is determined between the performance of the *LSTM* model trained on historical car sales, the *LSTM* model trained on historical car sales and Google Trends data and the *LSTM* model trained on historical car sales and Instagram data. Hence, Google Trends data and Instagram data are equally informative. This finding is surprising as the research of Geva et al. (2013, 2015) found that models based on search trends data statistically outperformed models based on forum data.

Thirdly, the *LSTM* model utilizing the historical car sales data, Google Trends data and Instagram data as input significantly outperformed the other *LSTM* models, on average, in 38,57 percent of the cases in terms of *MAE* and in 36,30 percent of the cases in terms of *MSE*. This could be clarified by the fact that search trends data and social media data complement one another due to their difference in nature (Santillana et al., 2015; Geva et al., 2013, 2015). More specifically, search trends data tends to reveal the true interests of the customers while not affecting the mindset of others, whereas social media exposes the interests of customers to their social environment and consequently affecting the social environment's purchasing behavior (Geva et al., 2013).

Finally, for some car models, the different *LSTM* models predicted negative car sales. This mostly occurred when the actual car sales of the models are close to or equal to zero. Hence, a potential explanation for this phenomenon is that, since an *LSTM* model is not aware that car sales cannot be negative, the models predicted a decrease in car sales causing predictions close to zero to become negative over time.

7 Conclusion

As it is of crucial importance for car manufacturers to know the future sales of car models (Fantazzini & Toktamysova, 2015), the purpose of this master thesis is to determine whether the incorporation of Google Trends data and Instagram data in a model based on historical car sales data improves the model's forecasting accuracy.

To conclude this research, I will provide an answer to my proposed research questions.

- Is it possible to predict car sales making use of Instagram features?

Firstly, I found that the inclusion of Instagram features does not significantly improve the model's predictive power to forecast car sales, both in terms of *MAE* and *RMSE*.

- Is it possible to predict car sales making use of Google Trends search data?

Secondly, based on the results of this research, Google Trends data does not significantly enhance the forecasting power of the model to predict car sales, both in terms of *MAE* and *RMSE*.

- Is a higher predictive performance present when combining Instagram features and Google Trends data to predict car sales?

Thirdly, I conclude that the predictive performance of the forecasting algorithm using a combination of historical car sales data, Instagram data and Google Trends data is, on average, significantly higher in 38,58 percent of the cases in terms of *MAE* and in 36,30 percent of the cases in terms of *RMSE* than the forecasting algorithms using either historical car sales data, both car sales data and Google Trends data or both car sales data and Instagram data.

7.1 Limitations and suggestions for future research

This section covers the encountered limitations of this master thesis and provides some suggestions for further research.

In this research, some limitations in terms of collecting the relevant data are present. Firstly, on average, 75 percent of all hashtags that are present in the caption of a picture are not directly related to the content of that picture (i.e. stophashtags) (Giannoulakis and Tsapatsoulis, 2015, 2016a). Since the collection of Instagram data in this master thesis is based on hashtags, a high probability exists that not all collected Instagram posts are directly related to the relevant hashtag. Secondly, as Google Trends is sensitive to accent marks and spelling errors, the search queries utilized in this dissertation to find all search data related to a car model is not exhaustive (Mavragani & Ochoa, 2019). Thirdly, a drawback of car sales data is the relatively large granularity compared to the Google Trends data and the Instagram data. More precisely, the car sales data is available at a monthly level, whereas Google Trends data is available at a weekly basis and Instagram data at the level of seconds. Hence, I was forced to aggregate the Google Trends data and the Instagram data at a monthly basis. Consequently, the number of training observations is reduced to twelve (i.e. one observation for each month). Furthermore, the forecasting models were allowed to predict negative car sales. In order to solve this, the car sales can undergo a logarithmic transformation (Hyndman & Athanasopoulos, 2018).

For further research, features extracted from images and videos by means of deep learning techniques such as convolutional neural networks might be an interesting variable to extract from Instagram posts (LeCun et al., 2015; Paolanti, Kaiser, Schallner, Frontoni, & Zingaretti, 2017). On top of that, other forecasting algorithms such as an ensemble of *ARIMA* and *LSTM* could be utilized to test whether this model leads to better forecasts. Moreover, a different algorithm than *ARIMA* can be applied to predict the external regressors, such as *LSTM* or an ensemble of *ARIMA* and *LSTM*. On top of that, training the *LSTM* and *ARIMA* algorithm on all 146 car models instead of solely on the twelve training observations might improve the forecasting power of these algorithms. Lastly, an alternative performance measure that does not depend on the magnitude of the car sales, such as the symmetric mean absolute percentage error, could be utilized since it could make the forecasting accuracy more easily to compare (Venkatesh et al., 2014).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Ghemawat, S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467.*

Ahn, H. I., & Spangler, W. S. (2014, April). Sales prediction with social media analysis. In *2014 Annual SRII Global Conference* (pp. 213-222). IEEE.

Alin, A. (2010). Multicollinearity. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(3), 370-374.

Asur, S., & Huberman, B. A. (2010, August). Predicting the future with social media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01* (pp. 492-499). IEEE Computer Society.

Barreira, N., Godinho, P., & Melo, P. (2013). Nowcasting unemployment rate and new car sales in south-western Europe with Google Trends. *NETNOMICS: Economic Research and Electronic Networking*, *14*(3), 129-165.

Bayer, J., Wierstra, D., Togelius, J., & Schmidhuber, J. (2009, September). Evolving memory cell structures for sequence learning. In *International Conference on Artificial Neural Networks* (pp. 755-764). Springer, Berlin, Heidelberg.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157-166.

Benthaus, J., & Skodda, C. (2015). Investigating consumer information search behavior and consumer emotions to improve sales forecasting.

Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, *191*, 192-213.

Bingham, G., Macke, W., & Miikkulainen, R. (2020). Evolutionary optimization of deep learning activation functions. *arXiv preprint arXiv:2002.07224*.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of computational science*, *2*(1), 1-8.

Box, G. E., & Jenkins, G. M. (1970). Time series analysis: Forecasting and control Holden-Day. *San Francisco*, 498.

Bravo-Marquez, F., Mendoza, M., & Poblete, B. (2014). Meta-level sentiment models for big social data analysis. *Knowledge-based systems*, *69*, 86-99.

Briggs, J. (2017, July 10). How do you compare large numbers of items in Google Trends? Retrieved from https://digitaljobstobedone.com/2017/07/10/how-do-you-compare-large-numbers-of-items-in-google-trends/

Brownlee, J. (2018, November 14). How to Develop LSTM Models for Time Series Forecasting. Retrieved from https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

Brownlee, J. (2019, January 9). A Gentle Introduction to the Rectified Linear Unit (ReLU). Retrieved from https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

Butina, D. (1999). Unsupervised data base clustering based on daylight's fingerprint and Tanimoto similarity: A fast and automated way to cluster small and large data sets. *Journal of Chemical Information and Computer Sciences*, *39*(4), 747-750.

Cao, J., Li, Z., & Li, J. (2019). Financial time series forecasting model based on CEEMDAN and LSTM. *Physica A: Statistical Mechanics and its Applications*, *519*, 127-139.

Carrière-Swallow, Y., & Labbé, F. (2013). Nowcasting with Google Trends in an emerging market. *Journal of Forecasting*, *32*(4), 289-298.

Chaovalit, P., & Zhou, L. (2005, January). Movie review mining: A comparison between supervised and unsupervised classification approaches. In *Proceedings of the 38th annual Hawaii international conference on system sciences* (pp. 112c-112c). IEEE.

Choi, H., & Varian, H. (2009). Predicting initial claims for unemployment benefits. *Google Inc*, 1-5.

Choi, H., & Varian, H. (2012). Predicting the present with Google Trends. *Economic record*, *88*, 2-9.

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

Chu, S. C., & Kim, Y. (2011). Determinants of consumer engagement in electronic word-of-mouth (eWOM) in social networking sites. *International journal of Advertising*, *30*(1), 47-75.

Clement, J. (2020, July 24). Instagram: distribution of global audiences 2020, by age group. Retrieved from https://www.statista.com/statistics/325587/instagram-global-age-group/

Cochrane, C. (2018, May 19). Time Series Nested Cross-Validation. Retrieved from https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9

Colliander, J., & Marder, B. (2018). 'Snap happy'brands: Increasing publicity effectiveness through a snapshot aesthetic when marketing a brand on Instagram. *Computers in Human Behavior*, *78*, 34-43.

Daoud, J. I. (2017, December). Multicollinearity and regression analysis. In *Journal of Physics: Conference Series* (Vol. 949, No. 1, p. 012009). IOP Publishing.

de Best, R., (2020, March 4). Market share distribution of search engines in the Netherlands from 2009 to 2019. Retrieved from https://www.statista.com/statistics/688737/market-shares-of-search-engines-in-the-netherlands/

Dhaoui, C., Webster, C. M., & Tan, L. P. (2017). Social media sentiment analysis: lexicon versus machine learning. *Journal of Consumer Marketing*.

Ding, X., Liu, B., & Yu, P. S. (2008, February). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining* (pp. 231-240).

Dodourova, M., & Bevis, K. (2014). Networking innovation in the European car industry: Does the Open Innovation model fit?. *Transportation Research Part A: Policy and Practice*, *69*, 252-271.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, *12*(7).

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American statistical association*, *56*(293), 52-64.

Fantazzini, D., & Toktamysova, Z. (2015). Forecasting German car sales using Google data and multivariate models. *International Journal of Production Economics*, *170*, 97-135.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, *11*(1), 86-92.

General Mills (2016). Pseudo API for Google Trends. Retrieved from: https://github.com/GeneralMills/pytrends

Geva, T., Oestreicher-Singer, G., Efron, N., & Shimshoni, Y. (2013). Do customers speak their minds? using forums and search for predicting sales.

Geva, T., Oestreicher-Singer, G., Efron, N., & Shimshoni, Y. (2015, November). Using forum and search data for sales prediction of high-involvement products. In *Tomer Geva, Gal Oestreicher-Singer, Niv Efron, Yair Shimshoni." Using Forum and Search Data for Sales Prediction of High-Involvement Products"-MIS Quarterly, Forthcoming*.

Gezici, G., Dehkharghani, R., Yanikoglu, B., Tapucu, D., & Saygin, Y. (2013, June). Su-sentilab: A classification system for sentiment analysis in twitter. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (pp. 471-477).

Giannoulakis, S., & Tsapatsoulis, N. (2015, September). Instagram hashtags as image annotation metadata. In *IFIP International Conference on Artificial Intelligence Applications and Innovations* (pp. 206-220). Springer, Cham.

Giannoulakis, S., & Tsapatsoulis, N. (2016a). Evaluating the descriptive power of Instagram hashtags. *Journal of Innovation in Digital Ecosystems*, *3*(2), 114-129.

Giannoulakis, S., & Tsapatsoulis, N. (2016b, October). Defining and identifying stophashtags in instagram. In *INNS Conference on Big Data* (pp. 304-313). Springer, Cham.

Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., & Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature*, *457*(7232), 1012-1014.

Google Trends (2020). Google Trends. Retrieved from: https://www.google.com/trends

Google Trends: Understanding the data. (n.d.). *Google News Initiative*. Retrieved from https://newsinitiative.withgoogle.com/training/lesson/4876819719258112?image=trends&tool=Google%20Trends

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6645-6649). IEEE.

Guo, T., Xu, Z., Yao, X., Chen, H., Aberer, K., & Funaya, K. (2016, October). Robust online time series prediction with recurrent neural networks. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 816-825). Ieee.

Helmini, S., Jihan, N., Jayasinghe, M., & Perera, S. (2019). Sales forecasting using multivariate long short term memory network models. *PeerJ PrePrints*, *7*, e27712v1.

Highfield, T., & Leaver, T. (2015). A methodology for mapping Instagram hashtags. *First Monday*, *20*(1), 1-11.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Hoffman, D. L., & Fodor, M. (2010). Can you measure the ROI of your social media marketing?. *MIT Sloan management review*, *52*(1), 41.

Hu, Y., Manikonda, L., & Kambhampati, S. (2014, May). What we instagram: A first analysis of instagram photo content and user types. In *Eighth International AAAI conference on weblogs and social media*.

Hutto, C. J., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*.

Hyndman, R. J. (2020). Forecasting Functions for Time Series and Linear Models. Retrieved from https://cran.r-project.org/web/packages/forecast/forecast.pdf

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Hyndman, R. J., & Khandakar, Y. (2007). *Automatic time series for forecasting: the forecast package for R* (No. 6/07). Clayton VIC, Australia: Monash University, Department of Econometrics and Business Statistics.

Ili, S., Albers, A., & Miller, S. (2010). Open innovation in the automotive industry. *R&d Management*, *40*(3), 246-255.

Jordan, J. (2018, March 1). Setting the learning rate of your neural network. Retrieved from Jeremy Jordan: https://www.jeremyjordan.me/nn-learning-rate/

Kalogirou, S. A. (2000). Applications of artificial neural-networks for energy systems. *Applied energy*, *67*(1-2), 17-35.

Kelleher, J. D. (2019). *Deep Learning*. Mit Press.

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

Kreinovich, V., Nguyen, H. T., & Ouncharoen, R. (2014). How to estimate forecasting quality: a system-motivated derivation of symmetric mean absolute percentage error (SMAPE) and other similar characteristics.

Landrum, G. (2020). rdkit.ML.Cluster.Butina module. Retrieved from https://www.rdkit.org/docs/source/rdkit.ML.Cluster.Butina.html

Lassen, N. B., Madsen, R., & Vatrapu, R. (2014, September). Predicting iphone sales from iphone tweets. In *2014 IEEE 18th International Enterprise Distributed Object Computing Conference* (pp. 81-90). IEEE.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

Levenshtein, V. I. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (Vol. 10, No. 8, pp. 707-710).

Mavragani, A., & Ochoa, G. (2019). Google Trends in infodemiology and infoveillance: methodology framework. *JMIR public health and surveillance*, *5*(2), e13439.

Meire, M., Ballings, M., & Van den Poel, D. (2016). The added value of auxiliary data in sentiment analysis of Facebook posts. *Decision Support Systems*, *89*, 98-112.

Mudinas, A., Zhang, D., & Levene, M. (2012, August). Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the first international workshop on issues of sentiment discovery and opinion mining* (pp. 1-8).

Murphy, K. P. (2012). A Probabilistic Perspective. *Text book*.

Necas, D. (2014). python-Levenshtein. Retrieved from https://pypi.org/project/python-Levenshtein/

Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 2018). San Francisco, CA: Determination press.

Nymand-Andersen, P., & Pantelidis, E. (2018). *Google econometrics: nowcasting euro area car sales and big data quality requirements* (No. 30). ECB Statistics Paper.

Ogasawara, E., Martinez, L. C., De Oliveira, D., Zimbrão, G., Pappa, G. L., & Mattoso, M. (2010, July). Adaptive normalization: A novel data normalization approach for non-stationary time series. In *The 2010 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

Ortigosa, A., Martín, J. M., & Carro, R. M. (2014). Sentiment analysis in Facebook and its application to e-learning. *Computers in human behavior*, *31*, 527-541.

Pai, P. F., & Liu, C. H. (2018). Predicting vehicle sales by sentiment analysis of Twitter data and stock market values. *IEEE Access*, *6*, 57655-57662.

Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10* (pp. 79-86). Association for Computational Linguistics.

Paolanti, M., Kaiser, C., Schallner, R., Frontoni, E., & Zingaretti, P. (2017, September). Visual and textual sentiment analysis of brand-related social media pictures using deep convolutional neural networks. In *International Conference on Image Analysis and Processing* (pp. 402-413). Springer, Cham.

Phi, M., (2018, September 24). Illustrated Guide to LSTM's and GRU's: A step by step explanation. Retrieved from https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Pittman, M., & Reich, B. (2016). Social media and loneliness: Why an Instagram picture may be worth more than a thousand Twitter words. *Computers in Human Behavior*, *62*, 155-167.

Polgreen, P. M., Chen, Y., Pennock, D. M., Nelson, F. D., & Weinstein, R. A. (2008). Using internet searches for influenza surveillance. *Clinical infectious diseases*, *47*(11), 1443-1448.

Rogers, S., (2016, July 1). What is Google Trends data — and what does it mean? Retrieved from https://medium.com/google-news-lab/what-is-google-trends-data-and-what-does-it-mean-b48f07342ee8

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

Santillana, M., Nguyen, A. T., Dredze, M., Paul, M. J., Nsoesie, E. O., & Brownstein, J. S. (2015). Combining search, social media, and traditional data sources to improve influenza surveillance. *PLoS computational biology*, *11*(10).

Santillana, M., Nsoesie, E. O., Mekaru, S. R., Scales, D., & Brownstein, J. S. (2014). Using clinicians' search query data to monitor influenza epidemics. *Clinical infectious diseases: an official publication of the Infectious Diseases Society of America*, *59*(10), 1446.

Schmidhuber, J., Wierstra, D., Gagliolo, M., & Gomez, F. (2007). Training recurrent networks by evolino. *Neural computation*, *19*(3), 757-779.

Scikit-learn: sklearn.preprocessing.MinMaxScaler (2018). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Scikit-learn: sklearn.model_selection.TimeSeriesSplit (2020). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html

Seebach, C., Pahlke, I., & Beck, R. (2011). Tracking the digital footprints of customers: How firms can improve their sensing abilities to achieve business agility.

Siami-Namini, S., & Namin, A. S. (2018). Forecasting economics and financial time series: ARIMA vs. LSTM. *arXiv preprint arXiv:1803.06386*.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, *37*(2), 267-307.

Tan, S., Wang, Y., & Cheng, X. (2008, July). Combining learn-based and lexicon-based techniques for sentiment detection without using labeled examples. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 743-744).

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. *International journal of forecasting*, *16*(4), 437-450.

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

Tomczyk, E., & Doligalski, T. (2015). Predicting new car registrations: nowcasting with Google search and macroeconomic data. *E. Tomczyk, T. Doligalski, Predicting New Car Registrations: Nowcasting with Google Search and Macroeconomic Data,[in:] Sł. Partycki (ed.), E-społeczeństwo w Europie Środkowej i Wschodniej. Teraźniejszość i perspektywy rozwoju, Wydawnictwo KUL, Lublin*, *2015*, 228-236.

Trawiński, B., Smętek, M., Telec, Z., & Lasota, T. (2012). Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *International Journal of Applied Mathematics and Computer Science*, *22*(4), 867-881.

Turney, P. D. (2002, July). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 417-424). Association for Computational Linguistics.

Venkatesh, K., Ravi, V., Prinzie, A., & Van den Poel, D. (2014). Cash demand forecasting in ATMs by clustering and neural networks. *European Journal of Operational Research*, *232*(2), 383-392.

Wachter, P., Widmer, T., & Klein, A. (2019, September). Predicting Automotive Sales using Pre-Purchase Online Search Data. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 569-577). IEEE.

Wang, C. F., (2019, January 8). The Vanishing Gradient Problem: The Problem, Its Causes, Its Significance, and Its Solutions. Retrieved from https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

Weytjens, H., Lohmann, E., & Kleinsteuber, M. (2019). Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet. *Electronic Commerce Research*, 1-21.

Wilcoxon, F. (1946). Individual comparisons of grouped data by ranking methods. *Journal of economic entomology*, *39*(2), 269-270.

Wilson, D. R., & Martinez, T. R. (2003). The general inefficiency of batch training for gradient descent learning. *Neural networks*, *16*(10), 1429-1451.

Wu, L., & Brynjolfsson, E. (2009). The future of prediction: how Google searches foreshadow housing prices and quantities. *ICIS 2009 Proceedings*, 147.

Yu, X., Liu, Y., Huang, X., & An, A. (2010). Mining online reviews for predicting sales performance: A case study in the movie domain. *IEEE Transactions on Knowledge and Data engineering*, *24*(4), 720-734.

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., & Liu, B. (2011). Combining lexicon-based and learning-based methods for Twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, *89*.

Appendices

Appendix 1. 200 car models and their total sales of 2019

Rank	Car model	Total sales
1	Tesla Model 3	29.948
2	Volkswagen Polo	12.964
3	Ford Focus	10.517
4	Volkswagen Golf	9.296
5	Kia Niro	9.249
6	Renault Clio	9.056
7	Ford Fiesta	8.945
8	Toyota Aygo	8.659
9	Peugeot 108	8.434
10	Kia Picanto	8.014
11	Opel Karl	7.991
12	Hyundai Kona	7.153
13	Volkswagen Up	7.153
14	Opel Crossland X	6.182
15	Toyota Yaris	6.084
16	Skoda Octavia	6.023
17	Opel Astra	5.963
18	Volkswagen Tiguan	5.871
19	Volkswagen T-Roc	5.734
20	Renault Captur	5.733
21	Nissan Qashqai	5.725
22	Peugeot 208	5.144
23	BMW 3-serie	5.092
24	Toyota Corolla	4.816
25	Peugeot 3008	4.811
26	Hyundai i10	4.712
27	Citroën C1	4.637
28	Opel Grandland X	4.553
29	Volvo XC40	4.461
30	Volvo V40	4.432

(Continued)

Rank	Car model	Total sales
31	Mini Mini	4.268
32	Mercedes-Benz A-klasse	4.264
33	Audi E-tron	4.119
34	Mazda CX-5	4.101
35	Citroën C3	3.966
36	Nissan Leaf	3.817
37	Peugeot 308	3.787
38	Renault Mégane	3.775
39	Toyota C-HR	3.673
40	Volvo V60	3.650
41	Volkswagen T-Cross	3.644
42	Opel Corsa	3.413
43	Seat Ibiza	3.240
44	Skoda Fabia	3.216
45	Nissan Micra	3.205
46	Skoda Kodiaq	3.191
47	Peugeot 5008	3.190
48	Audi A3	3.124
49	Fiat 500	2.899
50	BMW 1-serie	2.888
51	Mercedes-Benz C-klasse	2.867
52	BMW i3	2.861
53	Peugeot 2008	2.771
54	Kia Ceed	2.759
55	Suzuki Swift	2.746
56	Mitsubishi Space Star	2.665
57	Toyota RAV4	2.532
58	Skoda Karoq	2.511
59	Ford EcoSport	2.510
60	Mazda 2	2.494
61	BMW 5-serie	2.448
62	Mercedes-Benz B-klasse	2.385

(Continued)

Rank	Car model	Total sales
63	Seat Arona	2.303
64	Mercedes-Benz CLA	2.299
65	Audi A1	2.254
66	Renault Zoe	2.210
67	Citroën C5 Aircross	2.178
68	Suzuki Ignis	2.152
69	Seat Ateca	2.100
70	Kia Stonic	2.081
71	Renault Twingo	2.058
72	Hyundai Ioniq	2.056
73	Mitsubishi Outlander	2.003
74	Hyundai i20	1.977
75	Mazda CX-3	1.933
76	Seat Leon	1.923
77	BMW X1	1.878
78	Mercedes-Benz Sprinter	1.849
79	Audi A4	1.824
80	Volvo XC60	1.698
81	Citroën C3 Aircross	1.696
82	Mini Countryman	1.690
83	Renault Kadjar	1.645
84	BMW 2-serie Tourer	1.641
85	Suzuki Celerio	1.554
86	Opel Insignia	1.512
87	Mercedes-Benz E-klasse	1.511
88	Skoda Superb	1.501
89	Volkswagen Passat	1.413
90	Renault Scénic	1.404
91	Suzuki Vitara	1.380
92	Citroën C4 Cactus	1.332
93	Audi Q2	1.288
94	Peugeot 508	1.269

(Continued)

Rank	Car model	Total sales
95	BMW 4-serie	1.211
96	Mazda 3	1.202
97	Ford Kuga	1.180
98	Skoda Scala	1.159
99	Mercedes-Benz Vito	1.155
100	Mini Clubman	1.140
101	BMW X3	1.123
102	Skoda Citigo	1.120
103	Citroën C4 SpaceTourer	1.081
104	Audi A5	1.079
105	Dacia Duster	1.056
106	Dacia Sandero	1.056
107	Kia Sportage	1.048
108	MG ZS	1.020
109	Mercedes-Benz GLC	964
110	Jeep Compass	960
111	Mercedes-Benz GLA	925
112	Hyundai Tucson	920
113	Mitsubishi Eclipse Cross	920
114	Audi Q3	903
115	Kia Rio	899
116	Audi A6	884
117	Opel Ampera-e	882
118	Mitsubishi ASX	880
119	Dacia Logan	805
120	BMW X5	798
121	Ford Mondeo	772
122	Jaguar I-Pace	770
123	Mazda CX-30	770
124	Seat Mii	720
125	Opel Mokka	703
126	Volvo V90	698

(Continued)

Rank	Car model	Total sales
127	Toyota Auris	658
128	Kia Proceed	656
129	Hyundai i30	651
130	Opel Adam	619
131	Suzuki S-Cross	605
132	Ford Transit Custom	603
133	Volvo S60	591
134	Volvo XC90	575
135	BMW X2	572
136	Volkswagen Transporter	546
137	Seat Tarraco	527
138	Tesla Model S	527
139	Honda Jazz	504
140	Volkswagen Touran	468
141	Tesla Model X	467
142	Volkswagen Arteon	467
143	Ford Ka+	455
144	Nissan Juke	435
145	Fiat 500X	413
146	Mazda 6	411
147	Jeep Renegade	398
148	Honda CR-V	390
149	Fiat Panda	383
150	Porsche Macan	382
151	DS 7 Crossback	373
152	Porsche 911	367
153	Land Rover Range Rover Evoque	343
154	Skoda Rapid	341
155	Nissan X-Trail	336
156	Land Rover Range Rover Sport	332
157	Porsche Cayenne	331
158	Mercedes-Benz V-klasse	325

(Continued)

Rank	Car model	Total sales
159	Ford C-MAX	315
160	DS 3 Crossback	313
161	BMW X4	311
162	Renault Talisman	311
163	Honda HR-V	307
164	Subaru Forester	298
165	BMW Z4	288
166	Smart Forfour	285
167	Alfa Romeo Giulia	274
168	Kia Optima	272
169	BMW 2-serie	266
170	Lexus CT	265
171	Lexus UX	263
172	Honda Civic	259
173	Volkswagen Golf Sportsvan	256
174	Land Rover Range Rover Velar	234
175	Suzuki Baleno	234
176	Volkswagen Caddy	222
177	Audi Q5	221
178	Volvo S90	221
179	Land Rover Range Rover	214
180	BMW 7-serie	211
181	Alfa Romeo Mito	210
182	Toyota Proace	208
183	Porsche Panamera	206
184	Mazda MX-5	186
185	Mercedes-Benz GLE	180
186	Renault Espace	180
187	Fiat Tipo	177
188	Suzuki Jimny	171
189	Peugeot Rifter	168
190	Alfa Romeo Stelvio	163

(Continued)

Rank	Car model	Total sales
191	Lexus ES	159
192	Land Rover Discovery Sport	156
193	Toyota Prius	156
194	BMW 8-serie	150
195	Mercedes-Benz S-klasse	150
196	BMW 6-serie GT	147
197	Opel Movano	145
198	Subaru XV	142
199	Dacia Lodgy	140
200	Jaguar XE	140

Appendix 2. 200 car models and their related hashtags

Car model	Hashtags
Alfa Romeo Giulia	#alfaromeogiulia
Alfa Romeo Mito	#alfaromeomito
Alfa Romeo Stelvio	#stelvio
Audi A1	#audia1
Audi A3	#audia3
Audi A4	#audia4
Audi A5	#audia5
Audi A6	#audia6
Audi E-tron	#etron
Audi Q2	#audiq2
Audi Q3	#audiq3
Audi Q5	#audiq5
BMW 1-serie	#bmw1series
BMW 2-serie	#bmw2
BMW 2-serie Tourer	#bmw2seriesactivetourer
BMW 3-serie	#bmw3series
BMW 4-serie	#4series
BMW 5-serie	#5series
BMW 6-serie GT	#bmw6gt
BMW 7-serie	#7series
BMW 8-serie	#8series
BMW i3	#bmwi3
BMW X1	#bmwx1
BMW X2	#bmwx2
BMW X3	#bmwx3
BMW X4	#bmwx4
BMW X5	#bmwx5
BMW Z4	#bmwz4
Citroën C3 Aircross	#c3aircross
CitroënC1	#citroenc1
CitroënC3	#citroenc3

(Continued)

Car model	Hashtags
CitroënC4 Cactus	#c4cactus
CitroënC4 SpaceTourer	#c4spacetourer
CitroënC5 Aircross	#c5aircross
Dacia Duster	#daciaduster
Dacia Lodgy	#lodgy
Dacia Logan	#dacialogan
Dacia Sandero	#daciasandero
DS3 Crossback	#ds3crossback
DS7 Crossback	#ds7crossback
Fiat 500	#fiat500
Fiat 500X	#fiat500x
Fiat Panda	#fiatpanda
Fiat Tipo	#fiattipo
Ford C-MAX	#fordcmax
Ford EcoSport	#ecosport
Ford Fiesta	#fordfiesta
Ford Focus	#fordfocus
Ford Ka+	#fordkaplus
Ford Kuga	#fordkuga
Ford Mondeo	#fordmondeo
Ford Transit Custom	#transitcustom
Honda Civic	#hondacivic
Honda CR-V	#hondacrv
Honda HR-V	#hondahrv
Honda Jazz	#hondajazz
Hyundai i10	#hyundaii10
Hyundai i20	#hyundaii20
Hyundai i30	#hyundaii30
Hyundai Ioniq	#ioniq
Hyundai Kona	#hyundaikona
Hyundai Tucson	#hyundaitucson
Jaguar I-Pace	#ipace

(Continued)

Car model	Hashtags
Jaguar XE	#jaguarxe
Jeep Compass	#jeepcompass
Jeep Renegade	#jeeprenegade
Kia Ceed	#kiaceed
Kia Niro	#kianiro
Kia Optima	#kiaoptima
Kia Picanto	#kiapicanto
Kia Proceed	#kiaproceed
Kia Rio	#kiario
Kia Sportage	#kiasportage
Kia Stonic	#stonic
Land Rover Discovery Sport	#landroverdiscoverysport
Land Rover Range Rover	#landroverrangerover
Land Rover Range Rover Evoque	#evoque
Land Rover Range Rover Sport	#rangeroversport
Land Rover Range Rover Velar	#velar
Lexus CT	#lexusct
Lexus ES	#lexuses
Lexus UX	#lexusux
Mazda 2	#mazda2
Mazda 3	#mazda3
Mazda 6	#mazda6
Mazda CX-3	#mazdacx3
Mazda CX-30	#cx30
Mazda CX-5	#mazdacx5
Mazda MX-5	#mazdamx5
Mercedes-Benz A-klasse	#mercedesaclass
Mercedes-Benz B-klasse	#mercedesbclass
Mercedes-Benz C-klasse	#cclass
Mercedes-Benz CLA	#mercedescla
Mercedes-Benz E-klasse	#eclass
Mercedes-Benz GLA	#mercedesgla

(Continued)

Car model	Hashtags
Mercedes-Benz GLC	#mercedesglc
Mercedes-Benz GLE	#gle
Mercedes-Benz S-klasse	#mercedessclass
Mercedes-Benz Sprinter	#mercedessprinter
Mercedes-Benz Vito	#mercedesvito
Mercedes-Benz V-klasse	#mercedesvclass
MG ZS	#mgzs
Mini Clubman	#miniclubman
Mini Countryman	#minicountryman
Mini Mini	#minicooper
Mitsubishi ASX	#mitsubishiasx
Mitsubishi Eclipse Cross	#eclipsecross
Mitsubishi Outlander	#mitsubishioutlander
Mitsubishi Space Star	#mitsubishispacestar
Nissan Juke	#nissanjuke
Nissan Leaf	#nissanleaf
Nissan Micra	#nissanmicra
Nissan Qashqai	#nissanqashqai
Nissan X-Trail	#nissanxtrail
Opel Adam	#opeladam
Opel Ampera-e	#amperae
Opel Astra	#opelastra
Opel Corsa	#opelcorsa
Opel Crossland X	#crosslandx
Opel Grandland X	#grandlandx
Opel Insignia	#opelinsignia
Opel Karl	#opelkarl
Opel Mokka	#opelmokka
Opel Movano	#opelmovano
Peugeot 108	#peugeot108
Peugeot 2008	#peugeot2008
Peugeot 208	#peugeot208

(Continued)

Car model	Hashtags
Peugeot 3008	#peugeot3008
Peugeot 308	#peugeot308
Peugeot 5008	#peugeot5008
Peugeot 508	#peugeot508
Peugeot Rifter	#rifter
Porsche 911	#porsche911
Porsche Cayenne	#cayenne
Porsche Macan	#porschemacan
Porsche Panamera	#panamera
Renault Captur	#renaultcaptur
Renault Clio	#renaultclio
Renault Espace	#renaultespace
Renault Kadjar	#kadjar
Renault Mégane	#renaultmegane
Renault Scénic	#renaultscenic
Renault Talisman	#renaulttalisman
Renault Twingo	#renaulttwingo
Renault Zoe	#renaultzoe
Seat Arona	#seatarona
Seat Ateca	#seatateca
Seat Ibiza	#seatibiza
Seat Leon	#seatleon
Seat Mii	#seatmii
Seat Tarraco	#seattarraco
Skoda Citigo	#skodacitigo
Skoda Fabia	#skodafabia
Skoda Karoq	#karoq
Skoda Kodiaq	#kodiaq
Skoda Octavia	#skodaoctavia
Skoda Rapid	#skodarapid
Skoda Scala	#skodascala
Skoda Superb	#skodasuperb

(Continued)

Car model	Hashtags
Smart Forfour	#smartforfour
Subaru Forester	#subaruforester
Subaru XV	#subaruxv
Suzuki Baleno	#suzukibaleno
Suzuki Celerio	#celerio
Suzuki Ignis	#suzukiignis
Suzuki Jimny	#jimny
Suzuki S-Cross	#suzukiscross
Suzuki Swift	#suzukiswift
Tesla Model 3	#teslamodel3
Tesla Model S	#teslamodels
Tesla Model X	#modelx
Toyota Auris	#toyotaauris
Toyota Aygo	#toyotaaygo
Toyota C-HR	#toyotachr
Toyota Corolla	#Corolla
Toyota Prius	#prius
Toyota Prius+	#priusplus
Toyota Proace	#proace
Toyota RAV4	#toyotarav4
Toyota Yaris	#toyotayaris
Volkswagen Arteon	#arteon
Volkswagen Caddy	#vwcaddy
Volkswagen Golf	#volkswagengolf
Volkswagen Golf Sportsvan	#golfsportsvan
Volkswagen Passat	#passat
Volkswagen Polo	#volkswagenpolo
Volkswagen T-Cross	#tcross
Volkswagen Tiguan	#volkswagentiguan
Volkswagen Touran	#volkswagentouran
Volkswagen Transporter	#volkswagentransporter
Volkswagen T-Roc	#troc

(Continued)

Car model	Hashtags
Volkswagen Up	#volkswagenup
Volvo S60	#s60
Volvo S90	#s90
Volvo V40	#volvov40
Volvo V60	#volvov60
Volvo V90	#volvov90
Volvo XC40	#volvoxc40
Volvo XC60	#xc60
Volvo XC90	#volvoxc90

Appendix 3. 200 car models and their related search terms

Car model	Search term
Alfa Romeo Giulia	alfa romeo giulia
Alfa Romeo Mito	alfa romeo mito
Alfa Romeo Stelvio	alfa romeo stelvio
Audi A1	audi a1
Audi A3	audi a3
Audi A4	audi a4
Audi A5	audi a5
Audi A6	audi a6
Audi E-tron	audi e-tron
Audi Q2	audi q2
Audi Q3	audi q3
Audi Q5	audi q5
BMW 1-serie	bmw 1-serie
BMW 2-serie	bmw 2-serie
BMW 2-serie Tourer	bmw 2-serie tourer
BMW 3-serie	bmw 3-serie
BMW 4-serie	bmw 4-serie
BMW 5-serie	bmw 5-serie
BMW 6-serie GT	bmw 6-serie gt
BMW 7-serie	bmw 7-serie
BMW 8-serie	bmw 8-serie
BMW i3	bmw i3
BMW X1	bmw x1
BMW X2	bmw x2
BMW X3	bmw x3
BMW X4	bmw x4
BMW X5	bmw x5
BMW Z4	bmw z4
Citroën C3 Aircross	citroën c3 aircross+citroen c3 aircross
Citroën C1	citroën c1+citroen c1

(Continued)

Car model	Search term
Citroën C3	citroën c3+citroen c3
Citroën C4 Cactus	citroën c4 cactus+citroen c4 cactus
Citroën C4 SpaceTourer	citroën c4 space tourer +citroen c4 space tourer
Citroën C5 Aircross	citroën c5 aircross+citroen c5-aircross
Dacia Duster	dacia duster
Dacia Lodgy	dacia lodgy
Dacia Logan	dacia logan
Dacia Sandero	dacia sandero
DS3 Crossback	ds3 crossback
DS7 Crossback	ds7 crossback
Fiat 500	fiat 500
Fiat 500X	fiat 500x
Fiat Panda	fiat panda
Fiat Tipo	fiat tipo
Ford C-MAX	ford c-max
Ford EcoSport	ford ecosport
Ford Fiesta	ford fiesta
Ford Focus	ford focus
Ford Ka+	ford ka+
Ford Kuga	ford kuga
Ford Mondeo	ford mondeo
Ford Transit Custom	ford transit custom
Honda Civic	honda civic
Honda CR-V	honda cr-v
Honda HR-V	honda hr-v
Honda Jazz	honda jazz
Hyundai i10	hyundai i10
Hyundai i20	hyundai i20
Hyundai i30	hyundai i30

(Continued)

Car model	Search term
Hyundai Ioniq	hyundai ioniq
Hyundai Kona	hyundai kona
Hyundai Tucson	hyundai tucson
Jaguar I-Pace	jaguar i-pace
Jaguar XE	jaguar xe
Jeep Compass	jeep compass
Jeep Renegade	jeep renegade
Kia Ceed	kia ceed
Kia Niro	kia niro
Kia Optima	kia optima
Kia Picanto	kia picanto
Kia Proceed	kia proceed
Kia Rio	kia rio
Kia Sportage	kia sportage
Kia Stonic	kia stonic
Land Rover Discovery Sport	land rover discovery sport
Land Rover Range Rover	land rover range rover
Land Rover Range Rover Evoque	land rover range rover evoque
Land Rover Range Rover Sport	land rover range rover sport
Land Rover Range Rover Velar	land rover range rover velar
Lexus CT	lexus ct
Lexus ES	lexus es
Lexus UX	lexus ux
Mazda 2	mazda 2
Mazda 3	mazda 3
Mazda 6	mazda 6
Mazda CX-3	mazda cx-3
Mazda CX-30	mazda cx-30
Mazda CX-5	mazda cx-5
Mazda MX-5	mazda mx-5
Mercedes-Benz A-klasse	mercedes-benz a-klasse
Mercedes-Benz B-klasse	mercedes-benz b-klasse

(Continued)

Car model	Search term
Mercedes-Benz C-klasse	mercedes-benz c-klasse
Mercedes-Benz CLA	mercedes-benz cla
Mercedes-Benz E-klasse	mercedes-benz e-klasse
Mercedes-Benz GLA	mercedes-benz gla
Mercedes-Benz GLC	mercedes-benz glc
Mercedes-Benz GLE	mercedes-benz gle
Mercedes-Benz S-klasse	mercedes-benz s-klasse
Mercedes-Benz Sprinter	mercedes-benz sprinter
Mercedes-Benz Vito	mercedes-benz vito
Mercedes-Benz V-klasse	mercedes-benz v-klasse
MG ZS	mg zs
Mini Clubman	mini clubman
Mini Countryman	mini cooper
Mini Mini	mini countryman
Mitsubishi ASX	mitsubishi asx
Mitsubishi Eclipse Cross	mitsubishi eclipse cross
Mitsubishi Outlander	mitsubishi outlander
Mitsubishi Space Star	mitsubishi space star
Nissan Juke	nissan juke
Nissan Leaf	nissan leaf
Nissan Micra	nissan micra
Nissan Qashqai	nissan qashqai
Nissan X-Trail	nissan x-trial
Opel Adam	opel adam
Opel Ampera-e	opel ampera-e
Opel Astra	opel astra
Opel Corsa	opel corsa
Opel Crossland X	opel crossland x
Opel Grandland X	opel grandland
Opel Insignia	opel insignia
Opel Karl	opel karl
Opel Mokka	opel mokka

(Continued)

Car model	Search term
Opel Movano	opel movano
Peugeot 108	peugeot 108
Peugeot 2008	peugeot 2008
Peugeot 208	peugeot 208
Peugeot 3008	peugeot 3008
Peugeot 308	peugeot 308
Peugeot 5008	peugeot 5008
Peugeot 508	peugeot 508
Peugeot Rifter	peugeot rifter
Porsche 911	porsche 911
Porsche Cayenne	porsche cayenne
Porsche Macan	porsche macan
Porsche Panamera	porsche panamera
Renault Captur	renault captur
Renault Clio	renault clio
Renault Espace	renault espace
Renault Kadjar	renault kadjar
Renault Mégane	renault mégane+renault megane
Renault Scénic	Renault scénic+renault scenic
Renault Talisman	renault talisman
Renault Twingo	renault twingo
Renault Zoe	renault zoe
Seat Arona	seat arona
Seat Ateca	seat ateca
Seat Ibiza	seat ibiza
Seat Leon	seat leon
Seat Mii	seat mii
Seat Tarraco	seat tarraco
Skoda Citigo	skoda citigo
Skoda Fabia	skoda fabia
Skoda Karoq	skoda karoq

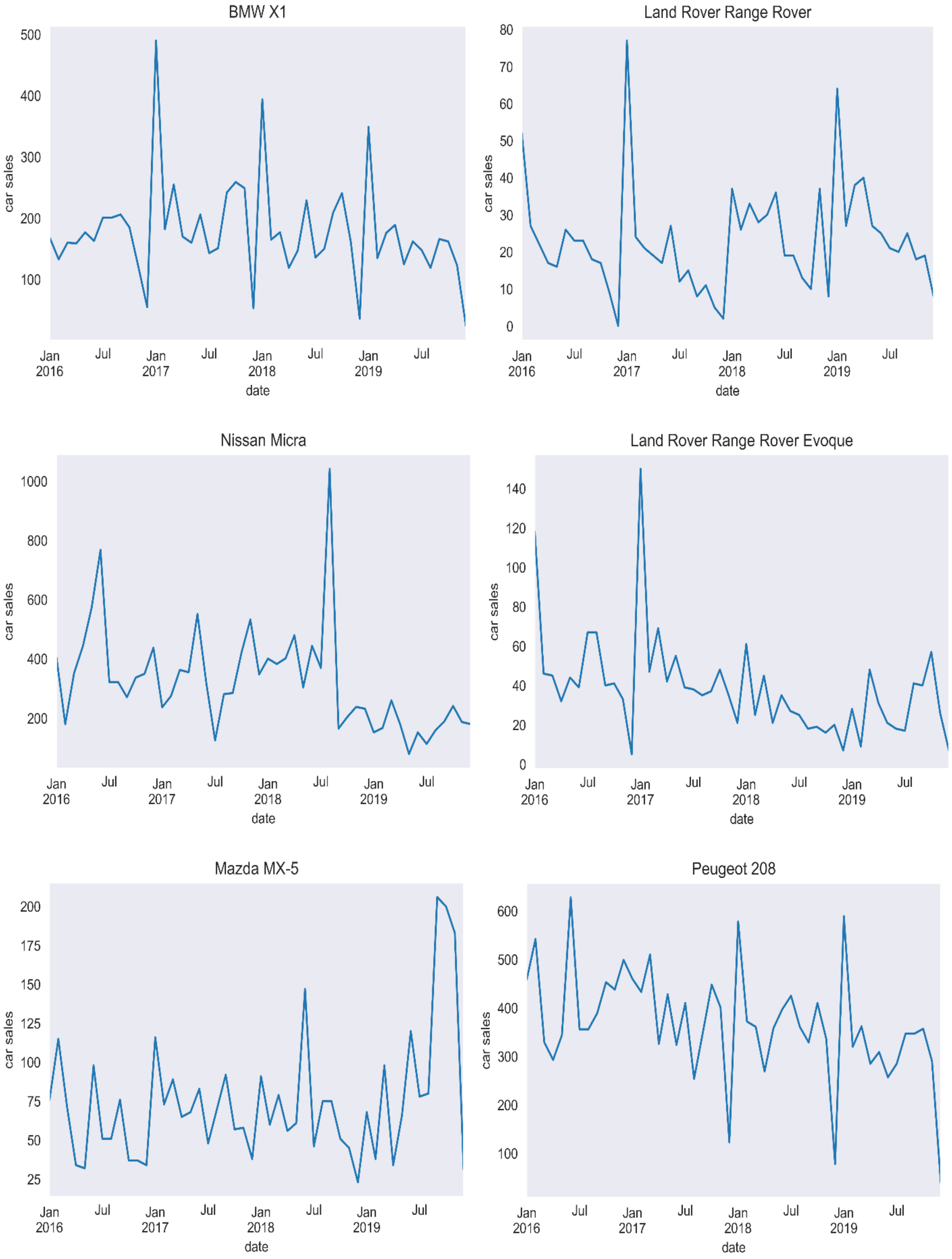
(Continued)

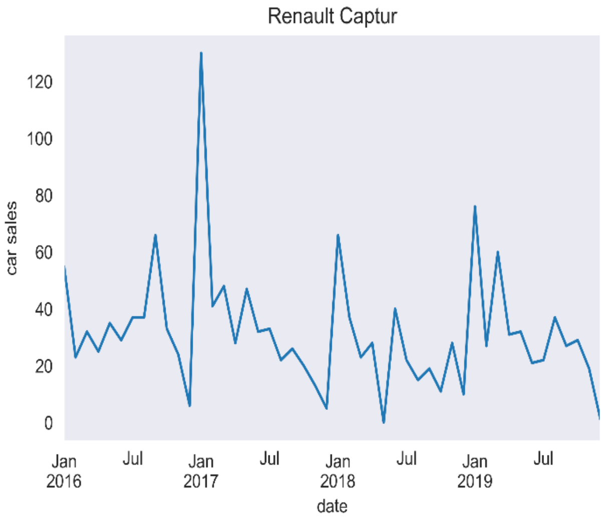
Car model	Search term
Skoda Kodiaq	skoda kodiaq
Skoda Octavia	skoda octavia
Skoda Rapid	skoda rapid
Skoda Scala	skoda scala
Skoda Superb	skoda superb
Smart Forfour	smart forfour
Subaru Forester	subaru forester
Subaru XV	subaru xv
Suzuki Baleno	suzuki baleno
Suzuki Celerio	suzuki celerio
Suzuki Ignis	suzuki ignis
Suzuki Jimny	suzuki jimny
Suzuki S-Cross	suzuki s-cross
Suzuki Swift	suzuki swift
Tesla Model 3	tesla model 3
Tesla Model S	tesla model s
Tesla Model X	tesla model x
Toyota Auris	toyota auris
Toyota Aygo	toyota aygo
Toyota C-HR	toyota c-hr
Toyota Corolla	toyota corolla
Toyota Prius	toyota prius
Toyota Prius+	toyota prius s+
Toyota Proace	toyota proace
Toyota RAV4	toyota rav4
Toyota Yaris	toyota yaris
Volkswagen Arteon	volkswagen arteon
Volkswagen Caddy	volkswagen caddy
Volkswagen Golf	volkswagen golf
Volkswagen Golf Sportsvan	volkswagen golf sportsvan
Volkswagen Passat	volkswagen passat
Volkswagen Polo	volkswagen polo

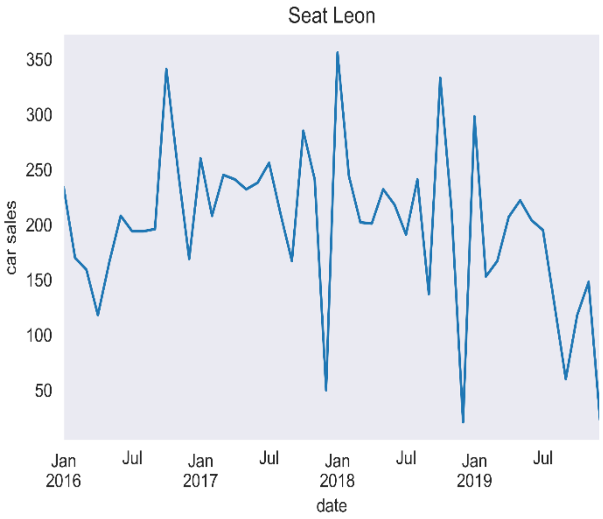
(Continued)

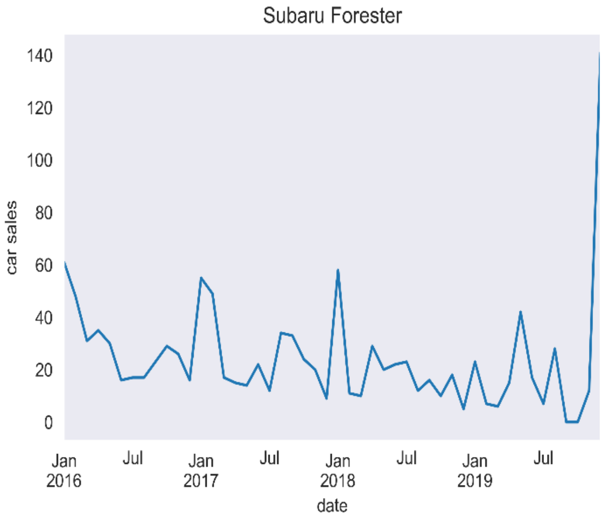
Car model	Search term
Volkswagen T-Cross	volkswagen t-cross
Volkswagen Tiguan	volkswagen tiguan
Volkswagen Touran	volkswagen touran
Volkswagen Transporter	volkswagen transporter
Volkswagen T-Roc	volkswagen t-roc
Volkswagen Up	volkswagen up
Volvo S60	volvo s60
Volvo S90	volvo s90
Volvo V40	volvo v40
Volvo V60	volvo v60
Volvo V90	volvo v90
Volvo XC40	volvo xc40
Volvo XC60	volvo xc60
Volvo XC90	volvo xc90

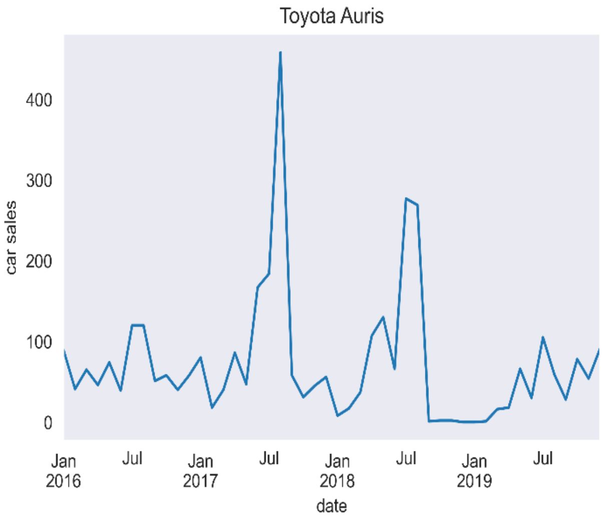
Appendix 4. Clustering: Car sales from January 2016 until December 2019 of the cluster centroids

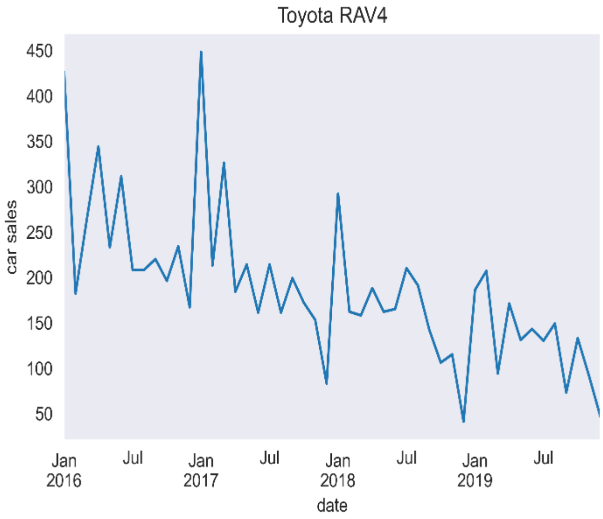


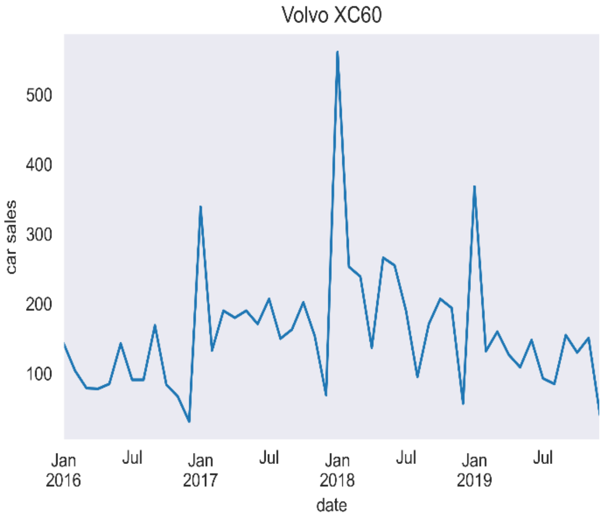

Renault Captur


Seat Leon


Subaru Forester


Toyota Auris


Toyota RAV4


Volvo XC60

Appendix 5. Clustering: Cluster distribution of the car models

Cluster number	Car model
1	Land Rover Range Rover
1	Audi A1
1	Audi A3
1	Audi A6
1	Audi Q3
1	BMW 2-serie
1	BMW 3-serie
1	BMW 7-serie
1	BMW X4
1	BMW Z4
1	Citroën C4 Cactus
1	Dacia Lodgy
1	Dacia Sandero
1	Ford Fiesta
1	Ford Kuga
1	Hyundai i10
1	Kia Optima
1	Kia Sportage
1	Land Rover Range Rover Sport
1	Mazda 2
1	Mazda 3
1	Mazda CX-3
1	Mazda CX-5
1	Mercedes-Benz Vito
1	Mini Countryman
1	Mitsubishi Outlander
1	Nissan Leaf
1	Nissan Qashqai
1	Opel Adam
1	Opel Astra
1	Peugeot 308
1	Peugeot 508
1	Porsche Cayenne
1	Renault Mégane
1	Renault Talisman
1	Seat Ibiza
1	Skoda Octavia
1	Smart Forfour
1	Subaru XV
1	Tesla Model S
1	Toyota Aygo
1	Volkswagen Golf
1	Volkswagen Passat
1	Volkswagen Polo
1	Volkswagen Transporter

(Continued)

Cluster number	Car model
1	Volvo V40
1	Volvo XC90
2	Nissan Micra
2	Audi A4
2	Toyota Prius
2	Volkswagen Caddy
2	Audi Q5
2	BMW 2-serie Tourer
2	Volkswagen Touran
2	BMW X3
2	Volvo V60
2	Fiat 500X
2	Ford Focus
2	Ford Mondeo
2	Honda CR-V
2	Kia Ceed
2	Land Rover Discovery Sport
2	Mazda 6
2	Mercedes-Benz A-klasse
2	Mercedes-Benz CLA
2	Mercedes-Benz GLC
2	Mercedes-Benz S-klasse
2	Mercedes-Benz V-klasse
2	Mitsubishi ASX
2	Nissan X-Trail
2	Porsche Macan
2	Renault Scénic
2	Renault Twingo
2	Skoda Citigo
2	Suzuki Celerio
2	Suzuki Jimny
2	Suzuki Vitara
3	Seat Leon
3	Mercedes-Benz C-klasse
3	Volkswagen Tiguan
3	Mini Clubman
3	BMW 4-serie
3	Volvo S60
3	BMW X5
3	Fiat Panda
3	Ford C-MAX
3	Renault Clio
3	Honda Jazz
3	Renault Kadjar
3	Renault Zoe
3	Seat Mii

(Continued)

Cluster number	Car model
3	Skoda Fabia
3	Skoda Rapid
3	Suzuki Swift
4	Peugeot 208
4	Ford EcoSport
4	Audi A5
4	Toyota Yaris
4	Porsche Panamera
4	Volkswagen Golf Sportsvan
4	Honda Civic
4	Renault Espace
4	Hyundai i20
4	Kia Picanto
4	Citroën C1
4	Dacia Duster
4	Lexus CT
4	Peugeot 108
5	BMW X1
5	Mercedes-Benz B-klasse
5	Mercedes-Benz GLA
5	BMW 5-serie
5	Volkswagen Up
5	Mini Mini
5	Mitsubishi Space Star
5	BMW i3
5	Opel Corsa
5	Opel Mokka
5	Fiat 500
5	Peugeot 2008
6	Mazda MX-5
6	Peugeot 5008
6	Mercedes-Benz GLE
6	Honda HR-V
6	Mercedes-Benz Sprinter
6	Skoda Superb
6	Dacia Logan
7	Toyota RAV4
7	Jeep Renegade
7	Opel Insignia
7	Opel Karl
7	Hyundai i30
8	Land Rover Range Rover Evoque
8	Citroën C3
8	Hyundai Tucson
8	Suzuki S-Cross
8	Jaguar XE

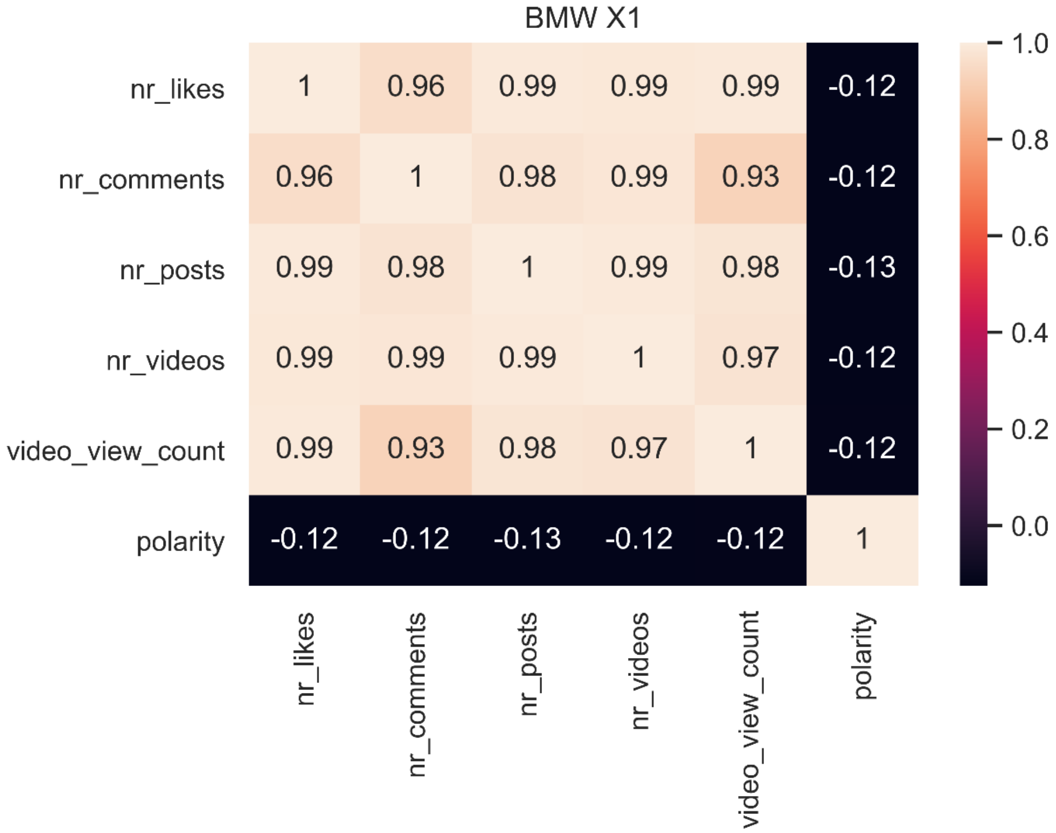
(Continued)

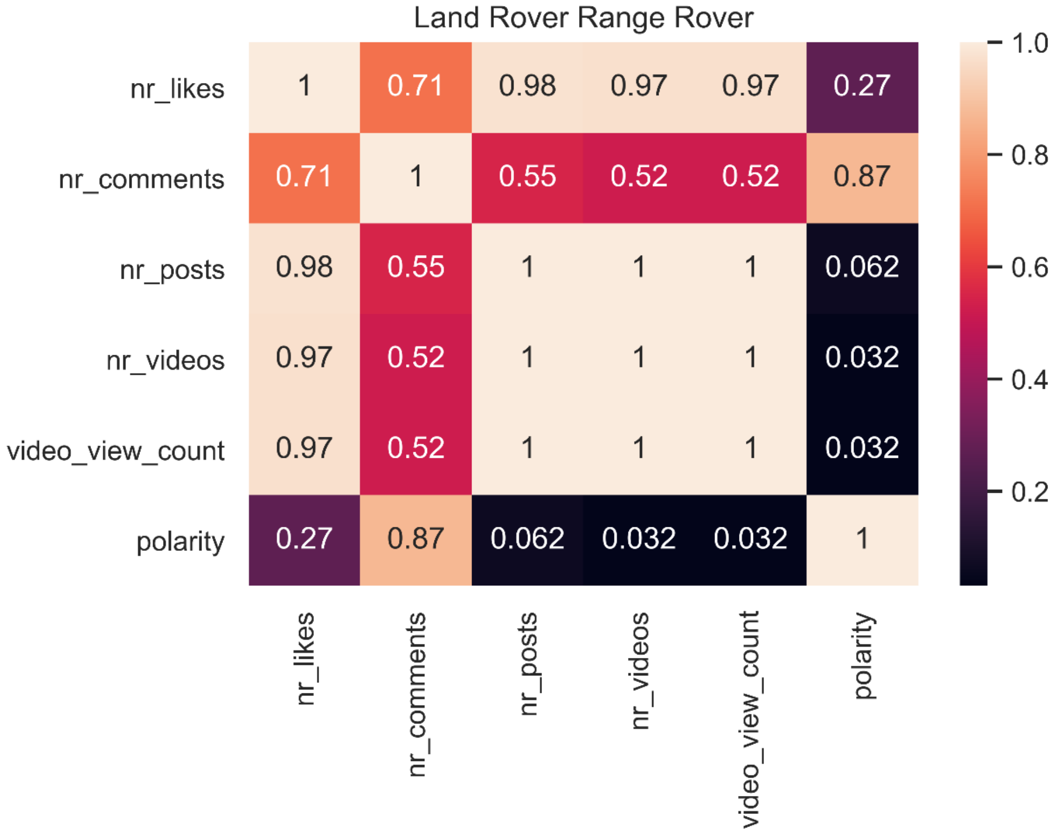
Cluster number	Car model
9	Renault Captur
9	Kia Rio
9	Mercedes-Benz E-klasse
10	Volvo XC60
10	Nissan Juke
11	Toyota Auris
11	Alfa Romeo Mito
12	Subaru Forester
12	BMW 1-serie

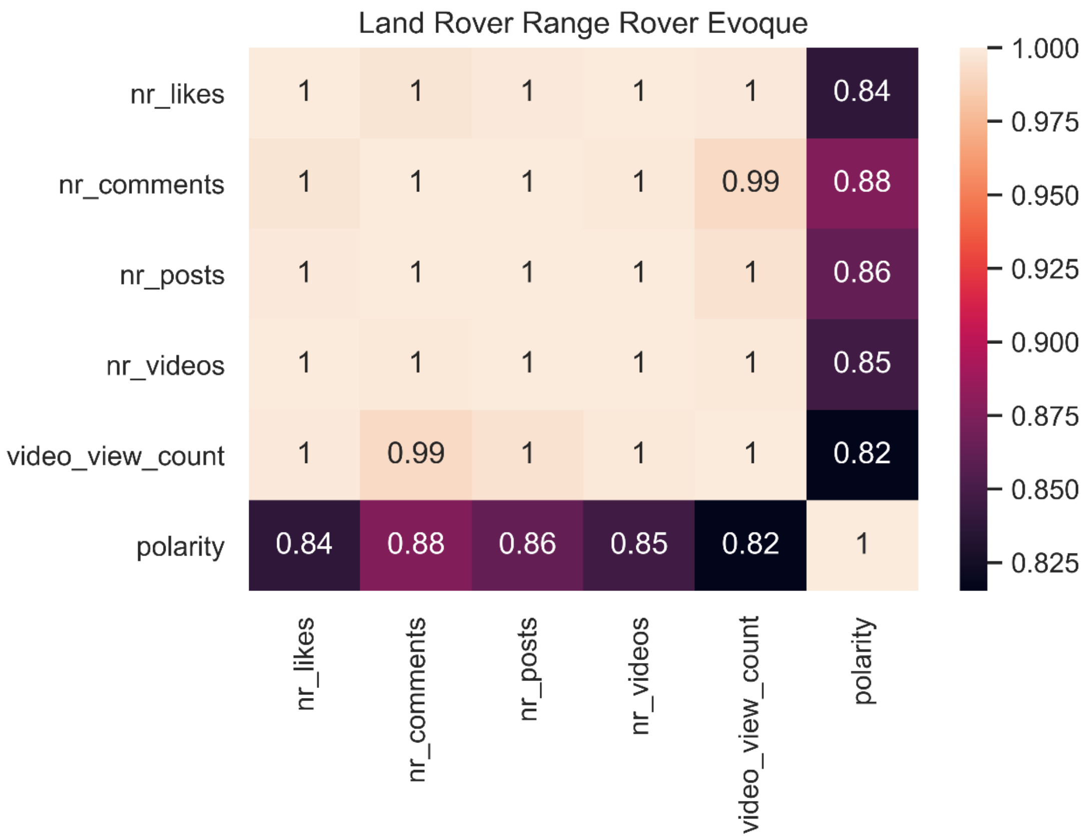
Appendix 6. *VIF*

	VIF	***VIF***	***VIF***	***VIF***	***VIF***	***VIF***	**Average**
Car model	**nr_likes**	**nr_comments**	**nr_posts**	**nr_videos**	**video_view_count**	**polarity**	***VIF***
BMW X1	460,67	305,25	178,78	778,12	354,58	1,02	346,40333
Land Rover Range Rover	∞	∞	∞	∞	∞	∞	∞
Land Rover Range Rover Evoque	∞	∞	∞	∞	∞	150,94	∞
Mazda MX-5	39516	2072	5980	13404	200	1,04	10195,507
Nissan Micra	∞	∞	∞	∞	∞	∞	∞
Peugeot 208	12895290	1238991	1323633	4940351	205208,9	1,29	3433912,5
Renault Captur	∞	∞	∞	∞	∞	∞	∞
Seat Leon	∞	∞	∞	∞	∞	∞	∞
Subaru Forester	86835	∞	87186	∞	∞	5,53	∞
Toyota Auris	268,8	263,7	1517,51	1945,21	42,46	1,06	673,12333
Toyota RAV4	8145433	1738018	2014538	9475561	666342,7	1,18	3673315,6
Volvo XC60	∞	∞	∞	∞	∞	∞	∞

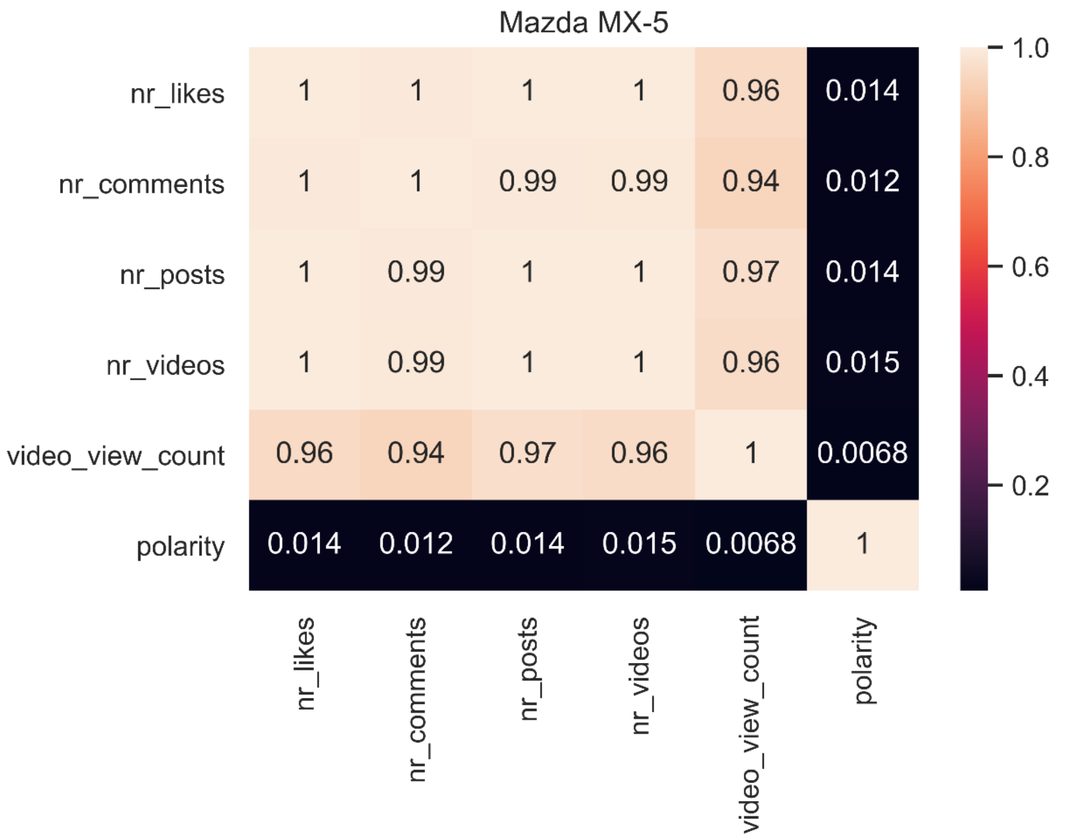
Appendix 7. Pearson correlation between the Instagram features



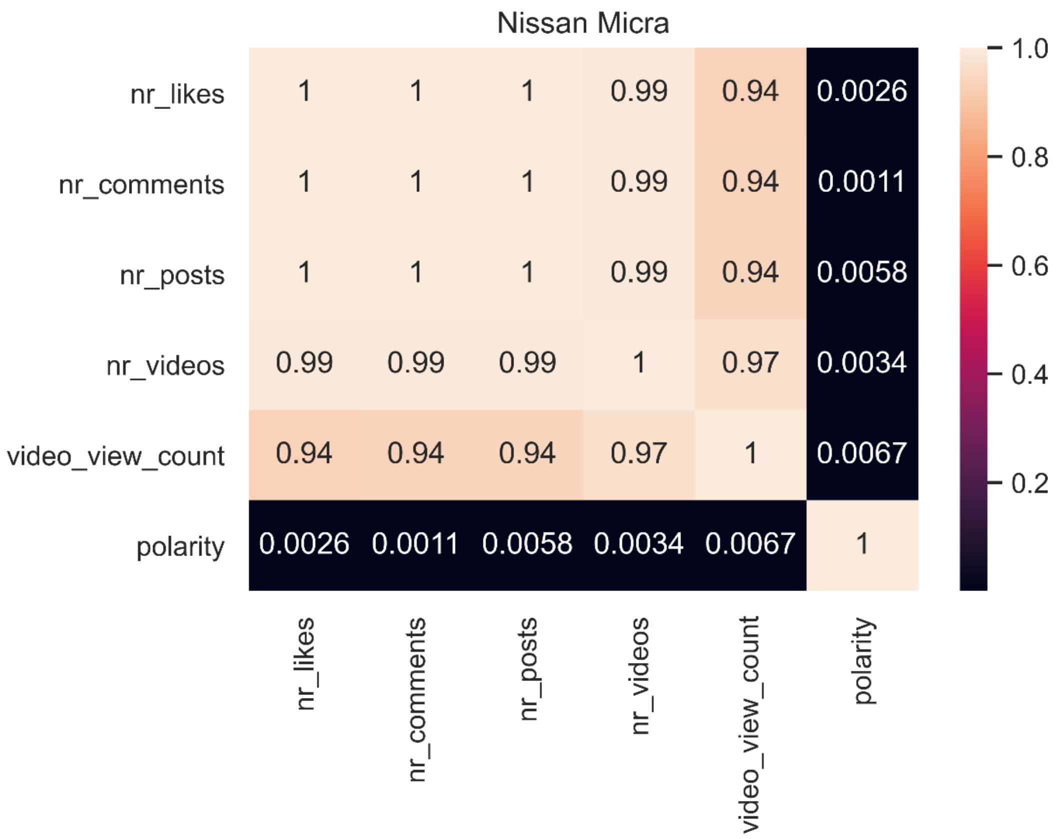


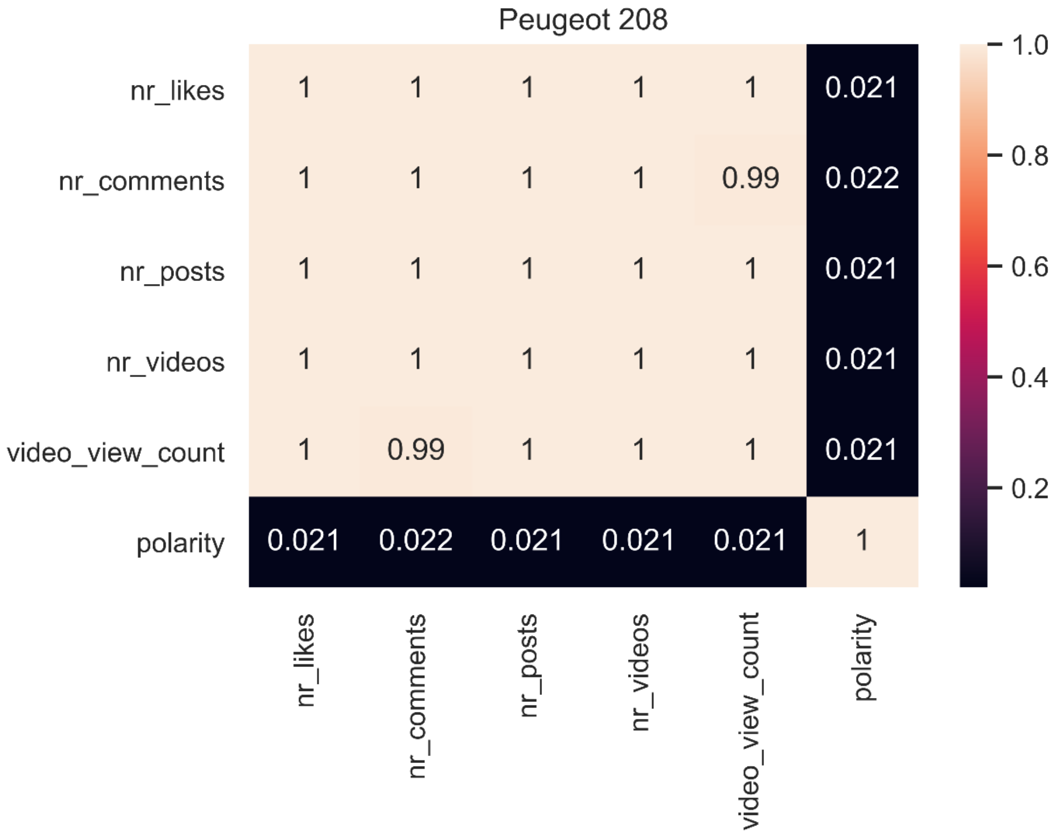


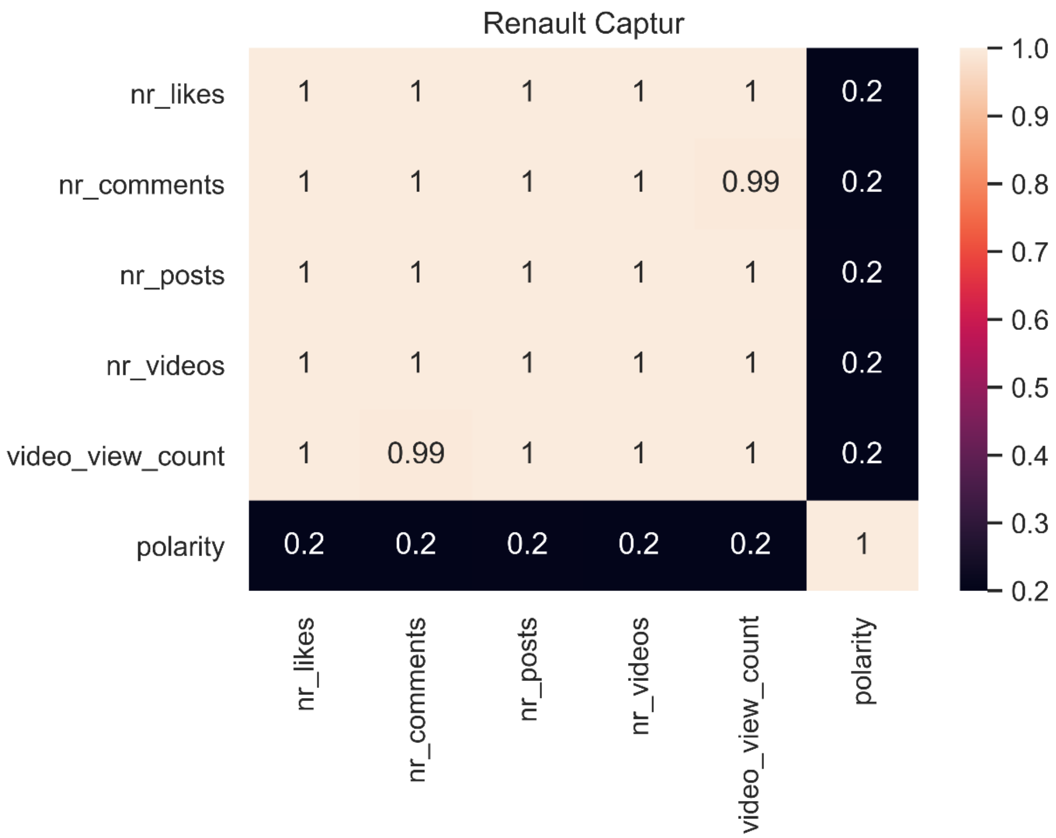
Land Rover Range Rover Evoque

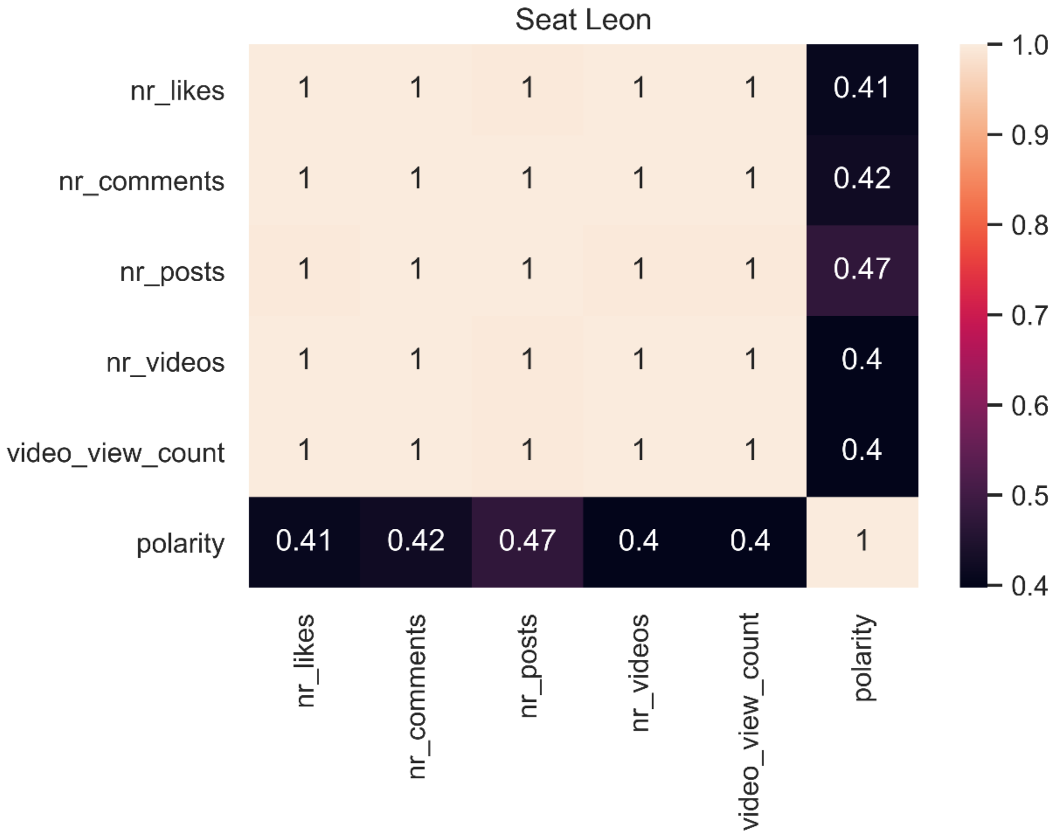


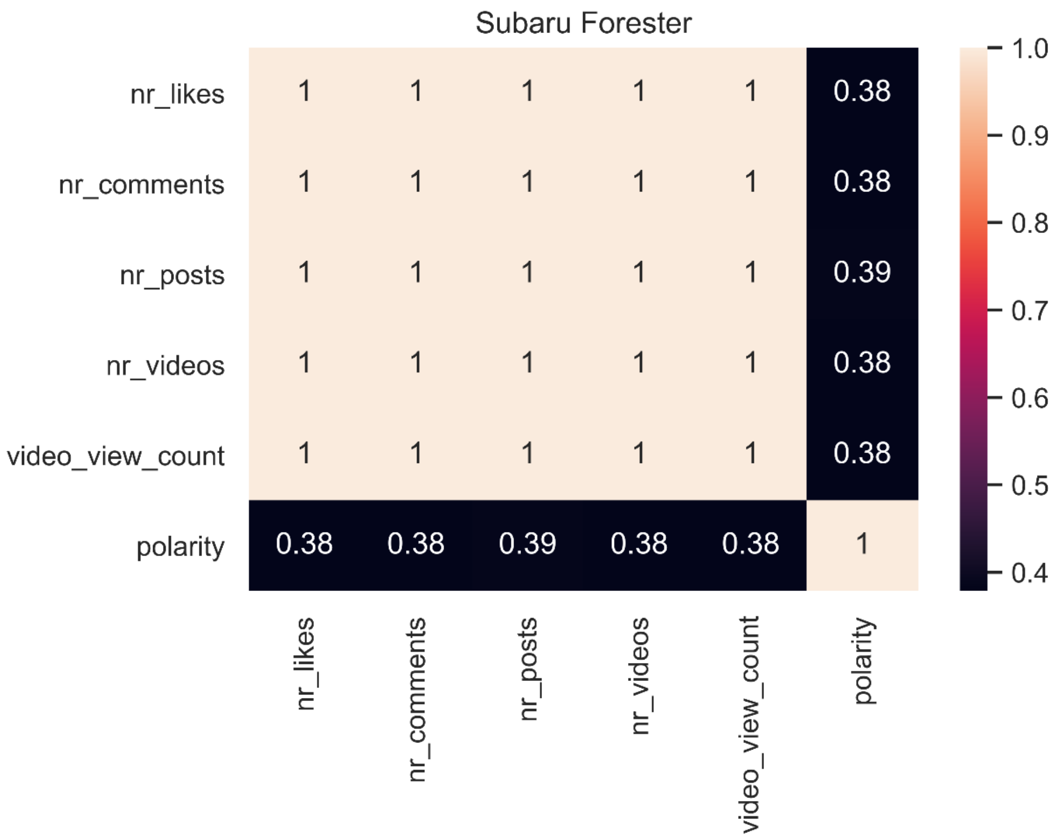
Mazda MX-5


Nissan Micra
nr_likes
nr_comments
nr_posts
nr_videos
video_view_count
polarity
1 1 1 0.99 0.94 0.0026
1 1 1 0.99 0.94 0.0011
1 1 1 0.99 0.94 0.0058
0.99 0.99 0.99 1 0.97 0.0034
0.94 0.94 0.94 0.97 1 0.0067
0.0026 0.0011 0.0058 0.0034 0.0067 1
nr_likes
nr_comments
nr_posts
nr_videos
video_view_count
polarity
1.0
0.8
0.6
0.4
0.2

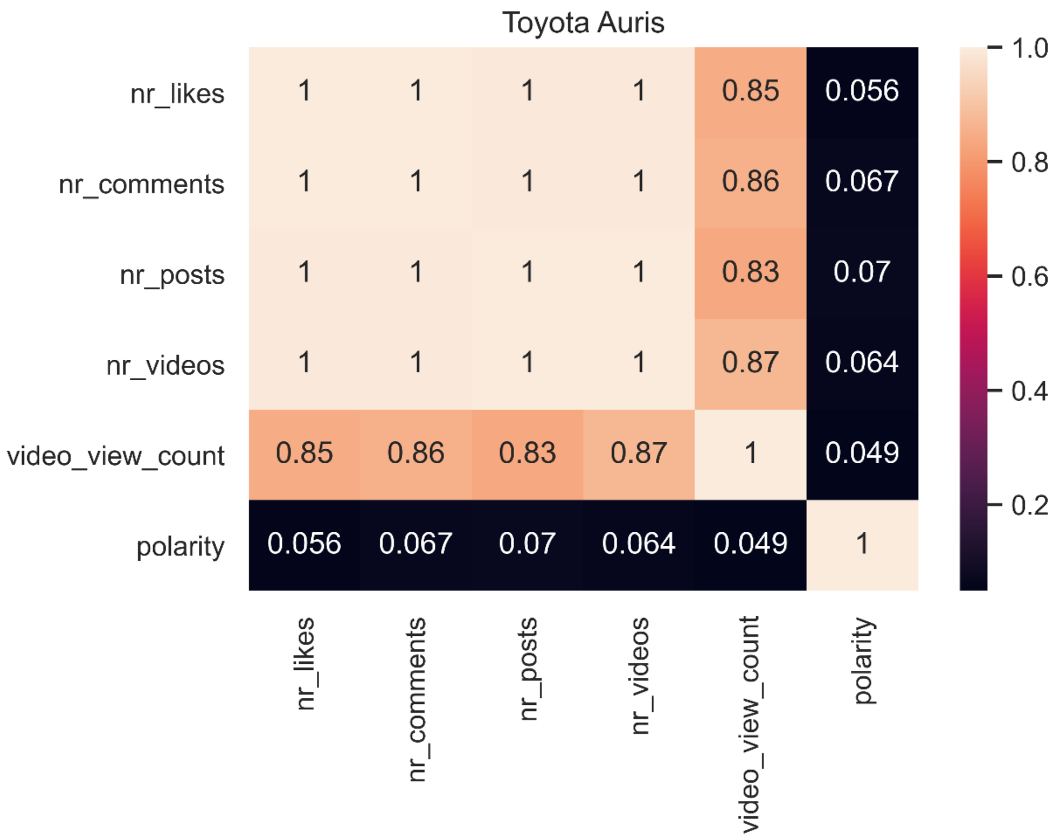

Peugeot 208
nr_likes
nr_comments
nr_posts
nr_videos
video_view_count
polarity
1 1 1 1 1 0.021
1 1 1 1 0.99 0.022
1 1 1 1 1 0.021
1 1 1 1 1 0.021
1 0.99 1 1 1 0.021
0.021 0.022 0.021 0.021 0.021 1
nr_likes
nr_comments
nr_posts
nr_videos
video_view_count
polarity
1.0
0.8
0.6
0.4
0.2


Renault Captur
nr_likes
nr_comments
nr_posts
nr_videos
video_view_count
polarity
1 1 1 1 1 0.2
1 1 1 1 0.99 0.2
1 1 1 1 1 0.2
1 1 1 1 1 0.2
1 0.99 1 1 1 0.2
0.2 0.2 0.2 0.2 0.2 1
nr_likes
nr_comments
nr_posts
nr_videos
video_view_count
polarity
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2

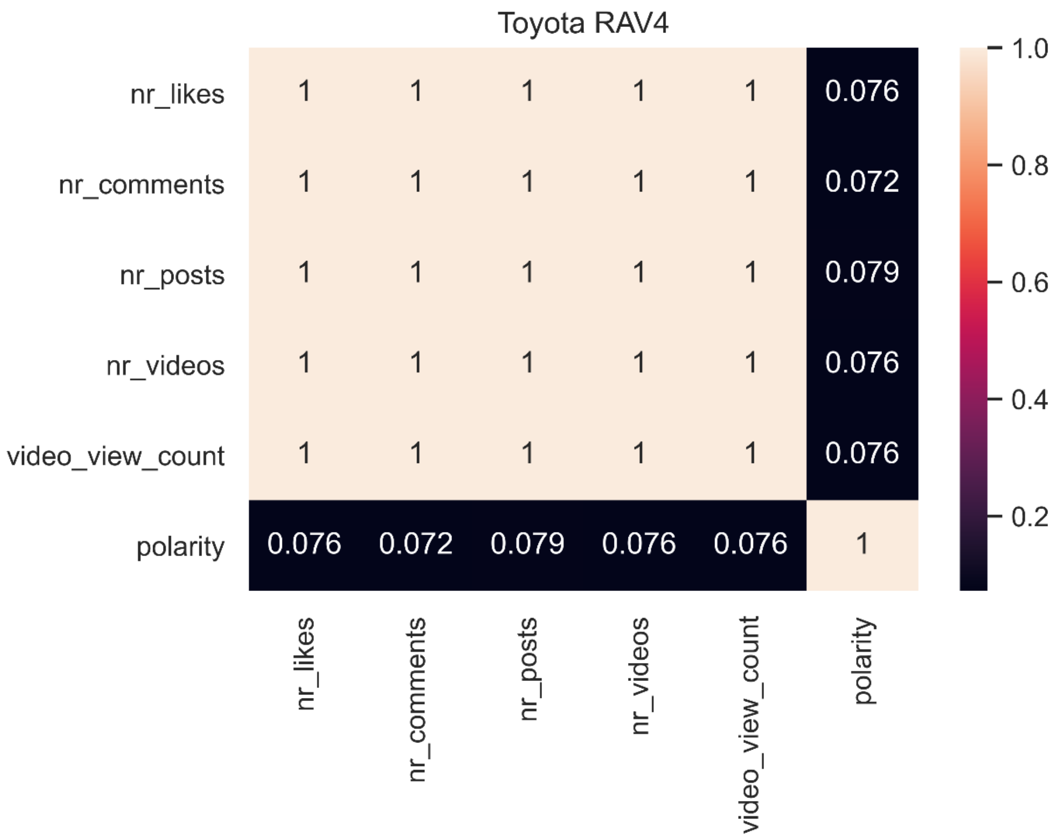

Seat Leon
nr_likes
nr_comments
nr_posts
nr_videos
video_view_count
polarity
1 1 1 1 1 0.41
1 1 1 1 1 0.42
1 1 1 1 1 0.47
1 1 1 1 1 0.4
1 1 1 1 1 0.4
0.41 0.42 0.47 0.4 0.4 1
nr_likes
nr_comments
nr_posts
nr_videos
video_view_count
polarity
1.0
0.9
0.8
0.7
0.6
0.5
0.4

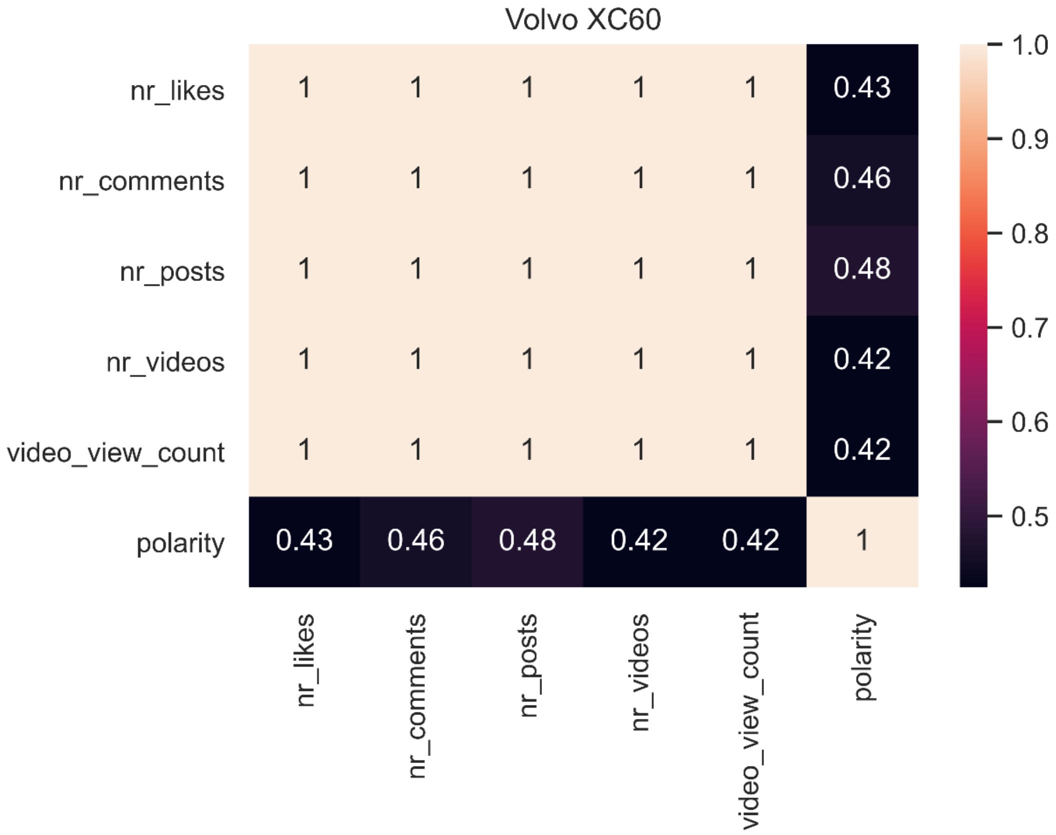


Subaru Forester



Toyota Auris


Toyota RAV4
nr_likes
nr_comments
nr_posts
nr_videos
video_view_count
polarity
1 1 1 1 1 0.076
1 1 1 1 1 0.072
1 1 1 1 1 0.079
1 1 1 1 1 0.076
1 1 1 1 1 0.076
0.076 0.072 0.079 0.076 0.076 1
nr_likes
nr_comments
nr_posts
nr_videos
video_view_count
polarity
1.0
0.8
0.6
0.4
0.2


Volvo XC60
nr_likes
nr_comments
nr_posts
nr_videos
video_view_count
polarity
1 1 1 1 1 0.43
1 1 1 1 1 0.46
1 1 1 1 1 0.48
1 1 1 1 1 0.42
1 1 1 1 1 0.42
0.43 0.46 0.48 0.42 0.42 1
nr_likes
nr_comments
nr_posts
nr_videos
video_view_count
polarity
1.0
0.9
0.8
0.7
0.6
0.5

Appendix 8. Hyperparameters of *LSTM* using dataset 1

Car model	***LSTM* size**	**Batch size**	**Learning rate**
BMW X1	128	10	0,01
Land Rover Range Rover	32	11	0,1
Land Rover Range Rover Evoque	128	11	0,3
Mazda MX-5	32	10	0,3
Nissan Micra	64	8	0,3
Peugeot 208	8	8	0,2
Renault Captur	64	2	0,3
Seat Leon	64	6	0,1
Subaru Forester	64	10	0,2
Toyota Auris	64	6	0,1
Toyota RAV4	128	4	0,2
Volvo XC60	64	11	0,01

Appendix 9. Hyperparameters of *LSTM* using dataset 2

Car model	***LSTM* size**	**Batch size**	**Learning rate**
BMW X1	128	4	0,01
Land Rover Range Rover	32	11	0,01
Land Rover Range Rover Evoque	64	8	0,3
Mazda MX-5	32	6	0,3
Nissan Micra	128	10	0,01
Peugeot 208	64	10	0,3
Renault Captur	128	4	0,3
Seat Leon	128	2	0,01
Subaru Forester	64	6	0,3
Toyota Auris	128	11	0,2
Toyota RAV4	64	2	0,3
Volvo XC60	32	10	0,01

Appendix 10. Hyperparameters of *LSTM* using dataset 3

Car model	***LSTM* size**	**Batch size**	**Learning rate**
BMW X1	128	4	0,01
Land Rover Range Rover	16	1	0,3
Land Rover Range Rover Evoque	128	10	0,3
Mazda MX-5	128	11	0,01
Nissan Micra	128	10	0,3
Peugeot 208	64	6	0,1
Renault Captur	64	2	0,3
Seat Leon	128	4	0,1
Subaru Forester	32	10	0,3
Toyota Auris	8	8	0,2
Toyota RAV4	32	2	0,3
Volvo XC60	8	1	0,1

Appendix 11. Hyperparameter values of *LSTM* using dataset 4

Car model	*LSTM* size	Batch size	Learning rate
BMW X1	32	10	0,01
Land Rover Range Rover	8	11	0,2
Land Rover Range Rover Evoque	8	4	0,2
Mazda MX-5	128	8	0,3
Nissan Micra	16	10	0,01
Peugeot 208	128	11	0,3
Renault Captur	64	8	0,01
Seat Leon	8	1	0,3
Subaru Forester	128	8	0,2
Toyota Auris	32	8	0,1
Toyota RAV4	16	1	0,01
Volvo XC60	8	2	0,2

Appendix 12. Hyperparameter values of *ARIMA* using dataset 4

Car model	***p***	***d***	***q***	***P***	***D***	***Q***	***S***
BMW X1	0	0	4	0	0	0	12
Land Rover Range Rover	0	1	4	0	0	0	12
Land Rover Range Rover Evoque	0	0	1	0	0	0	12
Mazda MX-5	0	0	2	0	0	0	12
Nissan Micra	0	0	2	2	0	0	12
Peugeot 208	0	0	2	1	0	0	12
Renault Captur	0	0	4	1	0	0	12
Seat Leon	0	0	4	1	0	0	12
Subaru Forester	0	0	2	1	0	0	12
Toyota Auris	0	0	2	1	0	0	12
Toyota RAV4	0	0	1	1	0	0	12
Volvo XC60	0	0	4	0	0	0	12

Appendix 13. Performance of *ARIMA* using dataset 4

Car model	MAE	RMSE
Alfa Romeo Mito	19,16853172	22,34119731
Audi A1	194,1533983	202,8615189
Audi A3	289,6633451	309,1393044
Audi A4	104,7231532	130,8617898
Audi A5	63,41043571	71,73124888
Audi A6	62,46635613	65,59004511
Audi Q3	102,8930959	112,5473266
Audi Q5	10,77308993	14,61737087
BMW 1-serie	226,1861641	327,0372408
BMW 2-serie	183,0399817	223,5389284
BMW 2-serie Tourer	134,5714374	144,3875082
BMW 3-serie	508,3445333	540,6804823
BMW 4-serie	125,6340427	152,5337587
BMW 5-serie	84,8719619	115,4167834
BMW 7-serie	31,51195183	33,42349353
BMW i3	233,2069107	276,8032271
BMW X1	71,33924742	98,50176561
BMW X3	93,22553438	112,3767811
BMW X4	46,90825169	47,82254301
BMW X5	87,07650069	103,4034488
BMW Z4	31,84770371	34,4652456
Citroën C1	296,1351136	316,8682532
Citroën C3	301,5684126	325,4519842
Citroën C4 Cactus	100,4886157	104,4615617
Dacia Duster	35,43454848	38,16553228
Dacia Lodgy	27,51198711	36,41816541
Dacia Logan	47,21845584	54,45667754
Dacia Sandero	65,27454375	73,03058197
Fiat 500	151,8675098	159,1802365
Fiat 500X	28,89106003	36,16898168
Fiat Panda	37,46257288	40,04095575
Ford C-MAX	32,96078725	36,86690155
Ford EcoSport	161,7236138	177,9296782
Ford Fiesta	440,7832187	476,4513299
Ford Focus	547,8002065	632,249468
Ford Kuga	194,4268719	233,1514893
Ford Mondeo	50,0833382	63,09055427
Honda Civic	23,16998816	26,93175883
Honda CR-V	26,20418096	30,81760668
Honda HR-V	11,34166064	12,79806021
Honda Jazz	33,877479	37,19406495
Hyundai i10	291,4215878	311,8247219
Hyundai i20	71,94433854	87,54348718
Hyundai i30	23,4269263	27,43991866
Hyundai Tucson	44,37953861	50,23041206

(Continued)

Car model	MAE	RMSE
Jaguar XE	47,54191897	52,51981798
Jeep Renegade	23,23111851	26,90513538
Kia Ceed	78,44299831	90,24839851
Kia Optima	16,78751766	17,91127041
Kia Picanto	416,6565639	453,9355596
Kia Rio	27,58697269	33,10486987
Kia Sportage	73,99810536	77,79544467
Land Rover Discovery Sport	14,09672525	16,23540032
Land Rover Range Rover	26,99986161	27,41557107
Land Rover Range Rover Evoque	20,31601088	23,86092923
Land Rover Range Rover Sport	44,98492433	45,94577144
Lexus CT	12,22097778	12,77982885
Mazda 2	119,2154326	128,5892136
Mazda 3	470,5681897	488,8941561
Mazda 6	22,70726438	25,26311878
Mazda CX-3	91,01085062	96,71093583
Mazda CX-5	240,4380848	266,3165611
Mazda MX-5	31,82571822	36,58866692
Mercedes-Benz A-klasse	197,9752424	216,9813042
Mercedes-Benz B-klasse	123,6711832	145,6182171
Mercedes-Benz C-klasse	174,0605579	202,3312826
Mercedes-Benz CLA	222,086651	254,478744
Mercedes-Benz E-klasse	90,71128063	107,7487898
Mercedes-Benz GLA	77,13471361	92,83731777
Mercedes-Benz GLC	104,5142247	121,84759
Mercedes-Benz GLE	12,68447644	14,48842655
Mercedes-Benz S-klasse	10,96008656	12,32091849
Mercedes-Benz Sprinter	222,2083296	269,9203353
Mercedes-Benz Vito	84,43076626	89,57106866
Mercedes-Benz V-klasse	27,10034029	33,57686821
Mini Clubman	85,1739881	87,86848917
Mini Countryman	130,075961	138,2402435
Mini Mini	482,225687	500,1651878
Mitsubishi ASX	102,7979573	113,0824645
Mitsubishi Outlander	139,3636966	147,3932941
Mitsubishi Space Star	207,3825288	252,1205745
Nissan Juke	134,5656761	185,8592069
Nissan Leaf	365,4289525	400,5364058
Nissan Micra	130,3432677	166,8618044
Nissan Qashqai	317,60721	331,3906167
Nissan X-Trail	86,5161861	93,4818119
Opel Adam	20,72524221	21,8966095
Opel Astra	527,020403	544,7990999
Opel Corsa	381,5613907	548,1569909
Opel Insignia	84,86815133	111,450964

(Continued)

Car model	MAE	RMSE
Opel Karl	864,3713517	956,7801923
Opel Mokka	47,83843529	53,5993458
Peugeot 108	242,1370701	251,1754272
Peugeot 2008	128,8142961	149,0660777
Peugeot 208	252,9264603	310,3709046
Peugeot 308	321,0525247	332,5626342
Peugeot 5008	95,20939502	109,6367186
Peugeot 508	114,0597086	122,0022336
Porsche Cayenne	128,6617192	135,3254443
Porsche Macan	29,76015708	32,75428485
Porsche Panamera	9,352680199	10,57829367
Renault Captur	213,6749532	226,602179
Renault Clio	395,0786074	417,8878398
Renault Espace	21,34685334	24,29798228
Renault Kadjar	104,4474314	119,9625419
Renault Mégane	504,1651816	523,5489815
Renault Scénic	178,5556877	187,1531867
Renault Talisman	25,48134956	26,26134362
Renault Twingo	68,5633706	82,59673654
Renault Zoe	422,3714671	483,8970022
Seat Ibiza	285,6774014	315,8210385
Seat Leon	92,3749165	116,4846271
Seat Mii	45,52908142	54,01599965
Skoda Citigo	117,7238578	134,5341166
Skoda Fabia	127,1105967	147,0934892
Skoda Octavia	408,0426298	441,0062598
Skoda Rapid	120,0361673	125,8495408
Skoda Superb	108,1459177	129,7859829
Smart Forfour	10,91598457	13,54845944
Subaru Forester	9,839618325	13,96594052
Subaru XV	18,19094107	19,66285211
Suzuki Celerio	104,3484629	109,5642289
Suzuki Jimny	10,58940349	11,24286401
Suzuki S-Cross	25,57311865	26,70148501
Suzuki Swift	215,7222323	248,9347383
Suzuki Vitara	108,0561684	117,5168063
Tesla Model S	80,44138209	86,95895153
Toyota Auris	83,22829917	96,20070796
Toyota Aygo	384,7403244	420,432586
Toyota Prius	10,3993055	12,27903311
Toyota RAV4	98,93312961	133,5903332
Toyota Yaris	263,3555295	326,4607498
Volkswagen Caddy	13,20125891	16,65051964
Volkswagen Golf	827,6115503	882,1248339
Volkswagen Golf Sportsvan	38,8258068	47,92690208
Volkswagen Passat	138,2141561	146,8705081

(Continued)

Car model	MAE	RMSE
Volkswagen Polo	1057,942127	1188,999481
Volkswagen Tiguan	189,3592443	213,9525843
Volkswagen Touran	64,41230759	72,62153602
Volkswagen Transporter	28,06357695	29,13653872
Volkswagen Up	192,0766684	260,2010248
Volvo S60	82,92702978	101,419722
Volvo V40	232,583158	239,7532243
Volvo V60	303,2876938	366,069724
Volvo XC60	46,75268932	51,55668035
Volvo XC90	62,32995786	68,2144774

Appendix 14. Performance of *LSTM* using dataset 4

Car model	MAE	RMSE
Alfa Romeo Mito	14,18421091	14,27467563
Audi A1	2938,825203	3324,412212
Audi A3	5942,981112	6609,593559
Audi A4	65,52638899	86,63387187
Audi A5	97,9489054	106,4257609
Audi A6	1392,948948	1473,578589
Audi Q3	1799,658427	2016,213934
Audi Q5	13,58637265	19,0874545
BMW 1-serie	183,2910287	254,2004322
BMW 2-serie	2909,843779	3398,266546
BMW 2-serie Tourer	53,53282057	62,06899571
BMW 3-serie	8005,098479	8916,845161
BMW 4-serie	65,19615609	76,06384901
BMW 5-serie	75,50079673	88,23740067
BMW 7-serie	417,920574	466,6234094
BMW i3	165,2157963	169,9885922
BMW X1	55,36737497	65,09913562
BMW X3	50,09640884	67,43720522
BMW X4	903,520528	1012,295362
BMW X5	52,89320265	60,15825583
BMW Z4	333,7894315	447,9190103
Citroën C1	98,61852417	103,5698710
Citroën C3	78,18916516	83,41857158
Citroën C4 Cactus	87,58447889	90,21878855
Dacia Duster	48,68768338	53,62624995
Dacia Lodgy	23,51698161	27,51684135
Dacia Logan	88,65372031	90,1823326
Dacia Sandero	2010,500968	2196,947324
Fiat 500	154,0091313	164,0397316
Fiat 500X	10,69517463	12,67864487
Fiat Panda	13,48250798	19,52295791
Ford C-MAX	30,73572677	30,73707763
Ford EcoSport	130,010182	137,6288065
Ford Fiesta	8063,974533	9060,252248
Ford Focus	239,7298933	341,7330373
Ford Kuga	2863,685569	3314,678366
Ford Mondeo	33,04763086	40,85174966
Honda Civic	29,79306902	30,69392053
Honda CR-V	7,669060843	12,17882602
Honda HR-V	36,1406814	37,16689093
Honda Jazz	18,82625634	22,14920847
Hyundai i10	5831,248004	6479,135834
Hyundai i20	97,86691965	102,7410219
Hyundai i30	18,02723503	19,690104
Hyundai Tucson	55,37755694	55,60655519

(Continued)

Car model	MAE	RMSE
Jaguar XE	24,46841235	27,00351952
Jeep Renegade	12,06595148	12,99693378
Kia Ceed	78,4548645	81,94568771
Kia Optima	713,9807797	778,7415105
Kia Picanto	773,0930808	789,0340724
Kia Rio	16,5106005	19,15536213
Kia Sportage	2117,948752	2241,783355
Land Rover Discovery Sport	5,763035502	7,614565388
Land Rover Range Rover	461,9728074	521,9942632
Land Rover Range Rover Evoque	16,15034703	22,58209836
Land Rover Range Rover Sport	577,9698432	646,2863913
Lexus CT	16,62481577	18,06791694
Mazda 2	2977,651978	3447,701166
Mazda 3	158,4168167	163,1861797
Mazda 6	7,074017797	8,902396565
Mazda CX-3	3385,956611	3786,863324
Mazda CX-5	6572,968284	7332,914418
Mazda MX-5	36,70963124	36,87216036
Mercedes-Benz A-klasse	160,9297682	196,1310323
Mercedes-Benz B-klasse	89,35169656	92,44159185
Mercedes-Benz C-klasse	62,028365	75,77321246
Mercedes-Benz CLA	98,99502128	122,9266234
Mercedes-Benz E-klasse	67,37234497	71,41614165
Mercedes-Benz GLA	55,04215676	57,10969978
Mercedes-Benz GLC	32,91017696	39,14264388
Mercedes-Benz GLE	58,39907728	61,46267938
Mercedes-Benz S-klasse	2,285565649	2,793358856
Mercedes-Benz Sprinter	289,5207193	290,5141852
Mercedes-Benz Vito	2811,166915	3184,739372
Mercedes-Benz V-klasse	8,622075626	8,882771014
Mini Clubman	41,919087	53,70159048
Mini Countryman	2183,230652	2431,444541
Mini Mini	205,0560897	216,6989354
Mitsubishi ASX	47,00187683	57,45933913
Mitsubishi Outlander	2929,374709	3266,873618
Mitsubishi Space Star	148,1788755	203,3997226
Nissan Juke	73,1383885	97,21238683
Nissan Leaf	11137,4118	12559,98494
Nissan Micra	56,01877485	63,01354836
Nissan Qashqai	5626,308051	6257,224691
Nissan X-Trail	23,15698448	27,18981614
Opel Adam	716,0447946	768,1382171
Opel Astra	10909,42035	12011,72433
Opel Corsa	281,1054077	445,711355
Opel Insignia	56,92158944	64,20774531

(Continued)

Car model	MAE	RMSE
Opel Karl	545,1723393	572,6695218
Opel Mokka	45,90787724	46,91425858
Peugeot 108	418,9368635	457,7387421
Peugeot 2008	100,0821915	121,8015907
Peugeot 208	440,164517	467,0329556
Peugeot 308	6607,494727	7135,526095
Peugeot 5008	405,5892639	417,3899491
Peugeot 508	1919,900652	2128,662358
Porsche Cayenne	2043,878148	2329,291373
Porsche Macan	7,393605777	11,27077133
Porsche Panamera	22,8273934	23,86632638
Renault Captur	120,9352984	134,9056668
Renault Clio	239,9393572	275,7086584
Renault Espace	10,6605602	10,68356302
Renault Kadjar	71,06450326	75,69921483
Renault Mégane	230,1851897	238,5168161
Renault Scénic	112,1531867	119,1118197
Renault Talisman	502,2468778	567,5936972
Renault Twingo	43,72544752	55,84813581
Renault Zoe	379,1947981	383,8624787
Seat Ibiza	5278,086549	5858,5138
Seat Leon	79,89526585	95,39480204
Seat Mii	36,71216856	40,288824
Skoda Citigo	77,83329882	100,4189481
Skoda Fabia	53,61403329	65,74163633
Skoda Octavia	9505,384782	11132,83532
Skoda Rapid	73,356587	73,35747035
Skoda Superb	170,1550075	176,0099973
Smart Forfour	302,0158697	342,1596145
Subaru Forester	23,51180458	33,68430487
Subaru XV	294,7640996	328,896411
Suzuki Celerio	38,44147164	41,29478708
Suzuki Jimny	4,609247753	7,243615488
Suzuki S-Cross	21,15579987	23,09259417
Suzuki Swift	88,002485	104,5800432
Suzuki Vitara	41,15616817	46,15618674
Tesla Model S	2308,111238	2733,653037
Toyota Auris	40,70228222	45,59281234
Toyota Aygo	205,6161667	220,0561767
Toyota Prius	3,712707179	5,235451835
Toyota RAV4	72,09261867	80,70056301
Toyota Yaris	326,816449	362,5970444
Volkswagen Caddy	5,018622194	5,897803841
Volkswagen Golf	19586,04347	22360,26377
Volkswagen Golf Sportsvan	25,42108318	25,53242917
Volkswagen Passat	2452,198201	2828,677397

(Continued)

Car model	MAE	RMSE
Volkswagen Polo	21624,64921	23811,85491
Volkswagen Tiguan	167,7158726	195,3738879
Volkswagen Touran	18,3382661	23,16679017
Volkswagen Transporter	874,557939	939,0207043
Volkswagen Up	242,4616394	267,0642826
Volvo S60	39,26424408	45,01797082
Volvo V40	5989,441515	6553,654205
Volvo V60	107,8541456	134,0949344
Volvo XC60	50,11415427	69,96114212
Volvo XC90	1222,111625	1341,048627

Appendix 15. Performance of *LSTM* using dataset 1

Car model	MAE	RMSE
Alfa Romeo Mito	13,04574476	13,28216294
Audi A1	53,81411525	75,40800398
Audi A3	98,66697475	131,6456515
Audi A4	70,10183934	78,4512519
Audi A5	34,74183982	42,08114087
Audi A6	21,71595274	26,17565058
Audi Q3	36,56094197	48,59290814
Audi Q5	16,48588617	17,93188687
BMW 1-serie	196,4476798	273,407109
BMW 2-serie	84,12002019	105,6159586
BMW 2-serie Tourer	56,44673593	68,93746724
BMW 3-serie	181,2630702	222,0271421
BMW 4-serie	49,5364794	58,44868038
BMW 5-serie	64,84209115	74,32113375
BMW 7-serie	11,50038637	14,55422516
BMW i3	152,6302774	153,2988343
BMW X1	52,93919155	58,96806817
BMW X3	55,5774231	56,59793104
BMW X4	10,59222821	11,79644356
BMW X5	45,48638153	56,61358344
BMW Z4	10,30439159	13,01523411
Citroën C1	116,5308862	131,8160985
Citroën C3	105,5587456	112,5684818
Citroën C4 Cactus	35,15984523	38,15169812
Dacia Duster	20,69557517	22,70437273
Dacia Logan	26,9008097	28,84574582
Dacia Logan	13,28531756	16,71144019
Dacia Sandero	33,31715993	36,66191997
Fiat 500	149,869097	160,0877505
Fiat 500X	20,53969029	24,0214694
Fiat Panda	14,02920641	22,72562385
Ford C-MAX	25,39542852	25,4311133
Ford EcoSport	51,15391432	60,42270794
Ford Fiesta	188,0981184	189,5572822
Ford Focus	353,3388149	397,5658118
Ford Kuga	77,56768908	98,2820399
Ford Mondeo	33,13828823	37,1146283
Honda Civic	7,333491462	8,471816131
Honda CR-V	19,34591184	20,29973819
Honda HR-V	7,460623877	10,57166992
Honda Jazz	17,72272709	20,05261692
Hyundai i10	114,6760864	129,7180069
Hyundai i20	43,60012163	51,9055736
Hyundai i30	27,65292249	30,84743831
Hyundai Tucson	30,72879342	31,18499067

(Continued)

Car model	MAE	RMSE
Jaguar XE	26,71618318	34,13186137
Jeep Renegade	13,40250887	16,94170532
Kia Ceed	29,05846732	33,96506142
Kia Optima	15,87169266	15,92748665
Kia Picanto	124,324864	177,0532707
Kia Rio	26,9191731	35,81183539
Kia Sportage	21,35994829	24,60491411
Land Rover Discovery Sport	8,443043845	9,000472508
Land Rover Range Rover	5,744615419	7,46782072
Land Rover Range Rover Evoque	36,66895933	42,95037866
Land Rover Range Rover Sport	12,44972338	16,36630284
Lexus CT	6,608043943	7,531169771
Mazda 2	61,65457589	65,1825212
Mazda 3	142,1313138	150,6167967
Mazda 6	14,39043835	15,46388699
Mazda CX-3	106,9745952	107,6504072
Mazda CX-5	119,0191084	126,6412823
Mazda MX-5	7,450141089	7,860369127
Mercedes-Benz A-klasse	150,3157218	156,4155018
Mercedes-Benz B-klasse	91,12888009	92,1535697
Mercedes-Benz C-klasse	125,1859785	141,3916408
Mercedes-Benz CLA	135,3659624	162,1198479
Mercedes-Benz E-klasse	45,12987954	53,99973457
Mercedes-Benz GLA	51,72367096	52,39865533
Mercedes-Benz GLC	101,1408648	108,2016496
Mercedes-Benz GLE	13,15707833	21,09126994
Mercedes-Benz S-klasse	13,12612179	13,39392079
Mercedes-Benz Sprinter	106,8198874	109,2236113
Mercedes-Benz Vito	53,83792332	55,16658335
Mercedes-Benz V-klasse	40,66754314	41,19753415
Mini Clubman	42,8634807	51,84057524
Mini Countryman	50,4250303	57,70814877
Mini Mini	403,0316187	415,1316177
Mitsubishi ASX	82,33903503	89,12099546
Mitsubishi Outlander	55,76971	62,35345285
Mitsubishi Space Star	153,8360334	199,0634099
Nissan Juke	96,06704003	127,6561862
Nissan Leaf	217,5877816	226,6114476
Nissan Micra	103,2179217	120,2690036
Nissan Qashqai	98,90263585	102,2086298
Nissan X-Trail	38,13817962	42,13131861
Opel Adam	37,70580782	37,8639686
Opel Astra	77,69032942	93,19771499
Opel Corsa	296,7270115	411,773543

(Continued)

Car model	MAE	RMSE
Opel Insignia	85,7742048	91,70087755
Opel Karl	1037,366577	1037,366577
Opel Mokka	47,47164045	47,84917535
Peugeot 108	155,4722159	194,1248799
Peugeot 2008	87,93117196	105,2488235
Peugeot 208	152,8841553	175,6210706
Peugeot 308	71,28131321	95,67176046
Peugeot 5008	67,36982073	75,24532275
Peugeot 508	39,90973009	54,16424627
Porsche Cayenne	38,02423695	57,64946067
Porsche Macan	24,00990132	25,96510308
Porsche Panamera	5,941364152	7,285163542
Renault Captur	101,4264657	123,777663
Renault Clio	223,1019897	268,5626146
Renault Espace	11,10181277	11,1235543
Renault Kadjar	52,65994372	60,09760084
Renault Mégane	135,1316817	145,1786167
Renault Scénic	46,13817962	48,41935641
Renault Talisman	6,897852216	8,602472159
Renault Twingo	54,37437657	63,21601769
Renault Zoe	279,0239999	285,8545198
Seat Ibiza	115,1118338	145,6005474
Seat Leon	77,25710188	91,48558739
Seat Mii	31,10953304	36,81172785
Skoda Citigo	87,24642726	93,65176747
Skoda Fabia	46,65025766	59,37964108
Skoda Octavia	142,6173662	155,8926976
Skoda Rapid	59,38590949	59,52035392
Skoda Superb	51,43120575	57,12068746
Smart Forfour	9,447419303	10,36943383
Subaru Forester	22,17838832	23,81140377
Subaru XV	6,506807055	8,465368365
Suzuki Celerio	74,51723371	82,91298207
Suzuki Jimny	7,765205656	8,35237375
Suzuki S-Cross	36,24038778	42,7801498
Suzuki Swift	74,22402518	84,86121489
Suzuki Vitara	54,51618617	59,13818617
Tesla Model S	50,72352764	51,45432492
Toyota Auris	48,05579921	50,24428301
Toyota Aygo	207,3634469	227,8660964
Toyota Prius	6,198393277	7,360368013
Toyota RAV4	110,3444301	116,38191
Toyota Yaris	149,3300999	157,2395612
Volkswagen Caddy	15,40779631	16,7615992
Volkswagen Golf	285,7545253	331,426011
Volkswagen Golf Sportsvan	27,04748726	27,15170869

(Continued)

Car model	MAE	RMSE
Volkswagen Passat	44,9344417	58,02378233
Volkswagen Polo	432,113351	504,627605
Volkswagen Tiguan	144,9880415	186,4225542
Volkswagen Touran	37,06947654	37,5656041
Volkswagen Transporter	28,98089681	29,3223127
Volkswagen Up	227,2401341	250,3953866
Volvo S60	36,23841095	45,22248567
Volvo V40	140,8342002	147,0576504
Volvo V60	220,7438224	247,8844427
Volvo XC60	69,33938599	90,55775709
Volvo XC90	24,96082306	30,12847411

Appendix 16. Performance of *LSTM* using dataset 2

Car model	MAE	RMSE
Alfa Romeo Mito	2,688583	2,688583
Audi A1	64,44721	95,39719
Audi A3	84,13389	142,3894
Audi A4	60,96253	79,69995
Audi A5	29,65709	47,34523
Audi A6	18,7694	22,73658
Audi Q3	34,05237	51,11616
Audi Q5	14,03098	15,76699
BMW 1-serie	197,0688	271,8514
BMW 2-serie	72,5132	112,6891
BMW 2-serie Tourer	40,62655	53,62244
BMW 3-serie	198,7831	250,7908
BMW 4-serie	63,3826	69,41096
BMW 5-serie	68,05129	81,13371
BMW 7-serie	13,50888	17,13884
BMW i3	216,3146	224,4241
BMW X1	63,62419	72,11706
BMW X3	47,54658	49,18165
BMW X4	7,210617	12,51222
BMW X5	41,69908	46,12957
BMW Z4	11,82601	16,25857
Citroën C1	102,1729	146,4705
Citroën C3	87,56841	92,59616
Citroën C4 Cactus	28,56158	30,12568
Dacia Duster	21,17318	23,32426
Dacia Lodgy	15,51616	18,19781
Dacia Logan	13,23084	16,66036
Dacia Sandero	16,43485	17,19168
Fiat 500	147,0093	154,8158
Fiat 500X	25,98377	28,99668
Fiat Panda	12,82912	24,30557
Ford C-MAX	29,64483	30,14054
Ford EcoSport	40,24256	46,13266
Ford Fiesta	186,6624	204,8056
Ford Focus	399,0624	442,0851
Ford Kuga	93,78203	115,9505
Ford Mondeo	43,03595	50,66352
Honda Civic	5,6724	7,645909
Honda CR-V	22,82723	24,06925
Honda HR-V	8,519448	9,430871
Honda Jazz	17,08253	20,28467
Hyundai i10	101,3954	145,3559
Hyundai i20	31,81335	35,29173
Hyundai i30	27,82206	31,05168
Hyundai Tucson	21,24901	21,90355

(Continued)

Car model	MAE	RMSE
Jaguar XE	27,05618	34,51318
Jeep Renegade	13,53488	17,0548
Kia Ceed	47,14359	47,83989
Kia Optima	8,957	9,109629
Kia Picanto	213,738	259,0746
Kia Rio	21,11834	24,49567
Kia Sportage	8,716058	14,08022
Land Rover Discovery Sport	5,264362	6,037593
Land Rover Range Rover	8,290458	9,752539
Land Rover Range Rover Evoque	32,85374	39,7449
Land Rover Range Rover Sport	18,42941	21,83057
Lexus CT	5,136916	7,136537
Mazda 2	20,80032	23,46062
Mazda 3	131,1158	139,1617
Mazda 6	15,1841	16,81205
Mazda CX-3	57,04478	58,85998
Mazda CX-5	71,01367	74,59172
Mazda MX-5	7,58839	8,034631
Mercedes-Benz A-klasse	119,8251	133,5454
Mercedes-Benz B-klasse	108,1716	111,2092
Mercedes-Benz C-klasse	121,7616	137,115
Mercedes-Benz CLA	129,5531	159,5014
Mercedes-Benz E-klasse	122,15	125,6976
Mercedes-Benz GLA	68,60766	73,13294
Mercedes-Benz GLC	54,25956	63,06613
Mercedes-Benz GLE	11,89405	18,2835
Mercedes-Benz S-klasse	8,478099	8,629187
Mercedes-Benz Sprinter	48,39575	54,02215
Mercedes-Benz Vito	27,67499	33,39242
Mercedes-Benz V-klasse	20,03458	21,30748
Mini Clubman	37,37487	47,358
Mini Countryman	52,62479	64,25598
Mini Mini	235,1318	247,2138
Mitsubishi ASX	60,54032	71,50249
Mitsubishi Outlander	46,42804	55,14323
Mitsubishi Space Star	178,3516	221,9411
Nissan Juke	80,85658	105,7359
Nissan Leaf	114,5547	129,3314
Nissan Micra	138,3892	151,8273
Nissan Qashqai	103,9173	120,9023
Nissan X-Trail	25,51687	31,18967
Opel Adam	26,05911	26,21847
Opel Astra	75,68004	93,9193
Opel Corsa	305,2893	398,7424
Opel Insignia	86,41822	92,30355

(Continued)

Car model	MAE	RMSE
Opel Karl	1042,481	1042,481
Opel Mokka	57,79114	58,13851
Peugeot 108	181,047	222,9352
Peugeot 2008	84,85755	96,65757
Peugeot 208	180,5065	212,6137
Peugeot 308	76,65357	89,92483
Peugeot 5008	84,3123	93,70316
Peugeot 508	40,71768	55,4674
Porsche Cayenne	48,83382	70,07252
Porsche Macan	19,83115	21,21637
Porsche Panamera	5,292277	7,362578
Renault Captur	208,7354	236,6985
Renault Clio	261,1806	297,8567
Renault Espace	7,171917	7,205983
Renault Kadjar	59,42975	66,43659
Renault Mégane	168,1387	189,1784
Renault Scénic	56,55878	61,62017
Renault Talisman	7,483475	8,303945
Renault Twingo	44,75843	52,72337
Renault Zoe	223,9118	236,3358
Seat Ibiza	107,6013	147,754
Seat Leon	71,0994	80,62947
Seat Mii	28,55696	35,04404
Skoda Citigo	95,64382	99,80271
Skoda Fabia	72,70835	85,0424
Skoda Octavia	104,2369	129,0473
Skoda Rapid	67,00157	69,68484
Skoda Superb	35,77036	46,77087
Smart Forfour	5,689032	6,168331
Subaru Forester	22,98502	24,56447
Subaru XV	5,607011	9,133027
Suzuki Celerio	75,90133	84,62069
Suzuki Jimny	7,97228	8,334592
Suzuki S-Cross	32,4223	39,60693
Suzuki Swift	88,56063	108,6709
Suzuki Vitara	90,20458	96,20471
Tesla Model S	21,373	24,17678
Toyota Auris	11,04252	11,04806
Toyota Aygo	195,1929	232,2701
Toyota Prius	6,365475	7,346896
Toyota RAV4	111,073	117,2423
Toyota Yaris	141,0294	169,6728
Volkswagen Caddy	13,81324	14,59705
Volkswagen Golf	156,2064	185,6134
Volkswagen Golf Sportsvan	17,59109	17,75162
Volkswagen Passat	40,41972	61,63801

(Continued)

Car model	MAE	RMSE
Volkswagen Polo	345,8267	509,9328
Volkswagen Tiguan	198,1432	214,2131
Volkswagen Touran	32,71026	36,21219
Volkswagen Transporter	16,70208	17,43886
Volkswagen Up	253,7589	264,8143
Volvo S60	45,56316	57,48711
Volvo V40	88,07135	93,95093
Volvo V60	251,9011	264,3223
Volvo XC60	41,57621	49,83735
Volvo XC90	23,60211	32,43746

Appendix 17. Performance of *LSTM* using dataset 3

Car model	MAE	RMSE
Alfa Romeo Mito	28,03453772	28,29747184
Audi A1	59,30181449	79,90967735
Audi A3	133,6413116	150,8481866
Audi A4	201,8428879	215,3842358
Audi A5	41,86387416	43,82503623
Audi A6	30,06414141	33,25902536
Audi Q3	42,03370013	50,06521113
Audi Q5	44,52331189	47,10340954
BMW 1-serie	221,8140368	314,3542924
BMW 2-serie	94,02648272	106,581135
BMW 2-serie Tourer	181,8901585	187,8356071
BMW 3-serie	159,1227679	203,2832705
BMW 4-serie	72,53446851	83,69810528
BMW 5-serie	66,91522544	76,56054019
BMW 7-serie	10,99936513	13,36892938
BMW i3	218,1783077	220,835904
BMW X1	58,93859972	65,91879485
BMW X3	175,4547751	180,9326141
BMW X4	18,1548451	18,77713525
BMW X5	51,48788221	63,97905731
BMW Z4	11,26231303	11,91866831
Citroën C1	131,7294684	153,5442662
Citroën C3	78,98741988	83,51981981
Citroën C4 Cactus	26,32473293	27,83993994
Dacia Duster	22,36405509	25,37703569
Dacia Lodgy	9,156181897	11,18919679
Dacia Logan	17,78648458	23,80782768
Dacia Sandero	55,42976815	57,36159152
Fiat 500	148,2042236	159,0535772
Fiat 500X	49,57370867	51,09776919
Fiat Panda	16,11390959	20,0261201
Ford C-MAX	31,86434071	36,07287302
Ford EcoSport	74,49839347	83,94804447
Ford Fiesta	178,8004499	195,316559
Ford Focus	789,5180817	856,3966052
Ford Kuga	77,55841173	95,61124564
Ford Mondeo	79,02397891	86,22922653
Honda Civic	11,08812441	13,18324017
Honda CR-V	30,61502429	32,71367563
Honda HR-V	11,19886017	14,1517353
Honda Jazz	21,10467584	25,69167236
Hyundai i10	130,7270508	152,3758452
Hyundai i20	63,91493007	72,21231086
Hyundai i30	12,97887502	16,42805887
Hyundai Tucson	26,34238979	26,88088574

(Continued)

Car model	MAE	RMSE
Jaguar XE	18,81635175	20,89846153
Jeep Renegade	9,363180161	10,42749515
Kia Ceed	247,8188771	249,9541937
Kia Optima	21,55479622	21,58503113
Kia Picanto	141,4568743	174,013833
Kia Rio	26,76263537	28,32320258
Kia Sportage	37,94270434	38,95838627
Land Rover Discovery Sport	17,13441672	18,80889573
Land Rover Range Rover	8,467437472	9,628853353
Land Rover Range Rover Evoque	13,74039323	26,24730186
Land Rover Range Rover Sport	10,99915341	14,16933999
Lexus CT	8,210710253	8,971395821
Mazda 2	94,39704023	96,00523572
Mazda 3	103,368784	105,5171476
Mazda 6	26,54301616	28,11148387
Mazda CX-3	153,3298623	153,3551042
Mazda CX-5	190,65305	198,8579046
Mazda MX-5	2,119197709	3,481787733
Mercedes-Benz A-klasse	325,9608962	361,7734183
Mercedes-Benz B-klasse	88,92342159	89,59840218
Mercedes-Benz C-klasse	193,7445352	209,9528636
Mercedes-Benz CLA	282,4592198	300,9469923
Mercedes-Benz E-klasse	169,0221601	171,6034846
Mercedes-Benz GLA	68,30402483	71,5989832
Mercedes-Benz GLC	125,8544906	131,6536866
Mercedes-Benz GLE	21,52999292	28,83007541
Mercedes-Benz S-klasse	13,24842112	13,53379009
Mercedes-Benz Sprinter	16,70989663	26,18476179
Mercedes-Benz Vito	87,53380149	88,23995929
Mercedes-Benz V-klasse	32,75618281	33,41038161
Mini Clubman	46,56174251	56,16670026
Mini Countryman	52,57182748	62,91611352
Mini Mini	176,5698785	183,8603246
Mitsubishi ASX	114,7009054	130,5615402
Mitsubishi Outlander	83,48927307	89,05232732
Mitsubishi Space Star	149,9023743	196,2264646
Nissan Juke	70,45472935	95,96470562
Nissan Leaf	374,0402483	377,6984707
Nissan Micra	229,942918	238,0335542
Nissan Qashqai	103,1677115	124,5075214
Nissan X-Trail	25,36587412	28,31875268
Opel Adam	48,18303898	48,18446013
Opel Astra	134,8716431	151,7110647
Opel Corsa	294,4757342	412,0913083

(Continued)

Car model	**MAE**	**RMSE**
Opel Insignia	34,10938699	38,49193151
Opel Karl	520,8504028	520,8504028
Opel Mokka	53,52324731	54,39070712
Peugeot 108	162,6549334	186,598587
Peugeot 2008	80,74891881	97,5945017
Peugeot 208	126,479187	156,116309
Peugeot 308	113,5682504	130,854966
Peugeot 5008	67,14009094	79,29846267
Peugeot 508	44,82866124	54,83839255
Porsche Cayenne	39,89826311	53,14363462
Porsche Macan	45,57215663	47,31228302
Porsche Panamera	8,58794839	9,254455924
Renault Captur	312,3425816	335,1367837
Renault Clio	614,3728894	689,2687729
Renault Espace	14,82035092	14,88264212
Renault Kadjar	91,04626029	96,97734234
Renault Mégane	157,5161878	164,1378417
Renault Scénic	45,51806182	47,56692771
Renault Talisman	9,423491887	11,85912872
Renault Twingo	129,3013706	139,4728815
Renault Zoe	401,3578273	406,6416507
Seat Ibiza	144,8355495	169,3296295
Seat Leon	95,57633318	108,7721484
Seat Mii	46,60770965	63,78899198
Skoda Citigo	175,7082923	195,6233354
Skoda Fabia	83,09729222	96,70076371
Skoda Octavia	212,2143555	243,3544076
Skoda Rapid	83,37601689	83,89370804
Skoda Superb	41,28479549	60,27132111
Smart Forfour	13,8525922	14,48632093
Subaru Forester	9,88269043	10,81155433
Subaru XV	8,427301543	9,218856216
Suzuki Celerio	91,73179626	99,3679889
Suzuki Jimny	18,20040934	19,49795901
Suzuki S-Cross	21,63759477	26,30738373
Suzuki Swift	112,9540471	132,4769599
Suzuki Vitara	75,55168797	81,13879815
Tesla Model S	89,26063429	89,70559339
Toyota Auris	72,27409799	82,57626574
Toyota Aygo	208,1712472	232,4095386
Toyota Prius	15,20323767	15,91503581
Toyota RAV4	43,94570269	62,93389676
Toyota Yaris	161,9802464	170,2252983
Volkswagen Caddy	17,34218543	18,5372676
Volkswagen Golf	528,8858468	573,0611441
Volkswagen Golf Sportsvan	38,28487778	38,46050905

(Continued)

Car model	**MAE**	**RMSE**
Volkswagen Passat	65,1451961	71,65901109
Volkswagen Polo	557,3013742	592,8137664
Volkswagen Tiguan	160,4805908	196,6107186
Volkswagen Touran	44,99356297	52,00636283
Volkswagen Transporter	42,41757802	42,46421192
Volkswagen Up	285,1771153	301,7969445
Volvo S60	46,97207642	54,45141659
Volvo V40	175,3874294	184,906637
Volvo V60	410,1526685	436,0215503
Volvo XC60	50,84819903	70,99824166
Volvo XC90	29,20094844	34,796187