Master Thesis Data Science & Society – Spring 2022

Improving Airbnb listing price prediction with sentiment analysis, review recency and topic modelling.

Student details

Name:	Noor Cornelia Antonia Meijer
Student number:	U297733
Student email:	n.c.a.meijer@tilburguniversity.edu

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee

Supervisor:	Dr. Drew Hendrickson
Second reader:	Dr. Boris Čule

Tilburg University
School of Humanities & Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
July 8, 2022
Word Count: 8773

Abstract

Airbnb is one of the most popular and fastest growing sharing platforms in the world. However, its offer of peer-to-peer accommodation can make it difficult to properly price listings. This thesis provides new insights in price prediction by adding review information to a price prediction model with standard listing characteristics. Reviews are often overlooked in Airbnb prediction research but offer valuable insights. Using an open-source Airbnb dataset, several review features are mined from a large set of Airbnb listing reviews. The unsupervised learning method *topic modelling* is applied on reviews and included as predictor, which results in improved predictive performance for listing prices. In addition, (weighted) sentiment analysis features are obtained using VADER and transformed to features. However, they only marginally improve price prediction, due to the skewed distribution of sentiment scores. Both a Support Vector Regression and XGBoost model are a good fit for the Airbnb data, although XGBoost provides the best performance.

Data Source/Code/Ethics Statement

Work on this thesis (did/did not) involve collecting data from human participants or animals. The original owner of the data and code used in this thesis retains ownership of the data and code during and after the completion of this thesis. The code used in this thesis is publicly available [http://insideairbnb.com/get-the-data].

1 Introduction

While currently being a staple in the tourism industry, accommodation platform Airbnb did not even exist 15 years ago. Its unprecedented and explosive growth has been followed by an equally impressive growth of other peer-to-peer sharing platforms. The development of AI and big data, in combination with a customer demand for more sustainable consumption and less dependence on large multinationals has led to the popularity of the sharing economy (Cheng & Jin, 2019, Wirtz et al., 2019). These days, consumers can find shared alternatives for almost anything; accommodation (Airbnb or Homeaway), cars (Blablacar), transportation (Uber), and even pets (Borrowmydoggy) (Wirtz et al., 2019). The exponential growth of sharing platforms has been accompanied by a similar development in research on the sharing economy, but due to the vast amount and quick innovation, there are still many gaps (Hossain, 2020).

Research related to sharing platforms cannot simply be compared to its traditional equivalents, because the sharing economy has its own unique challenges such as bad working conditions, lack of regulation and higher risks for both provider and sharer (Malhotra & Van Alstyne, 2014). Therefore research that is specifically focused on sharing platforms is necessary to understand differences and how it changes today’s consumer patterns. Especially in the case of Airbnb, as trust is extremely important when booking accommodation that is likely in a place the consumer is not familiar with yet (Huurne et al., 2017).

One aspect of sharing platforms that has come to be of great importance are reviews. The internet and social media has made it especially easy to write and read reviews about anything. With an experiential good that is purchased online, such as hotels or Airbnb, a review becomes even more important and can provide a consumer with the confidence that the product or service they purchase will actually be delivered to them, in the state they expect (Lawani et al., 2019).

Therefore, it is essential that further research is done into sharing platform reviews, specifically Airbnb. Reviews are the basis for trust between the provider and the sharer, and can significantly affect both in different ways. This study will try to uncover the effect of reviews on the price of an Airbnb listing price, by adding review information to standard listing characteristics as predictors.

1.1 Relevance

Airbnb has over 200 million users worldwide and therefore many users could be impacted by research related to Airbnb (Cheng & Jin, 2019). With the importance of trust in the sharing economy, reviews are an especially relevant aspect of Airbnb. A bad review can lead users to opt out of renting to or from another user. However, reviews might not always be fair and can depend even on a user’s own personal background. Therefore Airbnb users could benefit from knowing the effect a review has on their own or other users’ profiles. Users might be missing out on rentals that perfectly fit their needs, but they disregard because of a single bad review. New renters could be missing out on valuable income due to the effects of bad reviews. Additionally, Airbnb itself might be able to adjust their policies and how they display reviews.

Scientific research has shown to still have a significant amount of gaps in literature when it comes to sharing platforms. While Airbnb is prominent in sharing economy research, there is still much to be explored. This thesis will provide insights into the effects of review information, besides standard features such as amount of reviews or aggregated ratings. Much information can be gathered from reviews, such as sentiment and topics, that are not often considered in price prediction. The findings could possibly be generalized to other sharing platforms as well, such as similar websites like Couchsurfing or Homeaway.

1.2 Research questions

Based on previous research and the existent gaps in the literature, this leads to the following research questions.

Main research question

Main Question: To what extent can Airbnb listing price prediction be improved by including review information?

Sub research questions

To answer this question, several sub research questions are formulated. The first set of sub questions (1a-c) focuses on the additional effect on price prediction of sentiment analysis, topic modelling and review recency. First the best standard features will be selected, based on a large set of features relating to host and property information, and used as input variables for a Support Vector Regression. The new review-based features are individually added to this model subsequently.

RQ1: To what extent can Airbnb listing price prediction with standard listing characteristics be improved by including review features into a Support Vector Regression?

> *RQ1a: To what extent can Airbnb listing price prediction with basic listing characteristics be improved by including sentiment analysis features into a Support Vector Regression?*
>
> *RQ1b: To what extent can Airbnb listing price prediction with basic listing characteristics be improved by including topic modelling features into a Support Vector Regression?*
>
> *RQ1c: To what extent can Airbnb listing price prediction with basic listing characteristics be improved by including review recency into a Support Vector Regression?*

Based on previous research, the SVR had the best result for a similar research question. However, other Machine Learning models should be compared to improve performance. Therefore the research will be repeated with different models, based on successful models in other research.

RQ2: Can Extreme Gradient Boosting or Ridge Regression with new review features improve price prediction performance of SVR?

2 Literature review

In this section the previous related work will be outlined. First, background and the current state of research on Airbnb rental price prediction will be given. In the subsequent section, the use of sentiment analysis for price prediction will be explored. Lastly, the two additional review features (review recency and topic modelling) will be discussed.

2.1 Airbnb Listing Price Prediction

While a vast amount of research has been performed in relation to Airbnb, nearly all studies that relate to the prediction of prices take a non-data science approach. Most research is based on theories from management or economics areas, such as hedonic price theory, and focuses on determining factors attributing to prices (Hossain, 2020, Dann, Teubner & Weinhardt, 2018,). However, recently researchers have started exploring Airbnb price prediction with Machine Learning and Deep Learning methods.

Chattopadhyay & Mitra (2019) compared three algorithms, OLS, random forest and a decision tree and their ability to predict listing prices with 137 amenities and 6 variables. They identified several city-specific price determinants across 11 cities in the US. Additionally, they were able to obtain composite scores of the variable importance by averaging the three models. While obtaining high R-SQUARED-scores with a random forest, the generalizability of this study is questionable since no train-test split is performed. There is also a lack of variables related to listing reviews, since the included variables are limited to review scores and number of reviews (per month and total).

Several researchers have incorporated sentiment analysis in their research as predictor variables. Most notably, Kalehbasti, Nikolenko & Rezaei (2021) created a model for listing price prediction using a combination of property characteristics, host information and review features. Sentiment analysis was performed on the reviews with the Python TextBlob library and the consequent scores were included as a new variable. A support vector regression outperformed ridge regression, gradient boosting, K-means + ridge regression, and neural network, and was able to illustrate a promising predictive performance for listing prices.

Trang, Huy & Le (2020) take a different approach to Airbnb price prediction, by first clustering the training data with k-means clustering, before building several classification models. Additionally, they perform a corset-based sampling of the data prior to the k-means analysis. Sentiment scores are determined by training a Long Short Term Memory (LSTM) network with labels of values 0 or 1 for negative and positive. They find that the combination of k-means with gradient boost and XGBoost both have a strong prediction performance. While performing well, the type of sentiment analysis used is in practice not very feasible since sentiment scores need to be labelled ahead of training the model.

Peng, Li & Qin (2020) seek to improve other price prediction models by using a multi-modality dataset that combines listing attributes and reviews with geographical position of the Airbnb. Reviews are processed into numeric variables using the Python NLTK, Sklearn and TextBlob libraries. After reducing the amount of variables with a Principal Component Analysis (PCA), an SVR, XGBoost and Neural Network are trained on the data. The XGBoost model with multi-modality proved to be superior over other models and modalities but was not able to outperform the other studies.

2.2 Text Analysis of Reviews for Price Prediction

A recent advance in sharing economy research is the inclusion of review features in price or booking predictions. Besides generic features such as number of reviews and star rating, a lot of information can be extracted from review text. Sentiment analysis is a commonly used method for extracting sentiment from any text (Kwon, Lee & Back, 2020). Sentiment scores can generally be obtained in two ways: a machine learning model can be trained to label sentiment or a pre-trained algorithm such as VADER or TextBlob can be used (Santos, Mota, Benevenuto & Silva, 2020).

Lawani et al (2019) use the AFINN lexicon to generate sentiment scores between -5 and +5 for each review. They use these sentiment scores as input for a hedonic spatial autoregressive model to explore the effects on rental prices, in addition to several other basic listing characteristics. They find that sentiment scores based on review text are better predictors than a unidimensional rating score, although they are not an improvement over disaggregated quality scores (e.g. cleanliness, location).

Shen, Liu, Chen & Ji (2020) use deep learning techniques to develop a text-based price recommendation system (TAPE) for new Airbnb listings. They trained a feedforward neural network with sentence embeddings from Airbnb descriptions to predict price. This model was used as an input, in combination with generic listing specifications and location data, to predict prices for new listings. They find that their model performs similarly to comparable models, but with less features.

In addition to sentiment, other features can be extracted from the text to better categorize a review. Structural topic modelling is an ML-based NLP method that can be used to find latent topics based on occurrences of words in texts (Kwon, Lee & Back, 2020). Sánchez-Franco & Alonso-Dos-Santos (2021) use structural topic modelling to identify hidden topics in Airbnb reviews, and their relationship with prices. This study has a special focus on the moderating role of gender and uses solely extreme gradient boosting for predictive analytics. Didarul Islam et al. (2022) use topic modelling to generate synthetic variables from Airbnb property descriptions and improve price prediction, in addition to special features. They found that both additions improve prediction accuracy.

2.3 Review Recency

While several studies have been performed to predict Airbnb listing prices with the use of review features, none of the ML-based research has included recency of the review. In other research areas, review recency has been proven to affect customer opinion on a review. Tandon, Aakash, Aggarwal & Kapur (2021) found that older reviews are viewed as less helpful than more recent reviews. Aakash & Jaiswal (2020) confirmed that recency has a positive effect on reviewer trustworthiness.

Wang, Zhu & Chen (2008) developed a new approach (RTBR) to representing Amazon reviews by incorporating review credibility and the time-decay of the review. Their model proved the importance of the recency of reviews, as it outperformed the Amazon review system on trustworthiness and was able to provide timely review results with a simple scheme.

3 Methodology

In this section the methods used for this thesis will be outlined. First, the models used for prediction are explored, i.e. a support vector regression, ridge regression and XGBoost. After, natural language processing methods (sentiment analysis and topic modelling) are discussed. The following conceptualization illustrates the pipeline of this thesis.

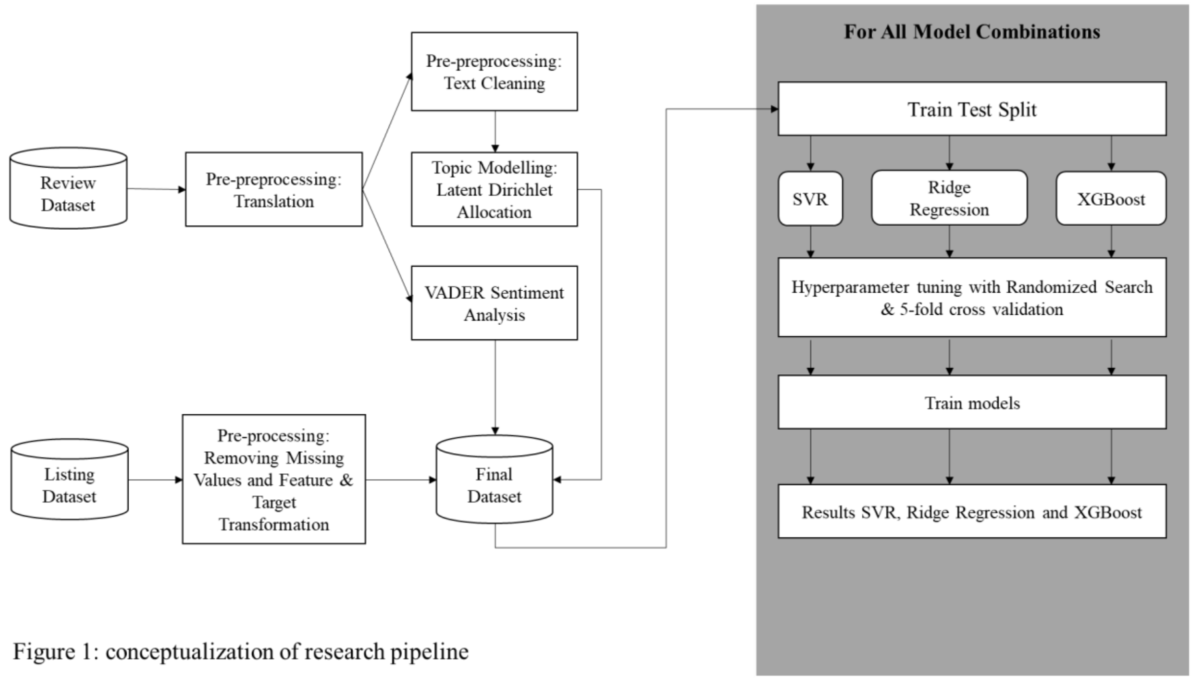
Review Dataset
Pre-preprocessing: Translation
Pre-preprocessing: Text Cleaning
Topic Modelling: Latent Dirichlet Allocation
VADER Sentiment Analysis
Listing Dataset
Pre-processing: Removing Missing Values and Feature & Target Transformation
Final Dataset
For All Model Combinations
Train Test Split
SVR
Ridge Regression
XGBoost
Hyperparameter tuning with Randomized Search & 5-fold cross validation
Train models
Results SVR, Ridge Regression and XGBoost

Figure 1: conceptualization of research pipeline

3.1 Support Vector Regression

As the basis of this thesis, a support vector regression (SVR) was used to test the first research questions (and first set of sub questions). The SVR algorithm is based on the principle of a Support Vector Machine (SVM), but when the prediction is a regression problem (as opposed to a classification problem). SVR has good accuracy, low computation times and relatively easy implementation and is therefore a perfect candidate for a first model. Its ability to model non-linear relationships on multi-dimensional spaces make it an appropriate algorithm for this research. Additionally, in a similar research of Kalehbasti, Nikolenko & Rezaei (2021) the performance of an SVR was superior compared to several other algorithms, such as gradient boosting, a neural network, and ridge regression. In similar price prediction models, an SVR was often chosen among several models.

3.2 Ridge Regression

Ridge regression is an extension of linear regression that addresses stability issues by adding a penalty for a model with large coefficients. The loss function is modified with an L2 penalty, which is based on the sum of squared coefficient values. Ridge regression is capable of handling variables with high multicollinearity, whereas this leads to issues in linear regression. With a large amount of variables in the dataset that are not unlikely to correlate due to similarity in nature, ridge regression could solve underlying correlation issues. In similar research, ridge regression performed well and scored close to or even outperformed more complicated algorithms such as an SVR or Gradient Boost (Kalehbasti, Nikolenko & Rezaei, 2021).

3.3 XGBoost

Gradient boosting is a powerful machine learning technique that creates an ensemble of models (usually decision trees) to minimize prediction error. XGBoost (eXtreme Gradient Boosting) is an implementation of the Gradient Boosting algorithm that has quickly gained popularity due to its good performance. It was designed to be fast to execute and have increased model performance. This leads to XGBoost being a common choice among price prediction research, with good results. Peng, Li & Qin (2020), Trang, Huy & Le (2020), Didarul Islam et al., (2022) all found that XGboost outperformed several other ML models in predicting Airbnb listing prices.

3.5 Sentiment Analysis

For this thesis, a sentiment analysis method called Valence Aware Dictionary and Sentiment Reasoner (VADER) is used. VADER is a lexicon- and rule-based model for general sentiment analysis developed by Hutto & Gilbert (2014), that outperforms many other sentiment analysis tools. It scores a text on the general sentiment with a score in the range of -1 to 1 (negative to positive). A unique feature of VADER is that it takes several heuristics into account, such as punctuation and capitalization. Besides accounting for punctuation, VADER is able to understand non-conventional text such as emoticons, capitalized words and conjunctions. Standard text pre-processing are not necessary, as VADER is able to properly analyze texts without these transformations (PyPi, n.d.). These characteristics make this sentiment analysis method perfect for reviews, as these are written by individuals and likely to contain emoticons, slang, and multiple punctuation marks. Additionally, VADER is often found to have better precision, recall and accuracy than other popular lexcion-based methods (e.g. TextBlob, SentiWordNet and NLTK) in a wide range of applications such as tweets, reviews and forum posts (Bonta, Kumaresh & Janardhan, 2019, He & Zheng, 2019, Mujahid, Lee, Rustam, Washington, Ullah, Reshi & Ashraf, 2021).

3.6 Topic Modelling

Topic modelling is a text mining technique that discovers latent topics in texts. A commonly used method for topic modelling is Latent Dirichlet Allocation (LDA). LDA is an unsupervised machine learning method that finds topics based on word frequencies and outputs the probabilities of texts belonging to a topic (Zhang et al, 2022). This method is the most commonly used type of topic modelling due to its efficiency and ability to handle big data (Guo, Barnes & Jia, 2017). It is therefore especially suited to the current review dataset, which has a size of 272056 rows.

4 Experimental setup

This chapter will cover the experimental setup that was followed in this thesis. In the following sections, the dataset (4.1), exploratory data analysis and pre-processing methods (4.2), evaluation metrics for the models and the baseline (4.3) and the experimental procedure, including hyperparameter tuning, are discussed.

4.1 Dataset

To answer the research questions, an open-source dataset with publicly available information from the Airbnb website will be used. The data is independently scraped and made available on http://insideAirbnb.com/index.html. This website provides data on Airbnb listings from several large cities across the world, sorted by city. The data has already been verified, analyzed, cleansed and aggregated where appropriate by the data provider. The available data is a snapshot of the current listings on Airbnb, therefore the data used in this thesis represents all Airbnb listings on March 16, 2021. All variables therefore represent the value on this date, e.g. the price of the listing at that date. For this thesis, the main focus will be on the city Amsterdam and therefore data from only this city will be used.

Several files belong to the dataset, related to the listings, reviews, calendar and location. In this thesis the data about listings and reviews will be used. From now on, these will be referred to as the listing dataset and review dataset. Both datasets are linked by the key 'listing ID'. The listing dataset has a large amount of features about all listings since August 2020, with rental characteristics such as amount of bedrooms, neighborhood and information about the host. This also includes information about price, availability of the listing and (aggregated) review scores. The variables are in a variety of formats: continuous, bools and categorical. This dataset contains a total of 5597 rows (listings) and 74 columns (features). Several irrelevant, uninformative or unusable features (e.g. URLs and duplicates) from this dataset are dropped, the remaining set of features can be found in Appendix A. The review dataset contains all reviews of the listings up until the downloaded date. This dataset has 272056 reviews and 6 features. The main feature is the review text, which is taken exactly as it was written, including punctuation marks, symbols and emoticons. In addition, it contains the listing ID, review ID reviewer ID, reviewer name, and date of the review.

4.2 Exploratory Data Analysis & Pre-processing

The raw data from the Airbnb datasets is not appropriate for the required analyses and therefore several tasks need to be performed to prepare the data properly. Additionally, to gain more insights into the dataset, exploratory data analysis is performed. This section is divided into two parts, first the listing dataset is discussed, second the review dataset.

4.2.1 Listing Dataset

In this section the listing dataset will be explored and pre-processing is explained. Missing values and their treatment are discussed, followed by an exploration of the data distribution. Finally all transformation of the features and the target variable is outlined.

Missing values

Appendix A shows the percentage of missing values for each of the features in the reduced listing dataset. The highest amount of missing values for a single variable is 36 %, which amounts to 2015 missing data points. However, this variable (host_about) will be transformed to a dummy variable where 1 = not missing and 0 = missing, so missing values are avoided. Of the other variables, several review-related variables have 10 or 9 % missing values due to listings not having reviews yet. These listings are dropped, since analysis needs to be performed on reviews in this research and this is not possible when no reviews are available. This leads to 527 reviews or 9.42% of the total listing being dropped. The variable 'beds' contains 2% or 101 missing values, these are dropped as it is unclear why these values are missing (if there are no beds or these are missing at random).

Furthermore, missing values are due to wrong imputation and are corrected. The ‘bedrooms’ variable contains 314 NAN values, however after further investigation we can conclude that these NAN values are for studio apartments. This happens due to the fact that these type of apartments are a single room and have no bedroom. This can be gathered from the fact that 0 does not occur as a value in this column, the minimum value is 1 (see also Appendix A). Additionally, an analysis of the listing with a NAN value for *bedrooms,* confirms that 173 of these 314 listings have the word ‘studio’ appear in their listing name or descriptions. Therefore NAN values for this feature are imputed as 0.

Data distribution

Appendix A contains the distribution of several numeric variables in the listing dataset. This table indicates that some variables have skewed distributions and outliers. *Host listings count, accommodates, bedrooms, beds, minimum nights, maximum nights, reviews per month*, and *number of reviews,* show large differences in range, while the mean is low. These features contain multiple outliers, however they are most likely true values since the data has been properly cleaned and checked by the data collectors (Inside Airbnb). Due to a wide variation in popularity, some Airbnb listing could have hundreds of reviews while others have none. Also, some large houses exist that have many bedrooms or beds. As such, these are natural outliers and not due to any errors. Therefore none of the outliers will be removed. However, the large ranges, especially across features, can lead to issues with modelling. To prevent any problems, variables are normalized by scaling.

Feature transformation

The features in the dataset are in different formats and many are not appropriate for analysis (Appendix A). Therefore several transformations are performed. First, the variable ‘host_since’ is transformed from a date to numeric variables. The variable is first transformed to date format from a string format, and subsequently the amount of days from the processing date since the ‘host_since’ date are calculated. Several variables are then transformed to dummy variables. Furthermore, two textual variables, amenities and property type, are one-hot-encoded to new dummy variables. The property type column contains a string with a host-specified property type. Since there is a wide range of slightly different categories, this leads to a large amount of 58 unique values of property_type. Therefore each listing is put in one of 5 categories based on the text the host wrote, e.g. text containing ‘shared’ or ‘hostel’ is allocated to the ‘shared’ category. After all listings are categorized, they are one-hot-encoded into 5 new dummy variables for each category.

The amenities column contains lists with all amenities that a listing has. Since hosts can decide themselves which amenities the listing has, there are 1016 unique types of amenities in the dataset. 606 of those only appear once, and it is not feasible to include all of them. Therefore only amenities that occur more than 2000 times in the dataset are included, which sums up to 28 unique amenity types that are one-hot-encoded into new dummy variables. All included amenities can be found in Appendix A.

For the amount of bathrooms, the dataset contains only a columns with the amount of bathrooms in string format with 22 unique values. To avoid adding a large amount of one-hot-encoded variables to the dataset, this variable is transformed into two new variables by analysing the text. Bath_shared is a dummy variable that signifies if a listing has a shared bathroom (=1) or not (=0). Bath_number is a numeric variable that contains the number of bathrooms in a range of 0 to 5.

Furthermore, the host_verifications and host_about features are transformed. Host_verifications contains lists with all verifications per host. These are transformed to a numeric variable with the amount of verifications per host. As described in the missing values section, the host_about variable is transformed to a dummy variable, where 1 = host has an ‘about’ text and 0 = host has no an ‘about’ text.

The full list of variables in the new listing dataset can be found in Appendix A. The dataset after transformation exists of 4780 rows and 84 columns.

Target transformation

The target variable in this thesis is the price of an AirBnb listing per night, however the values in the price column of the dataset are not adequate for prediction. Prices were given in a textual format with € signs and were therefore converted to numerical values. The distribution of all prices can be found in table 1. The prices have a large range of values, with a minimum of 0 and a maximum of 6477. Since it is unlikely that an Airbnb listing is free, all rows with prices of 0 are removed. To deal with the large range and extreme values, logarithms of the prices were taken. Taking the logs can deal with these issues and improve price prediction. The table below illustrates the change from price to log_price. The range of log_price is much smaller and median value is now very close to the mean, whereas the difference was much larger (135 to 164) for price. The table also shows that 7 rows were dropped with a value of 0.

	Price	**Log_price**
count	5597	5590
mean	164	4.923
median	135	4.905
std	162	0.565
min	0	2.890
25%	95	4.553
50%	135	4.905
75%	198	5.292
max	6477	8.776

Table 1 – Price vs. Log_price distribution

4.2.2 Review Dataset

In this section the review dataset will be explored and pre-processing is explained. The distribution of different languages and their translation is discussed. Missing values and their treatment are discussed, followed by an exploration of the data distribution. Finally pre-processing of reviews is outlined.

Language Distribution & Translation

Due to the Airbnb's core service of tourism, the reviews are from tourists all over the world and therefore written in many different languages. Language detection of the reviews shows that only 209308 or 76.9% of the reviews are in English, while the other reviews are in 45 different languages. Appendix B shows the full distribution of languages. As we feel this is a too large subset to drop, comments are translated using Deep_Translator (Deep-Translator, n.d.). Deep_Translator offers access to Google Translate, which is able to automatically detect the language of a text and translate it to English. Translating all reviews to the same language will also allow for easier exploration of the data and make it easier to compare. GoogleTranslator is not able to handle emoticons and reviews with exclusively punctuation (e.g. reviews that only contain '.'), so those reviews cannot be translated. Reviews that are in another language but cannot be translated are therefore dropped during the translation process, since they also are not useful for topic modelling. After translation, 271,276 out of 272,055 reviews remain due to this removal.

Exploratory Data Analysis

With reviews translated, the dataset can be properly explored. The review dataset contains no NAN values, i.e. cells with N/A or that are empty. Therefore no other reviews are dropped from the review dataset.

	Length review	Number of words
mean	27	47
std	25	45
min	100	100
25%	104	170
50%	206	360
75%	357	630
max	595	999

Table 2 – Review Distribution

Table 2 shows the distribution of the reviews. Reviews have an average length of 269 characters and 47 words, showing that the average review is quite short. The range is quite large, with some reviews only having a single word while the longest has 999 words (Airbnb review limit is a 1000 words).

	positivity_score	negativity_score	compound_score	neutral_score
mean	0.345459	0.009838	0.839362	0.644703
std	0.170332	0.030277	0.237142	0.165129
min	0	0	-0.997000	0
25%	0.232000	0	0.822100	0.562000
50%	0.319000	0	0.926700	0.672000
75%	0.433000	0	0.967000	0.755000
max	1	1	0.999600	1

Table 3 – Sentiment Score Distribution

VADER sentiment values are divided into categories positive, negative and neutral scores. Additionally a compound score of all scores is calculated. Table 3 illustrates how sentiment values are distributed in the dataset. Figure 2 & 3 show the distribution of all sentiment scores. As the table illustrates, reviews are overwhelmingly positive. Further investigation shows that in fact 216,155 or 79.7% of the reviews have a negativity score of 0. This is reflected in the distribution graph of the negativity scores, which is heavily skewed to the right. However, positivity scores do not seem extremely unbalanced, although there are skewed slightly to the left. This combination also leads to a heavily skewed compound_score, as can be seen in figure 3. Neutral scores are relatively balanced, although also skewed to the left. There is also a jump at a neutral score of 1 and a positivity score of 0, which could be explained by short reviews of one word that do not have any meaning and are therefore just classified as neutral.

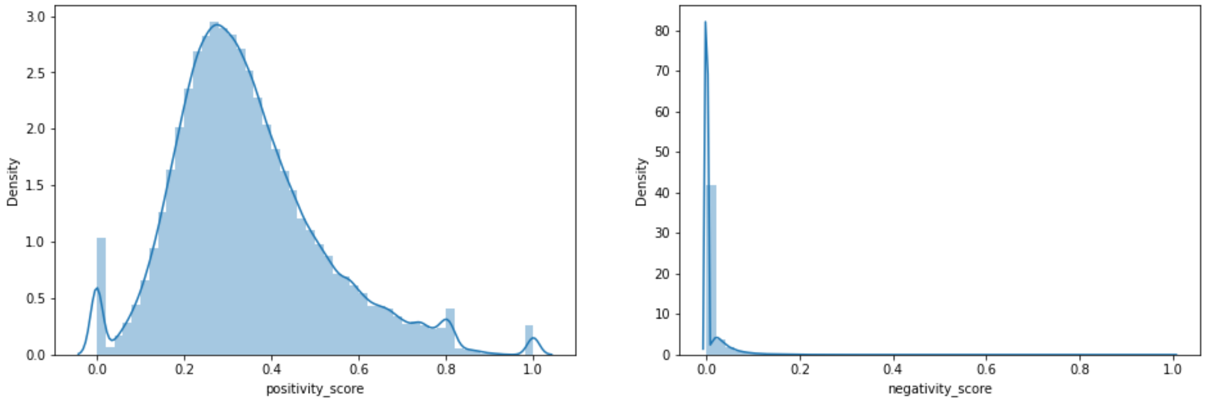

Figure 2 - Distribution of positivity_score and negativity_score (VADER)

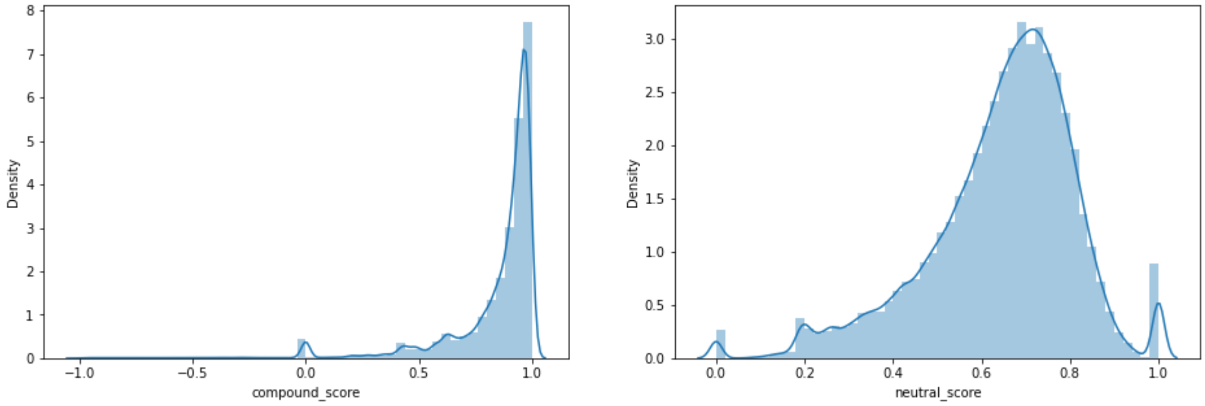

Figure 3 - Distribution of compound_score and neutral_score (VADER)

Pre-processing of Reviews

To prepare the review text for analysis, several cleaning steps are taken. While VADER is designed to handle text that is not pre-processed, for effective topic modelling it is necessary to prepare the text. First, all text is put in lower case and all stop words are removed from the text. This done using the spaCy library, which contains 326 English stop-words (spaCy, n.d.). All punctuation is removed using the string library and words are lemmatized using NLTK WordNetLemmatizer. After visual inspection of the reviews, it is noted that two unusual letters/symbols appear. First the string '\r<br>', which signifies a line break in the text. Furthermore the letter 'u' appears a lot, which after further inspection seems to be a results of lemmatizing 'us'. Both these characters are also removed from the text. Finally, the remaining text is split into words. Every review is transformed to a list of words. This will allow for processing the text effectively for LDA.

4.3 Evaluation Metrics and Baseline

To be able to compare the models against different studies, three evaluation metrics were used: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R-squared. The main evaluation metric will be R-squared, as the purpose of this thesis to find the additional effect of several review variables and compare models. However, errors also are discussed, as it is important that predicted prices do not have high errors. RMSE is more sensitive to extreme error values that have a large distance from the mean, while MAE weighs all errors equally. RMSE therefore provides more insight into errors that are extremely off the predicted value, which MAE does not. Additionally, R-squared indicates the proportion of variation in the data that is explained by the model, i.e. how well the model fits the data. This is an important addition to the error values, since those do not say anything about the fit of the model. Vice versa, R-squared could be high while there is large error in the model, which is unknown without MAE/RMSE.

A Support Vector Regression (SVR) will serve as the baseline model for this thesis. In this model, only standard listing variables from the listing dataset will be included. The full list of included variables can be found in Appendix A.

4.4 Experimental Procedure

This section explains the procedure followed to answer all research questions. First, the two datasets are merged. The review dataset has scores for each review, but several reviews can be linked to a single listing. Therefore, review scores are aggregated and for each listing an average sentiment score is calculated. Average negative, positive, neutral and compound sentiment scores are added to the review dataset for each listing. The new full dataset is then divided into a train-test split in advance of creating the models. 80% is allocated as training data, the remaining 20% is testing data. A baseline is obtained by including all standard listing variables from the original Airbnb listing dataset in a Support Vector Regression model. To test the first research question, the same model is run with aggregated sentiment analysis scores added as 4 new features. In turn, topic modelling and review recency variables are added. How these variables are obtained is described in more detail in section 4.5.2 and 4.5.3 below. The results are compared to the baseline based on evaluation metrics as described in section 4.3.

To answer research question 2, different models are tested. All analyses that were performed with SVR are now reproduced with both Ridge Regression and XGBoost. Performance is compared to the SVR results based on the evaluation metrics as described in section 4.3. The section below describes hyperparameter tuning for all models.

4.5.1 Hyperparameter Tuning

For all models, hyperparameters were tuned with Randomized Search. Although Randomized Search does not test every single hyperparameter combination, it is much more efficient than Grid Search. It therefore does not secure the best hyperparameter setting for every model, but due to a large amount of models and combinations to be tested this was the most efficient tuning method in the available time. For the hyperparameter search, 5-fold cross validation was used to validate hyperparameter values. With K-fold cross validation, the data is divided into k folds and each iteration a different holdout set is used as validation set. With this method, the training dataset can be re-used to test the best performing hyperparameter values while no data needs to be solely designated as a validation set. For all hyperparameter tuning, the main evaluation metric R-squared is used as a performance measure.

The tables below indicate which hyperparameter values were tested for each model, as well as the values for topic modelling.

SVR

Parameter	Values
C	[15,10,5,1]
gamma	[0.001,0.01,0.1,0.0001,'scale', 'auto']
epsilon	[0.2,0.1,0.05,0.01]
kernel	['rbf','poly',linear']

Table 4 – Hyperparameter values SVR

Ridge Regression

Parameter	Values
Alpha	Range(0, 1, 0.01)

Table 5 – Hyperparameter values Ridge Regression

XGBoost

N_estimators	[10,50,100,300,400,500]
min_split_loss	[0,0.2,0.5]
max_depth	[2,3,5,10,15]
booster	['gbtree','gblinear','dart']
learning_rate	[0.05,0.1,0.3,0.5]
min_child_weight	[1,2,3]
subsample	[0.5,0.7,1]
base_score	[0.25,0.5,1]

Table 6 – Hyperparameter values XGBoost

Topic modelling

Before transforming the topic scores to features, the number of topics K needs to be determined, as well as several hyperparameters. Because LDA is an unsupervised learning technique, it is not possible to use regular evaluation metrics (such as accuracy or R-squared) to test the performance of the model. For LDA, a measure that is often used to tune hyperparameters is coherence score. To obtain this score, the coherence between all documents and topics is calculated. One type of coherence score is Umass Coherence, which was proposed by Mimno, Wallach, Talley, Leenders & McCallum (2011). The formula for Umass Coherence is given below

$$C_{UMASS}\,(w_i, w_j, \varepsilon) = \log \frac{P\left(w_i\,, w_j\right) + 1}{P(w_j)}.$$

Where $P\left(w_i\,, w_j\right)$ is the number of documents containing words w_i and w_j, $P\left(w_j\right)$ is the number of documents containing word w_i. The Umass metric outputs a score between -14 and 14. The closer the absolute value is to 0, the better the coherence of the model.

Using the Umass score as an evaluation metric, hyperparameters K, alpha and eta are tuned. The following values are tested:

Parameter	**Values**
Alpha	[0.1,0.2,0.3,0.4,'symmetric']
Eta	[0.01,0.1,0.2,'symmetric']

Table 7 – Hyperparameter values LDA

4.5.2 Topic Modelling

After pre-processing, reviews are in a list format with only relevant words. All irrelevant words that do not give any meaning to a topic such as stop words, pronouns and often used verbs (have, are, make etc.) are removed as described in section 4.2.2. For the LDA, a dictionary is created where every unique term that appears in the reviews gets an index value. This dictionary is then used to create a Document Term Matrix from the reviews. The matrix and dictionary are then used to train the LDA model. The amount of topics needs to be pre-specified, so to find the optimal value the amount of topics k is tuned as described in the previous section. The LDA model provides us with a list of topics, their coefficients and the words that belong to the topic. These topics are used to create k new variables (one for each topic), where each review gets a score for each topic. The score signifies the correlation with the topic, i.e. how many words in the review correspond to words that belong to a certain topic. The scores are aggregated per listing and used as a new features in the listing dataset.

4.5.3 Review Recency

To obtain review recency, first the dates when the review was written are transformed to a numerical variable with the amount of days since that date. With the amount of days, a recency weight is calculated for each date. The formulas below signify how this weight is calculated. Each review is then multiplied by its recency weight to obtain a sentiment score with review recency included. Weighted scores are then grouped by listing to obtain an aggregate score. The weighted sentiment scores are used instead of normal sentiment scores for modelling.

4.6 Software and Algorithms

For all processing, packages Pandas, sklearn and numpy are used. For calculating summary statistics, sklearn and statistics packages are used. For plots and graphs, Seaborn, wordcloud, Yellowbrick and Matplotlib are used.

Several packages are used for feature transformation: datetime, Abstract Syntax Trees(literal_eval), json. For language detection and translation, Lang Detect and Deep-Translator are applied. For further processing of text data, sklearn, re, spaCy, NLTK, and VADER are used. For topic modelling with LDA, gensim package is used. Finally, for model building and hyperparameter tuning, sklearn and XGBoost are used.

5 Results

In this chapter results from the previously described methodology and experimental setup will be outlined. First, results from adding new review features (sentiment analysis, topic modelling & review recency) to a SVR will be provided. In section 5.3, the models described in sub-question 2 will be compared on their performance.

5.1 Baseline & Results Table

As described in section 4.3, an SVR with only listing features will serve as the baseline model. The results are includes in all results tables. The results from all model combinations can be found in table 8, a value in **bold** indicates the best score.

		SVR	*Ridge regression*	*XGBoost*
R-squared	Listing only	0.643	0.630	0.678
	+sentiment	0.644	0.630	0.680
	+best Topic Modelling	**0.656**	**0.650**	**0.681**
	+RR	0.646	0.632	0.675
MAE	Listing only	0.252	0.259	0.240
	+sentiment	0.251	0.258	0.242
	+best Topic Modelling	**0.249**	**0.252**	**0.238**
	+RR	0.251	0.259	0.241
RMSE	Listing only	0.328	0.336	0.313
	+sentiment	0.329	0.335	0.312
	+best Topic Modelling	**0.324**	**0.326**	**0.311**
	+RR	0.328	0.335	0.314

Table 8 – Results all model combinations

5.2 Results SVR with Review Information

In this section, results of the SVR model are discussed, for every addition of new review features. First, results from sentiment analysis are outlined, afterwards topic modelling and lastly review recency.

5.2.1 results sentiment analysis

To answer the first sub-question, sentiment analysis scores are added as features to a feature set of standard listing characteristics. As table 8 illustrates, the SVR with sentiment features outperforms the baseline model without sentiment features, but only marginally. Sentiment scores slightly improve R-squared and decrease error (both MAE and RMSE), all by 0.001.

The figure below illustrates the relation between predicted and actual values of the target variable. The best fit line seems to follow the distribution quite well and most data points are clustered around the line. However, as y increases, predicted values start to vary more and absolute errors become larger. The figure ‘actual values and absolute error’ illustrates the absolute errors for each value in the test data. This further shows that with more extreme values of y, especially high ones, the errors start increasing.

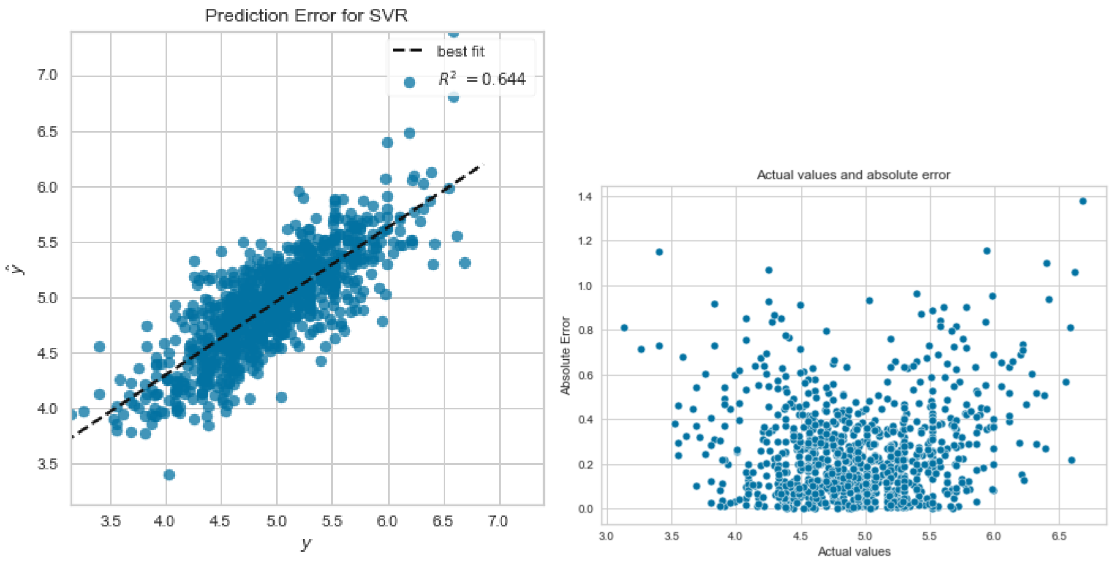

Figure 4 – SVR - Sentiment Analysis: Prediction Error & Actual Values vs. Absolute Error.

5.2.2 Results Topic Modelling

To uncover the effects of topic modelling on price prediction performance, topic features were added to the baseline feature set (listing features) and used for an SVR. The following tables illustrate the performance of 2-10 topics as opposed to a baseline with no topics included.

		SVR	Ridge regression	XGBoost
R-squared	No topics	*0.643*	*0.630*	*0.678*
	2 topics	0.644	0.630	0.680
	3 topics	0.641	0.631	0.673
	4 topics	0.639	0.629	0.670
	5 topics	0.647	0.639	0.681
	6 topics	0.656	0.650	0.681
	7 topics	0.641	0.632	0.674
	8 topics	0.641	0.630	0.675
	9 topics	0.642	0.632	0.672
	10 topics	0.646	0.630	0.670

Table 9 – Topic modelling results: R-squared

Before transforming the topic scores to features, the number of topics k needed to be determined, as well as several hyperparameters. This was done based on coherence scores, as described in section 4.5.1. However, tuning k showed that coherence scores increased with every extra topic. Since the goal of sub-question 1b is to find the effect of adding topic features, it is not feasible to pick the least amount of topics (this would imply only 1 topic). Based on previous research, the optimal amount of (general) topics is usually found to be 10 or lower (Hu, Zhang, Gao, & Bose, 2019, Didarul Islam et al. (2022). Therefore LDA was performed multiple times with a different number of topics, with k ranging from 2-10. For each amount k, the corresponding aggregated topic scores for each listing were used as input variables for the ML models. The results of each number of topics can be found in table 9,10 and 11, in comparison to a baseline (only listing variables, no topic variables included). The best performing scores are in **bold**.

MAE	No topics	*0.252*	*0.259*	*0.240*
	2 topics	0.251	0.259	0.241
	3 topics	0.253	0.259	0.244
	4 topics	0.253	0.259	0.243

	5 topics	0.252	0.256	0.241
	6 topics	**0.249**	**0.252**	**0.238**
	7 topics	0.251	0.256	0.245
	8 topics	0.253	0.259	0.242
	9 topics	0.252	0.258	0.243
	10 topics	0.251	0.259	0.246

Table 10 – Topic modelling results: MAE

While including 2 topics instead of none slightly increases R-squared (although only by 0.001), the scores decrease afterwards for 3 and 4 topics. Error scores follow a similar pattern in an inverse design. However, starting at k=5, R-squared starts growing again. Both the models with 5 and 6 topics perform better than the baseline model. For k=5, R-squared increases from 0.643 to 0.647 while MAE remains unchanged and RMSE slightly decreases. The model therefore is increasingly better at predicting variation in price, but does not necessarily improve on errors. For k=6, R-squared increases to 0.656 and the errors do improve, MAE decreases by 0.003 and RMSE decreases by 0.005. This model is therefore slightly better at decreasing errors that are very far off (RMSE) as opposed to average error (MAE). After k=6, R-squared starts to decrease again, and errors start to increase. However, the model with k=10 also outperforms the baseline, though R-squared is still lower than for k=6. Therefore the model with 6 topics is chosen as the best topic modelling model.

RMSE	No topics	*0.329*	*0.336*	*0.313*
	2 topics	0.329	0.336	0.312
	3 topics	0.331	0.335	0.316
	4 topics	0.332	0.336	0.317
	5 topics	0.328	0.331	0.312
	6 topics	**0.324**	**0.326**	**0.311**
	7 topics	0.331	0.335	0.315
	8 topics	0.330	0.335	0.315
	9 topics	0.330	0.335	0.316
	10 topics	0.328	0.335	0.316

Table 11 – Topic modelling results: RMSE

To further gather insights in topic modelling, the best model (i.e. k=6) is explored further. Table 12 illustrates the topics, their contribution, most important keywords and a sentence that is best representative of the topic. Figure 5 visually illustrates the words that are most closely identified with a certain topic.

Topic Number	**Topic % Contribution**	**Keywords**	**Representative Text**
0	0.9258	center, city, minute, walk, tram, station, amsterdam, close, away, restaurant	[relatively, easy, find, central, station, far, beautiful, area, canal, surrounded, coffee, shop...
	0.9242	amsterdam, place, come, home, experience, best, host, house, stay, time	[loved, staying, houseboat, week, carien, wonderful, host, amsterdam, amazing, city, better, way...
2	0.9306	apartment, located, clean, accommodation, recommend, nice, stay, good, pleasant, amsterdam	[neighborhood, good, apartment, ideally, placed, able, foot, apartment, clean, needed, present, ...
3	0.9425	stay, great, place, amsterdam, recommend, location, definitely, host, lovely, perfect	[amazing, time, maria, staying, boat, time, amsterdam, little, special, space, lovely, clean, pe...

4	0.9248	great, nice, location, place, clean, good, room, host, stay, super	[tamaras, place, perfect, location, easy, public, transportation, walking, distance, awesome, ba...
5	0.9351	room, hotel, staff, bed, bathroom, small, night, bit, shower, kitchen	[breakfast, sweetthe, stair, really, really, really, narrow, it, difficult, people, climb, large...

Table 12 - Topic Modelling: k=6 final topics

All topics are relatively similar in contribution, with high percentages ranging between 92% and 95%. The topics do not all have a clear general subject that can be identified. Topics 2, 3 and 4 are relatively similar and have positive wording (perfect, great, recommend) + the words 'Amsterdam'. However, some topics clearly focus on a certain theme. Topic 0 seems to relate to location (center, close, tram, walk), topic 1 appears to be related to service/social aspect (experience, host, home, best) and topic 5 indicates some theme associated with hotels/more commercial Airbnb's and amenities (hotel, staff, bathroom, kitchen).

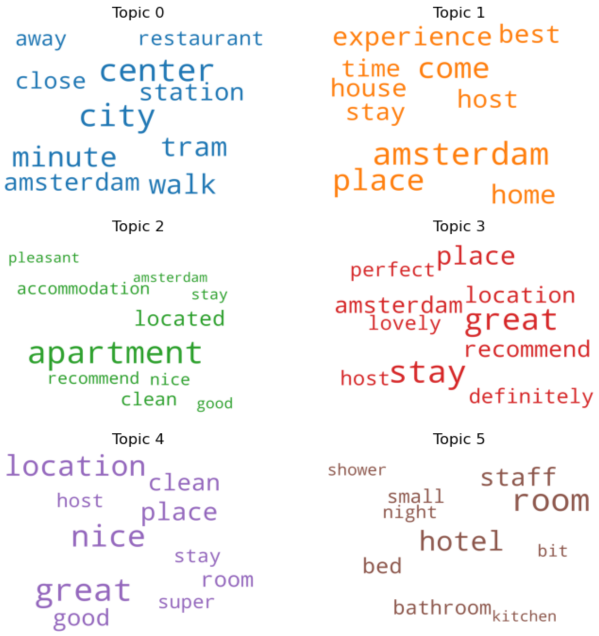

Figure 5 – Wordcloud of 6 topics

5.2.3 Results Review Recency

The model with review recency outperforms the baseline slightly, with an improvement of 0.003 in R-squared. Conversely, MAE is slightly decreased while RMSE stays the same. R- squared is also 0.002 higher than the model with only sentiment features (i.e. no recency weight on the features). Error metrics are the same or slightly lower (0.001). The figure below illustrates the relation between predicted and actual values of the target variable. Both figures show a similar pattern as the model with sentiment features, as errors seem to be increasing with higher values for y. The error values are almost identical, indicating that this model performs similarly in regards of price prediction error, which is validated by nearly similar MAE and RMSE scores.

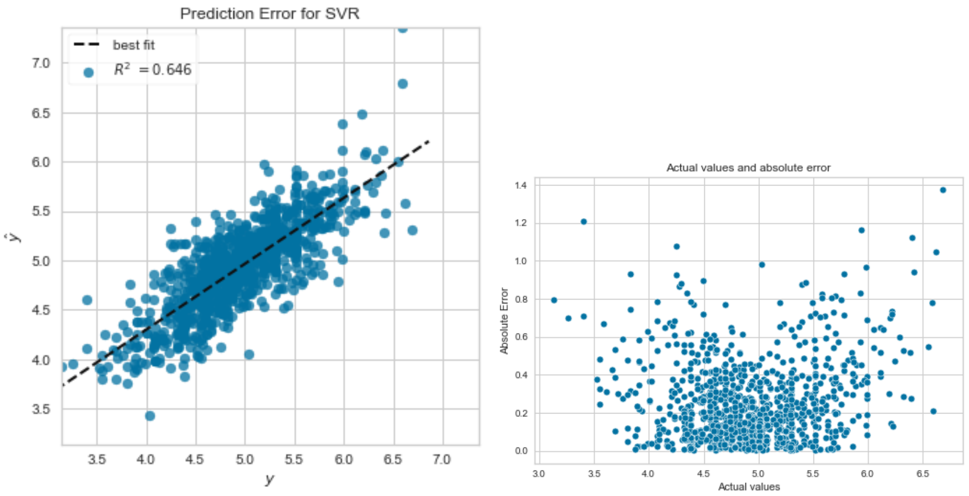

Figure 6 – SVR – Review Recency: Prediction Error & Actual Values vs. Absolute Error.

5.3 RQ2 model comparison

In the second sub-question, the performance of other ML models is compared to the SVR used in sub-question 1. In this section, the results of the Ridge Regression and XGBoost model are discussed.

Ridge Regression

The results of the ridge regression, for all models discussed in section 5.2, can be found in table 8. The results show that the ridge regression model has lower performance than the SVR for every feature set. The increase (of R-squared) and decrease (of MAE/RMSE) for different new feature additions is almost identical to that of the SVR. The highest R-squared for the ridge regression is 0.65, with the best combination of features being standard listing features + 6 LDA topics. Table 9, 10 and 11 show the results for topic modelling. These illustrate a similar pattern, where the change in evaluation metrics generally follows that of the SVR.

XGBoost

The results of the XGBoost, for all models discussed in section 5.2, can be found in table.. It is clear that XGBoost outperforms SVR with all combinations of features. For the model with sentiment and topic modelling features, the XGBoost model has a better R-squared score (0.680 and 0.681 respectively) and lower MAE and lower RMSE. However, the XGboost model shows some differing patterns for the other new feature additions, compared the SVR model. For the review recency feature, the XGBoost model does not outperform baseline R-squared score, while the SVR does. In addition, as table 9,10 and 11 show, for topic modelling XGboost has a different pattern. Although 6 topics is still the optimal solution, the increase in R-squared from 0.678 to 0.680 is much smaller than the increase of both lower than the decrease for Ridge Regression. This fact illustrates that the XGBoost model already has very good performance without any added features (i.e. listing only) and is hard to improve. The basic XGBoost model consequently outperforms any SVR and Ridge regression with new features, on R-squared (0.678), MAE (0.240) and RMSE (0.313).

The figure below illustrates the relation between predicted and actual values of the target variable. Again, the best fit line fits the data points well. Compared to the SVR error plots, the prediction error plot appears to have less error for extreme values of y. This is also illustrated by the fact that RMSE is significantly lower. The figure 'actual values and absolute error' illustrates the absolute errors for each value in the test data. Although similar to the plots of the SVR model, absolute error values appear to be slightly more clustered around the center and have less extreme values.

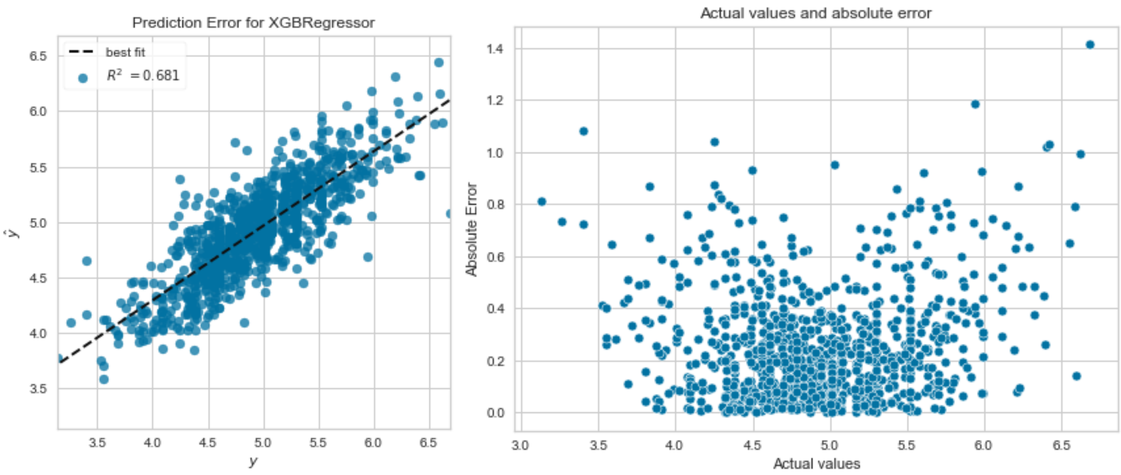

Figure 7 – XGBoost – Review Recency: Prediction Error & Actual Values vs. Absolute Error.

6 Discussion

The goal of this thesis was to discover the effects of including review features to standard listing characteristics as predictors for Airbnb listing price prediction. All additional features were found to improve price prediction, though for some feature combinations the improvement was marginal. The inclusion of topic modelling features provided the best performance. Although the SVR performed well, XGBoost proved superior in terms of all evaluation metrics. The ridge regression was not able to improve on the SVR for any of the feature combinations. In this chapter, results will be further discussed in relation to the research questions stated in chapter 1.

New review features

In this thesis, new features based on sentiment analysis, topic modelling and review recency were added to an SVR model with standard listing features. The results show that including sentiment analysis scores does improve price prediction performance, although the improvement is quite small. RMSE even increased, indicating that the inclusion of this feature produces more large errors. An issue with sentiment analysis in this particular case could be the distribution of sentiment across reviews. As the figures in section 4.2.2 show, reviews are overwhelmingly positive, and skewed. It is unknown what causes this, but it is not likely that almost no Airbnb users have a negative experience. Reasons for this could include social norms, under-reporting of bad experiences, or due to listings with negative reviews disappearing from Airbnb website altogether (Teubner and Glaser, 2018). This issue could be the reason that the sentiment analysis features do not have a significant effect on model performance.

Furthermore, additional features of weighted sentiment scores were added to basic listing features. The sentiment scores remained the same, but they were multiplied by a weight based the recency of the review. As the results show, this addition again improved performance of the model, but still marginally (a 0.003 increase in R-squared compared to baseline). Errors did not improve (RMSE) or only slightly (MAE). Considering the issue with positive reviews in Airbnb reviews, it is difficult to uncover the true effect of these features since they are based on the sentiment scores.

Finally, Latent Dirichlet Allocation (LDA) was performed on the reviews to obtain latent topics to include as new features. The optimal number of topics was found to be 6 topics, which significantly improved all evaluation metrics. While not all topics have a general theme, some clear distinctions can be made and illustrate the usefulness of the model.

Model comparison

To improve on the model, other Machine Learning methods were tested on the same feature combinations. In all feature combinations, the ridge regression model could not improve on performance of the SVR model. Only with topic modelling features was it able to outperform the baseline of an SVR with only standard listing variables. With these features, the ridge regression performance came very close to the SVR.

The XGBoost model was more successful in comparison to both other models, for every new feature addition. The XGBoost model with 6 topics included was the best overall model, according to all three performance metrics. The performance of this model is not surprising considering the effectiveness in similar studies. This model has an R-squared of 0.681, an MAE of 0.238 and RMSE of 0.311.

Contribution & Further Research

This thesis seeks to improve price prediction of Airbnb listings, through the inclusion of several review-based features. Although many studies have attempted to improve price prediction, there still remains a gap in research in studies based on Data Science methods. In particular, reviews are not often considered beyond basic characteristics such as ratings and reviews per month. This thesis makes a contribution by considering the effect of adding features based on sentiment analysis, review recency and topic modelling. Especially the inclusion of Topic Modelling provides promising improvements in predictive performance. In addition, XGBoost proves to be very adept at predicting Airbnb listing prices and is able to handle a large dataset with a variety of variable types. Further research could improve upon the XGBoost model by including other engineered features, either based on external data or reviews. Topic Modelling could also be further explored in several ways. The LDA model could be trained by adding more review data (e.g. from other cities) or testing for more topics. Additionally, other types of topic modelling techniques such as Latent Semantic Analysis or Non Negative Matrix Factorization could be used for feature engineering.

Limitations

Due to the large size of the review dataset, the large number of listing features and the necessary manual tuning of k topics, this thesis was limited in computational performance. Therefore, with the same dataset and features it might be possible to achieve better results in terms of predictive performance. For example, hyperparameter tuning was done through Randomized Search, which does not check all combinations and hence does not necessarily output the best possible parameters. Additionally, the sentiment distribution of Airbnb reviews is skewed, especially in the case of negative and compound scores. As discussed, the reason for this is unknown and could be a result of several circumstances. However, it is necessary to either discover what happens with negative reviews or consider this component when further analyzing Airbnb reviews.

7 Conclusion

In this thesis, the goal was to discover to what extent Airbnb listing price prediction could be improved by including review information. Based on the previous chapters, the research questions will be answered.

RQ1: To what extent can Airbnb listing price prediction with standard listing characteristics be improved by including review features into a Support Vector Regression?

> *RQ1a: To what extent can Airbnb listing price prediction with basic listing characteristics be improved by including sentiment analysis features into a Support Vector Regression?*
> To some extent, but only marginally. While sentiment scores improve price prediction with SVR, the increase in R-squared is very small. This is likely connected to the fact that Airbnb reviews are extremely positive and the data contains almost no negative reviews.
>
> *RQ1b: To what extent can Airbnb listing price prediction with basic listing characteristics be improved by including topic modelling features into a Support Vector Regression?*
> Topic modelling features provide a more successful increase in performance of the SVR. The best performing model contains 6 topics and outperforms the baseline. Although topic subjects are not all clearly divided in themes, they do improve performance.
>
> *RQ1c: To what extent can Airbnb listing price prediction with basic listing characteristics be improved by including review recency into a Support Vector Regression?*
> *Review recency only marginally improves price prediction performance, although more than sentiment scores individually. Similar to RQ1a, this is likely associated with the skewed sentiment scores.*

Overall, review features are able to improve price prediction of Airbnb listings with a Support Vector Regression model. However, the extent varies. Further research could improve by combining features or engineering different features from reviews.

RQ2: Can Extreme Gradient Boosting or Ridge Regression with new review features improve price prediction performance of SVR?

Extreme Gradient Boosting or XGBoost outperforms the SVR model on every feature combination. This model is more suited to the Airbnb data and provides the best performing model: and XGBoost with standard listing features + 6 topic features. Ridge Regression is not able to improve price prediction compared to the SVR model. However, for the optimal feature combination (i.e. standard listing features + 6 topics), the Ridge Regression scores are very close to those of the SVR.

Acknowledgements

Special thank to InsideAirbnb for providing and cleaning the data.

References

Aakash, A., & Jaiswal, A. (2020). Segmentation and Ranking of Online Reviewer Community. *International Journal of E-Adoption*, 12(1), 63–83.

Bonta, V., Kumaresh, N., & Janardhan, N. (2019). A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis. *Asian Journal of Computer Science and Technology*, 8(S2), 1–6

Md. Didarul Islam, Bin Li, Kazi Saiful Islam, Rakibul Ahasan, Md. Rimu Mia, Md. Emdadul Haque, (2022). Airbnb rental price modeling based on Latent Dirichlet Allocation and MESF-XGBoost composite model, *Machine Learning with Applications*, 7.

Chattopadhyay, M., & Mitra, S.K. (2019). Do airbnb host listing attributes influence room pricing homogenously? *International Journal of Hospitality Management*, 81, 54-65.

Cheng, M., & Jin, X., (2019). What do Airbnb users care about? An analysis of online review comments, *International Journal of Hospitality Management*, 76A, 58-70.

Dann, D., Teubner, T. & Weinhardt, C. (2019). Poster child and guinea pig – insights from a structured literature review on Airbnb. *International Journal of Contemporary Hospitality Management*, 31(1), 427-473.

Deep-Translator – Google Translate. Deep-Translator (n.d.). Retrieved from https://deep-translator.readthedocs.io/en/latest/usage.html#google-translate

Guo, Y., Barnes, S., & Jia, Q. (2017). Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation. *Tourism Management*, 59, 467-483.

Guttentag, D. (2019). Progress on Airbnb: a literature review. *Journal of Hospitality and Tourism Technology*, 10(4), 814-844.

He, L. & Zheng, K. (2019). How do General-Purpose Sentiment Analyzers Perform when Applied to Health-Related Online Social Media Data? *Stud Health Technol Inform.*, 264, 1208–1212.

Hong Trang, L., Duong Huy, T., & Ngoc Le, A. (2020) Clustering helps to improve price prediction in online booking systems. *International Journal of Web Information Systems*, 17(1), 45-53.

Hossain, M. (2020). Sharing economy: a comprehensive literature review. *International Journal of Hospitality Management*, 87.

Hu, N., Zhang, T., Gao, B., & Bose, I. (2019). What do hotel customers complain about? Text analysis using structural topic model. *Tourism Management*, 72, 417-426.

Huurne, M., Ronteltap, A., Corten, R., & Buskens, V. (2017). Antecedents of trust in the sharing economy: A systematic review. *Journal of Consumer Behaviour*, 16(6), 485–498.

Lawani, A. Reed, M.R., Mark, T., & Zheng, Y. (2019). Reviews and price on online platforms: Evidence from sentiment analysis of Airbnb reviews in Boston. *Regional Science and Urban Economics*, 75, 22-34.

Kalehbasti, P.R., Nikolenko, L., & Rezaei, H. (2021). Airbnb Price Prediction Using Machine Learning and Sentiment Analysis. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds) Machine Learning and Knowledge Extraction. CD-MAKE 2021. Lecture Notes in Computer Science(), vol 12844. Springer, Cham.

Kwon, W., Lee, M., & Back, K. J. (2020). Exploring the underlying factors of customer value in restaurants: A machine learning approach. *International Journal of Hospitality Management*, 91.

Malhotra, A. & Van Alstyne, M. (2014). The dark side of the sharing economy … and how to lighten it. *Communications of the ACM*, 57(11), 24–27.

Mujahid, M., Lee, E., Rustam, F., Washington, P.B., Ullah, S., Reshi, A.A., & Ashraf, I. (2021). Sentiment Analysis and Topic Modeling on Tweets about Online Education during COVID-19, *Appl.Sci.*, 11(8438), 1-25.

Mimno, D., Wallach, H., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing semantic coherence in topic models. *Proceedings of the 2011 Conference on Emperical Methods in Natural Language Processing*, 262–272.

nltk.stem.wordnet. NLTK (n.d.). Retrieved from https://www.nltk.org/_modules/nltk/stem/wordnet.html

Peng, N., Li, K., & Qin, Y. (2020). Leveraging Multi-Modality Data to Airbnb Price Prediction. Proceedings - 2020 2nd International Conference on Economic Management and Model Engineering, ICEMME 2020, 1066–1071.

Sánchez-Franco, M.J. & Alonso-Dos-Santos, M. (2021). Exploring gender-based influences on key features of Airbnb accommodations. *Economic Research-Ekonomska Istraživanja*, 34(1), 2484-2505.

Santos, G.V., Mota, F. S., Benevenuto, F.T., & Silva, H. (2020). Neutrality may matter: sentiment analysis in reviews of Airbnb, Booking, and Couchsurfing in Brazil and USA. 10, 45.

Spacy en_core_web_sm. spaCy (n.d.). Retrieved from https://spacy.io/models/en

String — Common string operations. Python (n.d.). Retrieved from https://docs.python.org/3/library/string.html

Shen, L., Liu, Q., Chen, G., & Ji, S. (2020). Text-Based Price Recommendation System for Online Rental Houses. Big Data Mining and Analytics, 3(2), 143.

Tandon, A., Aakash, A., Aggarwal, A.G, & Kapur, P.K. (2021). Analyzing the impact of review recency on helpfulness through conometric modeling. *Int J Syst Assur Eng Manag*, 12(1), 104–111.

Teubner, T. & Glaser, F. (2018) Up or out—The dynamics of star rating scores on Airbnb. *Research Papers*, 96.

VaderSentiment 3.3.2. PyPi (n.d.) Retrieved from https://pypi.org/project/vaderSentiment/

Wang, D. & Nicolau, J.L. (2017). Price determinants of sharing economy based accommodation rental: a study of listings from 33 cities on Airbnb.com. *International Journal of Hospitality Management*, 62, 120-131.

Wang, B. C., Zhu, W. Y., & Chen, L. J. (2008). Improving the Amazon review system by exploiting the credibility and Time-decay of public reviews. Proceedings - 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT Workshops 2008, 123–126.

Wirtz, J., So, K.K.F., Mody, M.A., Liu, S.Q. & Chun, H.H. (2019). Platforms in the peer-to-peer sharing economy. *Journal of Service Management*, 30(4), 452-483.

Zhang, S., Ly, L., Mach, N., & Amaya, C. (2022). Topic Modeling and Sentiment Analysis of Yelp Restaurant Reviews. *International Journal of Information Systems in the Service Sector*, 14(1), 1-16.

Appendices

Appendix A – Listing Dataset

Table 13 Included features in listing dataset

Variable name	**Format**
listing_id	int64
host_since	int64
host_about	int64
host_is_superhost	bool
host_listings_count	int64
host_total_listings_count	int64
host_verifications	int64
host_identity_verified	bool
accommodates	int64
Bathrooms_text	object
bedrooms	float64
beds	float64
price	float64
Property_type	object
amenities	object
minimum_nights	int64
maximum_nights	int64
availability_365	int64
number_of_reviews	int64
number_of_reviews_l30d	int64
review_scores_rating	float64
review_scores_accuracy	float64
review_scores_cleanliness	float64
review_scores_checkin	float64
review_scores_communication	float64
review_scores_location	float64
review_scores_value	float64
instant_bookable	bool
calculated_host_listings_count	int64
reviews_per_month	float64

Table 14 - Final Included features in listing dataset

Variable name	**Format**
listing_id	int64
host_since	int64
host_about	int64
host_is_superhost	bool
host_listings_count	int64

host_total_listings_count	int64
host_verifications	int64
host_identity_verified	bool
accommodates	int64
bedrooms	float64
beds	float64
price	float64
minimum_nights	int64
maximum_nights	int64
availability_365	int64
number_of_reviews	int64
number_of_reviews_l30d	int64
review_scores_rating	float64
review_scores_accuracy	float64
review_scores_cleanliness	float64
review_scores_checkin	float64
review_scores_communication	float64
review_scores_location	float64
review_scores_value	float64
instant_bookable	bool
calculated_host_listings_count	int64
reviews_per_month	float64
log_price	float64
bath_shared	int64
bath_number	float64
Bijlmer-Centrum	uint8
Bijlmer-Oost	uint8
Bos en Lommer	uint8
Buitenveldert-Zuidas	uint8
Centrum-Oost	uint8
Centrum-West	uint8
Aker-Nieuw Sloten	uint8
Baarsjes-Oud-West	uint8
De Pijp-Rivierenbuurt	uint8
Gaasperdam-Driemond	uint8
Geuzenveld-Slotermeer	uint8
IJburg-Zeeburgereiland	uint8
Noord-Oost	uint8
Noord-West	uint8
Oostelijk Havengebied - Indische Buurt	uint8
Osdorp	uint8
Oud-Noord	uint8
Oud-Oost	uint8
Slotervaart	uint8
Watergraafsmeer	uint8

Westerpark	uint8
Zuid	uint8
type_boat	uint8
type_entire_property	uint8
type_other	uint8
type_private_room	uint8
amenities_Wifi	int64
amenities_Essentials	int64
amenities_Smoke_alarm	int64
amenities_Heating	int64
amenities_Hangers	int64
amenities_Hair_dryer	int64
amenities_Hot_water	int64
amenities_Kitchen	int64
amenities_Long_term_stays_allowed	int64
amenities_Iron	int64
amenities_Shampoo	int64
amenities_Dishes_and_silverware	int64
amenities_Dedicated_workspace	int64
amenities_Coffee_maker	int64
amenities_Refrigerator	int64
amenities_Washer	int64
amenities_Bed_linens	int64
amenities_Cooking_basics	int64
amenities_Carbon_monoxide_alarm	int64
amenities_Fire_extinguisher	int64
amenities_First_aid_kit	int64
amenities_Private_entrance	int64
amenities_Oven	int64
amenities_Dishwasher	int64
amenities_Microwave	int64
amenities_Stove	int64
amenities_TV	int64
amenities_Dryer	int64

Table 15 – missing values listing dataset

Variable	% missing
host_since	0%
host_location	0%
host_about	36%
host_is_superhost	0%
host_total_listings_count	0%
host_verifications	0%
neighbourhood_cleansed	0%
property_type	0%
room_type	0%
accommodates	0%
bathrooms_text	0%
bedrooms	6%
beds	2%
amenities	0%
price	0%
minimum_nights	0%
maximum_nights	0%
availability_365	0%
calendar_last_scraped	0%
number_of_reviews	0%
first_review	9%
last_review	9%
review_scores_rating	9%
review_scores_accuracy	10%
review_scores_cleanliness	10%
review_scores_checkin	10%
review_scores_communication	10%
review_scores_location	10%
review_scores_value	10%
instant_bookable	0%
calculated_host_listings_count	0%
reviews_per_month	9%

Table 16 – variable distribution: selected variables with extreme values

	Host listing count	**accommodates**	**bedrooms**	**beds**	**Minimum nights**	**Maximum nights**	**Reviews per month**	**Number of reviews**
count	5.590	5.590	5.283	5.496	5.590	5.590	5.066	5.590
mean	3	2,94	1,57	1,95	3	535,81	1,14	49
std	14	1,44	0,92	1,64	15	524,74	2,18	90
min	0	1	1,0	1,0	1	1	0,01	0
25%	1	2	1,0	1,0	2	28	0,26	4
50%	1	2	1,0	1,0	2	365	0,54	17
75%	2	4	2,0	2,0	3	1.125	1,25	47
max	701	16	15,0	34,0	1.001	1825	83,69	913

Appendix B – review dataset

Table 17 - Language counts review dataset

Language	Count
en	209308
fr	21568
de	14451
es	7048
nl	6956
it	3479
pt	1460
ru	989
zh-cn	764
ro	661
Other	560
ko	553
af	539
ca	380
tl	365
da	352
no	282
so	272
cs	241
sv	193
pl	187
zh-tw	161
id	159
ja	157
tr	145
fi	136
cy	115
he	87
hu	85
hr	72
sw	68
vi	39
sl	38
el	36
et	32
sk	27
ar	25
lt	16
uk	11
bg	10
th	9
sq	9
lv	7

mk	3
fa	1