University of Twente

Financial Engineering and Management

Master's Thesis

Applying Text Mining Methods to Classify Maintenance Conditions for Real Estate Valuation

Author:

F.P. van Ingen

Supervisors:

B. Roorda

R.A.M.G. Joosten

Examination date:

July 10, 2020

External Supervisors:

A.J.A. Kort

Abstract

This study describes the potential of applying text mining methods for estimating the state of maintenance of real estate through advertisement texts. In order to succeed in the current growing housing market, mass appraisals are applying real estate valuation models on a large scale. However, these models are still striving for improvements in terms of accuracy, explainability and objectivity. Improving the classification of maintenance conditions by analyzing real estate advertisement through text mining methodologies could contribute to these estimation improvements.

Firstly, by implementing a dataset of approximately 65,000 real estate samples and combining count-based and word embeddings text mining methodologies with supervised machine learning classification algorithms, we created twelve models suitable for predicting maintenance conditions through advertisement texts. The results of this study show that the Term Frequency-Inverse Document Frequency (TF-IDF) model combined with the Logistic Regression (LR) classifier obtained best results (F1 = .717), where 28% of the predictions were differentiating from the maintenance scores estimated by appraisers and other experts. Secondly, the real estate valuation model of Ortec Finance is used as a benchmark model to estimate the contribution of text mining methods to existing real estate valuation models. Compared to the benchmark model using the original maintenance scores, which were set manually by appraisers, the model using text mining methodologies did not improve significantly. However, it did show an increase of the accuracy by 5.26%.

This study concludes that there is certain potential of applying text mining methods for real estate valuation in terms of automatization and explainability of real estate maintenance estimations. In addition, it contributes to normalize such practices and therefore shows potential for objectifying maintenance estimations. While considering the limitations and recommendations of this research, Ortec Finance could leverage this text mining classification tool to assist municipalities, appraisers and other experts for correcting real estate maintenance conditions in a more objective manner.

Keywords: real estate valuation, text mining, supervised machine learning algorithms, advertisements text, classification problem.

Acknowledgement

This report presents my graduation research for the master study Industrial Engineering and Management at the University of Twente. My internship at Ortec Finance was an enriching experience and I am thankful for the great opportunity the company provided for me. Due to COVID-19, I spent most of my internship at home, which felt a little bit odd in the beginning. I'm delighted that Ortec Finance still managed to motivate and support their interns in these uncertain and challenging times. I would like to thank my colleagues at the real estate department for their interest and knowledge. My gratitude especially goes out to Dirk-Jan for being such an involved and approachable supervisor, on both content as personal level.

Furthermore, my appreciation goes out to Berend and Reinoud for their constructive feedback and devoted time. In addition, I am especially thankful for the out-of-the-box questions they raised during our meetings, which made me rethink my work and which prepared me for the thesis defence.

At last, I would like to thank my family and friends. In particular, I would like to thank my girlfriend, Anne, who always encouraged me along the way. Furthermore, my thanks go out to my parents who unconditionally supported me during my time as a student.

The Hague, July 10th, 2020

Frank van Ingen

Contents

1 Introduction

1.1 Background

Real Estate received its own global industry classification in 2016 and makes up, on average, 5.1% of any institutional portfolio [21]. All property in the Netherlands has to be appraised yearly in order to determine real estate values for tax purposes. The size of the real estate tax or 'onroerendezaakbelasting'(OZB) depends on the value of the property. According to CBS [8], Dutch municipalities are expecting a total of 4.3 billion euro OZB yield in 2020, which is a 4.7% increase compared to 2019. The Real Estate Appraisal Law requires the properties to be appraised yearly before the 1st of January. Taxes in the current year are based on the assessed value in the preceding year. The number of Dutch real estate appraisers is simply too small to value the more than 7.9 million residential properties [3]. That is why yearly valuation is only possible with the help of automated valuation models (AVMs). This paper describes the potential of text mining methodologies to classify maintenance conditions of real estate, which can be used to improve current real estate valuation models.

Ortec Finance

Ortec Finance provides technology and advisory services for risk and return management. The company, headquartered in Rotterdam, was established in Rotterdam in 1981, and currently has over 250 employees with offices in Amsterdam, Londen, Toronto, Hong Kong, Zurich and Melbourne. They serve more than 500 clients in more than 20 countries with a total asset management of over 3 trillion euro [35].

Real Estate Valuation Model

Ortec Finance has over 25 years of experience in delivering real estate valuations. The real estate valuation department of Ortec Finance provides technology and solutions, using automated valuation models and computer-assisted mass appraisal, to enable real estate specialists to manage the complexity of real estate appraisal. Real estate valuations have crucial roles in the trade of municipalities, since they are the main factor that determines how much property taxes are raised.

The model used by the company, called the Hierarchical Trend Model (HTM) [15], is a statistical model and has already been operational for more than 25 years to value houses in different Dutch municipalities, varying from urban areas like Amsterdam to rural areas like Oldambt and

Voerendaal. HTM is an example of a state-space model, where some parameters are allowed to vary over time (time-dependent) and others are not allowed to vary over time (time-invariant).

In the HTM, the time-varying components are specified by trends at different levels, thus hierarchical trends. The model considers three levels: common, district and house type group trend. The district and house type group trends are modeled as deviations of the common trend by random walks. The random walk model assumes that the expected price level in the coming month is equal to that of the current month. As explained by Francke [15], the common trend has a more sophisticated specification, namely a local linear trend model, which assumes that the expected price change in the coming month is equal to the change in the current month. Both model specifications are flexible, but at the same time quite parsimonious with the number of variables needed. All three levels therefore do not require much information in order to predict real estate prices.

The time-invariant part of the HTM concerns the specification of the housing characteristics [14], for example, the size of the house and the condition of its maintenance. This part of the model is a non-linear specification that enables separation of the value of the house and the value of the land. The assumption that the coefficients of these variables are constant over time can be questioned, since the value of maintenance conditions may decrease over time. This non-linear specification can be seen as a way to adjust selling prices for differences in characteristics, thus giving standardized prices. The HTM can be paraphrased as follows:

$$\begin{aligned}
&Natural\ logarithm\ of\ the\ selling\ price\ of\ house\ i\ = \\
&\quad the\ level\ of\ the\ general\ trend\ at\ time\ t \\
&\quad +\ the\ level\ of\ district\ time\ component\ j\ at\ time\ t \\
&\quad +\ the\ level\ of\ house\ type\ time\ component\ k\ at\ time\ t \\
&\quad +\ the\ neighbourhood\ level\ l \\
&\quad +\ \textbf{the influence of the individual characteristics for house i} \\
&\quad +\ an\ error\ term
\end{aligned}$$

Francke and Vos [14] provided a detailed description of the HTM. As already mentioned, one individual housing characteristic is the state of maintenance. This component is currently estimated by appraisers and can therefore be used in real estate valuation models.

Text Mining

Text Mining (TM) in big data analytics is emerging as a powerful tool for harnessing the power of unstructured textual data by analysis. TM analyzes data to extract new knowledge and to identify patterns and correlations hidden in the data [19]. The goal of TM is to extract explicit and implicit concepts and semantic relations between concepts using Natural Language Processing (NLP) techniques. Thus, NLP is a component of text mining that performs a kind of linguistic analysis. TM draws on methods from statistics, machine learning, reasoning, information extraction, knowledge management, and other techniques for its discovery process [18].

In recent years, researchers have witnessed an increase in the quantities of available online textual data. Analyzing this data generates new insights and thereby opens up opportunities for research along various disciplines, such as cyber criminality, molecular biology domains and real estate valuation.

1.2 Research Proposal

Problem Statement

The process of determining values of real estate is time-consuming and demands various dependencies such as maintenance conditions and time-dependent features. In order to succeed in the current increasing housing market, mass appraisals are applying real estate valuation models on a large scale. However, these models can be improved even furtherly in terms of accuracy.

Nowadays, the state of maintenance is determined by means of photos and advertisement texts. Appraisers investigate the photos and texts, and determine the maintenance conditions themselves and thus manually. This step of the valuation process is time-consuming and dependent on the appraiser, which in turn results in inefficiency and subjectivity. In order to improve the process, text mining methodologies could be used instead, increasing efficiency and accuracy of the current real estate valuation model. Although text mining implications for predicting the state of maintenance look promising, Ortec Finance is currently not adopting such implications in their current valuation models. Moreover, the legal incentive is currently lacking to start using TM. Appraisers are obliged to meet specific requirements in order to be validated by the NRVT (Nederlands Register Vastgoed Taxateurs), which is necessary to guarantee high quality of appraisals. The validation standards do not specify the requirements concerning determination of the maintenance of real estate.

The firm may gain a lot from improving their valuation model by using TM methodologies. More precisely, TM involves more objective and automated estimations, which may increase the accuracy of the model. RTL Nieuws, a Dutch newspaper, concluded after research [10], that wrong real estate value estimations by municipalities could cause an immense amount of objection costs. In 2019, Dutch municipalities allocated 10 million euros to real estate valuation objection processes. 38.8% of all objections evaluated in 2019 were actually honored [33] due to possibly inadequate estimations. Overestimating or underestimating the state of maintenance could be an explanatory factor of inadequate estimations. Improving the accuracy of the valuation model, will result in better serving the municipalities as they can in turn raise taxes more adequately and decrease the number of objections. In addition, the ”Waarderingkamer”, supervisor of the act of raising property taxes, stated the growing importance of maintenance estimations within appraisals [33]. Maintenance conditions corresponding to property need to be controlled more frequently, which in turn emphasizes the importance of adequate and automated maintenance estimations to meet the growing demand of estimations in the future.

Research Goal

Our goal is to determine the potential of TM methodologies for classifying the state of maintenance through advertisement texts for real estate valuations. A dataset of approximately 65,000 house advertisement texts, including maintenance scores, with a time span from 2015 to 2020 is provided by Ortec Finance in order to properly test TM methodologies and train classification models.

First, an evaluation on text mining methodologies is performed in order to gain insight in the field of text mining. Text mining is a variation on a field called data mining [18], which generally refers to the process of extracting interesting and non-trivial information and knowledge from unstructured text. Unstructured texts can for instance be real estate advertisement texts provided by housing corporations and other institutions. As previously mentioned, current maintenance scores are subjective, since they depend on the appraiser. Automatically generated information from unstructured advertisement texts can predict maintenance conditions of real estate and can therefore positively contribute to the accuracy and consistency of existing AVMs.

Secondly, text mining methodologies and machine learning algorithms are investigated in order to generate a selection of textual features, which contribute mostly to the accuracy of predicting maintenance conditions. This selection will be fundamental for the predictive model we aim to create in this study.

Finally, we test the resulting model on its practical contribution. The model will be implemented in the current company's real estate valuation model in order to evaluate deviations in its accuracy for the prediction of real estate prices after implementation. The results will indicate whether text mining methodologies show potential in the improvement of accuracy of current real estate evaluation models.

Research Questions

In the context of our research goal, our main research question is as follows:

What is the potential of applying text mining methods for estimating the state of maintenance of Real Estate and its contribution to current Real Estate valuation models?

While this question sets the main premise, we break down the above question into several sub-questions to really define our scope. With respect to the identified goals, the sub-questions are formulated as follows:

Sub-question 1 (S1): *Which TM methodologies are suitable for predicting maintenance conditions through Real Estate advertisements?*

Sub-question 2 (S2): *Which set of textual features predict the state of maintenance most accurately?*

Sub-question 3 (S3): *To what extent can text mining methods contribute to current real estate valuation models?*

Relevance and Motivation

According to de Volkskrant [12], DNB advocates new measures to improve the quality and independency of appraisals. The use of AVMs certainly contribute to the wishes of the Dutch regulatory bank, but present valuation models fail to exclude the appraisers' manual tasks completely. Improving current valuation models by predicting specific housing characteristics, such as maintenance, automatically may contribute to the ideology of DNB.

To the best of our knowledge, no studies have been done on improving existing real estate valuation models by predicting the state of maintenance of real estate through advertisements using TM methodologies. Few studies have been done on directly valuing real estate prices through advertisement analysis. For example, Dick Stevens [29] attempted to directly predict selling prices through real estate advertisements, but classification and regression performances were lacking in terms of accuracy. We believe that existing AVMs perform significantly better, and that is why this paper will be dedicated to improving these models using text mining.

2 Literature Review

2.1 Text Mining

Data Mining is a field, which has seen rapid advances in recent years because of the immense advances in hardware and software technologies, which has led to the availability of different kinds of data. According to Aggarwal and Zhai [2], this is particularly true for the case of text data, where the development of hardware and software platforms for the web and social networks has enabled rapid creation of large repositories of different kinds of data. The increasing amounts of text data available from different sources has created a need for developments in algorithmic design, so interesting patterns from the data can be identified in a dynamic and scalable way.

Text mining is a variation on data mining. Data Mining and text mining differ in the type of data they handle. While data mining handles structured data coming from systems, such as databases, text mining deals with unstructured data found in documents, social media, and the web [19]. Thus, the difference is that in text mining patterns examined within texts, are extracted from natural language text rather than from structured databases of facts. Text mining techniques can be understood from the processes that go into mining the text and discovering insights from it. These techniques generally employ different text mining tools and applications for their execution. One popular technique is known as text categorization [32].

Text categorization, the activity of labeling natural language texts with thematic categories from a set arranged in advance, and a popular text mining technique, has accumulated an important status due to the availability of documents in digital form. The main aim of text categorization is the classification of documents into a fixed number of predetermined categories [6]. While utilizing machine learning, the main purpose is to learn classifiers through instances, which perform the category assignment automatically. The first step in text categorization is to transform documents, which typically are strings of characters, into a representation opt for the machine-learning algorithm. The task of constructing a classifier for documents does not differ a lot from other tasks of machine learning. The main issue is the representation of a document [11].

The document's representation is one of the preprocessing techniques that is used to reduce the complexity of the documents and make them easier to handle. The document has to be transformed from the full text version to a document vector [20]. For instance, a given corpus may be drawn from a lexicon of about 100,000 words, but a given text document can be represented as a sparse term-document matrix of size $n * d$, where n is the number of documents, and d is the size of the lexicon vocabulary [2].

Almost all the known machine-learning techniques for classification such as decision trees, Bayes methods, SVM classifiers, and neural networks have been extended to the case of text data. According to Aggarwal and Zhai [2], a considerable amount of emphasis has been placed on linear classifiers such as neural networks and SVM classifiers, with the latter being particularly suited to the characteristics of text data.

Performance of classification models is usually based on how well they are predicting outcomes for new data points. According to Sarkir [28], this performance is usually measured against a test or holdout dataset, which consists of data points that were not used to influence or train the classifier in any way. There are several metrics for the determination of a model's prediction performance like accuracy, precision, recall and F1-score.

2.2 Former Research on Real Estate Valuation

There are various empirical researches [31][9] conducted on the use of Artificial Intelligence (AI) for predicting real estate prices. In order to investigate the potential of text mining methodologies within real estate valuation models, we will review a study conducted at Ortec Finance. This study is regarded as most relevant, as the context is similar to this study.

Ceyhan [9] investigated the potential of applying modern machine learning approaches to existing real evaluation models. The author compared stand-alone black-box machine learning, full transparent and hybrid models, which combine the first two. The Hierarchical Trend Model (HTM)[15], which is the state-space model that Ortec Finance employs, was used as full transparent model. Neural network, random forest, and gradient boosting regression methods were used to evaluate the accuracy of black-box models, and are compared against the HTM. The proposed hybrid model replaced the time-invariant part in HTM with parts of the machine learning models. The results of Ceyhan's study show that models using machine learning obtained higher accuracy compared to the transparent model. The hybrid model, combining both models, obtained the highest accuracy. Ceyhan [9] recommended the company to explore the potential in machine learning models further within real estate valuation models.

The research showed to be a valuable contribution to existing house price predicting models. We therefore conclude that existing real estate valuation models can be improved using modern methodologies in the world of data science.

2.3 Former Research on Text Mining

Although existing valuation models manage to predict house prices using machine-learning methodologies, most studies excluded real estate advertisements and thus text mining methodologies within their models. Only few studies [1][29] included TM methods to produce or improve real estate valuation models. We review those studies and present the most interesting findings. In order to achieve more insights on the field of text analytics, we look at TM applications in different disciplines, such as social media [30][25], child abuse [4] and crowdfunding [34]. Therefore, we will explore the methodologies used in these researches and, again, present the most interesting findings.

Abdallah [1] proposed a 2-stage regression model that uses text mining to improve the prediction of the prices of real estate classifieds. After pre-processing the data, Abdallah [1] computed the term frequency-inverse document frequency (TF-IDF) in order to convert textual attributes to numerical features. After excluding high correlated features, the remainder was implemented in the linear regression model. The study showed that text mining reduces the error and improves the accuracy of real estate price prediction when comparing to simple linear regression models, excluding TM methodologies.

Stevens [29] focused on the prediction of selling price, asking price and price fluctuation, using text-mining techniques. In contrary to Abdallah [1], Stevens [29] performed classification on the real estate dataset, dividing real estate prices into different classes. In order to construct the dataset and generate features for further price predictions, Stevens created stemmed unigrams (single) and bigrams (double) from words, and linearly transformed the counts of the grams. This linear transformation (Vectorizing) converts the n-gram matrices into vectors, which therefore is used to generate the TF-IDF scores. For the classification task, Stevens considered the following five classifiers: Support Vector Classification (SVC), Linear SVC (LinSVC), K-Nearest

Neighbor Classifier (kNN), Multinomial Naive Bayes (MNB), and Decision Tree (DT) classifier. The study showed that the MNB classifier showed relatively good performance for both selling and asking price predictions. The LinSVC classifier achieved the best results for predicting the fluctuation of real estate prices. According to the author, accuracy values of the prediction models were below the results of comparable research. Stevens therefore recommends to consider other machine learning classifiers and configurations in order to achieve better performance results.

Beyond the scope of real estate valuations, Surjandari et al. [30] used TM methodologies to examine public sentiment of staple foods price changes in Indonesia based on twitter data. Sentiment analysis is used to extract the semantic value of text documents, which are features for prediction purposes. The researchers performed sentiment analysis by building word occurrence probability-model based on pre-classified documents. How these classified documents are obtained, is not mentioned in the paper. The authors considered three different classifiers: MNB, SVC and DT. Results showed that SVM classifier produced higher accuracy than MNB and DT.

Philander and Zhong [25] performed sentiment analysis as well in order to capture sentiment from integrated resort tweets. The study used a dictionary-based approach, also called a lexicon-based method, to analyze social media microblogging data from twitter. A sentiment lexicon was used to determine the sentiment orientation of the online text. The sentiment score was constructed by scoring the tweet text for positive and negative using the sentiment lexicon. As mentioned by Philander and Zhong [25], using smaller samples while considering the same lexicon, may lead to increasing values of miscategorized variables. The authors recommend to continuously measure validity while considering different sample sizes when using sentiment analysis.

Amrit et al. [4] proposed a decision support system for identifying child abuse based on structural and free-text data. They generated features for prediction by using the bag-of-words approach, which is a simplified representation of documents. The most important features used for this bag-of-words approach are the occurrences of words in the text. By applying univariate statistical tests, like the chi-squared test (X2) or analysis of variance (ANOVA), the authors considered a top number of features. For the classification part, the authors considered Bernoulli Naïve Bayes (BNB), MNB, Random Forest (RF) and SVM. The TF-IDF method was used in order to weight features in MNB and RF classifiers. One remarkable observation is that

TF-IDF does not seem to improve the performances of the MNB model, where implementing weighted features in the RF model results in a higher performance than non-weighted features. For SVM classification, the highest F1-score and lowest fall-out rate was achieved when using the polynomial kernel and penalty parameter of 0.2, two important input parameters of the SVM classifier. The authors used the Receiver Operating Characteristic (ROC) curves to measure the performances of the different classification algorithms. The study showed that incorporating unstructured data (text data) within the prediction models increases performance for predicting child abuse. The best performances were attained when using a boosted Decision Tree algorithm like RF, or when using SVM, that outperforms NB mainly on recall. The area under the ROC curve (AUC) showed that a tuned SVM algorithm performed the best for the prediction of abuse from unstructured data (text data). According to the authors, this is in line with the majority of the text mining literature that also proposes SVM as the best choice algorithm. In addition, they elaborate on the fact that a classifier based solely on structured data did not outperform the SVM classifier based on unstructured data, which indicates the potential of text mining methodologies.

Wang et al. [34] studied the impact of sentiment orientations on successful crowdfunding campaigns through text analytics. The study proved that positive sentiment in the blurb and detailed description promotes the successful campaigns. The authors considered Conditional Random Field (CRFs) and SVM as classifiers for prediction. As already mentioned by Amrit et al. [4], Wang et al. confirmed that SVM proves to be among the most effective classification methods, and is therefore used for predictive analysis. Their baseline model for prediction included all variables except for sentiment factors. The accuracy of the model increased after implementing the sentimental features.

2.4 Conclusion

Text mining is a variation on data mining and it is certain that text mining has gained a lot of reputation by researchers over the last years. Although different simple count-based text mining methodologies are already implemented in current real estate models, studies in different disciplines indicate that untouched and more complex methodologies have potential in the improvement of existing models. For instance, Wang [34] showed that a complex TM method like sentiment analysis contributes positively to models excluding this. However, sentiment methods are primarily used for separating negative from positive sentiment texts. When considering mostly positive sentiment advertisements within a multi-class classification problem, this method may be too soft. Real Estate models still have the possibility to use different and more advanced TM methodologies to capture hidden patterns within real estate descriptions, which then operate as features for the prediction of maintenance conditions. In addition, it seems that SVM classifiers produce higher performance in text analytics than other comparable classifiers, such as NB and LR classifiers. Instead of assuming the previous, we aim to deliver a confirmation. Therefore, this study will test different sets of classifiers and features in terms of performance.

3 Methodology

The goal of this study is to determine the potential of TM methodologies for classifying the state of maintenance through advertisement texts for real estate valuations. As contribution to this goal, we first dive deeper into the classification process. Secondly, we briefly discuss the different TM methodologies, ML algorithms, and the applied performance indicator. Lastly, we discuss the benchmark model conducted by the company in order to measure the practical significance of this research.

3.1 Classification

This research aims to classify advertisement texts automatically into pre-defined maintenance categories. Therefore, this problem is considered to be a text classification problem, also known as document classification problem in the domain of natural language processing [28]. The challenge involves classifying text documents, in this case real estate advertisements, into various predefined categories based on inherent properties or characteristics of each text document. Text or document classification is also often called text categorization. However, we will explicitly use the word "classification", since we focus on using a supervised approach using classification. Supervised learning refers to Machine Learning (ML) algorithms, which are particularly trained on pre-classified data, known as training data [28]. There are several variants of text classification tasks, based on the number of pre-defined classes in the dataset. Since this research will deal with more than two classes, multi-class classification tasks will be performed.

Figure 1 shows a detailed workflow of an automated text classification system. The workflow is divided into two boxes called training and prediction, which are the main practices of building a supervised text classifier. We test the classifier with an 80-20 ratio as seen in other ML research [9][29]. In other words, 80% of the total dataset is used for training and 20% for testing the classification model. During the training part we aim to train the classification model, using features extracted from pre-processed text data. These features are numeric arrays or vectors, since traditional machine learning algorithms cannot include raw unstructured textual data. In Section 3.2, different pre-processing and feature extraction methodologies will be discussed. The training part of the model is performed using supervised machine learning algorithms, which combine the extracted features and their pre-defined classes in order to train the classification model. The different machine learning algorithms will be discussed in Section 3.3. Training the model involves feeding the training feature vectors, or training document representations, from the training dataset and its corresponding training classes such that the ML algorithm can learn behaviors or patterns corresponding to each class. This knowledge can then be reused for predicting classes corresponding to the test dataset. The combination of features and the ML algorithm yields a classification model, which is the output of the training 'box'. In order to estimate the best performing configuration settings for each TM method and ML algorithm within the classification models, this study uses 5-fold cross-validation and the Scikit-Learn GridsearchCV package, which exhaustively considers all configuration setting combinations set by the user [28]. This validation method is used to check whether the model performs consistently across the validation folds of data.

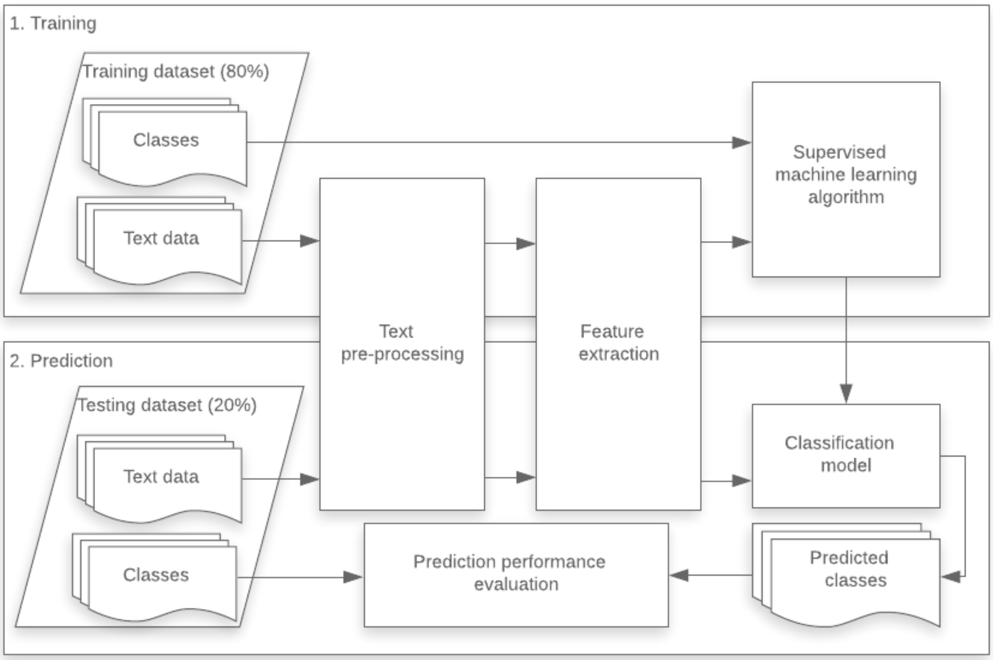


Figure 1: Detailed workflow for an automated text classification system.

The prediction part aims to either predict classes for new text data, or evaluate the predictive performance of the classification model. Predictions can be performed on the testing dataset using the same series of document transformation, namely text pre-processing and feature extraction. Afterwards, the predicted classes will be compared to the classes corresponding to the test dataset to evaluate the predictive performance of the model. This can be done using various performance metrics like F1-score, which will be discussed in section 3.4. Once the classification model is finalized, the last step is to implement the model in the current real estate valuation model of the company.

3.2 Text Mining Methods

3.2.1 Text Preprocessing

In the previous chapter, the workflow for an automated classification system is illustrated. Before feature extraction can be applied, raw text within documents needs to be converted into well-defined sequences of linguistic components that have similar structure and notation. In this chapter, we discuss several methods to normalize real estate advertisement texts. Besides the traditional preprocessing methods like tokenization, removing stop words and stemming, we also like to focus on removing unnecessary tokens, which will be applicable to this research.

Tokenization

Depending on the method of classification, texts need to be split up into several components, including sentences, which can be further broken down into sequences of words. Although splitting sentences into words sounds like an effortless task, it can be quite difficult to perform algorithmically. For this purpose, a tokenization package is considered. Tokenization can be described as the process of splitting textual data into smaller and more meaningful components [28]. We, therefore, use WordTokenization to split the text into words. This will allow the usage of models like n-grams, which uses combinations of n sequential words, in the analysis later on. The NLTK package in Python provides useful interfaces for word tokenization, which will be utilized during the analysis.

Stop Words

Stop words are words that are usually removed from corpora during preprocessing tasks, since these words carry little or no significant information [28]. Usually, these words occur most frequently when counting single words from a corpus. Therefore, stop words will be removed from the corpora. For the process of removing stop words automatically, we use the Dutch stop words list from the NLTK package.

Unnecessary tokens

In order to contribute to the linguistic value of advertisement text we try to intuitively remove tokens with no, little and/or overlapping significant value. As mentioned in the introduction of this paper, individual housing characteristics, like construction year and the surface of the house, are already included as features in the current automated valuation model of the company. This study aims to reduce the overlap and correlation between existing features and generated features by this model and therefore eliminate such values within the advertisement texts. Since this is computationally difficult and there is no existing package available that performs this task automatically, this part will be performed in an intuitive manner.

Next to overlapping information, punctuation marks will be separated from words and deleted as well. This is generally considered important, since ML algorithms will consider punctuation marks as numeric identities. These marks will not contribute to the predictive power, but will add to the computation time of such algorithms.

Stemming

Words in advertisement texts appear in many forms and tenses, whilst pointing to the same base form. Figure 2 shows how inflections are created by attaching affixes to the base form of the word. The process of obtaining the base form or stem word of inflected words is called stemming [28].

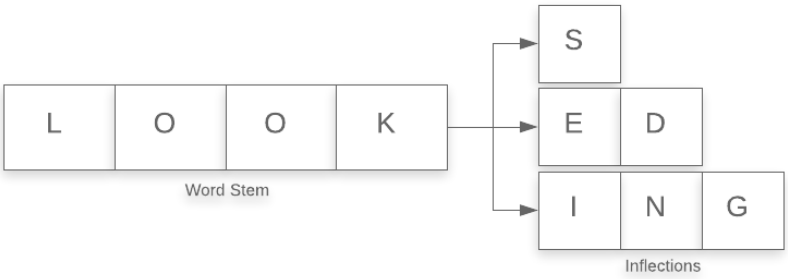


Figure 2: Word Stem and its inflections.

Thus, stemmers remove morphological affixes from words, leaving only the word stem. As already mentioned by Stevens [29], particular feature engineering models that include stemming have shown to be beneficial in a variety of studies. Therefore, the Dutch snowball model by Porter [26] will be applied, which is provided by the NLTK package as well. This model performs stemming on Dutch texts automatically.

3.2.2 Feature Engineering

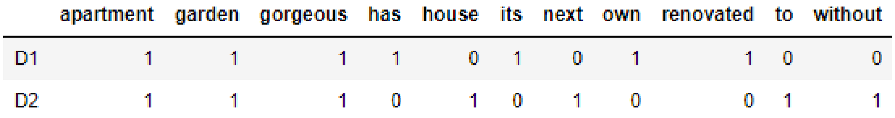
For the classification and machine learning part of the next section, preprocessed textual data has to be transformed into numeric representations of features for input. In this section, some popular and effective strategies for extracting meaningful features from text data are explored. These features can be used as a representation of advertisement texts and therefore be furtherly applied in building machine learning models for the automated classification task. In this section, the traditional (count-based) models like Bag of Words and more complex models like Word2Vec will be briefly discussed.

Bag of Words

A commonly used model for extracting feature vectors is the bag of words (BoW) or unigram model. This simple vector space model represents unstructured text as numeric vectors, such that each dimension of the vector is a specific word from the corpus [28]. The values corresponding to the dimensions are the occurrences, denoted by one or zero, of the word in that dimension of a particular text. Imagine we have two documents containing the following:

- D1: “renovated apartment has its own gorgeous garden”
- D2: “gorgeous house next to apartment without garden”

Figure 3 Illustrates how the documents are transformed into numeric vectors. The BoW model excludes the analysis of patterns within texts, like word order or grammar.

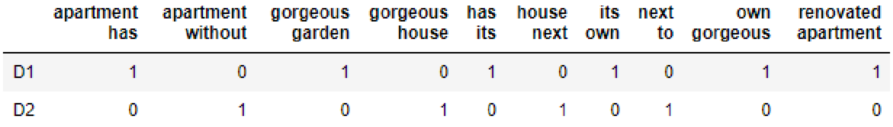


	apartment	garden	gorgeous	has	house	its	next	own	renovated	to	without
D1	1	1	1	1	0	1	0	1	1	0	0
D2	1	1	1	0	1	0	1	0	0	1	1

Figure 3: Bag of Words feature vectors for documents D1 and D2.

Bag of n-grams

As mentioned, the BoW model, also known as the unigram model, is invariant to word order. The bag of n-grams model can be used to generate numeric vector features, which obtain partial information about the word order in unstructured texts [28]. Thus, n-grams are collections of tokens (words) from a text document such that these collections occur in a sequence. Thus, bi-grams indicate collections of two words and tri-grams of three words. Figure 4 gives a partial example of a bi-gram, using the exact same text documents as the BoW model.



	apartment	apartment	gorgeous	gorgeous	has	house	its	next	own	renovated
	has	without	garden	house	its	next	own	to	gorgeous	apartment
D1	1	0	1	0	1	0	1	0	1	1
D2	0	1	0	1	0	1	0	1	0	0

Figure 4: Bag of n-grams feature vectors for documents D1 and D2.

TF-IDF

The vector features of N-grams and BoW models are solely based on absolute term frequencies. Terms that occur frequently across all documents may tend to overshadow other terms in the same feature set. TF-IDF tries to avoid this problem by combining term Frequency (tf) and inverse document frequency (idf) [28]. Term frequency is already computed in the BoW model and can be represented as the frequency f_{w_D} of word w in document D. The second step is to calculate the inversed document frequency. idf calculates for each term the inverse of the document frequency. It measures the word's degree of rareness across all documents and how important a word is to a specific document. Mathematically, it can be represented as follows:

$$idf(w) = log(\frac{N}{1 + df(w)}),$$

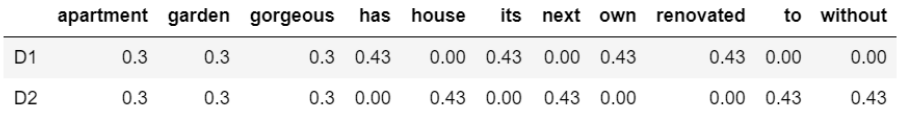
where the total number of documents N is divided by the frequency of documents df, containing the word w, and then applying logarithmic scaling to the outcome. The number 1 is added to the document frequency to prevent potential divisions by zero errors, and ignoring terms that might have zero $idf(w)$. The final step is to take the product of the tf and idf, in order to compute the term frequency-inverse document frequency ($TF–IDF$). It can be represented as follows:

$$TF–IDF(w, D) = f_{w_D} * log(\frac{N}{1 + df(w)}).$$

There are different versions of the TF-IDF model. We apply the normalized version of the TF-IDF matrix. This means that the squared elements of each vector in the matrix sum up to one. We will normalize the TF-IDF vectors by dividing it by the Euclidean L2 norm ($||TF–IDF||$). Mathematically, the final TF-IDF* feature vector is represented by:

$$TF–IDF^{*} = \frac{TF–IDF}{\|TF–IDF\|}.$$

Figure 5 gives an example of a TF-IDF, using the BoW model and the exact same text documents.



	apartment	garden	gorgeous	has	house	its	next	own	renovated	to	without
D1	0.3	0.3	0.3	0.43	0.00	0.43	0.00	0.43	0.43	0.00	0.00
D2	0.3	0.3	0.3	0.00	0.43	0.00	0.43	0.00	0.00	0.43	0.43

Figure 5: TF-IDF feature vectors for documents D1 and D2.

Word2Vec

While traditional models are nothing but pure frequency-based models, Word2Vec aims to understand the contextual meaning of text. It takes a text corpus as input and produces word vectors as output. The model was created by Google [17] in 2013 and, as mentioned by Sarkar [28], is a predictive deep learning based model to compute and generate high quality and distributed dense vector representations of words that capture contextual and semantic similarity. The resemblance between words can be generated by creating dense word embeddings for each word in the vector space representing the vocabulary of all text documents in the dataset. Thus, the total number of word vectors is essentially the size of the vocabulary of all text documents. There are two different model architectures that can be leveraged by Word2Vec for creating word embedding representations:

- The Continuous Bag of Words (CBOW) model
- The Skip-gram model

The CBOW model architecture tries to predict the target word (center word) based on the context words (surrounding words) and the Skip-Gram model architecture tries to achieve the inverse of what CBOW does, by predicting the source context words given a target word. Both model architectures are illustrated in Figure 6, where $w(t)$ is the center word and $w(t+2)$, $w(t+1)$, $w(t-1)$ and $w(t-2)$ the context words. According to Mikolov et al. [23], Skip-Gram models seem to produce higher accuracy than CBOW models. Therefore, we will leverage the Skip-Gram model in order to generate our word embeddings.

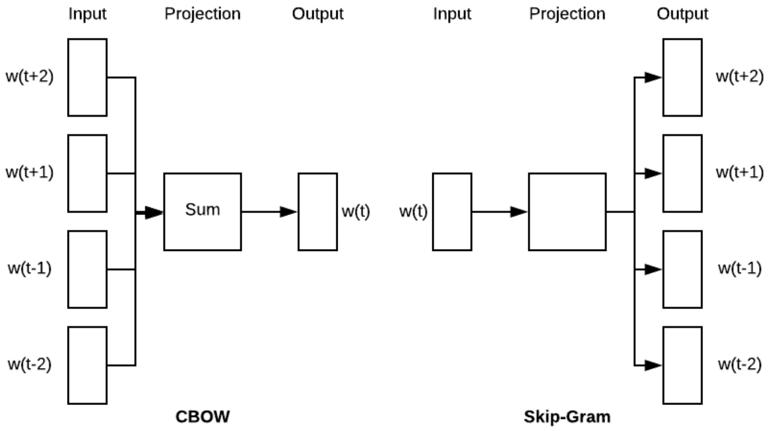


Figure 6: Word2Vec model architectures for creating word embedding representations.

As comprehensively described by Mikolov et al. [23], the training objective of the Skip-gram model is to find word representations that are useful for predicting the surrounding words in a sentence or a document. The objective of the model is to maximize the average log probability, given a sequence of training words w_1,w_2,. . . ,w_T. Mathematically, given a document of T words, the aim is to maximize the following:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c,j\neq 0} log\, p(w_{t+j}|w_t),$$

where $p(w_{t+j}|w_t)$ defines the output probability and c defines the window of context words (training words). Smaller c results in less training samples and thus can lead to less training time, at the expense of the accuracy. To generate the output probabilities, the model estimates a matrix, which maps the embeddings into a $|W|$-dimensional vector v_{w_i}. Using the softmax function, the probability of predicting the word w_o given the word w_i is defined as:

$$p(w_O|w_I) = \frac{exp({v'_{w_O}}^{\top} v_{w_I})}{\sum_{w=1}^{W} exp({v'_w}^{\top} v_{w_I})},$$

where v_w and $v_w^{'}$ are the input and output vector representations of w, and W is the number of words in the vocabulary. According to Mikolov et al. [24] this formulation is impractical, because the cost of computing the average log probability is proportional to the total amount of words in your vocabulary, which is often large. This problem will be tackled by using the hierarchical softmax objective function, which generally reduces the W output nodes in the neural network of the model. Gensim's Word2Vec Python package will be employed to train the model and create word embeddings for our vocabulary.

After generating word embeddings, using the Skip-Gram model architecture, we generate fixed averaged vector representations for each text document by average weighting the word vectors within that document. The remaining vector representations of all documents can therefore be used by machine learning algorithms for classification tasks.

FastText

The FastText model is an extension to Word2Vec and was proposed by Facebook in 2016 [13] and is based on the paper of Mikolov et al. [7]. Where Word2Vec treats each word in the corpus as an atomic entity and generates a vector representation for each word, FastText treats each word as composed of character n-grams. In other words, the FastText model tries to include the internal structures of words by proposing a different scoring function. Taking the word bike and tri-grams as an example, it will be represented by the character n-grams: <bi, bik, ike,

ke>. The word vector will therefore be represented by the sum of vector representations of its n-grams. Mathematically, the scoring function can be represented as follows:

$$s(w,c) = \sum_{g \epsilon G_w} {z_g}^{\top} v_c,$$

where $G_w \subset \{1, \ldots, G\}$ represents the set of n-grams appearing in word w, z_g the vector representation, which are associated to each n-gram g, and v_c the context vector. It is obvious that FastText adds a lot of additional computation to the training step of the model. However, the trade-off is a set of word-vectors that contain embedded sub-word information. Gensim's FastText Python package will be applied to produce new word vector embeddings for classification.

Doc2Vec

Doc2Vec is an extension to Word2Vec as well, and was introduced by Mikolov and Le [22] in 2014. The goal of Doc2Vec is to create a numeric representation of a document, regardless of its length. This vector will be added next to the input word vectors in the Word2Vec model. The Doc2Vec architecture models, Distributed Memory (DM) and Distributed Bag of Words (DBoW), are therefore slightly different, comparing to Word2Vec. Both models are illustrated in Figure 7.

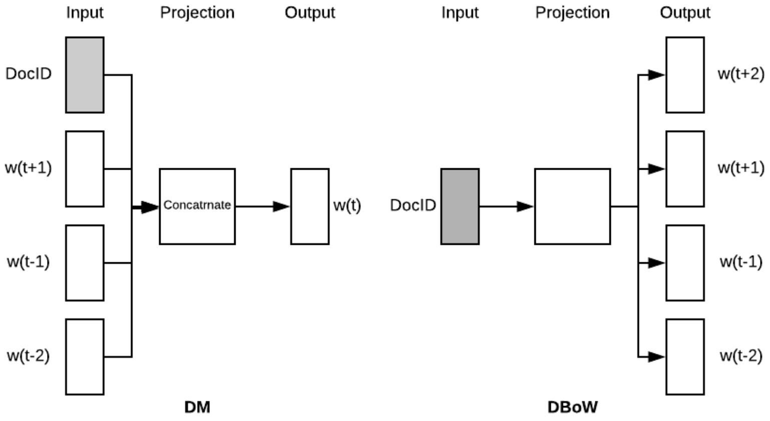


Figure 7: Doc2Vec model architectures for creating word embedding representations.

The DM model is analogous to the Word2Vec CBoW model. The difference is that DM maps a unique vector to each document DocID, represented by a column in a document matrix D. The document vectors and word vectors are therefore concatenated to predict the next word in a context. DBoW is analogous to the Word2Vec Skip-Gram model. Here, DBoW ignores the context words in the input and forces the model to predict words randomly sampled from the document in the output. The authors mention that DM alone usually works well for most tasks, with state-of-art performances, but in combination with DBOW is usually more consistent. Due to time constraints, this study will only leverage the first and stronger model, using Gensim's Word2Vec Python package. The combinations of different model architectures are left to future research, providing that DM obtains relative high performances across different TM feature engineering methodologies.

3.3 Machine Learning Methods

After the generation of numeric document representations, we are ready to use different supervised machine learning algorithms that are used for creating and learning this study's classification model. During this chapter, the training and tuning part of different algorithms will be discussed. Based on results of former research [1][29], the following classification algorithms will be briefly discussed:

- Multinomial Naive Bayes
- Logistic Regression
- Support Vector Machines

Multinomial Naive Bayes

The Naive Bayes (NB) algorithm is a supervised learning algorithm that puts the Bayes theorem into action [28]. This algorithm includes a "naive" assumption that each feature is completely independent of the others. Mathematically, this can be formulated, given a class variable y and a set of n features, in the form of a feature vector $\{x_1,x_2,\ldots,x_n\}$. Using the Bayes theorem, the algorithm denotes the probability of the occurrence of y given the features as follows:

$$P(y|x_1, x_2, ..., x_n) = \frac{P(y) * P(x_1, x_2, ..., x_n|y)}{P(x_1, x_2, ..., x_n)}.$$

Under the assumption that each feature is conditionally independent of every other feature and $P(x_1,x_2,\ldots,x_n)$ is constant, the conditional distribution over the class variable to be predicted, can be expressed as follows:

$$P(y|x_1, x_2, ..., x_n) = \frac{1}{Z}P(y) * \prod_{i=1}^{n} P(x_i|y),$$

where the evidence measure, $Z = p(x)$, is the constant scaling factor, which is dependent on the feature variable. Combining this equation with the Maximum a Posteriori (MAP) decision rule, which can be used to obtain a point estimate of an unobserved quantity, the Naive Bayes classifier can be built.

Multinomial Naive Bayes (MNB) is an extension to NB and usable when using multiple distinct classes. The distribution can be represented as $p_y = \{p_{y_1}, p_{y_2}, \ldots, p_{y_n}\}$ for each class label y and the total number of features n, which could be represented as the total vocabulary of distinct words or terms in text analytics. In this case, $P(x_i|y)$, or p_{y_i} will represent the probability of feature i that is connected to class y. Parameter p_{y_i} will be estimated with a smoothened version of maximum likelihood estimation (MLE) and represented as follows [28]:

$$\hat{p}_{y_i} = \frac{F_{y_i} + a}{F_y + an},$$

where

$$F_{y_i} = \sum_{x \epsilon TD} x_i, \qquad F_y = \sum_{i=1}^{|TD|} F_{y_i}.$$

Here F_{y_i} represents the frequency of occurrence for the feature i in a document with class label y in our training dataset TD. F_y represents the total frequency of all features for the class label y. a will represent the smoothening hyperparameter, which can be adjusted for tuning the model. The SciKit-Learn Python library provides an implementation of the MNB classifying algorithm and will therefore be applied. Because MNB assumes the variables to be non-negative, this study is not able to employ the MNB classifier for training the models using word embeddings, since these embeddings may contain negative values.

Logistic Regression

The Logistic Regression (LR) model is a statistical model that uses the logistic mathematical function to estimate parameter values. These parameters will serve as the coefficients of all our features. The model focuses on maximizing the likelihood of the predicted values to the observed (true) values. Consider a standard multiple linear regression model, depicted as follows:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n,$$

where $\{x_1, x_2, \ldots, x_n\}$ is a set of n features, in the form of a feature vector, and the model tries to estimate the coefficients $\{\beta_1, \beta_2, \ldots, \beta_n\}$ relating to the features. Considering the prediction of two categorical classes, we can represent this, using the logit of the probability p of predicting a specific class. The odds of p is $p/(1-p)$, which is the ratio of the favorable outcomes to the unfavorable outcomes. This is used as the standard unit of measurement for the log-odds scale and is represented as follows:

$$log(\frac{p}{1-p}) = \beta^\intercal x,$$

where

$$\beta^\intercal x = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n.$$

Here $\beta^\intercal$ is the weight vector and x the feature vector of one training sample. In order to obtain the class probability values that the LR model generates as output, we derive the following equation using the sigmoid function:

$$p = \frac{1}{1 + e^{-(\beta^\intercal x)}}.$$

Considering multiclass logistic regression with K classes, the probability that sample i belongs to class k given the feature vector x can be computed using the generalized softmax function and is represented as follows [5]:

$$p(y_i = k|x_i) = \frac{e^{\beta_k^\intercal x_i}}{\sum_{c=1}^{K} e^{\beta_c^\intercal x_i}}.$$

With the help of MLE we optimize and estimate the optimal coefficients or weights for each feature, which helps in maximizing the likelihood function. We will leverage the SciKit-Learn library, which provides an implementation of the LR model. The hyperparameters of the LR model, which will be tuned during the analysis, are penalty and the inverse of regularization. The penalty is used to specify the type of normalization used, and the inverse of regularization is used to specify the strength of regularization, which is a technique to discourage the complexity of the model [28].

Support Vector Machines

Support Vector Machine (SVM) is a supervised learning algorithm and can be used for both classification and regression. The SVM classification algorithm tries to construct a hyperplane of a collection of hyperplanes for the creation of a high-dimensional feature space [28]. Considering a two-class SVM, the algorithm takes in a training dataset of n data points $(x_1,y_1),\ldots,(x_n,y_n)$ such that the class variable $y_i \in \{1,\text{-}1\}$ where each value corresponds to the point x_i. Each data point x_i will represent a feature vector. The objective of the SVM is to find the max-margin hyperplane, which separates the set of data points having label 1 from the set of data points having label -1 [28]. The sample data points nearest to the hyperplane are known as support vectors.

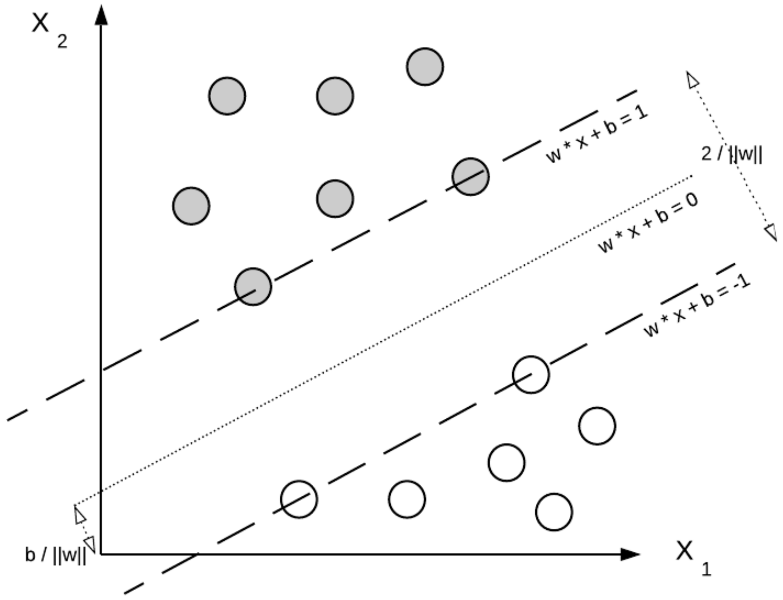


Figure 8: Two-class SVM including hyerplane and support vectors.

As illustrated in figure 8, the hyperplane is defined as the set of points x which satisfy $w * x + b = 0$, where w represents the normal vector to the hyperplane and $b/||w||$ the offset of the hyperplane from the origin. The data points in this illustration are linearly separable, and therefore we may construct hard margins, which are represented by the two hyperplanes $w * x + b = 1$ and $w * x + b = -1$. It is possible that a particular dataset cannot be separated linearly. In this case, we can use the hinge loss function, which can be represented as $max(0,1 - y_i(w * x + b))$[28], to obtain soft margins for the classification task.

When dealing with multiple classes n, for each class a binary classifier is trained to separate data points between the class and other $n-1$ classes. During the prediction of the data points, the distances to hyperplanes for each classifier are computed and the maximum score is chosen for selecting the class label. Thus, considering having three different classes, a total of three SVM classifiers will be trained for each class. Stochastic gradient descent is often used for minimizing the loss function in SVM algorithms [28]. This study will leverage the SciKit-Learn library, which provides an implementation of SVM. One important hyperparameter of the model is the kernel function, used to convert the existing feature space into a dimensional feature space, where data can be separated linearly. As mentioned before, this study will leverage the hinge loss function to obtain soft margins. Future research may explain which kernel function is favorable when tackling unstructured text representations, which already contain a huge number of dimensions.

3.4 Evaluation of Classification

Appropriate metrics are needed in order to measure the performances of classification models. For classification tasks, the F1-score is a popular performance metric and therefore extensively used by other studies [30][29][4]. This study therefore uses this performance metric to evaluate the classification models within this research.

The F1-score is often called the harmonic mean of precision and recall. Here, precision is defined as the number of predictions made that are actually correct or relevant out of all the predictions based on the positive class. In addition, recall is defined as the number of instances of the positive class that were correctly predicted [28]. The F1-score can be represented as follows:

$$F1_C = \frac{2 * Precision_C * Recall_C}{Precision_C + Recall_C},$$

$$Precision_C = \frac{TP_C}{TP_C + FP_C}, \qquad Recall_C = \frac{TP_C}{TP_C + FN_C},$$

where TP_C represents the number of correctly classified instances of class C, FP_C the number of falsely classified instances belonging to class C and FN_C the number of instances belonging to class C, but not correctly classified by the model. Since the dataset is imbalanced, as will be discussed in Section 4, this study will use the weighted average F1-score to normalize the scores regarding class proportion. The F1-score can range between 0 and 1, where 1 denotes perfect precision and recall. The set of features and ML algorithms with the highest F1-score will be elected as our final classification model.

3.5 The Benchmark Model

As mentioned in the introduction of this paper, the statistical valuation model used by the company is called the Hierarchical Trend Model [15] and will be used to evaluate the practical significance of the classification model including textual features. HTM is an example of a state-space model, where some parameters are allowed to vary over time (time-variant) and others are not allowed to vary over time (time-invariant). The time-invariant part of the HTM concerns the specification of the housing characteristics [14]. Maintenance condition characteristics belong to the time-invariant part of the model and are currently estimated manually by appraisers. The outcomes of the classification model may replace the estimations of the appraiser and tries to improve consistencies within the HTM model. Therefore, the usage of the classification model contributes to the reduction of prediction error, the difference between the estimated value and the true value. Hence, we have to examine whether automated maintenance valuations through text mining methodologies lead to a significant improvement within the HTM model. The HTM model can mathematically be represented as follows:

$$y_t = A_t + f(X_t, \beta),$$

where y_t is the log house selling prices for time period t, $f(X_t, \beta)$ is a partly nonlinear function of housing characteristics X_t with corresponding coefficients β, and A_t different other components of the HTM, such as district trends and other time-variant components. These are comprehensively described by Francke [14]. The nonlinear specification of housing characteristics can be further specified as follows:

$$f(X_t, \beta) = ln(x_1^{\beta_1} exp(Q\delta) + \beta_2 x_2 + \beta_3 x_3) + \beta_4 ln x_4,$$

where x_1 represents the size of the house, x_2 the lot size, x_3 parts other than land and buildings, like garages. The variable x_4 concerns values of both building and land, and Q contains remaining housing characteristics such as maintenance conditions.

Changing the maintenance housing characteristic will directly have impact on the house price and therefore may contribute to the reduction of prediction error. We estimate this error by using the Mean Absolute Percentage Error (MAPE) represented as follows:

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right|,$$

where N represents the total amount of sample observations, y_i the observed value, and $\hat{y}_i$ the predicted value of observation i. In this study, the MAPE will be measured before and after the implementation of our classification model in order to estimate deviations in the prediction error. To test whether the performance of the HTM including TM methodologies is significantly different, the Wilcoxon matched-pairs signed-ranks test will be performed. This is a non-parametric equivalent of the paired t-test with as null hypothesis (H_0) that the paired differences between two samples follow a symmetric distribution around zero [36]. Unlike the paired t-test, the paired differences of the Wilcoxon test do not need to follow a normal distribution, which makes this test more suitable for this study. The errors $e_{i,1} = |y_i - \hat{y}_{1,i}|$ and $e_{i,2} = |y_i - \hat{y}_{2,i}|$ will represent the pair of error i of the HTM including and excluding TM methodologies, respectively. After determining the *sign* of the differences $d_i = e_{i,1} - e_{i,2}$, the differences are ranked in order of absolute size with a rank of 1 assigned to the smallest difference. We reassign the signs of the differences to their respective ranks R_i and are therefore able to calculate the test statistic, which can be represented as follows:

$$W = \sum_{i=1}^{N^*} sign(d_i) * R_i,$$

where N^* is the reduced sample size, excluding differences equal to zero. For N^* larger than 20, the W test approximates the normal distribution. Therefore, the z-statistic will be used, which can be represented as follows:

$$z_{stat} = \frac{W - \frac{N^*(N^*+1)}{4}}{\sqrt{\frac{N^*(N^*+1)(2N^*+1)}{24}}}.$$

This study rejects H_0 if the computed z-statistic falls in the rejection region. Using a level of significance of 0.95, this study rejects H_0 if $z_{stat} > +1.96$ or $z_{stat} < -1.96$.

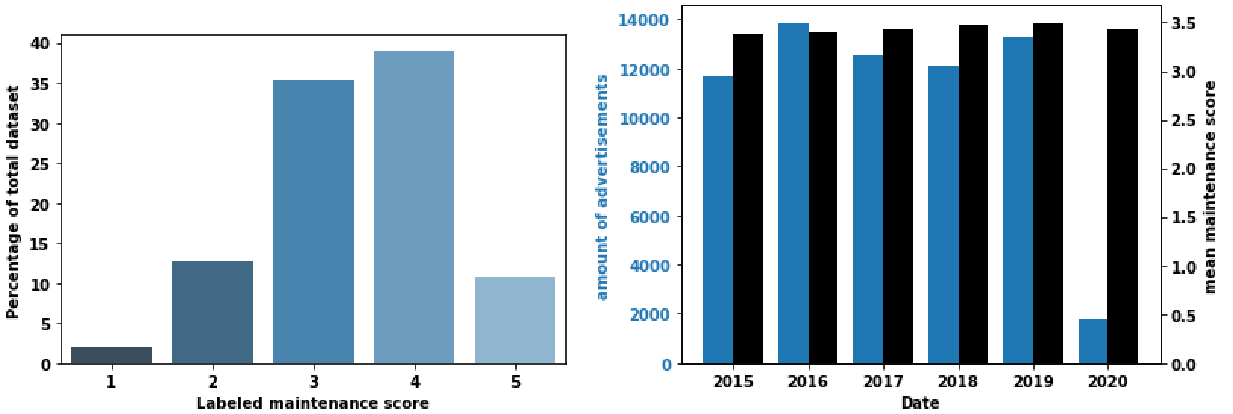
4 Data

The dataset used by this research consists of approximately 65,000 unique housing samples from April 2015 until February 2020 and has been constructed by Ortec Finance. All samples correspond to houses allocated in Amsterdam, the Netherlands, and contain data of individual housing characteristics, including advertisement texts. Table 1 shows an example of information of one dummy sample.

house_id:	324242
address:	Westerdoksdijk
residential_code:	3605
transaction_price:	409,000
model_price:	416,000
advertisement:	“INDELING: Het appartement is bereikbaar via een trappenhuis voorzien van een intercomsysteem. In de hal bevindt zich de meterkast, een ingebouwde garderobekast en de voordeur van het appartement. Eenmaal binnen bevindt u zich tussen keuken en woonkamer, die in directe verbinding staan met elkaar. Deze indeling zorgt ervoor dat er altijd licht van twee kanten binnenvalt. De royale woonkamer bevindt zich aan de voorkant van het huis, waar drie grote ramen uitzicht bieden op de straat. De halfopen keuken is voorzien van een ingebouwde kookplaat, afzuigkap en koel/vries combinatie. Er is ruimte voor het plaatsen van een aparte oven/magnetron en het is ook mogelijk om een vaatwasser in te laten bouwen. . . “
maintenance_score:	4.0
transaction_date:	14-05-2019

Table 1: Information on single instance in the dataset.

The transaction price is monitored on the date of transaction. The real estate valuation model, used by the company, determines the model price, and its deviation with the transaction price of the house represents the estimation error of the model. The maintenance scores, corresponding to the houses, are estimated by multiple appraisers and experts and rank from the ordinal data scale of one to five. Here, one refers to bad and five refers to good maintenance conditions assigned to the house. As illustrated in Figure 9a, the dataset can be labeled as imbalanced, because the frequencies of maintenance scores are not equally distributed. More specifically, the model already obtains an accuracy of 39% when all advertisements are automatically ranked as four. The number of samples and the mean maintenance score per year are shown in Figure 9b. This figure shows that the mean maintenance score over the years is almost constant.



(a) Proportion per maintenance score

(b) Number of advertisements and mean score per year

Figure 9: Descriptive analysis dataset.

The geographical points of the house IDs corresponding to houses in Amsterdam are shown in Figure 10. The points differ in color from green to red. Here, red refers to bad and green refers to good maintenance conditions assigned to the point. As can be seen in the figure, the center of the capital city possesses houses with better maintenance conditions in contrast to the suburbs.

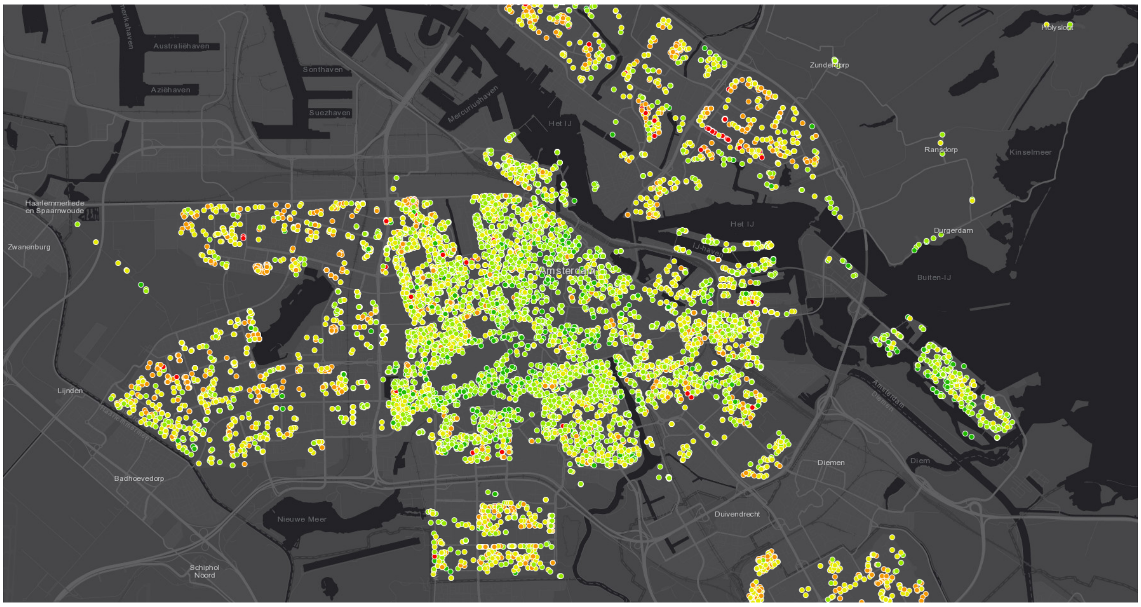


Figure 10: Geographical visualisation of the dataset. ● = good maintenance, ● = bad maintenance.

5 Results

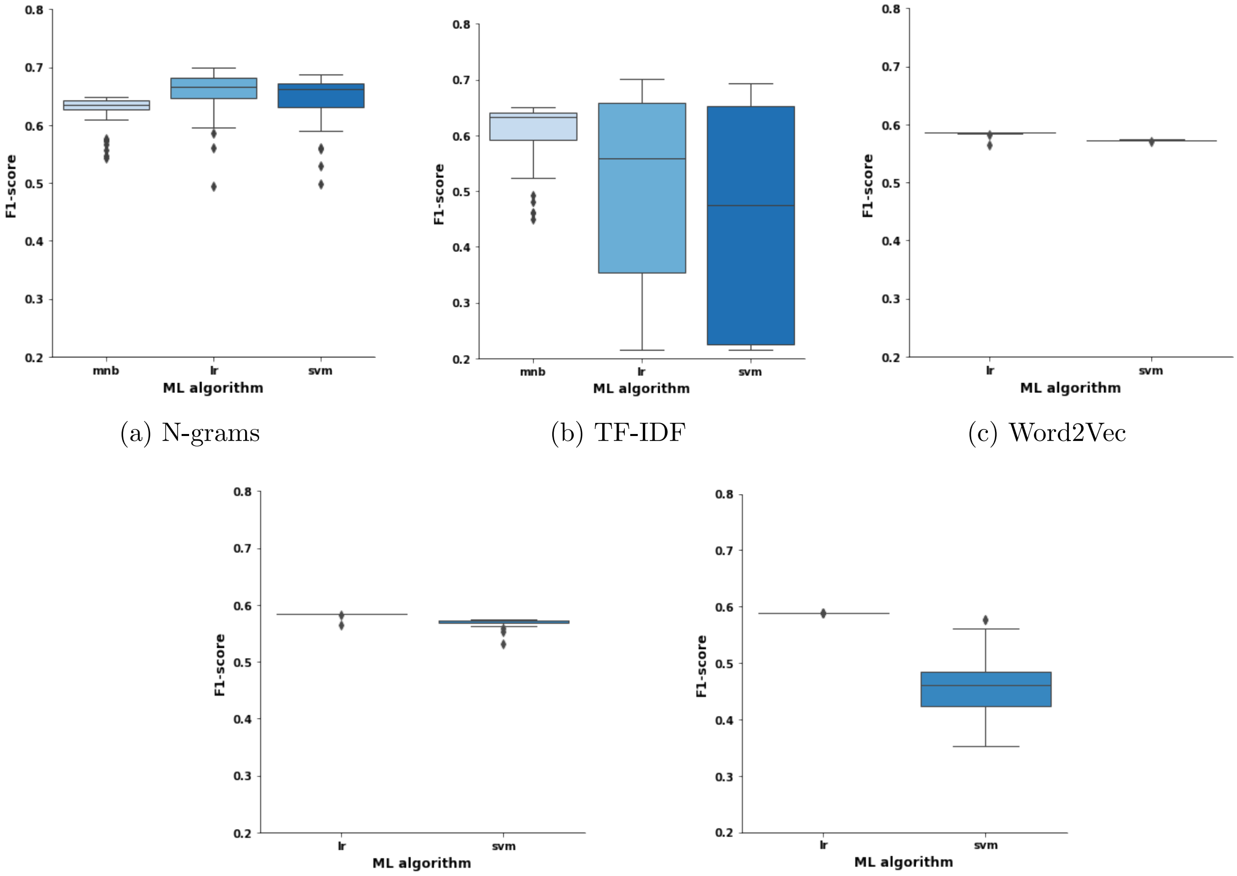
In this section, we present the results of this study. First, the results of the text mining methodologies combined with different supervised machine learning algorithms will be shown. As discussed in Section 3.4, the F1-score will be used as performance metric in order to measure the performances of the classification models. This is the harmonic mean of the model's precision and recall. Secondly, the results of the best performing classification model are used as input within the company's benchmark model. The explanation of the benchmark model can be found in Section 3.5. At this part, the MAPE will be used as performance metric and the Wilcoxon matched-pairs signed-ranks test to measure whether there is a significant difference between the errors of the models using the original maintenance values and the predicted values after implementation of the best performing classification model using TM methodologies.

5.1 Classification Results

The hyperparameters of different classification models combining TM methods with supervised ML are tuned using 5-fold cross-validation, as discussed in Section 3.1. The distributions of means of the cross-validation results per TM method are given in Figure 11. The BoW model is integrated into the n-grams model, since BoW is a stand-alone uni-gram model. Note that MNB classifier can only be applied to TF-IDF and n-grams, as explained in Section 3.3. Figure 11 shows that for all models the highest F1-score that is obtained, using the LR classifying algorithm. MNB performs worst for both TF-IDF and n-grams models. The best configuration settings per TM model are used for predictions on the test set. As shown in Table 2, best performance is achieved using the count-based term frequency-inverse document frequency method, using 1,2-grams and the Logistic Regression algorithm as supervised machine learning classification algorithm (F1= .717). TM models using word embeddings for the creation of feature vectors are performing worst, in particular Doc2Vec (F1= .549). Considering the fact that Doc2Vec and Fasttext are extensions of the Word2Vec model, they fail to contribute positively to the basic model.

TM model	Setting	Classifier	$N_{classes}$	F1-score
N-grams	1,2,3-grams	LR	5	.713
TF-IDF	1,2-grams	LR	5	**.717**
Word2Vec	SkipGram	LR	5	.588
FastText	SkipGram	LR	5	.586
Doc2Vec	DBoW	LR	5	.549

Table 2: Best performances per text mining methodology after tuning.



(a) N-grams

(b) TF-IDF

(c) Word2Vec

(d) FastText

(e) Doc2Vec

Figure 11: 5-Fold Cross-validation F1-scores per TM method.

The percentages of predicted values by the TF-IDF (1,2-grams) model with respect to the actual values are shown in Figure 12. In this matrix, we see that the model assigns different maintenance labels for 27.92% of the test samples within the dataset compared to the original values estimated by appraisers and experts. The highest amount of misclassifications is obtained predicting samples with maintenance score 1 (69.93%). Samples with a predicted maintenance score of 4 are most likely to equal the original values of the samples (81.05%). The colors within the boxes represent the number of samples classified correctly or incorrectly. As already illustrated in Figure 9a, we see that most samples have maintenance scores 4 and 3, which depicts that the distribution is skewed or imbalanced.

The best performing TF-IDF model uses 1,2-grams. In order to measure which uni-grams and bi-grams contribute mostly for measuring maintenance, we look at the n-grams with the highest correlations with different maintenance scores. These n-grams are shown in Table 3. We see that, for instance, household products like Quooker and Miele are highly correlated to real estate advertisements containing high maintenance scores. Uni-grams and bi-grams containing

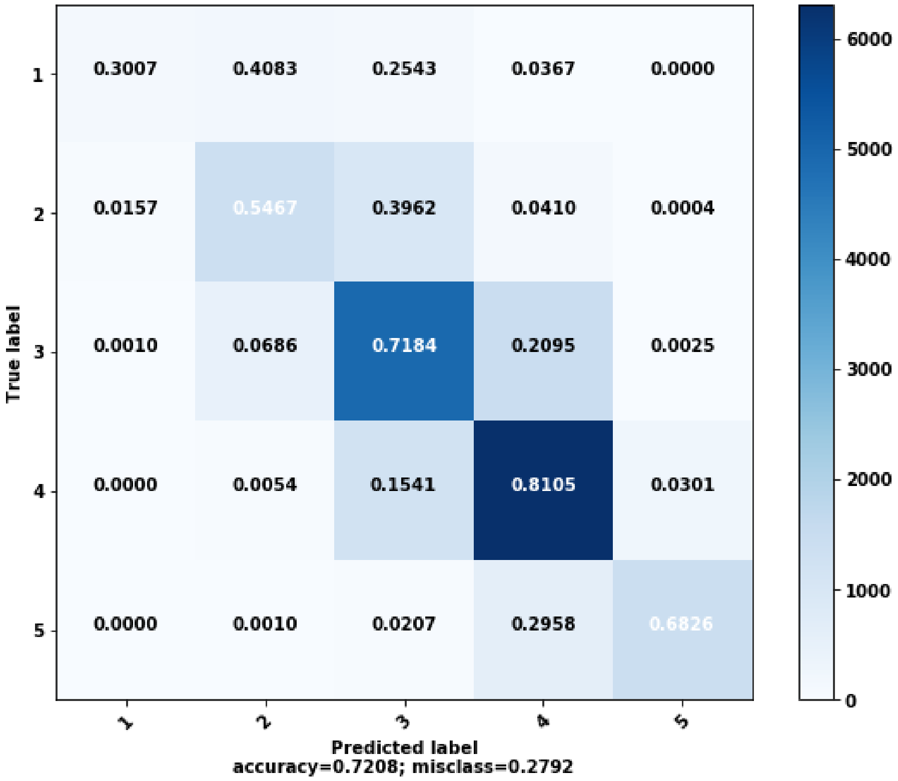


Figure 12: Classification report per predicted maintenance score for TF-IDF (1,2-grams) model.

words like ‘heating system’ are correlated to real estate with low maintenance scores.

Maintenance score	Uni-gram	Bi-gram
1	Verwarmingssysteem (heating system) Kluswoning (DIY home)	Ontspannen natuur (relaxed nature) Verwarmingssysteem gebaseerd (heating system based)
2	Huurwoning (rental home) Gratis (free)	Kopen zelfbewoning (buy self-habitation) Voormalige huurwoning (former rental house)
3	Materiaal (material) Hoogwaardig (high quality)	Nieuwe fundering (new foundation) Dubbele wastafel (double sinks)
4	Winkelcentrum (mall) Dakterras (roof terrace)	Amsterdam Poort* N/A Hoog plafond (high ceiling)
5	Quooker** N/A Miele** N/A	Balkon patio (balcony patio) Eigen badkamer (private bathroom)

Table 3: Uni- and Bi-grams highly correlated to maintenance scores (including English translations). * = place name, ** = product name.

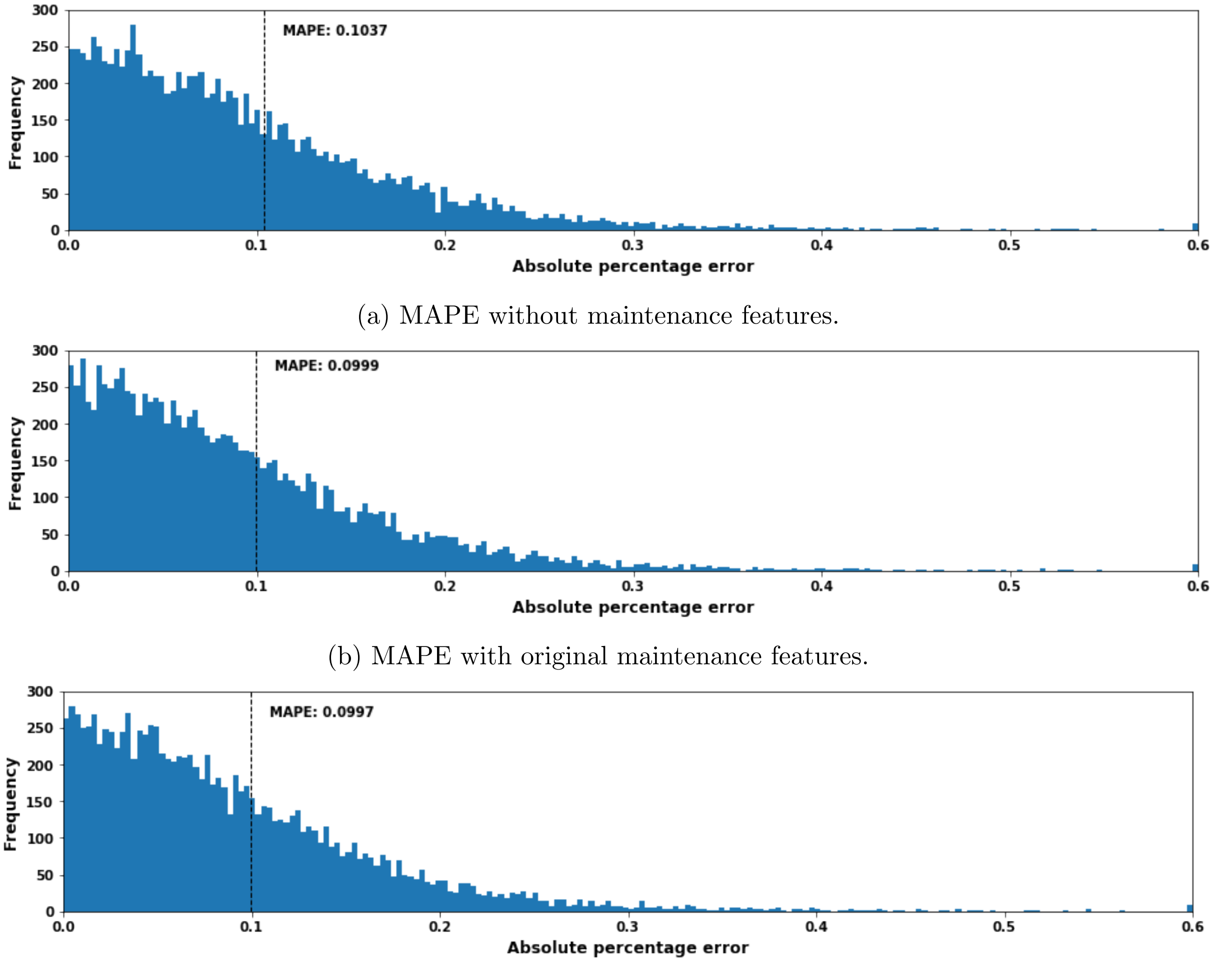
5.2 Benchmark Results

The current error of the real estate valuation model is estimated by the MAPE and is explained in Section 3.5. Note that the results in this section are based on the maintenance score predictions on the test set (out-of-the-sample). To illustrate the current contribution of maintenance features within the benchmark model, the MAPE without maintenance features is estimated at 10.37% and shown in Figure 13a. The predictions of this model are dependent on other individual housing characteristics and economic trends, as explained in Sections 1.1 and 3.5. Implementing the original maintenance features within the model results in an estimated MAPE of 9.99%. The distribution of the absolute percentage error of the model including the original maintenance scores is shown in Figure 13b. With respect to the model without maintenance features, the model incorporating maintenance feature reduces the MAPE by 0.38%. Furthermore, Figure 13c shows the distribution of the absolute percentage error of the model incorporating maintenance scores, predicted by using text mining methodologies. The MAPE of this model is estimated at 9.97%. With respect to the model without maintenance features, this model reduces the MAPE by 0.40%. This is an additional improvement of 5.26% in the MAPE compared to the model using the original maintenance scores.

We use the Wilcoxon matched-pairs signed-ranks test to measure whether there is a significant difference between results of the model using the original maintenance scores and the model using maintenance scores predicted by using TM methodologies. Further explanation about the statistical test can be found in Section 3.5 and Appendix A. The summarization of the test results is shown in Table 4. The z-value is estimated at 0.219, which means that the null hypothesis that the paired differences between the two samples follow a symmetric distribution around zero cannot be rejected. Hence, there is no significant difference between the results.

Test Statistic (W)	μ	σ_{adj}	z-value
$3.07 * 10^7$	$3.08 * 10^7$	$3.38 * 10^5$	0.219

Table 4: Results of the Wilcoxon matched-pairs signed-rank test.



(a) MAPE without maintenance features.

(b) MAPE with original maintenance features.

(c) MAPE with maintenance features predicted using TM.

Figure 13: MAPE and distribution of absolute percentage errors.

6 Conclusion

The Real Esate Appraisal Law requires real estate to be appraised yearly. The number of Dutch real estate appraisers is simply too small to value the more than 7.9 million residential properties. Therefore, automated valuation models are crucial for estimating real estate values. Accurate estimations of real estate by AVMs remain possible when incorporating individual housing characteristics, different economic trends and accurate maintenance conditions corresponding to the houses. The latter is currently estimated manually by several appraisers and other experts. Real Estate estimations may therefore contain subjective and inconsistent maintenance estimations, which can contribute negatively to the accuracy of current valuation models.

This study explores the potential of text mining methodologies within real estate valuation models by predicting maintenance conditions through real advertisement texts. Text mining is a variation on data mining, and used to discover patterns within real estate advertisement texts, which could describe the state of maintenance of particular real estate. Earlier research [1][29] already showed that simplistic TM methods proved its practical contribution to current real estate prediction models. In this study, approximately 65.000 samples including transaction prices and real estate advertisements, corresponding to houses in Amsterdam, were used to create and normalize document numeric feature vectors and extend current real estate research by using TM models that are more advanced. While utilizing different supervised machine learning classification algorithms, we found six classification models based on count-based text mining methodologies and six classification models by leveraging word embeddings suitable for predicting maintenance conditions though real estate advertisements. The results showed that more simplistic count-based models obtain higher accuracy. The TF-IDF model obtained the highest results (F1: .717), where approximately 28% of the predicted test results were different, with respect to the results of the appraisers and other experts. In addition, uni-grams containing luxury household products seemed to be highly correlated to advertisements with good maintenance conditions and uni-grams containing heating related features with bad maintenance conditions. In contrast to other research [29][30][4][34], the Logistic Regression classifier obtained higher performance than the Support Vector Machine classifiers.

We used the real estate valuation model of Ortec Finance as the benchmark model for estimating whether house price predictions using TM methodologies leads to significant improvements when compared to predictions using the original maintenance scores estimated by appraisers and other experts. Whereas including maintenance features in the company's real estate valuation model leads to a small reduction in the MAPE (0.38%), expanding the model by leveraging the TF-IDF model for predicting maintenance features increases this reduction by 5.26%. While utilizing the Wilcoxon matched-pairs signed-ranks test, this study concluded that there is no significant difference between the absolute percentage errors of the model including the TF-IDF methodology and the model leveraging the original maintenance scores.

Returning to the main research question, the automated classification model using TM methods scored high on F1-score when predicting the state of maintenance of Real Estate through real estate advertisements, but failed to improve the existing company's model significantly in terms of accuracy. Therefore, the results of this study show high potential of applying TM methods for the automatization of mass appraisal, since manually estimating maintenance conditions can be replaced by automated classification models using TM methods. In addition, as illustrated in table 3, it is possible to extract words and combinations of words highly correlated to specific maintenance conditions, which in turn shows potential in the explainability of such maintenance estimation. At last, the classification model is trained on real estate estimations of multiple appraisers and experts. Where one specific estimation may be labeled as biased or subjective, concatenating all opinions of the art of estimation contributes to the normalization of such practice and therefore shows potential for objectifying these estimations.

7 Discussion

In this chapter, the methods and results of this study followed by some limitations and suggestions for further research will be discussed. First, an explanation about the practical significance of this study is given as introduction to next section.

The classification model created in Section 3 and implemented in the benchmark model, will be adopted by Ortec Finance as an assisting score tool for estimating the state of maintenance for mass appraisal. This tool will help municipalities, appraisers and other experts for correcting real estate maintenance conditions in a more objective manner. Although the potential of such a text mining tool looks promising and its practical significance is luminous, a number of potential limitations and corresponding suggestions for future research can be listed.

7.1 Limitations

This study has several limitations. First, due to the limited time frame, all configuration settings of different text mining methodologies and machine learning classification algorithms were reduced to a limited set. Better and more extensive estimations remain possible when considering a broader set. For the text mining methodologies using n-grams features, combinations of words were limited to a maximum of tri-grams, which in turn reduces the potential of extracting meaningful sequential word collections of real estate advertisements. Our results show that tri-grams were superfluous for obtaining best results, and thus an increase in n of n-grams does not persistently increase the accuracy of prediction models. However, we cannot deny nor confirm that different combinations of n-grams would be more favorable in terms of accuracy. Furthermore, text mining methodologies leveraging word embeddings were limited in their model architectures. Due to the computation time for creating embeddings, this study excluded CBOW and DM. Hence, no concluding remarks of best performing architectures were presented by this research. For the machine learning part, configuration settings were only adjusted in terms of regularization. In addition, the SVM classifier only used the hinge loss function to optimize and build the model. Different kernel functions to convert the existing feature space into an even higher dimensional feature space, where data could be separated linearly, were excluded. Therefore, we cannot ascertain to have found the optimal combination of configuration settings for this particular dataset, with the purpose of obtaining the highest possible prediction accuracy of the classification model.

Secondly, there are limitations in terms of data interpretation, which need to be addressed. We trained a model based on a dataset of approximately 65.000 housing samples, containing advertisements texts and maintenance scores. The latter was estimated by appraisers and other experts, and thus may contain subjective data. Hence, it is uncertain whether the predicted maintenance scores are true projections of reality. If not, this may harm the internal validity of such a classification model, because it is trained by and therefore reliant on the maintenance scores estimated by appraisers and other experts. Nevertheless, we tried to improve current estimation practices instead of creating new definitions of maintenance conditions. Next to subjectivity, the relation between maintenance scores and the year of estimation may be questioned as well. Although Figure 9b shows a constant mean of maintenance scores throughout the years, the interpretation of maintenance could still differ between varying economic circumstances. Therefore, the results of this study represent the situation within the time span of the dataset and remains reluctant to circumstances beyond this scope.

At last, limitations occur in the interpretation of the results as well. Although this study concludes that there is no significant difference between the absolute percentage errors of the models including and excluding TM methodologies, this result is purely based on using the company's valuation model as a benchmark model and could differ when using alternative models. As illustrated in Figures 13a and 13b, incorporating maintenance features increases the model's prediction performance by a marginal amount, which depicts that the maintenance feature obtains relatively small beta in the prediction model. Increasing the beta causes increases in the degree of change in the outcome variable of the model and therefore changes the outcome of this study. In addition, we estimated the accuracy of the company's real estate valuation model using the MAPE. As hidden in the abbreviation, absolute percentages are taken and therefore we were not able to distinguish which real estate was responsible for overestimations or underestimations. This study would also like to address that all samples within the dataset correspond to houses allocated in Amsterdam. For reproducing similar results for different municipalities, the training part of this study has to be repeated on samples corresponding to houses allocated in that specific area.

7.2 Future research

Based on the limitations mentioned above, and the findings of this study, there still exists some untouched potential within the prediction of maintenance conditions through advertisement texts. First, beyond the extension of the limited set of configuration settings of the employed ML and TM methodologies, newer deep learning (DL) methodologies like neural networks (NN) classifiers could be used to further broaden the potential of TM methodologies for predicting maintenance conditions. DL has been delivering and is continuing to deliver continual success in text mining areas and it has enabled us to reach human-level accuracy in the last couple of years [28]. NN classifying models can leverage some form of transfer learning such that a pre-trained model, trained on a huge corpora of rich textual data, can be used to generalize representations on new text data, especially in classification problems with lack of data. This may be beneficial when predicting maintenance of real estate corresponding to municipalities with a low amount of training samples. Furthermore, it is possible to extend the basic single linear separator, which is similar to the separator of this study's implemented SVM classifier, with multiple layers of neurons in order to induce non-linear classification boundaries [2]. As shown in Table 2, we found difficulties in classifying maintenance conditions through averaged word embeddings as document feature vectors. Appendix B illustrates that the use of linear classification boundaries may be too soft, since embeddings highly correlated to different maintenance conditions are scrambled all over the place. Future research could investigate whether the use of non-linear classification boundaries using advanced NN classifiers results in better performances.

Secondly, as comprehensively described by Francke [14] and addressed in Section 1.1, the HTM model assumes the coefficients of maintenance features remain constant over time. This can be questioned, since maintenance deteriorates over time. Furthermore, Francke [16] concluded that on average older structures are assessed to be maintained less than newer structures, which depicts that there might be a relationship between the construction year of a house and the level of deterioration. Future research could investigate this further and implement such a time-dependent maintenance feature within the company's real estate valuation model, which in turn corrects maintenance scores over time and increases the model's robustness.

At last, we focused on determining the potential of text mining methodologies for classifying maintenance conditions. As discussed in Section 1.1, current estimations are not solely based on advertisement texts but on real estate pictures as well. Therefore, text-mining methodologies alone may not completely imitate the act of the appraiser. Quenxeng et al. [37] and Poursaeed et al. [27] already attempted to analyze real estate properties by image-based techniques. Incorporating these techniques in this study's classification model could be beneficial, since the act of the appraiser will be fully imitated. This study therefore recommends future research to explore the possibilities of combining different data science techniques for the estimation of maintenance.

References

1 Abdallah, S., & Khashan, D. A. (2017). Using text mining to analyze real estate classifieds. *Advances in Intelligent Systems and Computing*, *533*, 193–202.

2 Aggarwal, C., & Zhai, C. X. (2013). *Mining text data* (1st ed.). New York, USA: Springer.

3 Allecijfers.nl. (2019). Informatie over Nederland 2019 [Retrieved from: https://allecijfers.nl /nederland/].

4 Amrit, C., Paauw, T., Aly, R., & Lavric, M. (2017). Identifying child abuse through text mining and machine learning. *Expert Systems with Applications*, *88*, 402–418.

5 Balcan, M. F. (2019). Introduction to Machine Learning. Carnegie Mellon University: School of Computer Science [Retrieved from: https://www.cs.cmu.edu/ 10315/recitation/ rec3.pdf].

6 Bhavani Dasari, D., & Gopala Rao, V. K. (2012). Text Categorization and Machine Learning Methods: Current State of the Art. *Global Journal of Computer Science and Technology Software*, *12*, 36–46.

7 Bojanowski, T., Grave, E., Joulin, G., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

8 CBS. (2020). Gemeenten begroten 10,8 miljard euro aan heffingen in 2020 [Retrieved from: https://www.cbs.nl/nl-nl/nieuws/2020/05/gemeenten-begroten-10-8-miljard-euro-aan-heffingen-in-2020].

9 Ceyhan, C. (2017). *Evaluation of Machine Learning Techniques for House Valuations* (Master's thesis). Erasmus University Rotterdam.

10 DeNederlandscheBank. (2019). DNBulletin: Woningtaxaties overgewaardeerd [Retrieved from: https://www.dnb.nl/nieuws/nieuwsoverzicht-en-archief/DNBulletin2019/dnb382679 .jsp].

11 Edda, L., & Jörg, K. (2002). Text Categorization with Support Vector Machines. How to Represent Texts in Input Space? *Machine Learning*, *46*, 423–444.

12 Eerenbeemt, M. v. d. (2019). DNB: 'Taxateurs waarderen woningen systematisch te hoog' — De Volkskrant [Retrieved from: https://www.volkskrant.nl/nieuws-achtergrond/dnb-taxateurs-waarderen-woningen-systematisch-te-hoog bce6d24c/].

13 Facebook. (2016). FastText [Retrieved from: research.fb.com/blog/2016/08/fasttext/].

14 Francke, M. K. (2008). The hierarchical trend model. *Mass Appraisal Methods. An International Perspective for Property Values*, 164–180.

15 Francke, M. K. (2010). A State-Space Model for Residential Real Estate Valuation. *AENORM*, *18*, 4–7.

16 Francke, M. K., & Minne, A. M. v. d. (2017). Land, Structure and Depreciation. *Real Estate Economics*, *45*, 415–451.

17 Google. (2013). Word2Vec: Tool for computing continuous distributed representations of words [Retrieved from: https://code.google.com/archive/p/word2vec/].

18 Gupta, V., & Lehal, G. S. (2014). A Survey of Text Mining Techniques and Applications. *Journal of Emerging Technologies in Web Intelligence*, *1*, 60–76.

19 Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., & Yeganegi, M. R. (2020). Text Mining in Big Data Analytics. *Big Data and Cognitive Computing*, *4*, 1–34.

20 Khan, A., Baharudin, B., Lee, L. H., & Khan, K. (2011). A Review of Machine Learning Algorithms for Text-Documents Classification. *Open Computer Science*, *1*, 4–20.

21 Kok, N., Koponen, E., & Martinez-Barbosa, C. A. (2017). Big Data in Real Estate? *The Journal of Portfolio Management*, *43*, 202–211.

22 Le, Q., & Mikolov, T. (2014). Doc2Vec. *Proceedings of the 24th International Conference on World Wide Web*, *32*, 29–30.

23 Mikolov, T., Chen, K. C., & Dean, J. G. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1–12.

24 Mikolov, T., Chen, K., Corrado, G., Dean, J., & Sutskever, I. (2013). *Distributed Representations of Words and Phrases and their Compositionality* (tech. rep.). Google Inc.

25 Philander, K., & Zhong, Y. Y. (2016). Twitter sentiment analysis: Capturing sentiment from integrated resort tweets. *International Journal of Hospitality Management*, *55*, 16–24.

26 Porter, M. F. (2001). Snowball : A language for stemming algorithms [Retrieved from: http://snowball.tartarus.org/texts/introduction.html].

27 Poursaeed, O., Matera, T., & Belongie, S. (2018). Vision-based real estate price estimation. *Machine Vision and Applications*, *29*(4), 667–676.

28 Sarkar, D. (2019). *Text Analytics with Python* (2nd ed.). New York, USA: Springer.

29 Stevens, D. (2014). *Predicting Real Estate Price Using Text Mining Automated Real Estate Description Analysis* (Master's thesis). Tilburg University School of Humanities.

30 Surjandari, I., Naffisah, M. S., & Prawiradinata, M. I. (2014). Text Mining of Twitter Data for Public Sentiment Analysis of Staple Foods Price Changes. *Journal of Industrial and Intelligent Information*, *3*(3), 253–257.

31 Taffese, W. Z. (2007). Case-based reasoning and neural networks for real estate valuation. *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, AIA 2007*, 84–89.

32 UpGrad. (2019). What is Text Mining: Techniques and Applications — upGrad blog [Retrieved from: https://www.upgrad.com/blog/what-is-text-mining-techniques-and-applications/].

33 Waarderingskamer. (2020). Alles over de WOZ [Retrieved from: https://www.waarderingskamer.nl/alles-over-de-woz/].

34 Wang, W., Zhu, K., Wang, H., & Wu, Y. C. J. (2017). The impact of sentiment orientations on successful crowdfunding campaigns through text analytics. *IET Software*, *11*, 229–238.

35 Welie, T. (2020). About us — Ortec Finance [Retrieved from: https://www.ortecfinance.com/en/about-us].

36 Whitley, E., & Ball, J. (2002). Statistics review 6 : Nonparametric methods. *Critical Care*, *6*, 509–513.

37 You, Q., Pang, R., Cao, L., & Luo, J. (2017). Image-Based Appraisal of Real Estate Properties. *IEEE Transactions on Multimedia*, *19*, 2751–2759.

A : Wilcoxon Matched-pairs Signed-ranks Test

$e_{i,1}$	$e_{i,2}$	use?	abs(d_i)	sign(d_i)	rank*sign	rank ties freq (ft)	ties corr
12.38	12.46	yes	0.08	-1	-674	9	720
10.8	10.18	yes	0.62	1	4764	21	9240
4.03	4.48	yes	0.45	-1	-3626	27	19656
1.62	5.16	yes	3.54	-1	-8850.5	4	60
2.41	8.01	yes	5.6	-1	-9877	3	24
2.39	1.78	yes	0.61	1	4686	9	720
4.78	3.49	yes	1.29	1	6980.5	6	210
4.78	3.49	yes	1.29	1	6980.5	6	210
0.79	0.47	yes	0.32	1	2640	5	120

Table 5: Small portion of this study's Wilcoxon matched-Pairs signed-ranks test results

- Step 1: Remove pairs with equal errors and determine the absolute value of the difference between the two models for each case. Determine sign of the difference and rank the absolute difference from low to high (using average rank for ties).

- Step 2: Multiply the sign with the rank and determine the number, absolute sum and the average for both the negative and positive pairs. Determine the frequency of tied ranks and for each tied rank cube the frequency and subtract it once. Choose as test statistic (W) the minimum of the two absolute sums.

- Step 3: Determine the hypothesized sum of ranks in the population and the variance.

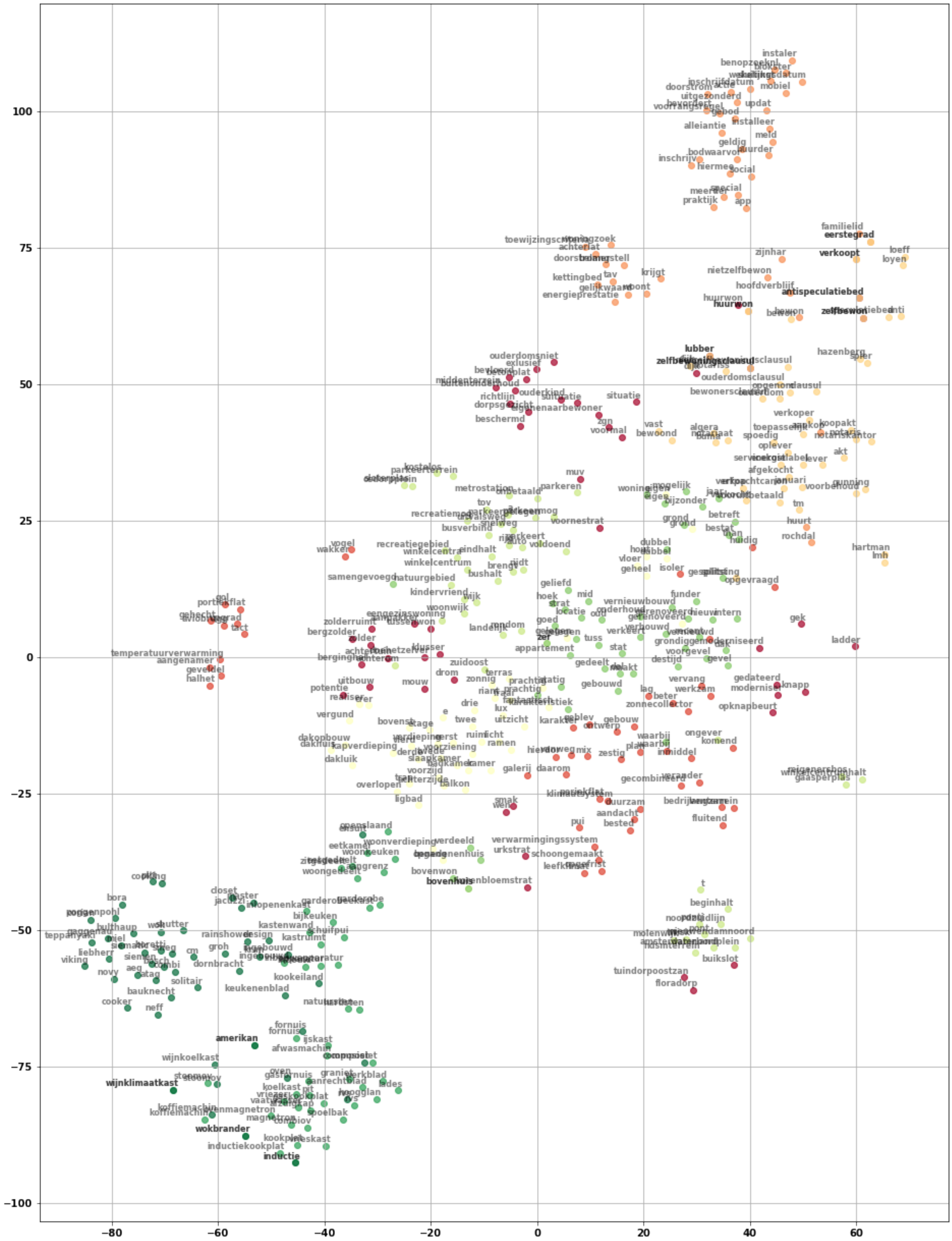
$$\mu = \frac{N(N+1)}{4}, \qquad \sigma^2 = \frac{N(N+1)(2N+1)}{24}$$

- Step 4: Determine the correction for tied rank, the adjusted standard error for tied ranks and the z-value.

$$T = \sum_{j=1}^{u} t_j^3 - t_j, \qquad \sigma_{adj} = \sqrt{\sigma^2 - \frac{T}{48}}, \qquad z = \frac{W - \mu}{\sigma_{adj}}$$

- Step 4: Determine the significance using the normal distribution and the proposed significance level.

B : 2-D Visualisation of Word2Vec Embeddings



Word2Vec word embeddings are plotted which are highly correlated to specific maintenance conditions. ● = good maintenance, ● = neutral maintenance ● = bad maintenance.