Erasmus University Rotterdam

Erasmus School of Economics

Complaints validity detection through natural language processing and machine learning

Author:
Daolue Lin
External supervisor:
Heny Selmi

Supervisor:
Prof. dr. ir. R. Dekker
Second assesor:
Dr. N.M. Almeida Camacho

December 9, 2022

Abstract

The aim of this research is to explore the feasibility of constructing models based on Natural Language Processing (NLP) and Machine Learning (ML) to classify a complaint's validity into Valid, Invalid, or Partially valid such that they can be prioritized or deprioritized in order to support the complaint handling process for CED Group. Three NLP techniques have been used to extract features from textual complaints, which are Sentiment Analysis, Latent Dirichlet Allocation, and Global Vectors for Word Representation. Subsequently, these features are used as input into four Machine Learning (ML) techniques, which are Multinomial Logistic Regression, Random Forest, XGBoost, and Support Vector Machine to classify the validity of complaints. The results have shown that the models are able to classify valid complaints effectively, while classification of invalid and partially valid complaints remain more challenging. The best performing model is the Support Vector Machine with the highest weighted average F1-score and accuracy followed by XGBoost, Random Forest and Multinomial Logistic Regression.

Contents

1 Introduction **2**

2 Literature **3**
- 2.1 Natural Language Processing and Machine learning in Customer Complaint Management . . . 3
- 2.2 Complaint validity detection . . . 4

3 Data **5**
- 3.1 Background information . . . 5
- 3.2 Data description . . . 5
- 3.3 Examples of complaints text . . . 6
- 3.4 Data analytics . . . 8
- 3.5 Data pre-processing . . . 9

4 Methodology **11**
- 4.1 Feature extraction . . . 12
  - 4.1.1 Sentiment Analysis . . . 12
  - 4.1.2 Latent Dirichlet Allocation (LDA) . . . 13
  - 4.1.3 Global Vectors for Word Representation (GloVe) . . . 14
- 4.2 Prediction models . . . 17
  - 4.2.1 Multinomial Logistic Regression (MNL) . . . 17
  - 4.2.2 Random Forest (RF) . . . 17
  - 4.2.3 eXtreme Gradient Boosting (XGBoost) . . . 18
  - 4.2.4 Support Vector Machine (SVM) . . . 20
- 4.3 Evaluation metrics . . . 20

5 Results **21**
- 5.1 Extracted features . . . 21
  - 5.1.1 Sentiment Analysis . . . 21
  - 5.1.2 LDA . . . 24
  - 5.1.3 GloVe . . . 27
- 5.2 Prediction models . . . 29
  - 5.2.1 Multinomial Logistic Regression . . . 29
  - 5.2.2 Random Forest . . . 30
  - 5.2.3 XGBoost . . . 32
  - 5.2.4 SVM . . . 34
  - 5.2.5 Overall results . . . 35
  - 5.2.6 Only two classes . . . 37
- 5.3 Deeper dive into the GloVe dimensions . . . 39

6 Conclusion and Implications **41**
- 6.1 Marketing and Managerial implications . . . 41
- 6.2 Conclusion . . . 42

7 References **43**

8 Appendix **46**

1 Introduction

Customer complaint management is an important process in Customer Relationship Management (CRM) for many companies and organizations. Customer complaint management can be viewed as the process of documenting and addressing customer complaints. This typically begins with the customer reporting an issue and providing a narrative of their efforts to resolve it (Galitsky et al., 2009). Addressing customer complaints is crucial for customer satisfaction, failing to solve or address complaints may result in e.g., customer churn and negative word of mouth.

Nowadays, companies are facing an immense number of complaints. This is due in part to the high expectations of customers', who have become more demanding in their expectations of high-quality products and services. As a results, companies are struggling to address these complaints effectively (Faed et al., 2016). Invalid complaints aggravate this process even more as it is difficult to distinguish a valid from an invalid complaint and the handler must determine its legitimacy based on the customer's narrative.

For CED Group[1], a challenge their customer service department has is effectively handling customer complaints due to large numbers and complexity of the complaints in the claims management business. A model to distinguish acceptable customer complaints from those which are not is desired to support their complaint handling process. In this way, their available resources can be allocated more towards handling valid complaints instead of invalid ones to increase effectivity and customer satisfaction. At the moment, no model is deployed to detect a complaint's validity. Complaints are received through various channels and summarized by a complaints handler. As complaints are in textual form and its validity is manually labelled into YES, NO, or Partial after the closure of a case, the feasibility of natural language processing (NLP) techniques combined with machine learning (ML) techniques should therefore be explored to analyze and predict a complaint's validity, i.e. a text classification problem.

The use of NLP and ML in supporting customer complaint management looks promising for many industries and research in this field is quite recent. However, not much research has been done in this field with regards to the insurance industry nor classifying a complaint's validity, and how different NLP techniques in combination with ML techniques perform. Such as extracting features from different NLP techniques as sentiment analysis Medhat et al. (2014), Latent Dirichlet Allocation (LDA) (Blei et al., 2003), Word2vec (Mikolov et al., 2013), and Global Vectors for Word Representation (GloVe) (Pennington et al., 2014). And afterwards feeding them into different ML methods.

This research aims to explore the feasibility of constructing models based on NLP and ML to classify a complaint's validity for CED Group. The following questions are addressed:

- How can a complaint's validity be modeled using various machine learning techniques and textual description of a complaint?

[1]CED Group is an insurance services and solutions platform for insurance claims management. https://www.cedgroup.eu/

- Which Natural Language Processing techniques can be used to process complaints text?
- Which machine learning model performs best in predicting a complaint's validity based on features extracted from Natural Language Processing?

This research will provide practical insights into how NLP and ML techniques can support in complaint management. Consequently, adding to the paradigm on how artificial intelligence (AI) will have noticeable impact on the way firms manage customer relationships, and CRM will transform into AI-CRM (Libai et al., 2020). In addition, by using real data, an indication of how models perform in real world scenarios can be given.

The remainder of the thesis is structured as follows. Section 2 gives an overview of existing and related literature. Section 3 discusses the data used from CED, and more information on the complaints handling process by CED. Section 4 discusses the methodology, where a description of NLP and ML algorithms is given, and also the evaluation metrics. In section 5 the models' results are discussed and compared. Finally, Section 6 concludes this research.

2 Literature

To lay the foundations for answering the research questions, a literature review is provided in this section in which the following is discussed: natural language processing, machine learning, customer complaint management, and complaint validity detection.

2.1 Natural Language Processing and Machine learning in Customer Complaint Management

As mentioned before, the use of NLP and ML in supporting customer complaint management is being explored by many industries, research in this field is quite recent and results showed that the models built seem promising for real-world application. Such as the use of NLP in processing customer complaints from water utilities (Tian et al., 2022). In addition, Liu et al. (2019) used Word2vec, Term Frequency - Inverse Document Frequency (TF-IDF), and GloVe as features to identify expectations to be fulfilled from complaints, expectation categories were selected by customers when filling a complaint. Moreover, Krishna et al. (2019) classified moderate and extreme complaints on social media posts to support CRM using TF-IDF , Word2vec, and linguistic features, the complaints were first manually classified into moderate or extreme. Furthermore, Yang et al. (2019) describes a model for detecting customer complaint escalation from text chat dialogues using recurrent neural networks for e-commerce companies, and HaCohen-Kerner et al. (2019) examined classifying complaint letters into categories according to given company categories using unigrams as features. Even though all of these studies use different approaches and models for processing complaints, their similarities lie in the fact that they successfully use extracted NLP features from complaints text for their end-goal. The NLP techniques used in this research are therefore based on the previously mentioned studies. Newer NLP techniques do exist, such as BERT (Devlin et

al., 2018) and ELMo (Peters et al., 1802), which are pre-trained on vast amounts of text. However, these models are argued to be not robust enough when used in a domain-specific setting (Kowsari et al., 2019).

2.2 Complaint validity detection

Literature specific on complaint validity detection is scarce, e.g., Galitsky et al. (2009) proposed a model for assessing the validity of a customer complaint based on the dialogue between the company representative and the customer. Complaints scenarios were labelled as directed graphs and similarity matching among graphs was applied to classify the complaint scenarios into valid or invalid. The results showed that, in their sample, 70% of invalid claims could be detected, and the accuracy was deemed satisfactory in this setting. Their model can not be used in our case as we do not make use of data in which a dialogue is present. In addition, the authors argued that analysis of complaints in textual form is difficult because of reasons, such as: (1) the structural and logical complexity of a complaint; (2) containing inconsistencies that are interconnected; (3) having a biased information representation; (4) descriptions with implicit and explicit goals; (5) containing a wide variety of domain-specific or technical terms; (6) having a poorly organized and emotional structure; and (7) containing many ambiguities and unclear references, and as a result, extracting insights from textual complaints has not gained much popularity from the research community. With advancements in NLP and ML techniques nowadays, the matter of invalid complaint detection should be re-examined, and whether modern techniques can be successfully applied in the real-world.

A related field with more available literature is detection of fake reviews (Crawford et al., 2015; Jiang et al., 2016; Mukherjee et al., 2013) and fake news detection (Figueira and Oliveira, 2017; Zhou et al., 2019). For example, fake reviews detection in the consumer electronics field using sentiment analysis and ML techniques (Barbado et al., 2019). Elmurngi and Gherbi (2017) studied fake reviews detection in online movie reviews using sentiment analysis and various ML techniques, and Jia et al. (2018) studied the fake reviews detection using word term frequencies, LDA, and Word2vec features. Though complaints differ from reviews, as reviews do no necessary require a response and can also be positive, their textual format is similar in the sense that they contain the same difficulties in analyzing them as mentioned by Galitsky et al. (2009). In addition, the final objective in fake reviews detection is to distinguish true from untrue text, which is also the case for complaint validity detection. Moreover, the model architectures used in fake reviews detection are comparable to those used for customer complaint management and should therefore also be applicable in our case. However, in all mentioned studies, the text used are entered by the complainant or reviewer self. In CED Group's case, the complaint can either be the original text of the complainant self or text entered by a handler in which the complaint is summarized and sometimes annotated. This implies that some of the difficulties in analyzing textual complaints as mentioned by Galitsky et al. (2009) might be eased, such as (1), (2), (4), (6), and (7) as complaints are summarized and annotated. For (3) this depends on how the handler interprets

the complaints. In short, validity detection of complaints is not a well-researched area, and fake review/news detection is a related field because of the same final objective. Also, the NLP and ML methods used in fake review/news detection are comparable to those used in customer complaint management. Therefore, the approach is this research will be based on the literature discussed in these two fields.

3 Data

This section will will discuss the data used in this research. Section 3.1 gives background information on the documentation process of complaints. Section 3.2 describes the dataset. Section 3.3 gives examples of the complaints in textual format. Section 3.4, focuses on data analytics of the dataset. Lastly, section 3.5 discusses how the data is processed for analysis.

3.1 Background information

In this research, complaints data from CED Group is used. Complaints are received through various channels, such as mail, phone, and online forms. After a complaint has been received, a case is opened in Microsoft Dynamics and the complaint is summarized in a description box and an interpretation by the handler can also be given, this is the also the text used in this is research. Afterwards, the complaint might be handled directly or handled at a later moment, which could be done by other employees. A first response must be given within 10 days as this is in their Service-level agreement with their corporate clients, this however does not mean that the case must be handled within 10 days, if a case needs more time, the complainant will be notified. The model aims to provide an indication on validity after the complaint is summarized such that human resources can be allocated more towards handling valid complaints to increase effectivity and customer satisfaction. After the closure of a case, the validity of the complaint is indicated by a handler which is based on whether the problem was CED Group's fault.

3.2 Data description

The dataset contains 2222 observations with a textual description of the complaint in dutch and validity of the complaint from 2015 until 2022. In total, 1189 complaints are labelled as valid, 628 as invalid, and 405 as partially valid. Other features in the dataset include:

- The business unit of CED Group: e.g., CED Advice, CED Repair, claims management, SOS International.
- Category of the complaint: e.g., behavior, accessibility, payment, communication, lead time, policy, quality of report, other.
- Specific sector of the business unit: e.g., Mobility, Property, Medical
- Which corporate client it relates to: e.g, ABN AMRO, Achmea, Aegon, Allianz, ASR, DSW, Nationale Nederlanden, OHRA, ONVZ.

- From whom the complaint comes: corporate client, principal, intermediary, policyholder, other.
- Who/which department received the complaint.
- Status of the complaint: active, inactive.
- Status reason: in progress, re-opened, cancelled, closed.
- Name of employee handling the complaint.
- Date received complaint.
- First date of reaction.
- Close date of complaint.
- Duration between received and first reaction in days.
- duration between received and closed date in days

3.3 Examples of complaints text

As all complaints are provided in Dutch, an English translation is added to make the text more comprehensible to the reader. After investigation by a handler into the complaint, the complaints are labelled valid, invalid, and partially valid.
Examples of textual description of complaints labelled as **valid**:

1. Original complaint in Dutch:

   Vanmorgen heb ik contact gehad met Dhr X. Hij heeft op DATUM melding gemaakt dat hij zijn sleutels kwijt is geraakt en dat de auto afgesleept moet worden naar de garage. Dit is gebeurd door de berger Y. Nu heeft hij de auto weer opgehaald en ziet dat hij schade heeft aan zijn voertuig. Alle 4 de wieldoppen zijn hierbij beschadigd en het linker voorwiel is zijn velg ook beschadigd. Hij wil graag dat dit opgelost gaat worden.

   Translated complaint to English:

   This morning I was in contact with Mr. X. He reported on DATE that he had lost his keys and that the car has to be towed to the garage. This was done by the recovery company Y. Now he has picked up the car again and sees that he has damage to his vehicle. All 4 wheel covers are damaged and the left front wheel's rim is also damaged. He wants this to be resolved.

2. Original complaint in Dutch:

   Expert verzocht de zaak met spoed op te pakken. De problematiek zat in de wetgeving op het gebied van dakbedekking waarvoor aanvullende vragen uitstonden naar BEDRIJF die was ingeschakeld voor nadere uiteenzetting. Nadat dit was ontvangen is een voorstel gedaan aan verzekerde en schadecijfers overlegt. Als hierover goed was gecommuniceerd was iedereen op de hoogte geweest.

Translated complaint to English:

Experts requested that the matter be dealt with urgently. The problem was in the legislation in the field of roofing, for which additional questions were raised to COMPANY, which was called in for further explanation. After this was received, a proposal was made to the insured and damage figures were submitted. If there had been good communication about this, everyone would have been informed.

By reading the first example, it becomes clear that the complaint has been summarized by a handler. In the second example, it can be seen that an interpretation of the complaint has been given by the handler.
Examples of textual description of complaints labelled as **partially valid**:

3. Original complaint in Dutch:

   Klagers kunnen zich niet vinden in de taxatiewaarden en vonden expert uit de hoogte. Na contact te hebben opgenomen zijn de meningen nog niet bijgesteld. Expert zal een kop koffie drinken met de rep om de zaak nog eens door te nemen.

   Translated complaint to English:

   Complainants do not agree with the appraisal values and found the expert supercilious. Opinions have not yet been adjusted after being contacted. Expert will have a cup of coffee with the rep to go over the matter again.

4. Original complaint in Dutch:

   Er is wel degelijk een tussenbericht verzonden. Echter wel te laat. De behandelaar heeft deze blijkbaar niet gezien. Nu alsnog de huidige stand van zaken doorgegeven.

   Translated complaint to English:

   An intermediate message has indeed been sent. However, too late. Apparently the practitioner has not seen this. Now passed on the current state of affairs.

Partially valid complaints imply that the problem of the complainant is not entirely CED Group's fault, in such cases CED Group still does its best to resolve the issue.
Examples of textual description of complaints labelled as **invalid**:

5. Original complaint in Dutch:

   Via Mandaat komt het verzoek van een belverslag. Patient claimt dat SOS niet volledig is geweest in haar infovoorziening. Aan de hand van de gespreksverslag blijkt dat eega van de patient volledig is geïnformeerd over de kosten e.d.. Het overgaan tot OK is een door hun zelf gemaakte keuze.

Translated complaint to English:

The request for a call report comes via Mandaat. Patient claims that SOS has not been complete in its information provision. Based on the interview report, it appears that the patient's spouse has been fully informed about the costs, etc. Going to OK is a choice they have made themselves.

6. Original complaint in Dutch:

Zaken zijn zéér uitgebreid besproken met alle betrokkenen en alle juristen van de partijen. Over de zaak is ook meer dan één rechtszaak geweest. Klacht als zodanig ongegrond.

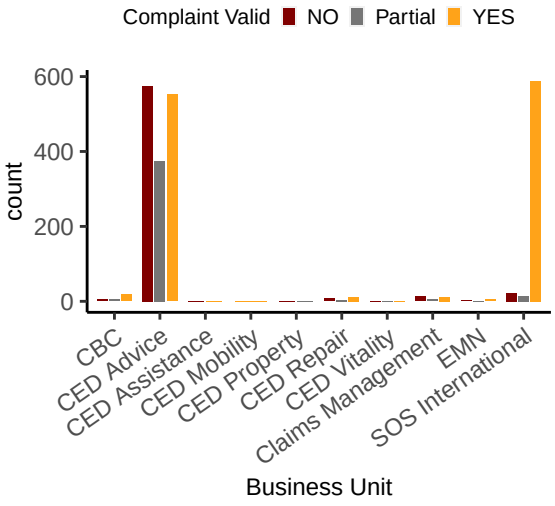
Translated complaint to English:

Matters were discussed in great detail with all those involved and all legal experts of the parties. The case has also been the subject of more than one lawsuit. Complaint as such unfounded.

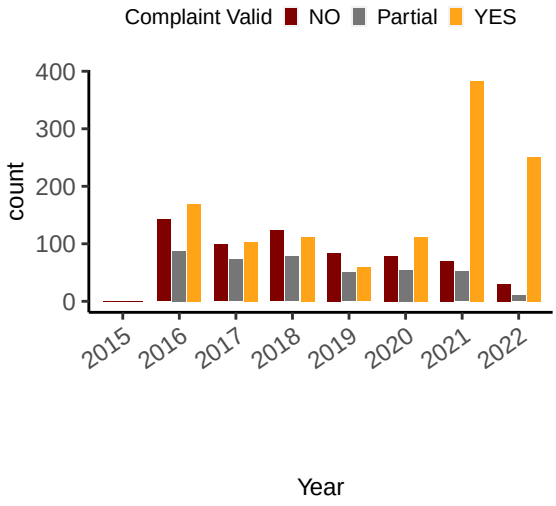
The given example texts give a good representation of Galitsky et al. (2009)'s argument as they cover all reasons of why analysis of complaints in textual form is difficult.

3.4 Data analytics

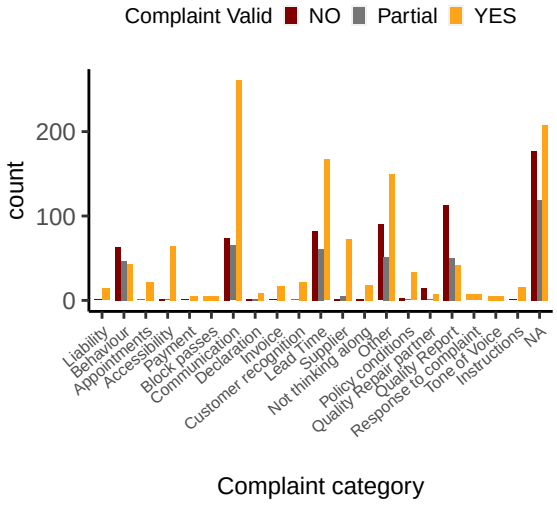
Figure 1 shows bar graphs of the number of complaints by validity across (a) the different business units, (b) years, (c) complaint categories, and (d) primary complainants. The complaints can be mainly divided between two business units of CED, namely CED Advice and SOS International. With CED Advice having 1501 complaints (552 Valid, 374 Partial, 575 Invalid) and SOS International having 624 complaints (588 Valid, 14 Partial, 22 Invalid), the remaining 97 complaints are spread among 8 other business units. For SOS International it appears that almost 95% of the received complaints are valid. In addition, looking at complaints across years, it can be noticed that the number of complaints has risen greatly since 2021. Complaints were received between December 2015 and May 2022, so the year 2015 shows a small number of complaints. Moreover, the categories with the most complaints are Behavior, Communication, Lead Time, Quality Report, Other, and NA. Many complaints have not been assigned or assigned to Other as the category of the complaint was not in the system. Furthermore, most complaints originate from the corporate clients as CED Group acts in name of their corporate client, and complaints are received by them first before being forwarded to CED Group.



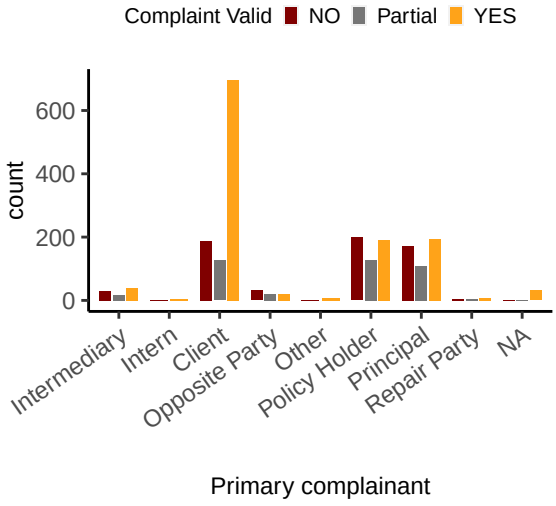
(a) Distribution across business units



(b) Distribution across years



(c) Distribution across categories



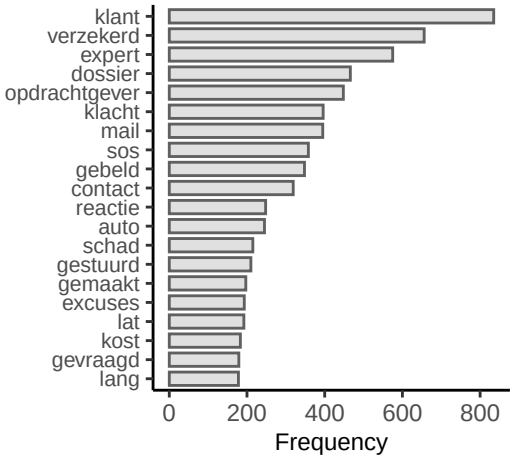
(d) Distribution across primary complainants

Figure 1: Basic statistics of complaints clustered by validity at CED Group, including the number of complaints across (a) the different business units, (b) years, (c) the complaint categories, and (d) the primary complainants.

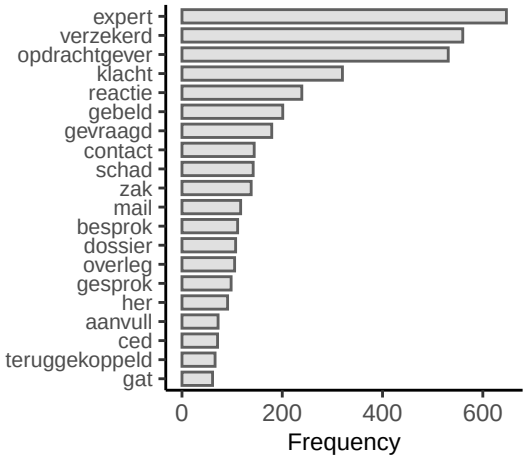
3.5 Data pre-processing

Textual data needs to be pre-processed before NLP techniques can be used. In this step, cleaning, tokenization, stop word removal, and stemming is applied to the textual data depending on the NLP algorithm, this is done following the text cleaning protocol (Berger et al., 2020; Kwartler, 2017) except for spell checking as this might change many domain specific words. First, cleaning includes removal of excess spaces, punctuations, numbers, special characters, and converting to lower cases. Second, tokenization breaks sentences down into word level such that the the frequency

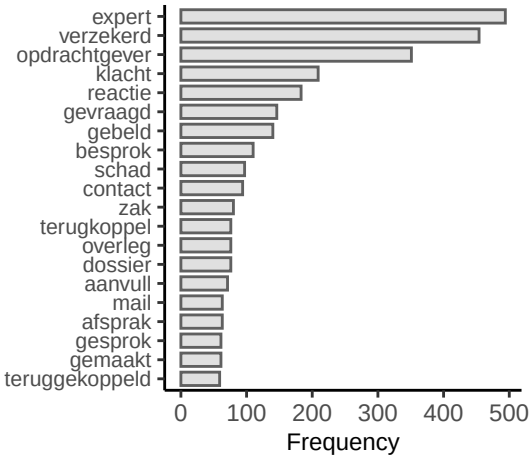
of word occurrences can be captured. Third, stop words are removed based on a list of dutch stopwords[2]. Lastly, words are stemmed using the Dutch stemming algorithm from the Snowball stemmer (*SnowballC* package) (Porter, 2001). Figure 2 shows the most frequent words by (a) Valid, (2) Invalid, and (3) Partially valid complaints after pre-processing. This is done by splitting the complaints text into tokens and counting them using *tidytext*.



(a) Valid complaints



(b) Invalid complaints



(c) Partially valid complaints

Figure 2: Top 20 most frequent words by (a) valid, (b) invalid, and (c) partially valid complaints

It can be seen that 'klant' (client) is the most frequent appearing word in valid complaints while it does not appear in the most frequent invalid nor partially valid complaints, the same goes for 'sos' (sos), 'auto' (car), excuses' (apologies), 'gestuurd' (sent), Also, the words 'dossier' (file), 'mail' (mail), appear more frequently in valid complaints.

[2]A list of the most comprehensive collection of stopwords for the dutch language is used. https://github.com/stopwords-iso/stopwords-nl

4 Methodology

In this section, the approach and methodology is described. The methods used are inspired by NLP and ML methods used following the literature on complaint management and fake reviews detection as they all successfully use extracted NLP features from complaints text for their end-goal, and should therefore also be applicable in our case. The NLP methods used are sentiment analysis following Elmurngi and Gherbi (2017), LDA topic modelling (Jia et al., 2018), and GloVe word embedding (Liu et al., 2019). The ML prediction models used are Random Forest (Barbado et al., 2019; Krishna et al., 2019), XGBoost (Krishna et al., 2019), and Support Vector Machine (SVM) Jia et al. (2018). These methods have all shown to achieve decent results and are therefore selected. Multinomial Logistic Regression (MNL) is used as a baseline for evaluation as this is one of the most commonly used model for classification problems. A flowchart of the methodology is shown in figure 3. Using the pre-processed complaint text as described in 3.5, features are extracted using NLP methods and concatenated for each complaint. Subsequently, they are fed into ML models and evaluated on their performance.

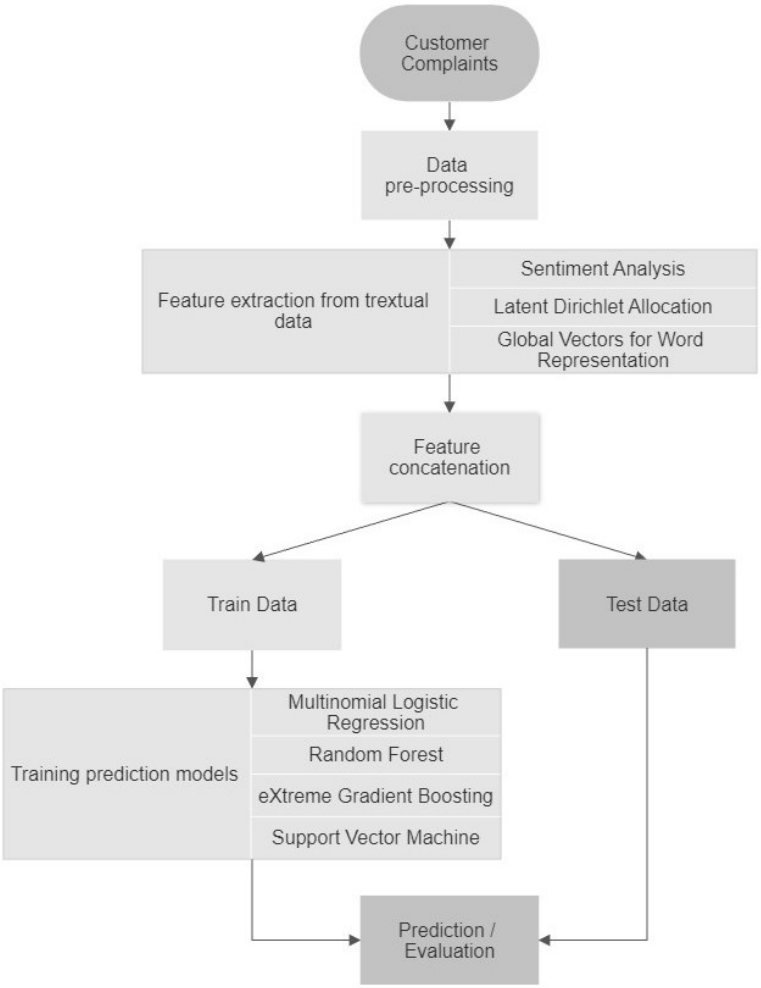


Figure 3: Flowchart of the methodology

4.1 Feature extraction

In this research, only the textual descriptions are used for feature extraction, the other features as described in 3.2 are not used because these are also labelled after a handler evaluated the descriptions.

4.1.1 Sentiment Analysis

Sentiment analysis extracts an author's emotional intent from text programmatically (Kwartler, 2017). In our case, the author is the handler summarizing the complaint. This does brings forth a new challenge as the handler might interpret the complaint differently or use other words that could change the sentiment from the original. For sentiment analysis, words in the text are not stemmed and punctuations are kept. A complaint is given a sentiment (polarity) score based on a sentiment dictionary containing positive and negative words (polarized words) (Liu, 2012). The sentiment score of a complaint is extracted following Rinker (2018) which also accounts for valence shifters (negators, amplifiers (intensifiers), de-amplifiers (downtoners), and adversative conjunctions) instead of only positive and negative words. A negator flips the sign of a polarized word (e.g., 'I do *not* appreciate the service'), in which *not* is the negator and 'appreciate' is the polarized word. An (de-)amplifier increases the effect of a polarized word (e.g., 'I *hardly/really* appreciate the service.'). And an adversative conjunction overrides the previous clause containing a polarized word (e.g., 'I appreciate the service *but* it was too slow .'). According to Rinker (2018), taking into account valence shifters increases performance in modelling sentiment as they occur fairly frequently text. A representation of the computation is as follows:

1. Each complaint ($c_i = \{S_1, S_2, \ldots, S_n\}$) constructed of S sentences, is deconstructed into sentence element ($s_i, j = \{w_1, w_2, \ldots, w_n\}$) where the words within the sentences are w. The words can be represented as $w_{i,j,k}$ (e.g., $w_{5,1,7}$ represents the seventh word, of the first sentence of the fifth complaint.)
2. Find words $w_{i,j,k}$ that appear in the sentiment dictionary, negative ($w^{-}_{i,j,k}$) and positive ($w^{+}_{i,j,k}$) words have a value of -1 and +1 respectively.
3. Consider 4 preceding and 4 following words.
4. Flip sign if a negation word ($w^{n}_{i,j,k}$) appears in the range.
5. Add/subtract 0.8 for every amplifier ($w^{a}_{i,j,k}$) or de-amplifier ($w^{n}_{i,j,k}$)
6. Multiply by $(1 + N_{adversative\ conjunctions} * 0.85)$ if the polarized word is located after an adversative conjunction.
7. Multiply by $(1 + N_{adversative\ conjunctions} * -1 * 0.85)$ if the polarized word is located before an adversative conjunction.
8. Sum over all polarity words in S_j.
9. Divide by $\sqrt{\text{number of words in } S_j}$.
10. Take the average sentiment score of all sentences S_j within c_i for the final complaint sentiment score.

As this method is only available in English and no equivalent method in Dutch is at hand, the complaints are translated to English for sentiment score extraction.

In addition, the emotions anger, fear, sadness, disgust, surprise, anticipation, trust, and joy are extracted using the NRC lexicon (Mohammad and Turney, 2013). This is done by counting the number of words in each customer complaint that are associated with the specific emotion.

4.1.2 Latent Dirichlet Allocation (LDA)

LDA topic modeling is a natural language processing technique that extracts latent topics from a collection of documents. The textual data from customer complaints is transformed into a document-term-matrix. Within this matrix, a latent distribution of documents over topics and topics over terms is assumed. Often mentioned words in the same documents represent the latent topics (Blei et al., 2003). For example, a topic with words often mentioned together in invalid complaints. Applying LDA to a complaints text generates a vector of probabilities for each topic, e.g., a complaints text has a probability of 0.9 corresponding to topic 1, and 0.1 of corresponding to topic 2. The topic labels are not given and have to be interpreted based on the output, this can be done by examining the most frequent words in each topic. The topic probabilities can be used as input features machine learning models after obtaining them.

In LDA, the textual dataset is refered to as a corpus, in our case, the dataset containing all complaints. Let the number of documents in the corpus be D (number of complaints), the number of words in complaint d be N_d, the size of the vocabulary be V, and the number of LDA topics be K. Each document is a sequence of N_d words w_i, so $w^{(d)} = \left(w_1^{(d)}, w_2^{(d)}, \ldots, w_{N_d}^{(d)}\right)$, where $w_i^{(d)}$ is the i-th word of document d, also N_d can vary across documents. In addition, $z^{(d)} = \left(z_1^{(d)}, z_2^{(d)}, \ldots, z_{N_d}^{(d)}\right)$ is the topic assignment vector for $w_i^{(d)}$ with $z_i^{(d)} = 1, \ldots, K$. Furthermore, θ denotes the document-topic proportions matrix $(D \times K)$, and ϕ denotes the topic-word probabilities matrix $(K \times V)$. The following generative process for each document is assumed by LDA:

1. Choose a topic distribution θ_d, $\theta_d \sim \text{Dir}(\alpha)$ for each document $d = 1, \ldots, D$, in which α is interpreted as the document-topic distribution.

2. Choose a topic-word distribution ϕ_k, $\phi_k \sim \text{Dir}(\beta)$ for each topic $k = 1, \ldots, K$, in which β is the topic-word distribution.

3. For each word w_i in each document d:

   (a) Choose a topic $z_i^{(d)}$, $z_i^{(d)} \sim$ Multinomial (θ_d).

   (b) Choose a word $w_i^{(d)}$, $w_i^{(d)} \sim$ Multinomial $(\phi_{z_i^{(d)}})$.

To estimate the optimal set of parameters in training an LDA model, a likelihood-function is set up in which the probability of generating the training documents is maximized (Crain et al., 2012). The likelihood-function is given by:

$$\mathcal{L}_{LDA} = \prod_{d=1}^{D} \prod_{k=1}^{K} \prod_{i=1}^{N_d} p\left(w_i^{(d)} \mid z_i^{(d)}, \phi\right) p\left(z_i^{(d)} \mid \theta_d\right) p\left(\theta_d \mid \alpha\right) p(\phi_k \mid \beta). \tag{1}$$

However, optimizing the likelihood-function directly is not achievable because $z_i^{(d)}$ are not directly observed. Within literature, there exist two main algorithms to solve the estimation problem; collapsed Gibbs sampling (Griffiths and Steyvers, 2004) and variational Expectation-Maximization (Blei et al., 2003). In this research the Gibbs sampling method is used as this is the most used method in literature (Jelodar et al., 2019).

Fine-tuning the LDA model can be done by tuning the hyperparameters α and β. A low α results in less topics per document, while a high α results in more topics per document. A low β results in the number of topics containing a small number of words, while a high level results in topics containing more words. As the number of topics k has to be provided beforehand and there is no rule of thumb to determine how many topics LDA should extract, the perplexity measure is used to determine to optimal k, in which the inverse of the geometric mean per-word likelihood of the test documents is calculated using a trained model (Newman et al., 2010). This is done by splitting the document-term matrix into a training and test set (80:20), and minimizing the perplexity measure on the test data when estimating LDA models.

4.1.3 Global Vectors for Word Representation (GloVe)

GloVe extracts the vector representation for words (word embedding) in D dimensions from a corpus. The meaning of a word is captured by the scores on the dimensions and can be used as input features for machine learning models after some processing. Similar words will have similar vectors. To illustrate this, an example is given below. Assume we train a three-dimensional embedding model ($D = 3$) on some corpus. The word vectors for the words 'coffee', 'tea', 'pizza', and 'man' resulting from the model are as follows:

$$\text{coffee} \begin{pmatrix} 0.9 \\ 0.8 \\ -0.9 \end{pmatrix}, \quad \text{tea} \begin{pmatrix} 0.8 \\ 0.9 \\ -0.7 \end{pmatrix}, \quad \text{pizza} \begin{pmatrix} 1.1 \\ -0.4 \\ -0.9 \end{pmatrix}, \quad \text{man} \begin{pmatrix} -0.8 \\ 0.3 \\ 1 \end{pmatrix}.$$

To give these vector scores more context, lets assume that dimension 1 tries to define whether the word means something related to food, dimension 2 tries to define whether something is related to liquid, and dimension 3 tries to define whether something is a living being. It can be noticed that 'coffee', 'tea', and 'pizza' have similar positive scores on dimension 1 and are indeed related to food while 'man' is not related to food and scores negative. On the second dimension, using the same intuition, the words 'coffee', 'tea', and 'man' score positive and are therefore more related to liquid while 'pizza' scores negative and is less related to liquid. Lastly, on the third dimension, only 'man' scores positive as it is the only word related to a living being.

Figure 4 illustrates these vectors. The words 'coffee', 'tea', and 'pizza' point towards the same direction and are therefore more similar to each other in comparison to the word 'man' as is points

towards the other direction. In reality, the number of dimensions used is much higher than 3 which makes the model far less interpretable.

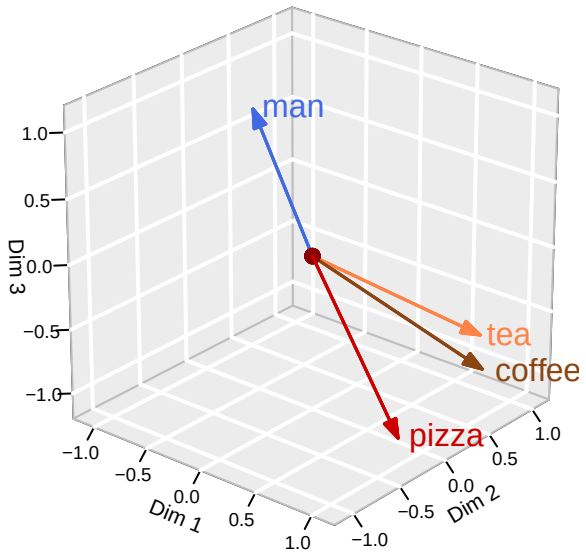


Figure 4: Three-dimensional embedding space example

GloVe is based on a combination of surrounding words (as in Word2vec) and the word-word co-occurrence matrix. Pennington et al. (2014) argue that by making use of both global statistics and local statistics of a corpus in Glove to capture the meaning of words, the best of both worlds is combined. GloVe finds two word vectors for each word in the vocabulary, which are named ordinary and context word vectors. The algorithm initially starts with two random vectors (the ordinary and context vector) per word and iteratively updates them by optimization of a loss function. After training a GloVe model, the two sets of vectors are collected in two $D \times V$ matrices, where D is the dimensionality of the embedding which has to be preset, and V is the number of unique words in the vocabulary. Pennington et al. (2014) suggest that the two word vectors should be combined by summing them to get one vector representation per word as they found evidence of improved results. Following their recommendation this is also done in our case.

To construct the GloVe model, a word-word co-occurence matrix X is created based on the corpus, which is the cleaned and stemmed text. The elements X_{ij} represent the number of times word j occurs in the context of word i, the context window is 6 by default, that is 3 words before and 3 words after. Following the notation by Pennington et al. (2014), let $X_i = \sum_k X_{ik} = \sum_k X_{ki}$ be the number of total occurrences of any word in the context of word i. Also, let $P_{ij} = P(j \mid i) = \frac{X_{ij}}{X_i}$ be the probability that word w_j appears in the context of word w_i. The co-occurence probability ratio is defined as $\frac{P_{ik}}{P_{jk}}$. This ratio is based on three words i, j, and k and captures the relationship between words. This is done by computing the co-occurance probability ratios of i and j with

various words k. The value of this ratio is expected to be high if k is related to i but not to j. Likewise, the ratio is expected to be low for k related to j but not to i. For k related or unrelated to both i and j, the ratio should be close to one. The GloVe model can be denoted as:

$$F\left(\left(w_i - w_j\right)^T \tilde{w}_k\right) = \frac{P_{ik}}{P_{jk}}, \tag{2}$$

where F is a function choice, $w \in \mathbb{R}^d$ are ordinary word vectors, and $\tilde{w} \in \mathbb{R}^d$ are context word vectors. To ensure the order of the word vector does not matter, that is changing $\tilde{w}$ with either w_i or w_j should result in the same value for F, is by specifying a ratio:

$$F\left(\left(w_i - w_j\right)^T \tilde{w}_k\right) = \frac{F\left(w_i^T \tilde{w}_k\right)}{F\left(w_j^T \tilde{w}_k\right)} = \frac{P_{ik}}{P_{jk}}. \tag{3}$$

Pennington et al. (2014) argued that the solution of equation (3) is $F = \exp$ and yields

$$w_i^T \tilde{w}_k = \log\left(P_{ik}\right) = \log\left(X_{ik}\right) - \log\left(X_i\right). \tag{4}$$

However, solution (4) is not symmetric for i and k because of the $\log(X_i)$ on the right-hand side. Therefore Pennington et al. (2014) replace $\log(X_i)$ with a bias b_i for w_i and $\tilde{b}_k$ for $\tilde{w}_k$ to establish symmetry, resulting in the final model equation:

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log\left(X_{ik}\right). \tag{5}$$

To fit the model, a least squares problem is formed using the model in equation (5) in which an additional weighting function f is introduced:

$$J_{GloVe} = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2, \tag{6}$$

where V is the size of the vocabulary. The weighting function serves two purposes, that is avoiding problems with $\log(0)$ when X_ij is small or zero, and avoiding too much importance for large values of X_ij. The proposed weighting function is

$$f(x) = \begin{cases} \left(x/x_{\max}\right)^\alpha & \text{if } x < x_{\max} \\ 1 & \text{otherwise,} \end{cases} \tag{7}$$

where x_{max} and α are hyperparameters. By default x_{max} and α are set to 100 and 0.75, respectively. Pennington et al. (2014) did not give motivation on the choice of x_{max}, but did motivate their choice for $\alpha = 0.75$ based on emprical tests.

4.2 Prediction models

4.2.1 Multinomial Logistic Regression (MNL)

Multinomial Logistic Regression is a generalized version of logistic regression when the dependent variable is a categorical variable with more than two classes ($K > 2$). The probabilities of the different classes y_i are modeled using a set of independent variables x_i with $i = 1, ..., N$ and coefficients β_k related to class k. In this research, recall that the classes are defined as:

$$y_i = \begin{cases} 1 & \text{if Valid complaint;} \\ 2 & \text{if Partially Valid Complaint;} \\ 3 & \text{if Invalid Complaint.} \end{cases} \tag{8}$$

For MNL, a single class has to be set as a reference, any of the K classes can be selected for this role. The choice of this reference class is unimportant for the outcomes of the predictions, only the interpretation and coefficient estimates will differ (James et al., 2013). The model is written as follows:

$$P(y_i = k \mid x_i) = \frac{e^{\beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kp}x_p}}{1 + \sum_{j=1}^{K-1} e^{\beta_{j0}+\beta_{j1}x_1+\cdots+\beta_{jp}x_p}}, \tag{9}$$

for $k = 1, ..., K-1$, and

$$P(y_i = K \mid x_i) = \frac{1}{1 + \sum_{j=1}^{K-1} e^{\beta_{j0}+\beta_{j1}x_1+\cdots+\beta_{jp}x_p}}, \tag{10}$$

for the reference class.

4.2.2 Random Forest (RF)

Random forest builds a collection of decision trees to predict the output for each instance, in our case the validity of a complaint. As this research deals with a classification problem, the majority vote of the individual trees in the collection determines the predicted class for each instance (Breiman, 2001). Each tree in the collection is grown by drawing bootstrapped samples from the training data to reduce variance. The Random Forest algorithm is shown in Algorithm 1 (Hastie et al., 2009).

Algorithm 1: Random Forest

1 **for** $b = 1$ **to** B **do**
2 From the training data, draw a bootstrap sample Z of size N.
3 Using Z, grow a random-forest tree T_b by repeating the following steps recursively for each unsplit node of the tree, until the minimum node size d_{min} is reached:
4 From the p available variables, select m variables .
5 Among the m variables, find the best split-point .
6 Split the nodes into two daughter nodes.
7 **end**

Output: The ensemble trees of $\{T_b\}_1^B$.

To make a prediction at a new point x:

Let $\hat{C}_b(x)$ be the prediction of the b^{th} random forest tree.

Then $\hat{C}_{rf}^B(x) = majority\ vote\left\{\hat{C}_b(x)\right\}_1^B$

For classification, the value of m (line 4) is by default $\sqrt{p}$. The best split-point (line 5) is found following the Gini index, which is a measure of total variance across K classes (James et al., 2013):

$$G = \sum_{k=1}^{K} p_{dk}\left(1 - p_{dk}\right), \tag{11}$$

where p_{mk} is the proportion observations of class k in node d. The impurity of a tree is minimized by finding the split-points with the largest decrease in Gini index. d_{min} can be tuned to control for the depth of a tree, a higher value prevents the tree from overgrowing and thus overfitting.

4.2.3 eXtreme Gradient Boosting (XGBoost)

XGBoost developed by Chen and Guestrin (2016) has proven to be successful in many applications because of its superior performance. In XGBoost, consecutive trees are built based on previously created trees. Using K additive trees, the output for each instance is predicted:

$$\hat{y}_i = \sum_{k=1}^{K} f_k\left(x_i\right), f_k \in F, \tag{12}$$

where $\hat{y}$ is the predicted class of a complaint, f_k is an individual tree, x are the features, F is the space of all individual trees, and $i = 1,\ldots, N$. Each leaf node on a tree contains a continuous score w, the final prediction is calculated by summing the scores on the resulting leafs based of the decision rules.

The following regularized objective function is optimized to train the set of independent trees:

$$\mathcal{L}_{xgboost} = \sum_{i=1}^{n} l\left(\hat{y}_i, y_i\right) + \sum_{k=1}^{K} \Omega\left(f_k\right), \tag{13}$$

where the loss function l measures the error between the predicted class of the complaint and the actual class of the complaint. For multi-class classification, the used loss function is cross-entropy:

$$l\left(\hat{y}_i, y_i\right) = \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log\left(\hat{y}_{ik}\right), \tag{14}$$

Ω in equation (13) regularizes model complexity to handle overfitting:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{15}$$

where the minimum loss reduction required for a node split is γ, and the number of leaf nodes in the tree is T. Setting γ higher will result in simpler trees, by default this is set to 0. In addition, to reduce overfitting, L1 and L2 regularization can be used, λ is the L2 parameter and is set by default to 1.

The objective function in equation (14) is optimized iteratively, the function is written as:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t\left(x_i\right)\right) + \Omega\left(f_t\right), \tag{16}$$

for the t-th iteration, After re-formulation the objective function for the t-th interation (tree) is written as:

$$\mathcal{L}^{(t)} = \sum_{j=1}^{T} \left[G_j w_j + \frac{1}{2}\left(H_j + \lambda\right) w_j^2\right] + \gamma T, \tag{17}$$

where $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$, with $I_j = \{i \mid q\left(x_i\right) = j\}$ being the set of instances on the j-th leaf node for a given tree structure q, and g_i and h_i being the first and second order Taylor expansion of the loss function, respectively.

The optimal weight w_j^* is then calculated for each leaf node j by:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}. \tag{18}$$

Substituting equation (18) into (17) gives us the equation for calculating the optimal value for a tree:

$$\mathcal{L}^* = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T. \tag{19}$$

A greedy algorithm is used to select the best tree structure, which starts with a single leaf and adds branches repeatedly since it is not possible to enumerate all possible trees (Chen and Guestrin, 2016). The formula used to evaluate the split candidates is as follows:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma, \tag{20}$$

where L is the left leaf node, and R the right leaf node. The split stops when the Gain is negative resulting in the final tree structure.

4.2.4 Support Vector Machine (SVM)

The SVM algorithm first introduced by Cortes and Vapnik (1995), classifies cases based on a linear decision rule (hyperplane). Given training data consisting of n pairs of features and classes (x_i, y_i), where $x_i \in \mathbb{R}^p$ and $y_i \in \{-1, 1\}$, a hyperplane is drawn through the feature space such that the margin between the hyperplane and the closest points of each class (support vectors) is maximized. Following the notation of Hastie et al. (2009), $x_i^T\beta + \beta_0 = 0$ is defined as a hyperplane with unit vector β and intercept β_0, the objective to be optimized is written as:

$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i \tag{21}$$

$$\text{subject to } \begin{cases} y_i\left(x^T\beta + \beta_0\right) \geq (1-\xi_i) \\ \xi_i \geq 0 \end{cases} \tag{22}$$

where the "cost" term C is constant and penalizes ξ, which is the proportional amount by which the prediction made by SVM is on the incorrect hyperplane side (also named slack variable), and $i = 1, \ldots, N$. A data point classified correctly will have $\xi = 0$, while an incorrect classified point will have $\xi > 1$. A small C allows for more points to cross the hyperplane to the incorrect side, thus reducing variance and over-fitting.

When data is not linearly separable, the data is projected into higher a higher dimensional space by SVM using a mapping function $\phi(x_i)$, also known as kernel function. The popular kernel functions are: linear, polynomial, and radial. The linear kernel works good for text classification, and also when the number of features is large (Hsu et al., 2003; Joachims, 1998), therefore the linear kernel is used in this research.

4.3 Evaluation metrics

For model evaluation, confusion matrices, consisting of the predicted class and the actual class are created. In a confusion matrix, a true positive (TP) indicates whether the actual positive class is predicted correctly. Likewise, a true negative (TN) indicates whether the actual negative class is predicted correctly. A false positive (FP) indicates whether the actual positive class is predicted incorrectly. And a false negative (FN) indicates whether the actual negative class is predicted incorrectly.

The Accuracy, Precision, Recall, and F1 scores are used for evaluating the models and are formulated as follows for $i = 1, ..., K$ classes:

$$Accuracy = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{23}$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i}, \tag{24}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i}, \tag{25}$$

$$F1_i = 2 \cdot \left(\frac{Precision_i * Recall_i}{Precision_i + Recall_i} \right). \tag{26}$$

First, Accuracy measures the number of correct predictions made among all predictions made. Second, Precision measures the number of correct positive predictions made among the total predicted positive predictions. Third, Recall measures the number of correct positive predictions among all samples that should have been predicted as positive. Lastly, the F1-score is a Harmonic Mean between Precision and Recall and therefore combines both measures. The weighted average for Precision, Recall, and F1-score are taken to describe the overall performance as we are dealing with a multiclass problem, this is done by calculating the mean of all per-class scores while considering the number of actual occurrences of the class in the data.

5 Results

In this section the results are discussed. First, the extracted features from the NLP methods are discussed in 5.1. Second, the results from the ML models using the extracted features are discussed in 5.2. Finally, a few important GloVe dimensions resulting from the ML models are discussed in 5.3.

5.1 Extracted features

5.1.1 Sentiment Analysis

For each complaint, a sentiment score is extracted and the number of words corresponding with each emotion category is counted following the methodology in 4.1.1. Figure 5 shows a density plot of the extracted sentiment score by validity.

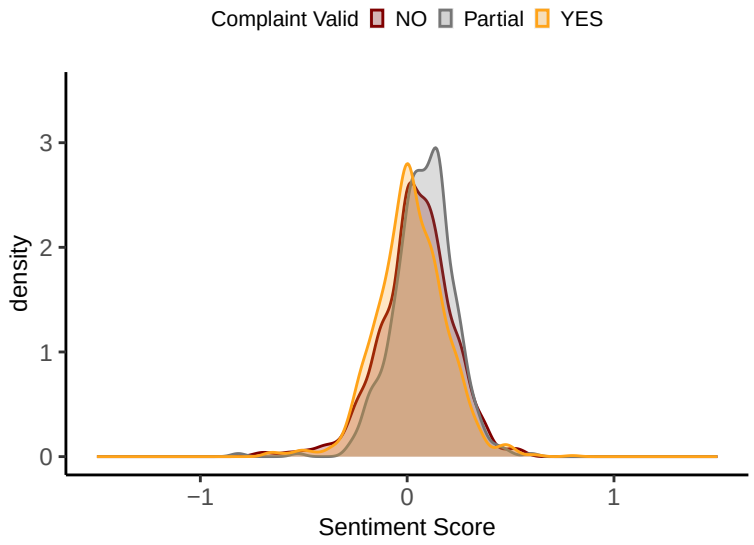
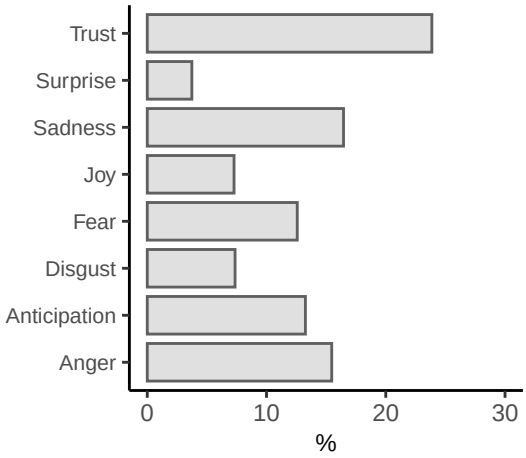


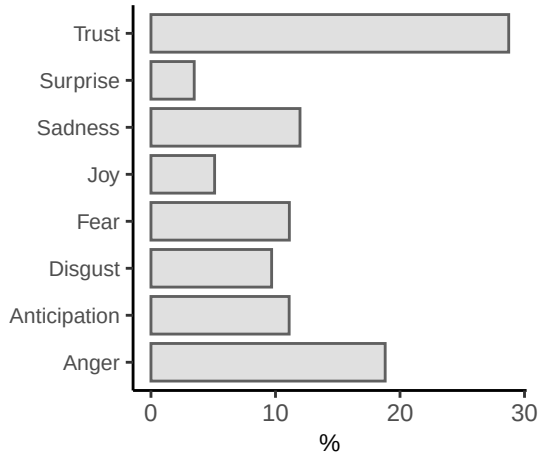
Figure 5: Sentiment score by validity

It can be seen that there is no clear sentiment distinction between the categories as they mostly overlap. Valid and invalid complaints seem to be more similar, while partially valid complaints are more positive. The average sentiment for all categories is positive with valid complaints having a mean of 0.018, while invalid and partially valid have a mean sentiment score of 0.041, and 0.079, respectively. Valid complaints are on average the least positive, while partially valid complaints are the most positive. The small difference between the categories is likely due to the use of many neutral and domain specific words, hence the values being close to zero. These results suggest that feature extraction using sentiment analysis on the textual complaint is challenging, and sentiment score on its own might not be a strong predictor.

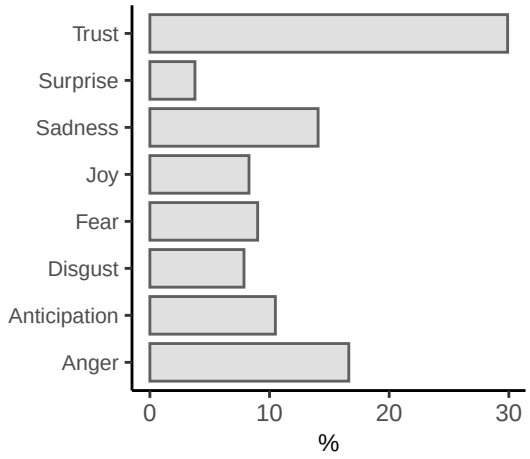
Furthermore, figure 6 shows the percentage of emotions present by (a) Valid, (2) Invalid, and (3) Partially valid complaints after extracting the emotion counts for each complaint. The emotion Trust has the highest prevalence among all complaint categories. The emotion Sadness and Anticipation seem to be more prevalent in valid complaints, while the emotions Anger and Disgust are more prevalent in invalid complaints.



(a) Valid complaints



(b) Invalid complaints



(c) Partially valid complaints

Figure 6: Percentage of emotions by (a) valid, (b) invalid, and (c) partially valid complaints

Both the extracted sentiment score and emotion count are used as input features for the prediction models.

5.1.2 LDA

Following the methodology in 4.1.2, topic probabilities for each complaint are obtained using LDA (*topicmodels* package). The hyperparameters α and β have not been tuned and kept at their default setting, which are $\alpha = 0.1$ and $\beta = 0.05$. Also, the Gibbs sampling method is used in the generative process. The number of topics k have been set at 40 as this is where the perplexity measure seems to be minimized as can be seen in figure 7.

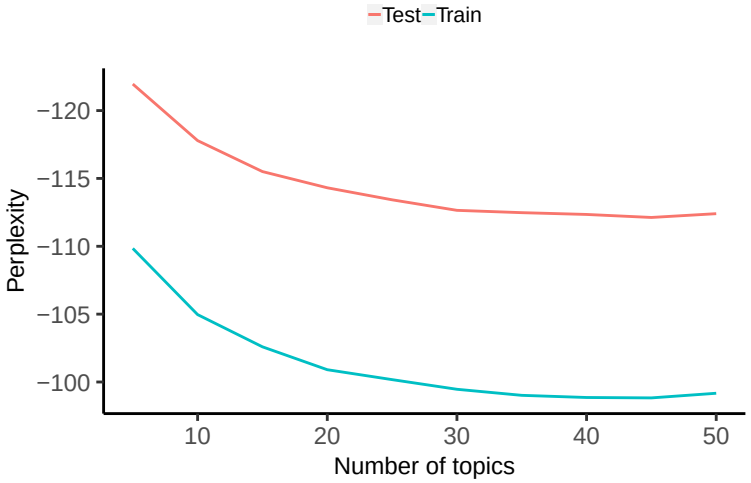
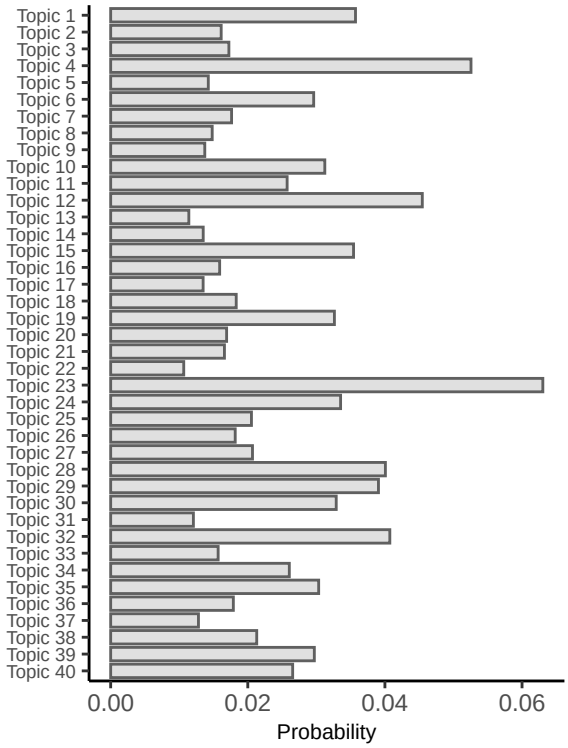


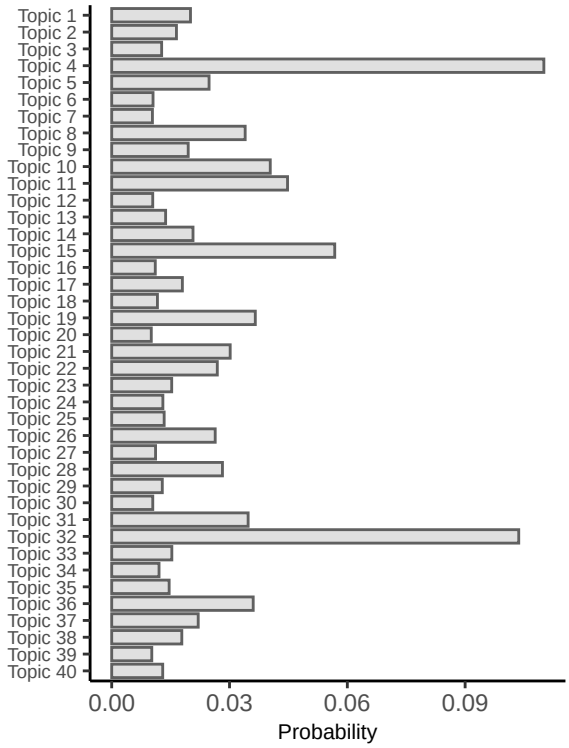
Figure 7: LDA Perplexity

Figure 8 shows the topic probabilities by (a) valid, (b) invalid, and (c) partially valid complaints. Among valid complaints, topic probabilities are more spread in comparison to invalid and partially valid complaints. The most prominent latent topics among valid complaints are 4, 12, and 23. Among invalid and partially valid complaints, topic probabilities are less spread with only a few dominant latent topic. In invalid complaints, topics 4, 15, and 32 seem to be most prominent, the same applies to partially valid complaints with addition of topic 10. Interestingly, the prominent topics in invalid and partially valid complaints also have a strong occurrence in valid complaints. This suggests that many invalid and partially valid complaints contain descriptions that are also present in valid complaints.

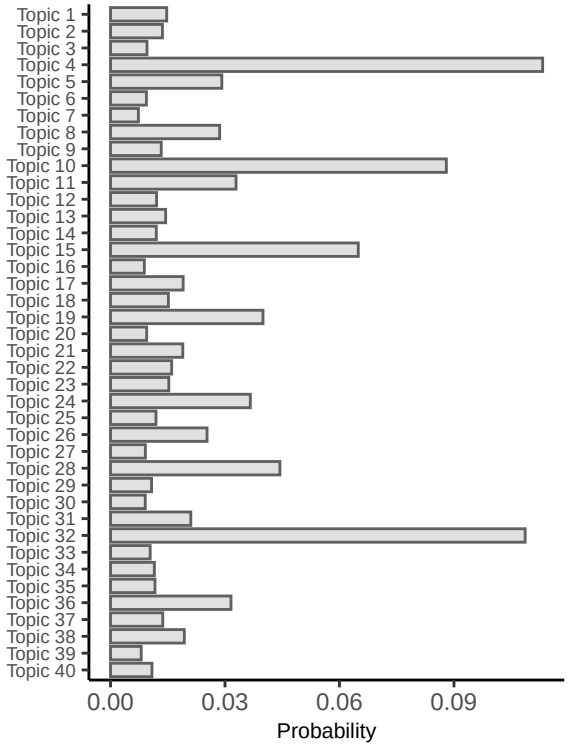
The most prominent topics and their top 10 terms among the complaint classes are shown in Table 1. In addition, table 13 Appendix shows all 40 topics and their top 10 terms.



(a) Valid complaints



(b) Invalid complaints



(c) Partially valid complaints

Figure 8: Probability of LDA generated topics by (a) valid, (b) invalid, and (c) partially valid complaints

Table 1: Most prominent LDA topics

Topic 4	**Topic 10**	**Topic 12**	**Topic 15**	**Topic 23**	**Topic 32**
verzekerd	verzekerd	lang	verzekerd	klant	expert
expert	klacht	klant	opdrachtgever	mail	reactie
opdrachtgever	gebeld	bon	expert	dossier	opdrachtgever
schad	expert	dossier	terugkoppel	gestuurd	gevraagd
klacht	gesprek	sos	gevraagd	lat	teruggekoppeld
overleg	tevred	wachttijd	reactie	geslot	klacht
gebeld	goed	bol	contact	wet	verzocht
aanvull	afgehandeld	klacht	gegev	asr	vrag
besprok	doorgesprok	mail	gezond	ontvang	terugkoppel
akkoord	gesprok	abn	gebeld	vrag	aanvull

As mentioned in the methodology, topic labels are not given and have to be interpreted based on the output by examining the most frequent words in each topic. To give an idea of what information these topics capture, the topic categories are determined for the most prominent topics based on the top 10 terms:

- Topic 4: Policy holder, expert, and client problem relating to damage
- Topic 10: Policy holder and expert problem
- Topic 12: Waiting time
- Topic 15: Policy holder, expert, and client problem
- Topic 23: Client problem
- Topic 32: Expert and client problem

The topic probabilities for each complaint which adds up to 1 is used as input features for the prediction models to predict the complaint class. As discussed before and seen in figure 8, valid complaints have multiple dominant topics while the most dominant topics in invalid and partially valid complaints also have a high probability of occurring in valid complaints (e.g., topics 4, 15, and 32). This suggests that predicting the invalid and partially valid complaint class based on LDA topics might show some difficulties.

5.1.3 GloVe

Based on the complaints, 100-dimension GloVe embeddings are obtained using 500 iterations in which the ordinary and context vector are combined as described in 4.1.3 (*text2vec* package). Though not visualized often in literature because of the many dimensions of word embeddings, a view of the GloVe word embedding in 2 dimensional space is shown in figure 9 to give the reader an idea of how the meaning of words are captured. This is done by reducing the 100 dimensions using Uniform Manifold Approximation and Projection (*umap* package) to 2 dimensions.

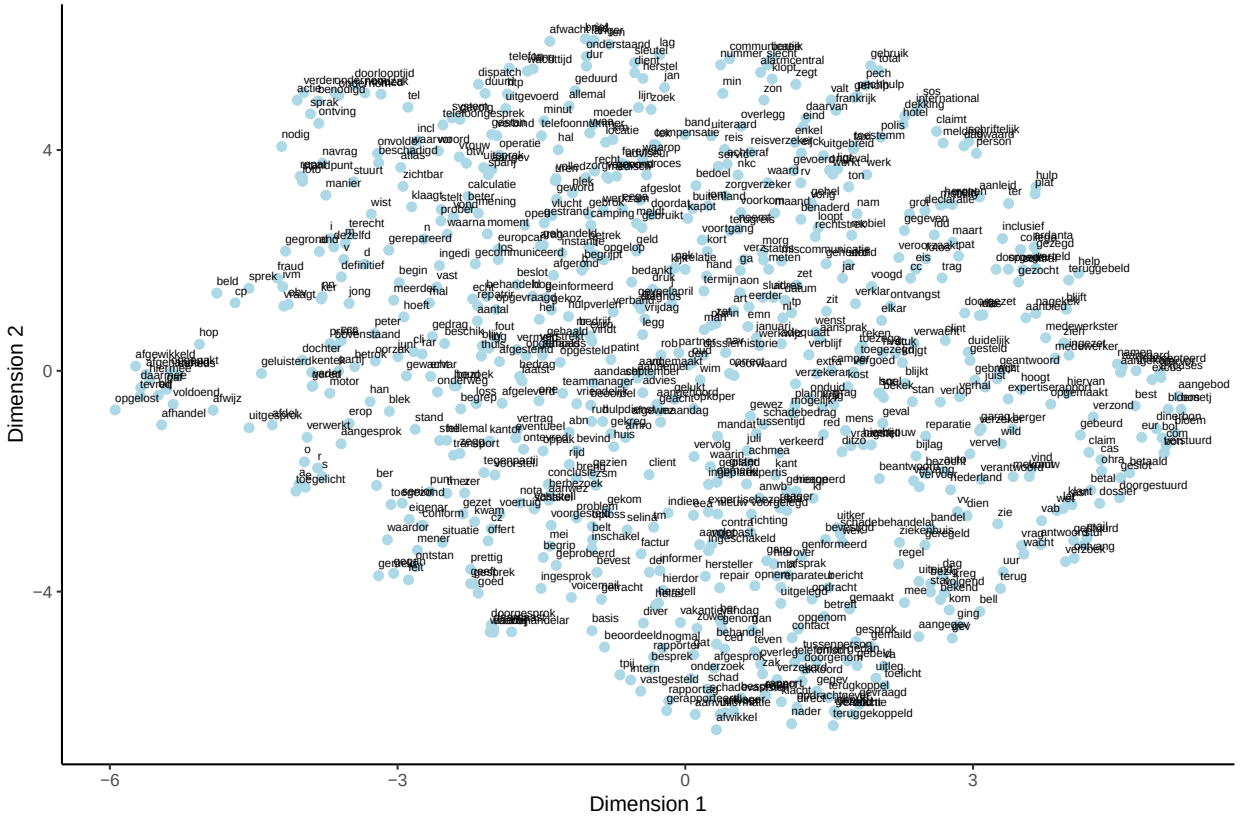
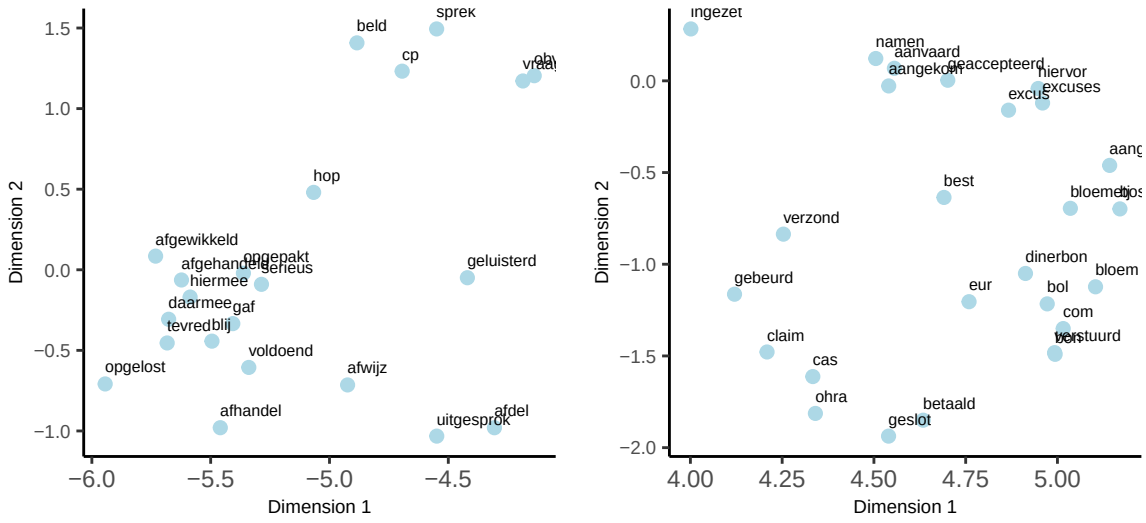


Figure 9: GloVe word embedding in 2D

Words that are more clustered to each other should have somewhat similar meaning, a zoomed in view is shown in figure 10 to elaborate on this. For example, on the far left we have words as 'afgehandeld' (done), 'afgewikkeld' (settled), 'opgelost' (solved), and 'tevreden' (satisfied). On the far right, words as 'excuses' (apologies), 'bloem' (flowers), 'bol' (a dutch webshop), 'bos' (bouquet), and 'dinerbon' (dinner voucher).



(a) Far left

(b) Far right

Figure 10: Partial views of the GloVe word embedding in 2D

As mentioned before, words are represented in a vector with $D = 100$ dimensions, this means that each complaint contains multiple words with its own unique vector. For an ML model to be able to use the word embeddings as features, the mean value of all word vectors for each complaint is calculated if the word is in the complaint following Liu et al. (2019). In this way, the gross vectors for each complaint is obtained and each complaint is represented as 100-dimension features.

To simply illustrate this, lets recall our three-dimensional embedding example (4.1.3) for the words 'coffee' and 'pizza',

$$\text{coffee} \begin{pmatrix} 0.9 \\ 0.8 \\ -0.9 \end{pmatrix}, \quad \text{pizza} \begin{pmatrix} 1.1 \\ -0.4 \\ -0.9 \end{pmatrix}.$$

If a complaint contains the words 'coffee' and 'pizza', the complaint will have a gross embedding of:

$$\text{Complaint} = \begin{pmatrix} (0.9 + 1.1)/2 \\ (0.8 + -0.4)/2 \\ (-0.9 + -0.9)/2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0.2 \\ -0.9 \end{pmatrix}.$$

5.2 Prediction models

The created features, which are, sentiment score, emotion count, LDA topics, and GloVe features are used to predict the complaint validity. The data is randomly divided into a training (75%) and test (25%) set based on the validity proportions to construct and evaluate the prediction models. For illustration, an example of the created features in a data frame is given in table 2. The complaint class is to be predicted from the created feature scores.

Table 2: Example of features.

	Complaint Class	Sentiment Score	Anger	...	Joy	LDA Topic 1	...	LDA Topic 40	GloVe Dimension 1	...	GloVe Dimension 100
Complaint 1	YES	-0.8164	9	...	2	0.0500	...	0.2500	0.1081	...	-0.4779
Complaint 2	YES	-0.6324	1	...	5	0.0333	...	0.1967	0.5007	...	0.0081
Complaint 3	NO	0.0368	0	...	3	0.6777	...	0.0046	0.0843	...	-0.1966
Complaint 4	Partial	0.5815	5	...	0	0.0687	...	0.1392	-0.3128	...	-0.3019
...	...	...	...	...	...	...	...	...	...	...	...

5.2.1 Multinomial Logistic Regression

The MNL model is built using the *nnet* package and Invalid Complaint as the reference class. Looking at figure 11, it can be seen that the valid complaints are classified well, but the model performs not so well on invalid and especially partially valid complaints. This is further supported by Table 3, which shows low Precision, Recall, and F1-score for invalid complaints and partially valid complaints. The accuracy of the model is 59.9%.

On the training data, the model has an accuracy of 67.2% which can be seen in figure 21 Appendix together with the corresponding evaluation metrics in table 14 Appendix. It is noticeable that the model initially struggled with correctly classifying partially valid complaints on the training data (Recall 0.28).

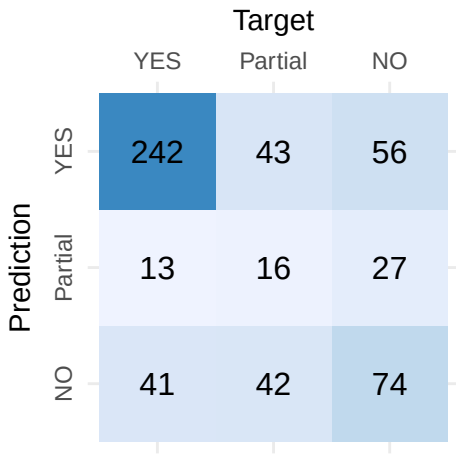


Figure 11: Confusion matrix MNL of predictions on test data with accuracy of 59.9%.

Table 3: Evaluation metrics Random Forest test data.

	Precision	Recall	F1
YES	0.7097	0.8176	0.7598
Partial	0.2857	0.1584	0.2038
NO	0.4713	0.4713	0.4713
Weighted	0.5648	0.5993	0.5767

5.2.2 Random Forest

The Random Forest model is built using the ranger implementation of Random Forest (*ranger* package) for faster computational time (Wright and Ziegler, 2015). 5-fold cross validation and a grid search is performed to find the optimal values for the hyperparameters. The optimal values of the hyperparameters are shown in Table 4 together with the description and search space. The resulting forest consist of 300 trees, uses 10 candidate variables at each split, and has a minimum node size of 50.

Table 4: Tuned hyperparameters for Random Forest.

Description	Hyperparameter	Search Space	Optimal Value
Number of trees	ntree	{100, 300, 500, 700}	300
Variables used for each split	mtry	{5, 10, 20, 30}	10
Minimum size of a node	min.node.size	{0, 25, 50, 75, 100}	50

Looking at the confusion matrix in figure 12, it can be noticed that valid complaints are classified well, but the model performs not so well on invalid and especially partially valid complaints. Table 7 further supports this, which shows low Precision, Recall, and F1-score for invalid complaints and partially valid complaints. The accuracy of the model is 62.6%.

On the training data, the model has an accuracy of 86.9% which can be seen in figure 22 Appendix together with the corresponding evaluation metrics in table 15 Appendix. The difference between the performance on the train and test set suggests that the model is overfitting on the training data even after tuning the parameters. Also, as with MNL, the model initially struggled with correctly classifying partially valid complaints on the training data (Recall 0.47).

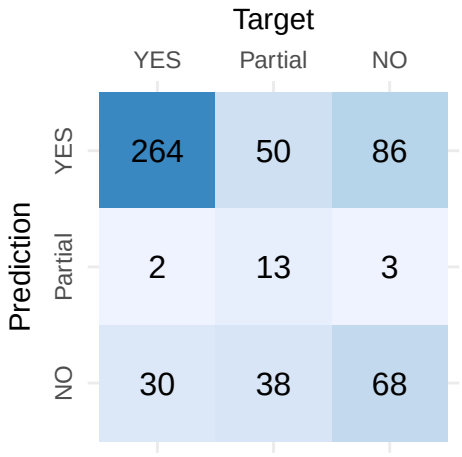


Figure 12: Confusion matrix Random Forest of predictions on test data with accuracy of 62.3%.

Table 5: Evaluation metrics Random Forest test data.

	Precision	Recall	F1
YES	0.66	0.8919	0.7586
Partial	0.7222	0.1287	0.2185
NO	0.5	0.4331	0.4642
Weighted	0.6260	0.6227	0.5767

Figure 13 shows the most important features of the Random Forest model. The importance is based on the Gini index (Equation (11)) It can be seen that the top important features are solely elements from the high-dimensional GloVe embedding model such as V3 (Dimension 3), V57 (Dimension 57), and V20 (Dimension 20).

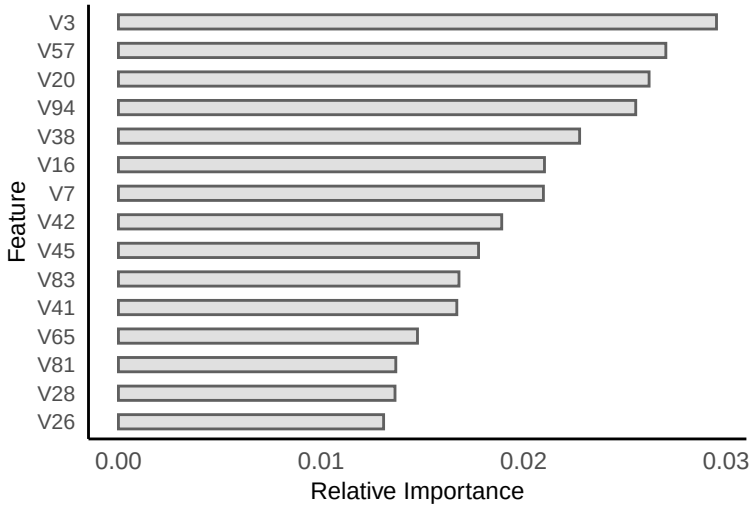


Figure 13: Feature importance Random Forest

5.2.3 XGBoost

The XGBoost model is built using the *xgboost* package, 5-fold cross validation, a maximum of 5000 trees, and the cross-entropy metric for validation. To increase model performance, a grid search is performed to find the optimal values for a few hyperparameters. The optimal values of the hyperparameters are shown in Table 6 together with the description and search space. The resulting XGBoost model consist of 50 trees, has a maximum tree depth of 3, a learning rate of 0.1, a subsample ratio of 0.7, an λ of 10, and a γ of 1.

Table 6: Tuned hyperparameters for XGBoost.

Description	Hyperparameter	Search Space	Optimal Value
Number of trees	nrounds	[1, 5000]	50
Maximum depth of a tree	max_depth	{2, 3, 4, 6, 9, 12}	3
Learning rate	eta	{0.1, 0.3, 0.6, 1}	0.1
Subsample ratio	subsample	{0.7, 1}	0.7
L2 regularization term	lambda	{1, 5, 10 ,30}	10
Minimum loss reduction	gamma	{0, .5, 1, 2, 3 }	1

Looking at the confusion matrix figure 14, it can be noticed that valid complaints are also classified well, but invalid and partially valid complaints not so well. The evaluation metrics are shown in Table 7. The accuracy of the model is 63.9%.

On the training data, the model has an accuracy of 76.6% which can be seen in figure 23 Appendix together with the corresponding evaluation metrics in table 16 Appendix. As with MNL and RF, XGBoost also seems to slightly overfit after tuning the parameters and struggled with

correctly classifying partially valid complaints on the training data (Recall 0.35).

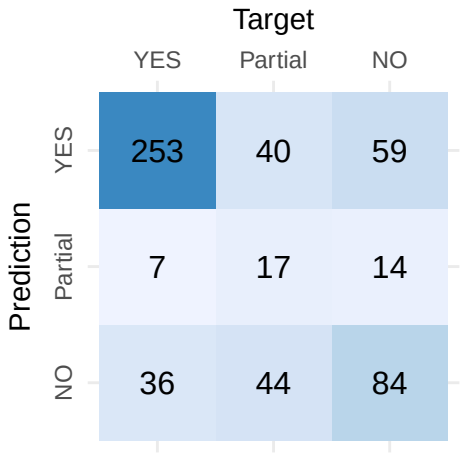


Figure 14: Confusion matrix XGBoost of predictions on test data with accuracy of 63.9%.

Table 7: Evaluation metrics XGBoost test data.

	Precision	Recall	F1
YES	0.7188	0.8547	0.7809
Partial	0.4474	0.1683	0.2446
NO	0.5122	0.535	0.5234
Weighted	0.6108	0.6390	0.6102

Figure 15 shows the most important features of the XGBoost model. The importance is calculated during the the construction of the trees based on Gain (Equation (20)), which is interpreted as the improvement in accuracy brought by a feature to the branches it is on. It can be seen that the top important features are also the elements from the high-dimensional GloVe embedding model with the addition of a few LDA topics (Topics 24 and 31). It is noticeable that the LDA topics in the top important list are not the most prominent topics as discussed in 5.1.2. Topic 24 and topic 31 can be interpreted by looking at the top terms in table 13 Appendix. Topic 24 contains many terms related to apologizing and topic 31 contains terms related to Second expert opinion.

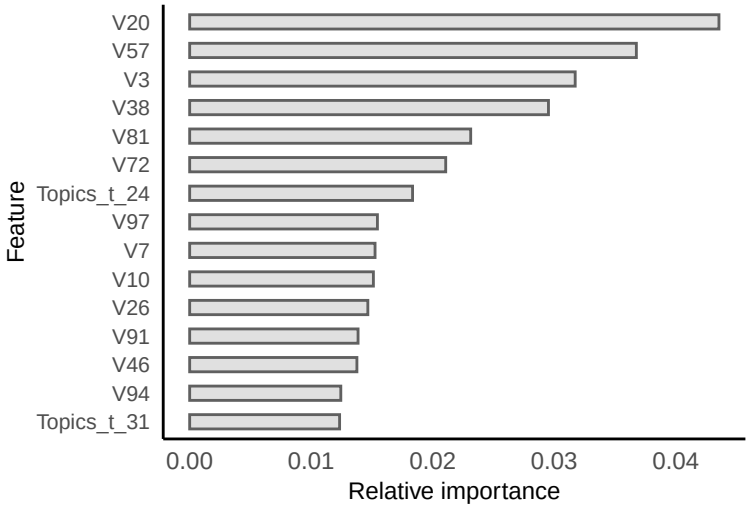


Figure 15: Feature importance XGBoost

5.2.4 SVM

The SVM is built using a linear kernel and 5-fold cross validation (*e1071* package). The penalty parameter C is tuned using a grid search (Figure 8).

Table 8: Tuned hyperparameter for SVM.

Description	Hyperparameter	Search Space	Optimal Value
Cost of constraints violation	cost	{0.01, 0.1, 1, 10, 100}	0.01

Looking at the confusion matrix in figure 16, it can be noticed that the SVM model performs quite similar to the previously discussed models. Though SVM seems to perform better on invalid complaints. The evaluation metrics are shown in Table 9. The accuracy of the model is 65.0%.

On the training data, the model has an accuracy of 69.6% which can be seen in figure 24 Appendix together with the corresponding evaluation metrics in table 17 Appendix. The model performs equally well on both train and test data, it can therefore be said that the SVM is a more generalized model with minimal tuning.

As a mapping function is used by SVM and the data is projected onto a higher dimensional space, a feature importance plot cannot be created as with Random Forest and XGBoost.

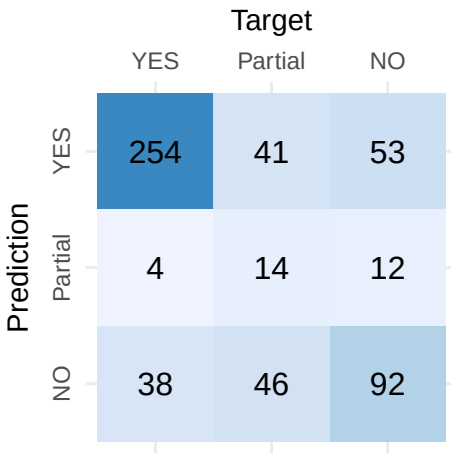


Figure 16: Confusion matrix SVM of predictions on test data with accuracy of 65.0%.

Table 9: Evaluation metrics SVM test data.

	Precision	Recall	F1
YES	0.7299	0.8581	0.7888
Partial	0.4667	0.1386	0.2137
NO	0.5227	0.586	0.5526
Weighted	0.6232	0.6498	0.6170

5.2.5 Overall results

Now that the models have been discussed individually, an overall summary of the results is given by comparing the models. A summary of the results based on the evaluation metrics for all models is given in Figure 17 and 10. The F1-score is indicated in orange, Precision in red, and Recall in blue. In addition, the valid complaint class is indicated by plus signs, partially valid complaint class by squares, invalid complaint class by triangles, and the weighted average for the values by circles.

Overall, SVM is the best performing model with an weighted average F1-score of 0.6170. In terms of accuracy, SVM also performs best with 65.0%. For valid complaints and invalid complaints, SVM also performs best with an F1-score of 0.79 and 0.55, respectively. For partially valid complaints, XGBoost performs best with an F1-score of 0.24. In addition, all models initially struggled with correctly classifying partially valid complaints on the training data (Figure 21, 22, 23, and 24). Upsampling has been tried for all models to resolve this issue but did not improve results. For invalid complaints, SVM and XGBoost are able to correctly classify more than the majority when looking at Recall. In theory, this means that such model might increase efficiency in handling complaints as the majority of invalid complaints can be deprioritized and the the number of false negative valid complaints is less than the number of true positive invalid complaints. When partially valid complaints are also allowed to be deprioritized, further efficiency gain might

be achieved as predictions made for this class mainly go into valid or invalid.

Performance on unseen (test data) invalid complaints seems to be challenging. As discussed in 5.1.1 and 5.1.2, many invalid and partially valid complaints might contain overlapping or ambiguous text descriptions which make them difficult to distinguish from valid complaints by the models. Moreover, according to the feature importance values from Random Forest and XGBoost, GloVe feature dimensions are shown to have the highest contribution to the predictive performance of the models, especially dimensions 57, 20, 38, and 3 (These dimensions are discussed below in 5.3). LDA appeared in the top important features for XGBoost, while features extracted from sentiment analysis did not appear at all in the top important list. Furthermore, based on the feature importance results, the models have been re-examined by solely using LDA or GloVe features as input. Performance of all models decreased when doing this, solely using LDA features decreased performance the most while solely using GloVe slightly decreased performance.

Relating our results back to literature, in general, SVM has shown to offer the best performance when using extracted text features (Crawford et al., 2015). Our results also suggest that SVM is the best performer based on the F1-score. In Liu et al. (2019), SVM was also selected as one of the better performing model while using word2vec and GloVe features to predict predefined complaint categories.

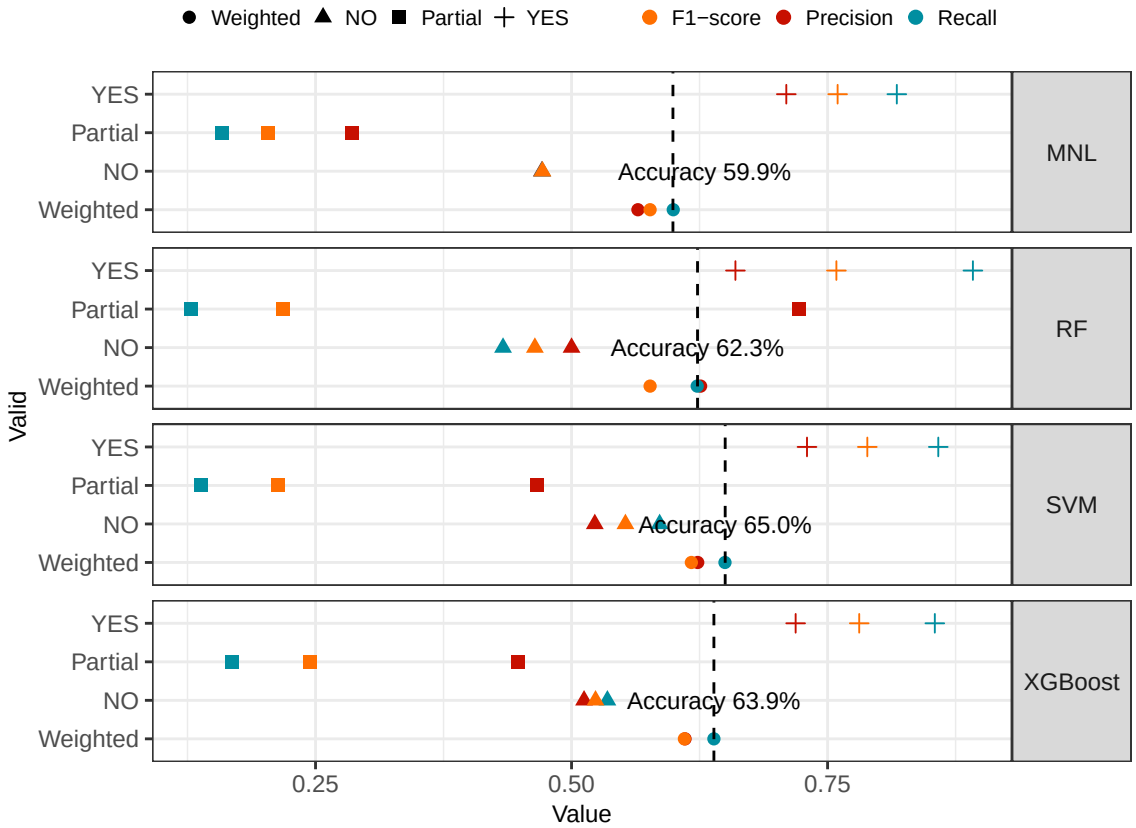


Figure 17: Overall results of the models

Table 10: Summary of evaluation metrics

		MNL	RF	SVM	XGBoost
YES	Precision	0.7097	0.6600	0.7299	0.7188
	Recall	0.8176	0.8919	0.8581	0.8547
	F1	0.7598	0.7586	**0.7888**	0.7809
Partial	Precision	0.2857	0.7222	0.4667	0.4474
	Recall	0.1584	0.1287	0.1386	0.1683
	F1	0.2038	0.2185	0.2137	**0.2446**
NO	Precision	0.4713	0.5000	0.5227	0.5122
	Recall	0.4713	0.4331	0.5860	0.5350
	F1	0.4713	0.4642	**0.5526**	0.5234
Weighted Average	Precision	0.5648	0.6260	0.6232	0.6108
	Recall	0.5993	0.6227	0.6498	0.6390
	F1	0.5767	0.5767	**0.6170**	0.6102

5.2.6 Only two classes

For partially valid complaints, arguments can be made to change their class label to valid or invalid. For example, in partially valid cases, CED Group still strives to resolve the issue even though the problem is not entirely CED Group's fault. To increase customer satisfaction, such complaints should still be prioritized and changed to valid. On the other hand, one could also argue that partially valid complaints are not really CED Group's fault and should therefore have lower priority and changed to invalid. Therefore, in this section, for experimentation, partially valid complaints are changed to valid and later to invalid, a SVM with linear kernel is trained to solve for the two classification problems. The optimal penalty parameter $C = 0.1$ is found for partially valid changed to valid, and $C = 0.01$ is found found for partially valid changed to invalid, both using a grid search. Figure 18 and 19 show the confusion matrices, and table 11 and 12 the evaluation metrics.

When changing partially valid complaints to valid, most of the complaints are classified as valid and the true positive invalid complaints decreased. When changing partially valid complaints to invalid, the false negative valid complaints increased. By looking at the ratio of true positive invalid complaints and false negative valid complaints, it is more preferable to change partially valid complaints to invalid as this ratio is higher.

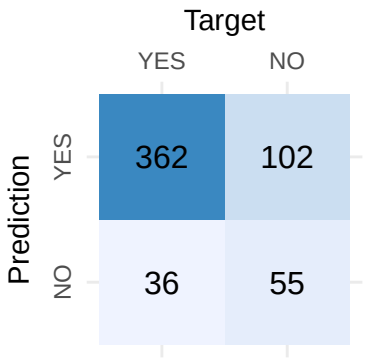


Figure 18: Confusion matrix SVM of predictions on test data with accuracy of 75.1% (Partially valid changed to valid).

Table 11: Evaluation metrics SVM test data (Partially valid changed to valid).

Precision	Recall	F1
0.7802	0.9095	0.8399

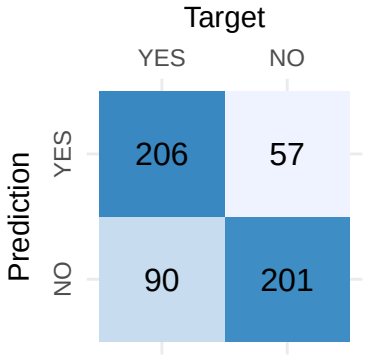


Figure 19: Confusion matrix SVM of predictions on test data with accuracy of 73.5% (Partially valid changed to invalid).

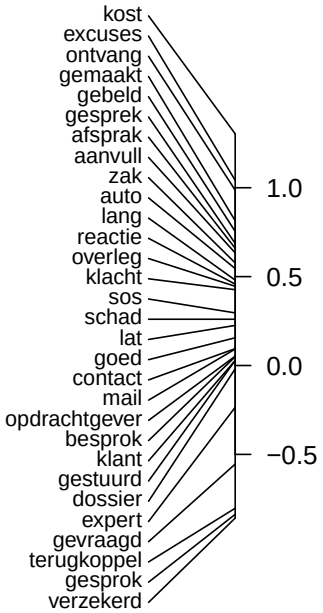
Table 12: Evaluation metrics SVM test data (Partially valid changed to valid).

Precision	Recall	F1
0.7833	0.6959	0.737

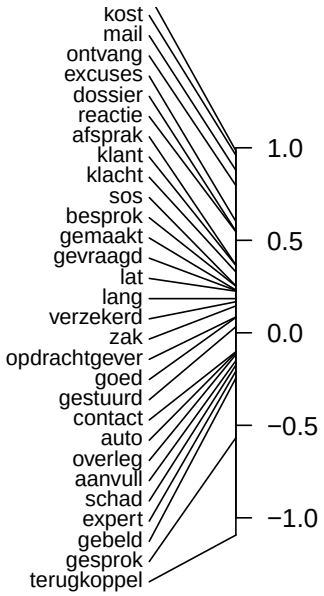
5.3 Deeper dive into the GloVe dimensions

From the results of the prediction models, we saw that top important features were elements from the high-dimensional GloVe embedding model (Figure 13 and 15). Unfortunately, those elements are normally not made for interpretation, it is however still interesting to attempt and identify what element of meaning is captured by the numbers of the embedding vectors to understand our data and models better. Therefore, this section attempts to interpret a few of those elements.

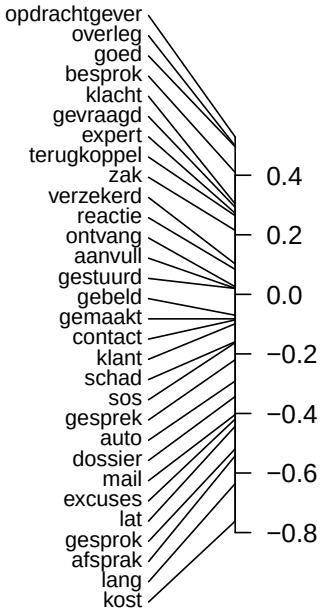
As explained before, the scores on the D GloVe dimensions capture the meaning of a word, and the gross vectors (dimensions) for each complaint is obtained by taking the mean value of all word vectors if the word is in the complaint. The top 30 frequent words in our complaints are selected to identify what meaning could be captured by some of the most important elements resulting from the models. Figure 20 shows the top 30 frequent words in our complaints and their value for dimension 57, 20, 38, and 3 as these were the most important dimensions resulting from both Random Forest and XGBoost.



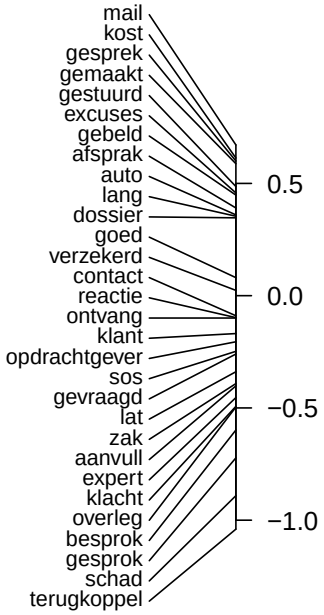
(a) Dimension 57



(b) Dimension 20



(c) Dimension 38



(d) Dimension 3

Figure 20: Elements of the high-dimensional GloVe embedding (a) 57, (b) 20, (c) 38, and (d) 3.

Dimension 57 shows that the words 'kost' (cost), 'excuses' (apologies), 'ontvang' (receive), and 'gemaakt' (incurred) are more related to each other in comparison to 'verzekerd' (insure), 'gesprok'

(spoken), and 'terugkoppel' (feedback) as they have high positive/negative values, this suggests that this dimension tries to capture the meaning of words related to communication and incurred costs by the insured person/party.

In comparison, dimension 20 shows that the words 'kost' (cost), 'mail' (mail), 'ontvang' (receive) are more related to each other in comparison to 'gesprok' (spoken), and 'terugkoppel' (feedback), while the value of 'gemaakt' (incurred) and 'verzekerd' (insure) are more close to zero. This suggests that this dimension tries to capture the meaning of words related to communication and incurred costs more generally, and less about the insured person/party.

Dimension 38 shows that the words 'opdrachtgever' (client), 'overleg' (discuss), and 'goed' (good) are more related to each other in comparison to 'kost' (cost), 'lang' (long), and 'afsprak' (appointment). This suggests that this dimension tries to capture the meaning of words relating to communication with the client.

Dimension 3 shows that the words 'mail' (mail), 'kost' (cost), 'gesprek' (conversation), and 'gemaakt' (incurred) are more related to each other in comparison to 'gesprok' (spoken), 'schade' (damage), and 'terugkoppel' (feedback). This suggest that this dimension also tries to capture the meaning of communication and incurred costs, but more related to damages.

6 Conclusion and Implications

6.1 Marketing and Managerial implications

In general, customer complaint management is an important aspect in marketing. To improve customer experience, satisfaction, trust, loyalty, and reputation, it is important to effectively address and resolve any issues that a customers may have with their products or services. The use of Natural Language Processing (NLP) and Machine learning (ML) can help businesses to more efficiently and effectively address and resolve customer complaints in the complaint management process. This research has focused on a specific part in improving this process by using real-world data from an insurance claims handler and NLP and ML to detect the validity of a complaint. Fake or invalid complaints can be a problem, as they can waste valuable time and resources which can potentially negatively affect a company. By being able to classify a complaint's validity, it is possible for businesses to allocate their resources more efficiently such as prioritizing legitimate complaints. In addition, having used real-world data provided a more accurate and relevant context for testing and evaluating the NLP and ML methods, which serves as a good indication of how such models perform in real world scenarios and could create value.

Our results have shown that the created models are able to classify valid complaints with a high Recall (above 0.8), that is, the proportion of actual valid complaints that was identified correctly. For invalid complaints, SVM and XGBoost are able to correctly classify more than the majority when looking at Recall. In theory, this means that implementing such model might increase efficiency in handling complaints as the majority of invalid complaints can be deprioritized and the number of false negative valid complaints is less than the number of true positive invalid complaints.

When partially valid complaints are also allowed to be deprioritized, further efficiency gain might be achieved as predictions made for this class mainly go into valid or invalid. Whether this theoretical gain holds in practice cannot be said with certainty as handling complaints includes various aspects which cannot be taken into account easily such as time and complexity. It is therefore recommended to experiment with caution and have analysts ready to keep track of the process.

6.2 Conclusion

The aim of this research has been to explore the feasibility of constructing models based on Natural Language Processing and Machine Learning to classify a complaint's validity into Valid, Invalid, or Partially Valid such that they can be prioritized or deprioritized in order to support the complaint handling process for CED Group.

In this research, three NLP techniques and four ML techniques have been examined for this purpose. First, Sentiment Analysis, Latent Dirichlet Allocation (LDA), and Global Vectors for Word Representation (GloVe) have been used to extract the sentiment score, LDA topic probabilities, and word embedding vectors from textual complaints. Subsequently, these are used as input features in a Multinomial Logistic Regression, a Random Forest, an XGBoost, and a Support Vector Machine (SVM) to classify a complaint's validity. The results have shown that the models are able to classify valid complaints effectively, while classification of invalid and especially partially valid complaints remain more challenging because of overlapping or ambiguous text descriptions. Following the feature importance measures, the high-dimensional GloVe embeddings seemed to be the most important features for our prediction models and performance of all models decreased when solely using LDA features, while solely using GloVe features only slightly decreased performance. In comparison to LDA and Sentiment Analysis, GloVe features are the least interpretable because of the many dimensions it consists of, an attempt has been made to interpret some of the most important dimensions comparing element values for specific words. Overall, the best performing model is the Support Vector Machine with the highest weighted average F1-score and accuracy followed by XGBoost, Random Forest and Multinomial Logistic Regression. This is in line with literature which suggests that SVM has shown to outperform many other models when using extracted features from text, yet differences were small in our research.

For future research, it is recommended to examine more ways for feature extraction such as using pre-trained NLP models and whether different ways of pre-processing the text will increase performance. In addition, this research only took the textual description and not the other given features in the dataset, further research could include more variables to examine whether predictive performance improves.

7 References

Barbado, R., Araque, O., Iglesias, C.A., 2019. A framework for fake review detection in online consumer electronics retailers. Information Processing & Management 56, 1234–1244.

Berger, J., Humphreys, A., Ludwig, S., Moe, W.W., Netzer, O., Schweidel, D.A., 2020. Uniting the tribes: Using text for marketing insight. Journal of Marketing 84, 1–25.

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. Journal of machine Learning research 3, 993–1022.

Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining. pp. 785–794.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine learning 20, 273–297.

Crain, S.P., Zhou, K., Yang, S.-H., Zha, H., 2012. Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond, in: Mining Text Data. Springer, pp. 129–161.

Crawford, M., Khoshgoftaar, T.M., Prusa, J.D., Richter, A.N., Al Najada, H., 2015. Survey of review spam detection using machine learning techniques. Journal of Big Data 2, 1–24.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Elmurngi, E., Gherbi, A., 2017. An empirical study on detecting fake reviews using machine learning techniques, in: 2017 Seventh International Conference on Innovative Computing Technology (INTECH). IEEE, pp. 107–114.

Faed, A., Chang, E., Saberi, M., Hussain, O.K., Azadeh, A., 2016. Intelligent customer complaint handling utilising principal component and data envelopment analysis (PDA). Applied Soft Computing 47, 614–630.

Figueira, Á., Oliveira, L., 2017. The current state of fake news: Challenges and opportunities. Procedia Computer Science 121, 817–825.

Galitsky, B.A., González, M.P., Chesñevar, C.I., 2009. A novel approach for classifying customer complaints through graphs similarities in argumentative dialogues. Decision Support Systems 46, 717–729.

Griffiths, T.L., Steyvers, M., 2004. Finding scientific topics. Proceedings of the National academy of Sciences 101, 5228–5235.

HaCohen-Kerner, Y., Dilmon, R., Hone, M., Ben-Basan, M.A., 2019. Automatic classification of complaint letters according to service provider categories. Information Processing & Management 56, 102102.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The elements of statistical learnin. Cited on 33.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., others, 2003. A practical guide to support vector classification.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An introduction to statistical learning.

Springer.

Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., Zhao, L., 2019. Latent dirichlet allocation (LDA) and topic modeling: Models, applications, a survey. Multimedia Tools and Applications 78, 15169–15211.

Jia, S., Zhang, X., Wang, X., Liu, Y., 2018. Fake reviews detection based on LDA, in: 2018 4th International Conference on Information Management (ICIM). IEEE, pp. 280–283.

Jiang, M., Cui, P., Faloutsos, C., 2016. Suspicious behavior detection: Current trends and future directions. IEEE intelligent systems 31, 31–39.

Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features, in: European Conference on Machine Learning. Springer, pp. 137–142.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., Brown, D., 2019. Text classification algorithms: A survey. Information 10, 150.

Krishna, G.J., Ravi, V., Reddy, B.V., Zaheeruddin, M., Jaiswal, H., Teja, P.S.R., Gavval, R., 2019. Sentiment classification of indian banks' customer complaints, in: TENCON 2019-2019 IEEE Region 10 Conference (TENCON). IEEE, pp. 429–434.

Kwartler, T., 2017. Text mining in practice with r. John Wiley & Sons.

Libai, B., Bart, Y., Gensler, S., Hofacker, C.F., Kaplan, A., Kötterheinrich, K., Kroll, E.B., 2020. Brave new world? On AI and the management of customer relationships. Journal of Interactive Marketing 51, 44–56.

Liu, B., 2012. Sentiment analysis and opinion mining. Synthesis lectures on human language technologies 5, 1–167.

Liu, Y., Wan, Y., Su, X., 2019. Identifying individual expectations in service recovery through natural language processing and machine learning. Expert Systems with Applications 131, 288–298.

Medhat, W., Hassan, A., Korashy, H., 2014. Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal 5, 1093–1113.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems 26.

Mohammad, S.M., Turney, P.D., 2013. Nrc emotion lexicon. National Research Council, Canada 2, 234.

Mukherjee, A., Venkataraman, V., Liu, B., Glance, N., others, 2013. Fake review detection: Classification and analysis of real and pseudo reviews. UIC-CS-03-2013. Technical Report.

Newman, D., Lau, J.H., Grieser, K., Baldwin, T., 2010. Automatic evaluation of topic coherence, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 100–108.

Pennington, J., Socher, R., Manning, C.D., 2014. Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 1802. Deep contextualized word representations. arXiv 2018. arXiv preprint arXiv:1802.05365 12.

Porter, M.F., 2001. Snowball: A language for stemming algorithms.

Rinker, T., 2018. Sentimentr.

Tian, X., Vertommen, I., Tsiami, L., Thienen, P. van, Paraskevopoulos, S., 2022. Automated customer complaint processing for water utilities based on natural language processing-case study of a dutch water utility. Water 14, 674.

Wright, M.N., Ziegler, A., 2015. Ranger: A fast implementation of random forests for high dimensional data in c++ and r. arXiv preprint arXiv:1508.04409.

Yang, W., Tan, L., Lu, C., Cui, A., Li, H., Chen, X., Xiong, K., Wang, M., Li, M., Pei, J., others, 2019. Detecting customer complaint escalation with recurrent neural networks and manually-engineered features, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers). pp. 56–63.

Zhou, X., Zafarani, R., Shu, K., Liu, H., 2019. Fake news: Fundamental theories, detection strategies and challenges, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. pp. 836–837.

8 Appendix

Table 13: Topics and terms extracted using LDA

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**
uur	verzekerd	mevrouw	verzekerd	eigenar
gebeld	grag	goed	expert	uitleg
bell	contact	lang	opdrachtgever	gegev
wacht	verzekerar	lat	schad	gesprok
terug	bericht	contact	klacht	opkoper
lang	opgenom	vond	overleg	dagwaard
kreg	hierbij	stond	gebeld	contra
gestan	stur	duurd	aanvull	tegenpartij
ker	hierop	wild	besprok	overleg
contact	verwacht	geregeld	akkoord	hoogt

Topic 6	**Topic 7**	**Topic 8**	**Topic 9**	**Topic 10**
contact	auto	verzekerd	emn	verzekerd
sos	motor	expert	gat	klacht
alarmcentral	nederland	contact	dossier	gebeld
ziekenhuis	transport	reparateur	ligt	expert
nederland	eijck	schad	jan	gesprek
vrag	afgeleverd	opgenom	blijv	tevred
reisverzeker	wek	gesprok	ligg	goed
huis	kost	wild	rapport	afgehandeld
regel	repatrir	gan	forensic	doorgesprok
lat	vervang	goed	opdracht	gesprok

Topic 11	**Topic 12**	**Topic 13**	**Topic 14**	**Topic 15**
opdrachtgever	lang	schad	clint	verzekerd
zak	klant	mogelijk	klacht	opdrachtgever
expert	bon	kom	ced	expert
dossier	dossier	ten	aangegev	terugkoppel
gesprok	sos	camping	das	gevraagd
lat	wachttijd	herstel	gedan	reactie
besprok	bol	blek	client	contact
aangegev	klacht	discussie	gebeld	gegev
tm	mail	jar	duidelijk	gezond
onderzoek	abn	gehel	dossier	gebeld

Topic 16	Topic 17	Topic 18	Topic 19	Topic 20
dochter	problem	vergoed	afsprak	dag
bloem	tussenperson	kost	gemaakt	volgend
aon	schad	btw	verzekerd	telefon
vader	waardor	factur	nieuw	kom
gesprek	lang	incl	gebeld	vakantie
verkeerd	doorlooptijd	nota	bereik	asr
gedan	betaald	bedrag	diver	zer
blijkt	opgepakt	akkoord	bezoek	afgeslot
gesprok	gebruikt	goed	telefonisch	compensatie
opd	geduurd	nl	contact	frankrijk

Topic 21	Topic 22	Topic 23	Topic 24	Topic 25
mail	klacht	klant	excuses	gat
zie	behandel	mail	aangebod	dossier
klacht	ced	dossier	gemaakt	communicatie
bijlag	betreft	gestuurd	gesprok	geval
reactie	dossier	lat	gebeld	tijd
verzoek	genom	geslot	aanvaard	goed
ontvang	teammanager	wet	goed	informatie
cas	ton	asr	uitgelegd	voorkom
waarin	onderzoek	ontvang	geaccepteerd	medewerker
graf	forensic	vrag	communicatie	verwacht

Topic 26	Topic 27	Topic 28	Topic 29	Topic 30
contact	gesprek	rapport	klant	klant
opgenom	melding	expert	dekking	sos
klager	gezegd	opdrachtgever	polis	camper
zak	ardanta	aanvull	pech	taxi
besprok	gegeven	verzond	hulp	dag
tp	collega	gemaakt	geholp	international
telefonisch	geeft	rapportag	dossier	garag
aangegev	aangegev	dossier	sos	kost
tel	medewerker	besprok	vind	hotel
opnem	polis	dispatch	auto	geregeld

Topic 31	Topic 32	Topic 33	Topic 34	Topic 35
zak	expert	gesprek	mail	kost
contra	reactie	actie	schad	betal
waarbij	opdrachtgever	mevrouw	gestuurd	vergoed
behandelar	gevraagd	gebeld	klant	ohra
info	teruggekoppeld	nodig	foto	akkoord
nn	klacht	ondernom	dossier	gedan
duidelijk	verzocht	verder	ontvang	betaald
overleg	vrag	schad	voertuig	gat
afgestemd	terugkoppel	wertenbroek	ohra	bedrag
standpunt	aanvull	verbind	ber	gemaakt

Topic 36	Topic 37	Topic 38	Topic 39	Topic 40
verzekerd	klacht	opdracht	pat	auto
ced	nn	gebeld	sos	klant
hersteller	dossier	gan	repat	berger
repair	claim	aangegev	klacht	garag
expert	stuk	direct	zie	vab
herstel	gezien	mevrouw	nl	lat
contact	ga	uitgelegd	cli	mee
schad	tijd	mee	rv	reparatie
opdrachtgever	uitgebreid	juni	collega	gebracht
ingeschakeld	gegrond	werk	gemaild	voertuig

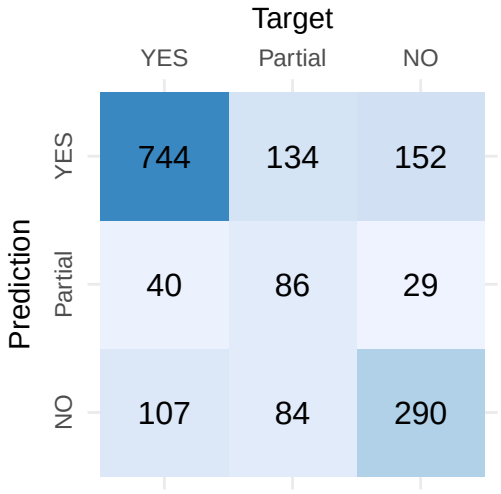


Figure 21: Confusion matrix MNL of predictions on train data with accuracy of 67.2%.

Table 14: Evaluation metrics MNL train data.

	Precision	Recall	F1
YES	0.7223	0.835	0.7746
Partial	0.5548	0.2829	0.3747
NO	0.6029	0.6157	0.6092
Weighted	0.6580	0.6723	0.6549

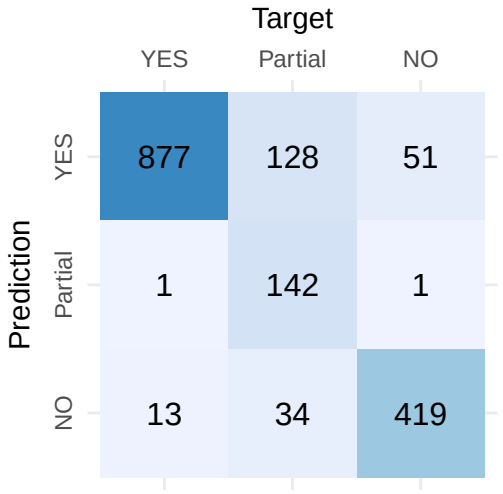


Figure 22: Confusion matrix Random Forest of predictions on train data with accuracy of 86.3%.

Table 15: Evaluation metrics XGBoost train data.

	Precision	Recall	F1
YES	0.8305	0.9843	0.9009
Partial	0.9861	0.4671	0.6339
NO	0.8991	0.8896	0.8943
Weighted	0.8783	0.8632	0.8503

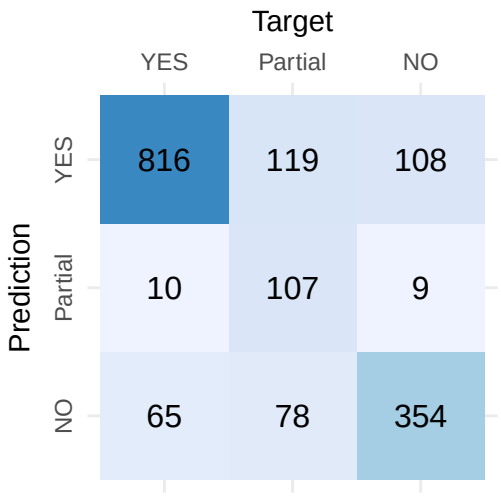


Figure 23: Confusion matrix XGBoost of predictions on train data with accuracy of 78.3%.

Table 16: Evaluation metrics XGBoost train data.

	Precision	Recall	F1
YES	0.7824	0.9158	0.8438
Partial	0.8492	0.352	0.4977
NO	0.7123	0.7516	0.7314
Weighted	0.7748	0.7665	0.7489

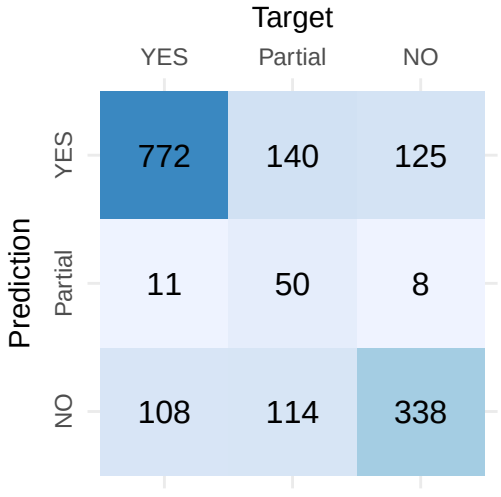


Figure 24: Confusion matrix SVM of predictions on train data with accuracy of 69.6%.

Table 17: Evaluation metrics SVM train data.

	Precision	Recall	F1
YES	0.7445	0.8664	0.8008
Partial	0.7246	0.1645	0.2681
NO	0.6036	0.7176	0.6557
Weighted	0.7010	0.6963	0.6626