Identifying effective affective email responses;

Predicting customer affect after email conversation

Erwin Huijzer

Master Thesis Business Analytics

VU VRIJE UNIVERSITEIT AMSTERDAM

anchormen
expert in data excellence

Identifying effective affective email responses;

Predicting customer affect after email conversation

Erwin Huijzer

Master Thesis Business Analytics

Vrije Universiteit Amsterdam

Faculty of Science

Business Analytics

De Boelelaan 1081a

1081 HV Amsterdam

Host organization:

Anchormen

Pedro de Medinalaan 11

1086 XK Amsterdam

July 2017

Preface

This thesis is written to conclude the Business Analytics Master programme. The external master project is usually carried out within a business, industry or research facility other than the departments of Mathematics and Computer Science.

This study is executed at Anchormen, a Groningen and Amsterdam-based IT-consultancy firm. Anchormen delivers services in the fields of big data, data science, artificial intelligence, data architecture, engineering, training and support.

The objectives of this study are to (1) set up domain specific sentiment detection machine learning models to detect sentiment and emotion in emails, (2) compare performance of these models with generic models for sentiment analysis (3) create a model to predict customer affect after an email conversation with CS.
The outcome of this study could aid in automatically generating email responses and measuring the performance of a chatbot platform.

I would like to thank my supervisor Davide Ceolin from the Faculty of Science of Vrije Universiteit Amsterdam for giving me helpful tips in tackling the various steps of the research process and giving useful feedback on my report. Furthermore, I would like to thank Anchormen, and in particular my supervisor Chris Pool, for giving me the opportunity and support to investigate this interesting subject. Also, I would like to thank Bram Gorissen from the Faculty of Science of Vrije Universiteit Amsterdam for being my second reader.
Last but not least, I would like to thank my family whose support made it possible to follow my aspiration of becoming a data scientist.

Erwin Huijzer
July 2017
Amsterdam

Summary

For many customers, a website form or email is the preferred channel to contact a company regarding questions and complaints. Handling these incoming messages is often a massive task for the company's Customer Support department. While handling the messages could have major influence of the customer satisfaction.

The purpose of this thesis is to perform affect analysis in the email domain. Focus will be on the interaction between Customer Support (CS) and a customer through email and the effect that CS email responses have on the customer sentiment. The main research question is: Which aspects of a CS email response affect customer sentiment? This research question can be divided into the following sub-questions; (1) what sentiment can be detected in customer emails, (2) does a domain specific sentiment detection machine learning model trained on a small set of emails, outperform a general model for sentiment analysis, (3) do CS response email features have predictive value for sentiment of a customer?

The data available for this study were emails originating from both UK based customers and Customer Support of a sportswear multinational. The first main step was annotating emails with sentiment (Pos, None, Neg, Mix) and emotions (Anger, Disgust, Fear, Joy, Sadness) before sentiment analysis and affect analysis could be done. Each email was annotated by three annotators out of a total of five annotators who each annotated different portions of the email data set. After annotation, as the next step, multiple models (Neural Net, Naive Bayes, Support Vector Machines, Random Forest, RAkEL, voting ensemble) were used to perform sentiment analysis and affect analysis. The analysis was approached as a classification problem, classifying sentiment and emotions for each email or email conversation between customer and CS.

The findings in this study show that voting ensembles of Neural Net, Random Forest and optionally Support Vector Machines can successfully identify Sentiment (kappa 0.45), Anger (kappa 0.51), Disgust (kappa 0.43) and Joy (kappa 0.61) in the domain they were trained for. Also, Sadness (kappa 0.36) and Fear (kappa 0.14) can be identified to a lesser extent. In this study, each of the domain specific models on based on a small set of annotated emails significantly outperforms ($p<0.01$) the commercial IBM Natural Language Understanding API which was trained on millions of news sources.

Predicting customer affect proofs to be difficult. While the models for Sentiment (Random Forest), Joy and Sadness (Naive Bayes) perform significantly ($p\leq0.01$) better than the benchmark, Anger ($p=0.13$) and Disgust ($p=0.07$) models do not. But, performance across all models could be considered low with an accuracy of 0.49 for Sentiment and F1 for emotions Anger, Disgust, Joy and Sadness ranging from 0.30 till 0.50.
Feature importance analysis revealed the following CS features to be important in affect analysis: day of the week (Anger), length of the message (Disgust), word-based tf-idf (Joy, Sadness), and character-based tf-idf (Sentiment). However, the machine learning models used, do not allow comprehending how these features exactly influence predictions.

While the email sentiment annotation process was not the primary focus of this study, still it was an important part as it supplied the labels for supervised sentiment analysis and affect analysis. The email sentiment annotation process revealed that humans find it difficult to agree on emotions displayed in email text. Future research should not underestimate the effort needed to gather high quality annotated data.

Results from this thesis could be used as a basis for quality improvement and automation of email handling of Customer Service response emails. Modelled probabilities of certain customer emotions to occur could be used as a quality measure of a manually drafted or automatically generated email.

Content

1 Introduction

Customers contact a company’s Customer Service for all kinds of reasons such as; questions or complaints on existing orders or purchased product, inquiries on payments, questions on website usage. For many customers, a website form or email is the preferred channel to contact a company. Handling these incoming messages is often a massive task for the company’s Customer Support (CS) department. Still, each customer contact offers the opportunity to improve the customer relation. To achieve this, great care needs to be spent on constructing an email that is effective in removing or at least reducing anger, disgust and other negative emotions. Or maybe even spark some joy with the customer. Gaining insight in customer emotions as expressed in email and assessing which affect is caused by the CS response email is essential in this process.

Throughout this thesis, ‘sentiment’ will refer to the polarity of the message: positive, negative, mix and neutral. The term ‘emotion’ refers to concepts such as anger, disgust, fear, joy and sadness. ‘Affect’ refers to sentiment or emotion induced by reading text.

The field of sentiment analysis and emotion detection in texts has been widely studied in the past few decades. Most of the research focuses on sentiment from writer’s point of view, while only few focus on reader’s point of view. Affect analysis has centred on the news domain (Lin et al., 2008; Yang et al., 2009, Ye et al., 2012). Affect analysis on emails is still a domain waiting to be studied.

The purpose of this thesis is to perform affect analysis in the domain of emails. Focus will be on the interaction between Customer Support and a customer through email and the effect that CS email responses have on the customer sentiment. The main research question is: Which aspects of a Customer Service email response affect customer sentiment? This research question can be divided into the following sub-questions; (1) what sentiment can be detected in customer emails, (2) does a domain specific sentiment detection machine learning model trained on a small set of emails, outperform a general model for sentiment analysis such as IBM Natural Language Understanding, (3) do CS response email features have predictive value for sentiment of a customer?

Getting insight in the expected effect of a Customer Service response email on customer sentiment could open the possibility to tailor the response in such a way that a certain desired customer affect can be achieved. Furthermore, predicted probabilities of customer sentiment and each emotion, can be used to measure CS response email quality. Besides quality improvement, also automation of email handling could be an area of application. Instead of manual email handling by CS employees, a significant reduction in manual handling can be achieved by automating email responses by means of an e-mail bot. To allow full automated email handling, customer sentiment needs to be considered upon email generation. A hybrid approach could also be an option where emails are automatically generated and only sent after a manual check.

In this study, machine learning techniques were applied on emails from both customers and Customer Support of a sportswear multinational. First, emails were annotated with sentiment and emotions to allow supervised machine learning models. Next, multiple models (neural net, naive Bayes, support vector machines, random forest, RAkEL, voting ensemble) were used to perform sentiment analysis and affect analysis. The analysis was approached as a classification problem, classifying sentiment and emotions for each email or email conversation between customer and CS.

This thesis has been organised in the following way. Chapter 2 presents related work to sentiment annotation, sentiment analysis and affect analysis. Next, chapter 3 gives some background information on the main machine learning models in this paper. The characteristics of the email dataset used can be found in Chapter 4. Chapter 5 describes the methods and data used. Results of annotation, sentiment analysis and affect analysis can be found in Chapter 6. Chapter 7 contains discussions. Finally, chapter 8 reports conclusions and further research.

2 Related work

2.1 Framework for sentiment and emotions in text

Sentiment and emotion analysis in text has been a very active research area in the last decade. While detecting sentiment and emotions is an extremely challenging task in itself, it starts with choosing the framework around sentiment and emotions.

The most prominent research in the field of emotions has been by Ekman starting in the 1960s. Ekman (1973) stated that six basic emotions can be recognized: Anger, Disgust, Fear, Joy, Sadness, and Surprise. These six emotions have been a popular basis for annotating emotions many domains such as news sentences (Bhowmick et al., 2010) and twitter messages (Roberts et al., 2012; Bosco et al., 2013). The latter two studies included a seventh emotion: 'Love'. Mohammad & Turney (2013) in their NRC emotion lexicon adopt the model of Plutchik (1980) which also includes categories anticipation and trust on top of the Ekman's six emotions. To minimize the annotation effort and to align with categories used in the baseline as set using IBM Natural Language Understanding, Ekman's model without the somewhat ambiguous Surprise category is adopted in this thesis

A sentiment framework appears to be the simpler task compared to an emotions framework. Commonly used sentiment labels are: Positive, Negative, Neutral. Mohammad (2016) proposes to use a mixed category to cover different sentiments towards different targets of opinion and to use a separate category to capture 'expressions of sarcasm, ridicule, or mockery'. Wiebe et al. (2005) describe sentiment as a much more complex model, distinguishing aspects as intensity, expression intensity, insubstantial and attitude type. In this thesis, annotation is done on email level. This increases the chance of mixed sentiment being present. The model of Mohammad (2016) is suited for this. Furthermore, adding an irony category as proposed by Mohammad will make annotation a bit simpler since it can take away confusion on how to annotate. Therefore, the five-category sentiment model of Mohammad is adopted.

2.2 Annotating emotions

The quality of annotation is highly dependent on clear annotation guidelines (Mohammad, 2016). Byron (2008) argues that "Email characteristics make miscommunication likely,… receivers often misinterpret work emails as more emotionally negative or neutral than intended". Mohammad (2016) sees additional causes that can give difficulties when annotating sentiment and emotions in text: "Speaker's emotional state; Success or failure of one side w.r.t. another; Neutral reporting of valanced information; Sarcasm and ridicule; Different sentiment towards different targets of opinion; Precisely determining the target of opinion; Supplications and requests; Rhetorical questions; Quoting somebody else or retweeting". However, not all the issues mentioned apply to annotating emails. Still, this summary indicates that agreement between annotators is an issue.

Mohammad (2016) proposes a simple sentiment annotation questionnaire to tackle most of the annotations issues. This questionnaire is used in this thesis and extended to also cover emotions. As the questionnaire is simple, it does not have extensive descriptions on exactly what each emotion comprises. It relies heavily on the annotator's interpretation of the emotions. To give some support to the annotators, related words per emotion are listed based on Goleman (1996).

Artstein and Poesio (2008) emphasize that agreement between annotators needs to be present to show the validity of the annotation scheme and results. Much research has been done on annotator agreement. To quantify results, many different metrics have been used across studies, kappa (Aman and Szpakowicz, 2007; Wiebe et al., 2005), precision, recall, F1 (Calvo et al., 2013, Gupta et al., 2013; Moraes et al., 2013, Xia, 2011), Krippenhoff's alpha (Bermingham, & Smeaton, 2009), accuracy (Wang et al., 2014; Moraes et al., 2013) and some less common statistics like IAA rate (Volkova et al., 2010) or MASI (Aman and Szpakowicz, 2007). As can be seen many statistics exist and no statistic is clearly preferred above all others.

Cohen's kappa (1960) could be considered as one of the most common metrics of agreement. The metric can have a maximum value of 1 indicating full agreement, while 0 or lower indicates no agreement. Since it is highly dependent on the data at hand, no single threshold can be determined to indicate whether agreement is sufficient or not. While McHugh (2012) does state ".", Landis and Koch (1977) give a milder interpretation of the kappa values, calling 0.41 moderate agreement.

2.3 Sentiment and affect analysis

The past twenty years have seen increasingly rapid advances in the field of sentiment analysis and emotion detection in texts. This development is driven by the explosive growth of digital messages on (social) media. Tweets, blogs, reviews, news articles can offer a wealth of information providing they can be interpreted correctly. Several studies focussed on detecting sentiment in email (Gupta et al., 2013; Mohammad and Yang, 2011). Gupta (2013) focused specifically on Customer Service emails and showed that emotions can be detected in emails using machine learning.

While most research on sentiment analysis focuses on writer's point of view, only little research focuses on the reader's point of view. Studies of Bradley and Lang (1999) and Strapparava and Valitutti (2004) targeted affect of words only, showing specific words can be labelled with affect labels. The only studies focussing on affect in full texts, are all in the news domain (Lin et al., 2008; Yang et al., 2009, Ye et al., 2012). Affect analysis on emails is a domain that has not been investigated. It is not known if sentiment and emotion induced by email can be detected and which features are most important in predicting these sentiment and emotions.

Affect analysis research of Yang et al. (2009) is limited to statistical analysis confirming a link between the text and reader emotion. Lin et al. (2008) and Chang et al. (2015) do use popular machine learning models SVM and Naive Bayes to extract and predict reader emotion. Bhowmick (2009) and Ye et al. (2012) demonstrate that the multilabel RAkEL method can give satisfactory results. To have the biggest chance to capture a potentially complex relation between email and reader emotion, this study uses SVM, Naive Bayes and RAkEL and on top of that some more popular models like Random Forest, Neural Net and voting ensemble. A high-level description of all models is given in sub-section 3.3.

One of the complexities a machine learning model may need to overcome in sentiment and affect analysis, is class imbalance in the data. How imbalance can be handled depends on how the data is used. In case of using the data as single label, oversampling the minority class or undersampling the majority class are the standard methods. For multilabel, sampling is not straightforward. Charte et al. (2015) describe multiple approaches. In this thesis the label power approach is used which treats each unique combination of multiple labels as a single label. These combined labels are then oversampled.

3 Background information

In this thesis, the kappa metric and various machine learning models are used:

- Neural Net
- Support Vector Machines
- Random Forest
- Naive Bayes
- Soft voting ensemble
- RAkEL

For readers not familiar with this metric or the machine learning models, this section provides some high-level background information.

3.1 Kappa metric

Kappa is a well-known and broadly used metric for inter-annotator agreement on categorical data and was first introduced by Cohen (1960). It was presented as an alternative to a simple agreement measure like percentage of agreement. The drawback of this simple measure is that some agreement is due to chance. Fewer categories or imbalanced classes will result in higher percentage agreement due to chance. Therefore, it is much more informative to adjust observed agreement to cope with chance. Cohen proposed the κ metric according to equation (1).

$$\kappa = \frac{A_o - A_e}{1 - A_e} \tag{1}$$

Here A_o is observed agreement and A_e is expected agreement. The intuition behind this formula is that it presents a ratio of found agreement that exceeds chance versus the maximum to be found agreement that exceeds chance. The S metric (Bennett et al., 1954), which is similar to the kappa metric, uses an expected agreement based on all categories being equally likely. Another similar metric, the π metric (Scott, 1955), assumes that category probability can differ per category and is the same for all annotators. The kappa metric however assumes that each annotator has its own probability distribution across categories reflecting its individual bias. The probability distribution per annotator is estimated using the actual annotation distribution.

Kappa has a maximum value of 1 and can (theoretically) have a very large negative value. In case kappa is negative, it means that observed agreement is even smaller than expected agreement. The maximum kappa value of 1 is only achieved in case of complete agreement. Which kappa value represents sufficient agreement is still under debate. Landis and Koch (1977) gave the following guidelines: < 0 no agreement, 0–0.20 slight, 0.21–0.40 fair, 0.41–0.60 moderate, 0.61–0.80 substantial, and 0.81–1 almost perfect agreement.

Below an example is given to illustrate the calculation of the kappa metric:

		Annotator 2		
		Anger	**No Anger**	**Total**
Annotator 1	**Anger**	20	20	40
	No Anger	10	50	60
	Total	30	70	100

Observed agreement: $A_o = diagonal\ values = 0.2 + 0.5 = 0.7$

Expected agreement: $A_e = Annotator\ 1\ Anger\ prob \cdot Annotator\ 2\ Anger\ prob + Annotator\ 1\ No\ Anger\ prob \cdot Annotator\ 2\ No\ Anger\ prob =$ $0.4 \cdot 0.3 + 0.7 \cdot 0.6 = 0.54$

Calculated kappa: $\kappa = \frac{A_o - A_e}{1 - A_e} = \frac{0.7 - 0.54}{1 - 0.54} = 0.35$

3.2 Text feature extraction

3.2.1 Tf-idf

Tf-idf (Term Frequency - Inverse Document Frequency) is a measure that gives an indication of the importance of a term (character or word n-gram) to a document in a corpus. It is the product of term frequency (tf) and inverse document frequency (idf).

Term frequency of term i in document j is the ratio of f_{ij} defined as the number of occurrences of term i in document j over the maximum on any term k in that same document. This results in a value of 1 for the most frequent term (see formula 2)

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}} \quad (2)$$

Inverse document frequency of term i in a corpus is defined as log^2 of the ratio of the total number of documents in the corpus N over the number of documents that contain a term i defined as n_i (see formula 3). IDF reduces the weight of terms that appear frequently across the corpus.

$$\mathrm{IDF_i} = \log^2\left(\frac{\mathrm{N}}{\mathrm{n_i}}\right) \quad (3)$$

The product $TF_{ij} \cdot IDF_i$ represents the tf-idf score for term i in document j. The highest tf-idf scores represent characteristic terms. Using this method, tf-idf scores can be calculated for every n-gram in the email corpus for every email.

3.2.2 Doc2Vec

Doc2vec is an extension of the Word2vec method. Word2vec is a method to produce word embeddings. The method uses neural networks to model each unique word from a corpus as a vector in a space. Mikolov et al. (2013) demonstrated that the word vectors are positioned in the vector space such that words that share common contexts in the corpus are near each other in the space. As an extension to Word2vec, Le and Mikolov (2014) developed Doc2vec which modifies the Word2vec algorithm to construct embeddings for larger blocks of text, such as sentences and entire documents.

3.3 Machine learning models

A number of machine learning models have been used in this thesis. Most models used are very common. In the sub-section only a high-level description will be given of these models. Further details about every model can be found in the papers that are referenced in the subsequent paragraphs.

3.3.1 Naive Bayes

Naive Bayes are probabilistic classifiers based on Bayes' theorem with naive independence assumptions between the features hereby disregarding any possible correlation between features. It works with conditional probabilities, determining what the probability of an observed event is given a certain condition. It is a supervised learning method using feature vectors and class labels. Naive Bayes has the advantage that it only requires few data records for training the classifier. In this thesis, a Gaussian naive Bayes is used. It handles continuous data under the assumption the values for each class are Gaussian distributed.

3.3.2 Support Vector Machines

The SVM algorithm as introduced by Vapnik (1995) is a supervised learning model. An SVM model presents data as points in space where its optimization target is to separate two classes by a hyperplane in such a way that the gap is as wide as possible. To predict classes of new data, this data is mapped into that same space. The prediction is determined by the side of the hyperplane the data point is located. SVM is a linear classifier, but SVM can also perform non-linear classification using the kernel trick. This maps the data into a higher dimensional space. In this thesis, the radial basis function kernel is used.

3.3.3 Neural Network

Neural networks are inspired on the brain and development already started in the 1950’s. In this thesis, a multilayer perceptron (MLP) neural network is used. A MLP is constructed from many simple processing units called neurons. A neuron can receive inputs from other neurons and based on a certain activation function it can fire a signal to other neurons. In an MLP, neurons are organized in multiple layers; input layer, hidden layer and an output layer. In this thesis, a single hidden layer was used. Training an MLP requires labelled data. Through an iterative process called backpropagation any errors in prediction of the training set are fed back into the system in backward direction, updating weights of neuron signals in the process.

3.3.4 Random Forest

The Random Forest method was first proposed by Ho (1995). Random forest is an ensemble method that uses a large amount of decision trees. The Random Forests method makes a random selection of features for each decision tree. This way variance in predictions is introduced between various decision trees. In case of classification, a label is assigned to the majority label across all decision tree predictions. Random forest corrects any overfitting of decision trees and reduces variance. An advantage of the method is that it usually generalizes well.

3.3.5 Soft voting ensemble

A voting ensemble is model that can combine multiple classifiers through a voting principle. A common principle is majority voting where predictions of all input classifier are counted and the majority label is assigned. A variant on this is soft voting, where all prediction probabilities are averaged and a certain label is assigned when the probability is above 0.5. This way models that result in more reliable predictions with higher probabilities have more weight in determining the final label. An advantage of this type of ensemble is that no train and test phase is needed and therefore no train and test data is needed. This allows for the full data set to be used for training and testing of the input classifiers.

3.3.6 RAkEL

RAkEL is an abbreviation of RAndom k-labELsets and is a method introduced by Tsoumakas and Vlahavas (2007). It is an ensemble method for multilabel classification such as multiple emotion labels per email or conversation as is the case within this thesis. The method uses supervised learning with label power classifiers. A label power classifier treats each unique combination of labels as a single label and effectively builds a single label classifier. RAkEL selects multiple subsets of k-labels and models a label power classifier on each subset. This step is performed in multiple iterations, randomly selecting different subsets each time. The ultimate step is combining the predictions via voting. Assigning a label when an average voting is above a set threshold.

4 Data

For investigating the impact of an email response on a client's sentiment and emotion, a dataset is selected from a UK based Customer Service department of a multinational specialized in sporting goods. The data set contains all incoming and outgoing emails from two different initiating channels: contact form, email. All emails are in English, apart from 2 emails which were removed. The emails span a period of more than three months (1-8-2016 till 21-11-2016). One would expect the numbers to be roughly the same since any incoming customer email should result in a response from CS department. However, the number of incoming emails in the set is much larger than the number of outgoing emails. The discrepancy is caused by emails with no text body. For unknown reasons, a large part of the incoming email records (23%) do not contain any text body. Excluding these 'no text body'-emails, the number of incoming and outgoing emails are balanced as expected. Table 1 shows some key characteristics of the email dataset.

Total Number of emails	88.829		
Incoming		50.433	
Incoming (with text body)			*38.798*
Incoming (no text body)			*11.635*
Outgoing		38.396	
Total number of threads (excl no text body emails)	26.674		

Table 1: Key characteristics of email dataset. Incoming and outgoing are balanced. Each incoming email receives and outgoing response email.

Upon reception of an email, a unique case number is assigned. Any follow-up emails are assigned the same case number, this way creating the possibility to analyse complete email threads. The number of emails (excluding emails with empty text body) in a thread ranges in this data set from 1 up to as much as 49. See Figure 1 for the distribution of number of emails in a thread. Since the email dataset has specific start and end times, the beginning or end of an email thread can be missing. Together with spam emails, this is the main reason of single-email threads. While most of threads contain two emails, the main interest in this study goes out to threads with three or more emails.

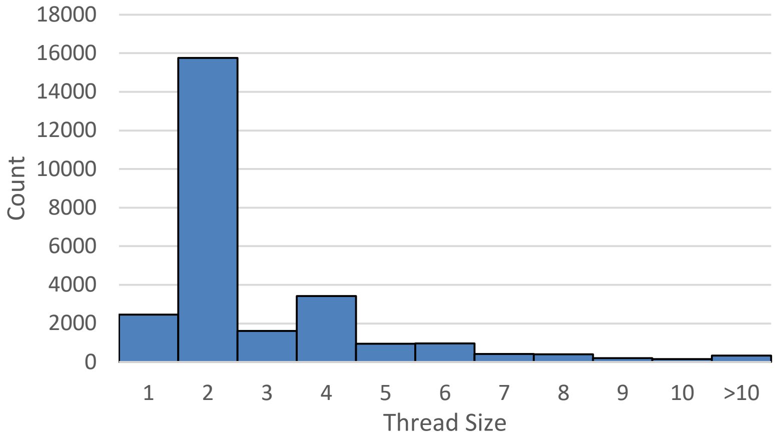


Figure 1: Distribution of thread size. Most threads are of size 2, containing single customer email and CS response. For thesis only threads with size>2 are used that contain client-CS-client conversations.

Only in threads of at least three emails, a “client – customer support – client” conversation can be seen. These conversations are necessary to be able to study the impact of CS email response on customer sentiment and emotion. If an email thread consists of five or more emails, each “client – customer support – client” sequence is counted as a separate conversation. A specific client email can therefore be part of two conversations; as final email for one conversation and as start email for the next. From Table 2 it can be concluded that 2.727 incoming emails are used in more than one conversation (2*9.500 – 16.273).

Number of emails part of conversations	25.773	
Incoming (used as start & end)		*16.273 (2.727)*
Outgoing		*9.500*
Number of conversations	9.500	
Number of threads with conversations	6.393	

Table 2: Characteristics of email conversations

The raw dataset contains a single record per email. Every record consists of the following fields:

Country	: Origin of the helpdesk. In this case always UK.
Contact Reason	: 11 distinct categories like Company Information, Existing Orders / Purchased Product, Payment / Credit
Case Origin	: Contact us form or Email
Is Incoming	: 0 (outgoing) or 1 (incoming)
Status	: New, Read, Draft, Sent, Replied
Message Date	: Date of format m/d/yyyy hh:mm
‘Created By: Full Name’	: System (incoming) or department name (outgoing)
Subject	: Single line of text
Text Body	: Email content including embedded URL links or any forwarded emails
Case Number	: 7-digit number

5 Research method and data gathering

This section describes the methods used in this study. In order to identify sentiment in emails and to predict customer affect, supervised machine learning was used. This required a clean email set labelled with sentiment and emotions. Therefore, the first two steps were data pre-processing and email annotation.

5.1 High-level approach

While this thesis focuses on customer affect analysis following a short email exchange with customer support, the method used can be applied more generically. The method can be used for any one-to-one email interaction and is not restricted to thesis subject of customer – Customer Service email interaction. Furthermore, the method can be applied to textual conversation other than email, such as chat, letters or even phone call or meeting transcripts.

The method consists of the following steps which are graphically presented in Figure 2:

1. **Data pre-processing**
   A data record is created for each conversation entity (e.g. email, chat message, part of phone call). Any non-relevant data is stripped (e.g. forwarded emails, advertisements). Basic features (number of words, message length etc.) and more advanced features (tf-idf, Doc2Vec) are created for each record. The features will be used as explanatory variables for both sentiment analysis of the conversation entity as well as affect analysis after the conversation.
2. **Annotation preparation**
   To allow for supervised machine learning, conversation entities need to be annotated with sentiment and emotions. The data is split into multiple work packages, each containing a manageable chunk of work that can be distribute the data across multiple annotators. If only part of the data can be annotated (e.g. related to insufficient capacity), then the work package split can best be done using stratified sampling. Thereby avoiding ending up with non-representative data which can cause issues in the machine learning approach for predicting sentiment and emotions.
3. **Manual annotation**
   Since annotating sentiment and emotions is a challenging task where agreement between annotators is an issue, it is necessary to have at least three annotators. Using majority voting, disagreement around sentiment and emotion classes can be solved. To maximize the agreement between annotators, it is key to have clear annotation instructions beforehand. Instructions used for this study can be found in Appendix A: Annotation guidelines.

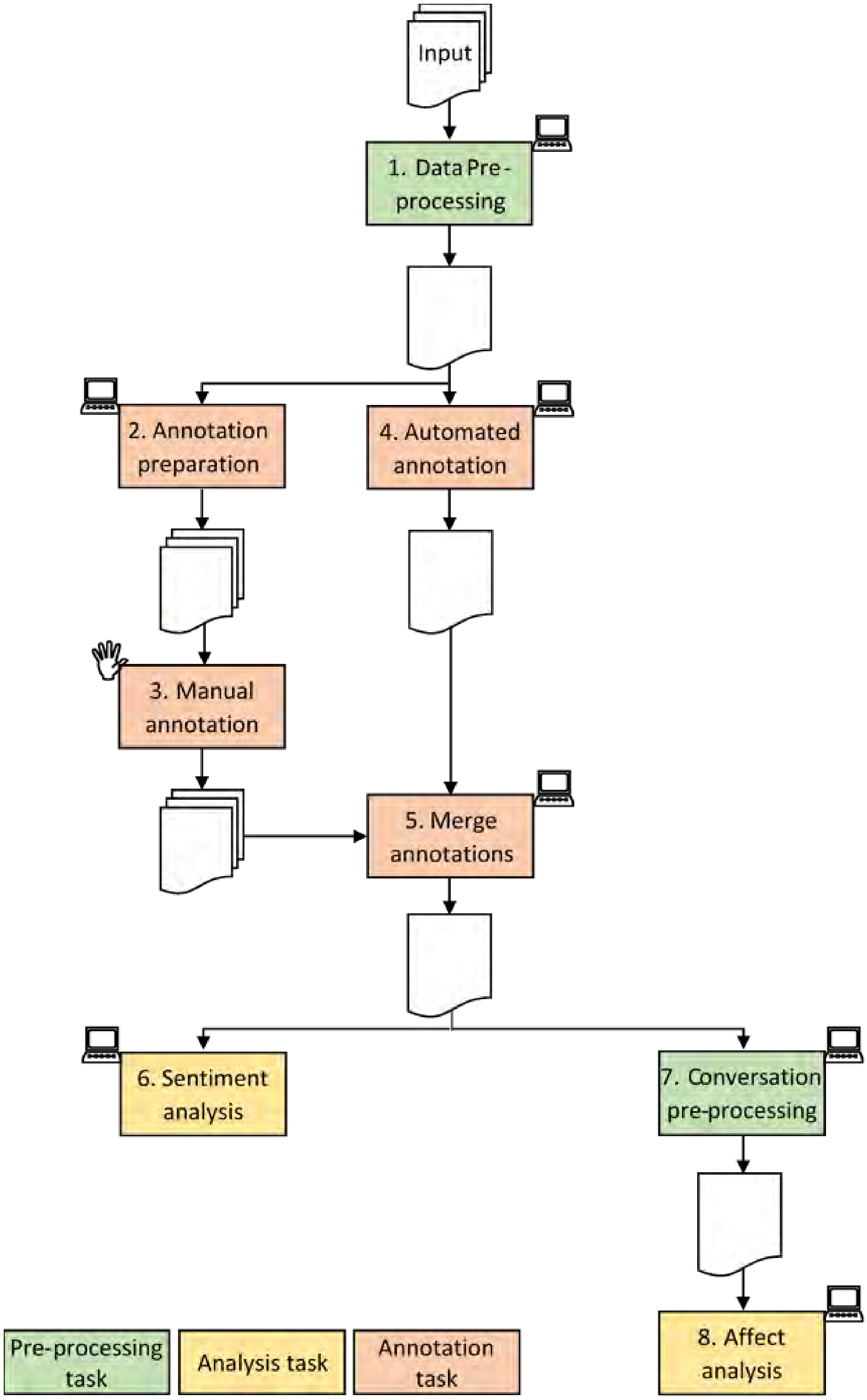


Figure 2: Overall high-level sentiment and affect analysis process. 💻indicates automated step. ✋indicates manual step. The process contains only a single manual step.

4. **Automated annotation**
   To avoid labour-intensive manual annotation, one could resort to using standardized automated annotation either lexicon based (e.g. NRC, LIWC) or machine learning based (e.g. IBM NLU). However, standardized automated annotation can have the major drawback that they are not performing well in the domain in question. Therefore, it is advisable to always perform some manual annotation as a baseline to determine the performance of automated annotation.
   Alternatively, manual annotation can be used to set up a domain specific machine learning based model which in turn can annotate new cases.
5. **Merge annotations**
   This step merges annotation results from the automated annotation step and all manually annotated work packages from each annotator. The result is a file which contains multiple annotations for each conversation entity.
6. **Sentiment analysis**
   Using the features from the pre-processed data as explanatory variables and manual annotated sentiment or emotions as the response variable, any classification method can be used for classifying sentiment or emotions categories in texts.
   Text sentiment can be none, positive, negative, mixed or irony. This makes it a multiclass category requiring a multiclass machine learning model.
   Throughout a text, multiple emotions can appear at the same time. The problem at hand regarding emotions is therefore a multilabel problem (opposed to a multiclass problem). This allows for two different approaches: either dedicated models per emotion or a multilabel model modelling all emotions simultaneously.
   In this study the following models have been tested: Neural Net, Random Forest, Naive Bayes, Support Vector Machines, soft voting ensembles of these same models and RAkEL.
7. **Conversation pre-processing**
   Conversations consist of multiple entities. Main purpose of this step is to merge entities which belong to the same conversation to create a joined feature vector. This includes incorporating the expected sentiment or emotion of the reader as response variable. The sentiment or emotion of the final annotated entity of a conversation, can be used as this response.
8. **Affect analysis**
   The methods and models used for affect analysis are the same as for sentiment analysis. The only difference is a more extensive feature vector used as input.

A detailed description of each step, specific to the thesis subject of affect analysis from email conversations, can be found in the rest of this chapter

5.2 Data pre-processing

To answer the main research question “Which aspects of a Customer Service email response affect customer sentiment?”, data gathering, data pre-processing and feature construction, are essential to model customer affect using machine learning models.

5.2.1 Email preparation

The raw email dataset required multiple pre-processing, annotation and analysis tasks. The full processing pipeline can be seen in Figure 2. The green tasks identify the pre-processing steps.

The pre-processing step is the first step in the processing pipeline. In this step, the following tasks were performed:

- Remove emails with **empty text body**
- Remove **non-English** emails (2x). The non-English emails have been identified as part of the annotation task which is described in the next chapter. The pre-processing does not include a generic language recognition step for all emails
- Add a **thread sequence number** to each email using Case Number and formatted Message Date. The first email in a thread has sequence number 0
- Remove any emails not part of a “client – customer support – client” **conversation**
- Extract and clean text body, hereby removing any **forwarded email** and any **embedded URL links**
- Expand **contractions** like “I’ve” to “I have”
- Replace **order numbers, article numbers** and **voucher codes** in text body by generic “order_no”, “article_no” or “coupon_code”
- Create simple features from text body:
    - Total number of **words** in the text body
    - Total number of **unique words** in the text body
    - Diversity of words. This is the **ratio** of **unique words** versus the **total** number of **words**
    - Number of **exclamation marks**
    - Number of words which are fully **capitalized**
    - Number of **sentences**
    - Number of **words per sentence**
    - Total **length** of the **message** in number of characters
    - Average **length** of a **word**
    - **Variance** in **word length**
- Create simple features from message date:
    - **Day** of the **week** (Monday, Tuesday,…)
    - **Part** of the **day** (Night, Morning, Afternoon, Evening)
- Create more complex features from text body:
    - Count of words occurring in **NRC emotion word** list (Mohammad & Turney, 2013) for each of the ten categories
    - Ratio of **correctly spelled words** / **total** number of **words**

  - **Tf-idf**
  - **Doc2Vec**

For detailed overview of variable names, description and descriptive statistics, see Appendix B: Data descriptive incoming emails.

5.2.1.1 Spelling check

All words in the text body are checked against the standard Unix English word list. The word list does not contain plurals or conjugations; therefore, words are lemmatized using NLTK Perceptron Tagger and the NLTK WordNet Lemmatizer[1]. Since the word list does not contain any contractions either, contractions are expanded to a full form. In several cases the contraction could be any of multiple forms; e.g. “he'll” could be a contraction of either “he will” or “he shall”. Correctly resolving the expansions is a complex task. However, in this study a simple list of common contractions is used containing a single expansion per contraction.

5.2.1.2 Tf-idf

Tf-idf scores can be calculated for every n-gram in the email corpus for every email. This can result in a very large number of scores per email which can lead to dimensionality problems in a later stage of machine learning. Another possible issue when generating tf-idf is the sparsity of resulting scores. Every email contains a number of n-grams which is far less than the total number of unique n-grams in the whole corpus. To reduce dimensionality the following actions are taken:

- Words are lemmatized using the NLTK Perceptron Tagger and the NLTK WordNet Lemmatizer.
- Stop words are excluded (e.g. 'i', 'too', before', 'are', 'what' etc.). The list of stop words from NLTK Stopwords Corpus is used. The list contains 153 English words.
- Order numbers, article numbers and voucher codes are generalized to the form: order_no, article_no, coupon_code.
- Settings are used to limit the number of features: maximum document frequency (0.5), maximum number of features (400), n-gram range (word: 1-3, character: 2-5)

Since incoming client emails are structured very different from outgoing customer support emails and since both emails serve a different purpose in the analysis, the tf-idf has been calculated separately for incoming and outgoing emails. This allows for focussing on n-grams that are specific for incoming or outgoing emails.

5.2.1.3 Doc2vec

This study uses the Gensim[2] Python implementation of Doc2vec. As input for the Doc2vec models a cleaned text body is used. This cleaning only includes removing forwarded email and any embedded URL links and generalizing order numbers, article numbers and voucher codes. Two Doc2vec models are trained, one for incoming emails (38k emails) and another for outgoing emails (38k emails).

[1] http://www.nltk.org/

[2] https://radimrehurek.com/gensim/about.html

5.2.2 Conversation data preparation

To allow affect analysis, the separate pre-processed incoming and outgoing emails need to be transformed to a single data record per “client – customer support – client” email exchange. Each record contains the following data fields:

- All features from pre-processed **originating client email**
- All features from pre-processed **CS response email**
- **Response Time**. Time between originating client email and CS response email rounded to whole days.
- Response variables: **Sentiment**, **Emotions**: Anger, Disgust, Fear, Joy, Sadness.
  These sentiment and emotions are taken from the annotated client response email.

Throughout a thread, 'Case Origin', 'Contact Reason’ remain the same. Since both incoming client email and CS response mail contain these fields, the fields are removed when combining into a record. Furthermore, the email sequence number occurs in both the first and second email (with a difference of one). The sequence number of the CS response mail is kept. In other cases where the same field occurs in both client email and CS response email, both fields are kept. The client email columns are renamed to avoid duplicate columns.

5.3 Email annotation

While data preparation supplies the essential inputs for machine learning models to answer the main research question “Which aspects of a Customer Service email response affect customer sentiment?”, generating correct outputs is just as essential for supervised learning of these models. These outputs consist of sentiment and emotions as annotated for each email by several annotators.

5.3.1 Sentiment and emotion framework

The first step in email annotation is determining the sentiment and emotion framework to adopt.

For sentiment, the following categories are used in line with Mohammad (2016). An email can have only one category at a time:

- **Positive** the speaker is using positive language, for example, expressions of support, admiration, positive attitude, forgiveness, fostering, success, positive emotional state
- **Negative** the speaker is using negative language, for example, expressions of criticism, judgment, negative attitude, questioning validity/competence, failure, negative emotion

- **Mix** the speaker is using positive language in part and negative language in part
- **None** the speaker is neither using positive language nor using negative language
- **Irony** the speaker is using expressions of sarcasm, ridicule, or mockery

Bosco et al. (2013) found that found that irony was often used to reflect negative sentiment. The presence of an Irony category can help to prevent confusion with the annotator when for example positive language is used to express negative sentiment.

For emotion, the framework of Ekman (1973) is used. The associated words listed with each emotion category are adopted from Goleman (1996):

- **Anger** the speaker is using language which expresses: animosity, annoyance, irritability, hostility, fury, outrage, resentment, wrath, exasperation, indignation, vexation, acrimony.
- **Disgust** the speaker is using language which expresses: contempt, disdain, scorn, abhorrence, aversion, distaste, revulsion.
- **Fear** the speaker is using language which expresses: anxiety, apprehension, nervousness, concern, consternation, misgiving, wariness, qualm, edginess, dread, fright, terror.
- **Joy** the speaker is using language which expresses: happiness, enjoyment, relief, contentment, bliss, delight, amusement, pride, sensual pleasure, thrill, rapture, gratification, satisfaction, euphoria, whimsy, ecstasy.
- **Sadness** the speaker is using language which expresses grief, sorrow, cheerlessness, gloom, melancholy, self-pity, loneliness, dejection, despair.

The emotion 'Surprise' is part of the Ekman framework too, however, it is not included here. The category is excluded for the following reasons:

- to align categories with automated benchmark of IBM Natural Language Understanding
- to reduce the complexity and total effort of manual annotation.

5.3.2 Automated annotation

To get some intuition on the quality of the manual annotation, automated annotation was performed to act as baseline. Two variants of automated annotation have been applied:

- **NRC emotion lexicon** (Mohammad & Turney, 2013) based.
  For sentiment, NRC lexicon contains positive and negative labels. If the email does not contain any words with a NRC sentiment label, the email is annotated with label None. If at least one word of either the positive or negative sentiment labels is present, the email receives that label. If words are present from both sentiment labels, the email gets label Mix. The label Irony is not used.
  For emotion, NRC lexicon contains eight labels. Of these eight labels, only Anger, Disgust, Fear, Joy, Sadness are used to perform a word count, in line with the chosen emotion model. If at least one word from the email appears in the NRC lexicon with an emotion label, the email is labelled as having the specific emotion.

- **IBM Natural Language Understanding**[3](IBM NLU).
  IBM NLU analysis text to extract meta-data among which sentiment and emotions. It provides an API which can analyse text of up to 10.000 characters. IBM NLU is based on a machine learning model trained on news sources.
  With regards to sentiment, the API returns the polarity, strength and a mixed-polarity indicator. The polarity strength is discarded in this study. If the mixed polarity indicator is present, the sentiment label is annotated as mixed. Otherwise, the sentiment label is annotated according to the IBM NLU sentiment type. IBM NLU does not indicate any irony, hence this sentiment label is not used.
  For emotions, the API returns per emotion a value between 0 and 1 indicating the probability of the text having the emotion. In case of a probability greater than 0.5, the emotion label is assigned. An email can have multiple emotion labels.

5.3.3 Manual annotation process

An email can have multiple concurrent emotions. The annotators were asked to label each emotion category on a four-level Likert scale indicating the intensity of the emotion. Using a Likert scale could be important for the machine learning phase of predicting resulting customer sentiment since the scale opens the opportunity for detailed analysis of changes in emotion.

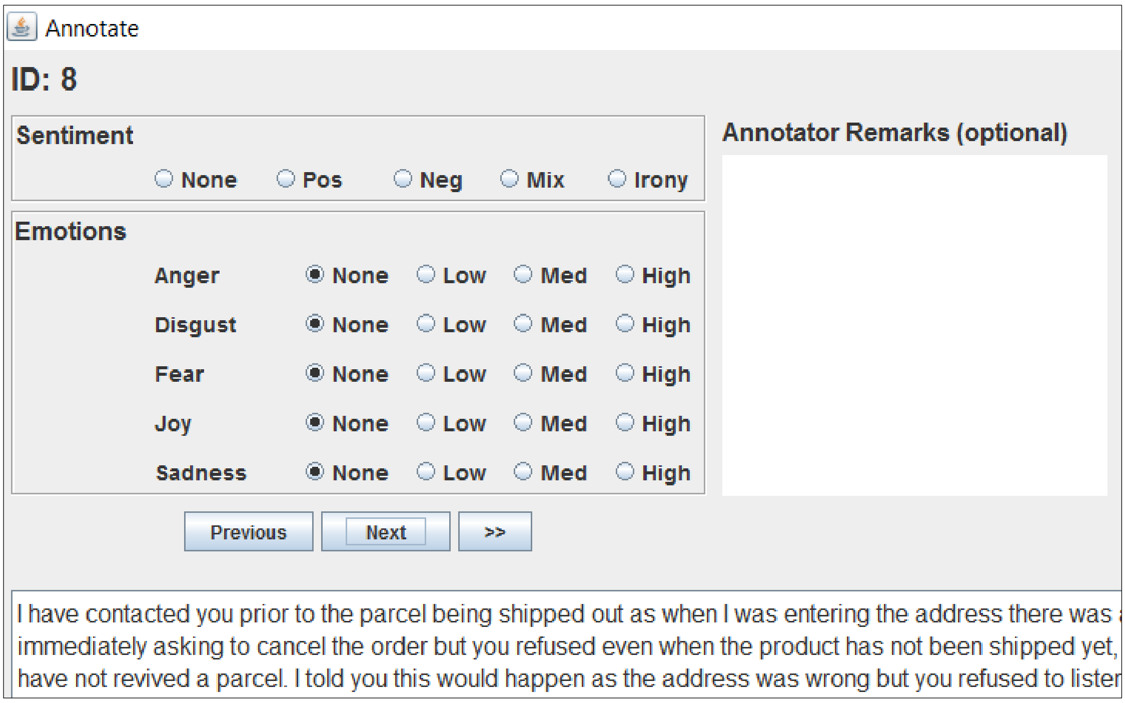
Annotation of sentiment and emotion in text can be done on multiple levels e.g. message level, sentence level or word level. The advantage of annotating on message level compared to more detailed level is that on message level there is more context available to help understand the true sentiment and emotions. Furthermore, the effort for annotating on message level is much less which will result in more annotated emails. However, the disadvantage is that a message can contain mixed sentiment and multiple emotions which results in a risk of causing confusion with the annotator and performance degradation in the machine learning phase.
In this study annotation is performed on message level, mainly due to the reduced annotation effort. The disadvantage of possible confusion with the annotator has been decreased by offering a mixed sentiment category and the possibility of annotating multiple emotions per email.
It should be noted that the whole email includes salutations and valedictions. While these opening and closing statements do not contain any content, they may contain indications of writer's sentiment or emotion e.g. 'Cheers' versus 'Yours very disgruntled shopper'. Even the absence of these elements can be an indicator.

Annotating sentiment and emotions can be a difficult task. Therefore, a single annotation on an email cannot be considered as a ground truth. Agreement between annotators on emotions in text is typically low. To cope with this uncertainty, three annotators are used for every email. While using more annotators is preferable, the number was restricted due to

[3] https://www.ibm.com/watson/developercloud/natural-language-understanding.html

constraints on the availability of capacity. In total five different annotators were involved in annotating emails. Mohammad and Turney (2013) used crowd sourcing for their emotion lexicon. However, this bypass of the capacity constraints was not an option since the confidentiality of the data prohibited sharing the data on an open platform.

Annotating emails with sentiment and emotions is not only difficult, it is also a laborious task. Tooling should support the annotators in their task. In this study, a simple Java application has been developed for this purpose (see Figure 3). The application allows annotators to assign labels to the whole email. Furthermore, an optional 'Annotator Remarks' is available which can be used to signal any specificities such as non-English emails or emails where the forwarded email is still attached.



Annotate

ID: 8

Sentiment

○ None ○ Pos ○ Neg ○ Mix ○ Irony

Emotions

Anger	● None	○ Low	○ Med	○ High
Disgust	● None	○ Low	○ Med	○ High
Fear	● None	○ Low	○ Med	○ High
Joy	● None	○ Low	○ Med	○ High
Sadness	● None	○ Low	○ Med	○ High

Annotator Remarks (optional)

I have contacted you prior to the parcel being shipped out as when I was entering the address there was
immediately asking to cancel the order but you refused even when the product has not been shipped yet,
have not revived a parcel. I told you this would happen as the address was wrong but you refused to lister

Figure 3: Sentiment and emotion annotation application

Using the annotation application, a small-scale test was done annotating several emails to determine the time needed for annotation. Based on this test, work packages were made containing 150 emails corresponding to about one hour of work. For any thread included in a work package, all incoming emails were included in the same work package. This way it was avoided that only one of the incoming emails in a conversation is annotated. As in that case an annotated email could be used for sentiment analysis, but not for predicting future customer sentiment and emotion.

Five annotators have annotated 2, 3 or 5 work packages of emails each. In total 745 emails have been annotated. Each email was annotated by three different annotators. Three of the emails were discarded since they contained only non-English text. The remaining set of 742 emails contained 455 client-CS-client interactions. This test set was used in all next steps from assessing annotator agreement to sentiment analysis and Affect analysis.

5.3.4 Combining multiple annotations

In this thesis only four (out of 742) emails were annotated with Irony by a single annotator. Examining these annotations showed that these emails all had negative sentiment. To reduce the number of categories and allow for better performance in the machine learning phase, the Irony category was included in the Negative category.

Initial analysis on annotator agreement on a binary scale (6.1.2) showed issues with regards to agreement. Therefore, the four levels (0-3) have been combined to a binary variable combining 1, 2 and 3 to True value. Furthermore, IBM NLU and NRC lexicon also use a binary emotion indicator which further supported the decision to use binary emotion classification.

Still, each of three annotators of an email can have annotations that differ from each other. Combining results of three annotators can be done in three ways:

- **Majority vote**. Assigned the label on which at least two annotators agree.
- **Full agreement**. Use only emails for which all annotators agree on a specific label. Any emails without agreement are discarded.
- **Average scoring**. The label value is the average over the three annotators.

Average scoring converts the problem into a regression problem. While the other two methods result in a classification problem.

In case of average scoring, each annotator score has the same weight in the final label value. However, this emphasizes any annotator disagreement. Hence Majority vote and Full agreement are the preferred methods of combining annotator results. The downside for Full agreement however is the loss of all data where full agreement is not present. With the limited number of emails (742) and conversations (455) available full agreement is more likely to produce poor machine learning results. Still, for sentiment analysis, both methods have been tested to determine which method gives the best result in this study. While for affect analysis, only majority vote has been used.

It should be noted the majority vote of three annotators may still lead to hung votes in the case of multiple sentiment classes. The combined Sentiment label is determined according to the following rules:

```
IF any #labels ≥ 2
     Combined label = majority label
ELSE IF  Annotator1 = 'POS' &
         Annotator2 = 'NEG' &
         Annotator3 = 'MIX'
     Combined label = 'MIX'
ELSE
     Combined label = 'NONE'
```

5.3.5 Measuring annotator agreement

Inter-annotator agreement is measured by using Cohen's kappa metric (see 3.1). Using kappa, a pairwise comparison of agreement is done between each of the annotators and automated annotations.

5.4 Method Sentiment analysis and Affect analysis

Sentiment analysis and affect analysis both comprise several steps. After initial data pre-processing as described in sub-section 5.2, some choices need to be made and actions need to be taken which are described in this section

5.4.1 Class imbalance handling

The majority of emails or conversations have sentiment None and lack most or all of the five emotions. Only a minority of the emails or conversations have a non-neutral sentiment label or an emotion label (see Table 5, Table 6, Table 14 and Table 37 through Table 41). This imbalance between the classes on each emotion or sentiment label may cause difficulties when training machine learning models. Depending on the model used, three main methods to cope with such an imbalance are:

- Assign a specific weight per class
- Undersample the majority class
- Oversample the minority class

Since not all models allow assigning a class weight, the class weight method is not used. Due to the small amount of annotated emails (742) and conversations (455), undersampling would likely result in too small amount of data to effectively perform machine learning. Therefore, oversampling is preferred in this case. Oversampling is implemented in Python using RandomOverSampler from the imbalanced-learn API (Lemaitre et al., 2017).

It should be noted that oversampling is only applied for sentiment models and models that model a single emotion at the time. In case of a multilabel model, modelling all emotions at the same time, sampling is not that easy. Sampling one of the labels would result in further imbalance of other labels. While Charte et al. (2015) do propose some methods for handling such class imbalance, multilabel modelling was executed without any imbalance countermeasures. While oversampling could improve machine learning results, both tests with and without oversampling were run to find maximum performance.

5.4.2 Categorical data handling

Most machine learning models require numeric input. However, features 'Case Origin', 'Contact Reason', 'dayOfWeek' and 'partOfDay' are categorical. Converting each category to a number has the drawback that is the variable is effectively converted into an ordinal variable. Therefore, each categorical feature is one-hot encoded resulting in each label for each feature being presented as a binary feature.

5.4.3 Single label or multilabel modelling of emotions

Each email or conversation record is labelled with five different emotions (and a 4-class sentiment). Modelling these emotions can be done in any of the following setups:

- Dedicated model for each emotion
- Multilabel model which models all emotions at the same time
- Ensemble of dedicated models for each emotion to models all emotions at the same time

The advantage of dedicated model is that it can specialise on a single emotion and thereby capture complex feature dependencies which may get lost when modelling multiple labels at the same time.

For modelling emotions multilabel modelling is a very interesting option as emotions for a single email may have correlation. For example, Anger is less likely to occur together with Joy in a same email. This correlation cannot be captured when modelling each emotion as a dedicated model.

The ensemble method has the advantages of both the dedicated and multilabel model but requires sufficient data to train & test the dedicated and multilabel models as well as train & test the ensemble of the models. With the limited number of annotated emails and conversations, this was not an option.

Both dedicated single label and multilabel models were tested to determine optimal model performance.

5.4.4 Model selection

When using a supervised machine learning approach for sentiment analysis or affect analysis, modelling does not differ from any other supervised machine learning classification method.

For this study, several popular machine learning models and ensembles for sentiment and affect analysis were selected. Among these models, RAndom k-labELsets multi-label classifier (RAkEL) was included since this model was identified by Ye et al. (2012) as the best performing model for affect analysis. Table 3 gives an overview of all models that were tested.

Model	Sentiment analysis	Emotion analysis	Affect analysis sentiment	Affect analysis emotion
Neural network (NN)	Multi class	Single label, multilabel	Multi class	Single label, multilabel
Naive Bayes (NB)	Multi class	Single label	Multi class	Single label
Support Vector Machines (SVM)	Multi class	Single label	Multi class	Single label
Random Forest (RF)	Multi class	Single label, multilabel	Multi class	Single label, multilabel
Soft Voting ensemble	NN + RF + NB + SVM NN + RF + NB NN + RF + SVM NN + RF	NN + RF + NB + SVM NN + RF + NB NN + RF + SVM NN + RF	RF + NB + SVM + NN RF + NB + SVM RF + NB RF + SVM	NB + RF + NN + SVM NB + RF + NN NB + RF NB + NN
RAkEL (Label Power)		Neural Net Random Forest		Neural Net Random Forest

Table 3: Selected models for experiments

The model combinations for the soft voting ensembles were selected after initial test with the individual models. Models showing the most potential were selected in multiple combinations.

Selecting which model to use for further feature selection is only one of the choices to be made. Other choices to be made were:

- How to combine annotations: **consensus** or **full agreement** (5.3.4)
- How to handle class imbalance: **no action** or **oversampling** (5.4.1)
- How to maximize evaluation result: use **default** 0.5 threshold or **adjusted threshold** (single label emotion analysis only)

In case of binary classification, assigning one label or the other is depending on the probability of each of the labels (which sum to one). The default for both labels is a probability threshold of 0.5. However, shifting the threshold in favour of one of the labels can improve performance. As part of the model selection, threshold optimisation was included. It was included for emotion only since for the 4-class sentiment, setting a threshold does not apply. In the latter case, a label is assigned which has the highest probability compared to the other three labels opposed to using a fixed threshold.

To determine which combination of above choices and which model had the most potential for optimal performance after feature selection, an initial test was done with all possible combination of each choice and each model. Each test encompassed 30 runs, each time with a different train (80%) and test (20%) sample. In case of setting optimal threshold, a further split of the data was done such that 60% was used to train the model, 20% was used to determine the threshold to maximize kappa and the remaining 20% was used the test the model performance. All the splits were made using stratified splitting on the response variable (scikit-learn 0.18.1 StratifiedShuffleSplit). This ensures that the distributions of the different classes of the response variable are the same in all data splits and hereby ensures that performance is not affected by non-representative splits.

The models with the highest average performance were selected to be used for feature selection. All models but one have been implemented using Python scikit-learn version 0.18.1 (MLPClassifier, RandomForestClassifier, SVC, GaussianNB, VotingClassifier). RAkEL was implement using scikit-multilearn version 0.0.5 (LabelPowerset). All model implementations use the default settings except for the maximum number of iterations for neural network (1000) and the number of trees from Random Forest (200).

5.4.5 Feature selection and feature importance

Feature selection

Feature selection is a key step in improving performance. Reducing the number of selected features can benefit model performance in two ways. Too many features will increase the risk of overfitting the model during training resulting in a model that does not generalize well. Furthermore, a high number of features creates a high dimensionality problem, which is inherently more complex.

For models like neural network, support vector machines it is difficult to quantify the importance of a specific feature. For an ensemble containing these models, it is even more difficult. Hence, using feature importance is not an option for feature selection. For this thesis, simple forward and backward incremental methods are used.

The backward approach starts with a full feature set. All available features from the set are tested separately, each time removing one of the features. The worst performing feature is removed from the feature set. This way the feature set is incrementally decreased until removing a feature does not further improve performance. Feature groups such as NRC word count, tf-idf or Doc2Vec, are treated as single features, including the group as a whole. To reduce the influence of random stratified test (80%) / train (20%) split on model performance, the kappa is averaged of 30 runs. The forward approach follows the same principles, but starting with an empty selected feature set. This time adding a feature at a time until no performance improvement can be reached. At the same time, the forward selected set is tested to determine if removing a feature may improve performance. The feature set from highest performing model, either backward or forward, is selected.

Feature importance

As indicated, feature importance is often very difficult to determine. Still, a simple method to get an impression of feature importance is the following. The method resembles the first step of the backward feature selection. First, a baseline is set which includes all selected features. Next, from the feature set a single feature is excluded. Feature importance is then quantified by the difference of the model result of the reduced feature set compared to the baseline. Each feature test consisted of 50 runs, each time using a different split of the data for training (80%) and testing (20%) and using the same data for all the features. The significance of each feature was then determined using a paired t-test for the feature versus the baseline containing all features. Significant performance degradation indicates a feature to be important. In this case the difference in kappa is solely due to a specific feature. The importance of a feature can be even greater but its effect may be masked by other features. If no significant change in performance is detected, it is undetermined what the feature importance is. The feature may be either obsolete with no impact on performance or any feature importance is masked due to high correlation with other features that result in approximately the same performance.

5.4.6 Model evaluation

The main metric for evaluating sentiment and emotion analysis models is the kappa metric (see 2.2). Since this metric is used for annotator agreement too, it helps to get some intuition of the performance compared to the human annotators. The kappa metric in this case presents the 'agreement' between predicted emotion and the combined annotated emotion.

Common and intuitive evaluation criteria for classification results are precision and recall. Precision and recall can be combined into a single measure; the F1 measure. F1 is the harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4}$$

All three metrics are used to present model performance too.

For affect analysis on emotion, finding conversations leading to certain emotions is of more interest than identifying conversations leading to no emotion at all. The F1 -measure on the minority class is a metric than can identify exactly that. While for affect analysis on sentiment there is no such minority class, overall accuracy can capture the ability to accurately identify the correct sentiment.

Table 4 gives an overview of the metrics that were used in this thesis for the various analysis types.

Analysis type	**Model selection metric**	**Other metrics**
Sentiment analysis	Kappa	Precision, Recall, F1
Emotion analysis	Kappa	Precision, Recall, F1
Affect analysis sentiment	Accuracy (=micro avg recall)	Precision, F1
Affect analysis emotions	F1-measure of minority class	Precision, Recall

Table 4: Model evaluation metrics per analysis type

After model selection (see 5.4.4) and feature selection (see 5.4.5), the final performance evaluation consisted of 100 runs of the best model and feature combinations for both sentiment and affect analysis on both sentiment and emotions. Each run was executed with a different stratified train (80%) and test (20%) sample. Based on the average predictions the various performance metrics were calculated.

6 Results

The results section consists of three main parts:

- Annotation results (6.1). Presents annotation distributions and annotator agreement.
- Sentiment analysis results (6.2). This sub-section includes results with regards to model selection, feature selection, feature importance, model performance and benchmark comparison.
- Affect analysis results (6.3). This sub-section is structured the same way as Sentiment analysis results, but first presents a high-level affect data overview.

6.1 Annotation results

The first research question is "*which sentiment and emotions can be detected in customer emails?*" The first step to answer this is to label emails with sentiment and emotions before in the next step a machine learning model can be trained to detect sentiment and emotions. However, the result of the prediction is only as good as the labels it's trained with. Therefore, the results of manual annotation are closely examined.

6.1.1 Annotation distributions

Sentiment annotation distribution

Table 5 shows the percentage of emails of all annotated emails with the indicated label. IBM NLU and NRC do not make use of the Irony category. Any Irony labels in the combined sentiment classes have been included in the negative sentiment label. Each email can have only one sentiment category (multi-class). The percentages therefore add up to 100 percent. Only 366 out of 742 (47%) of the emails have full agreement of all three annotators.

	None	Pos	Neg	Mix	Irony	Support
IBM NLU	11.6	35.3	15.5	37.6	-	742
NRC lexicon	19.7	36.1	6.9	37.3	-	742
Annotator 1	26.7	22.5	41.0	9.8	0.0	742
Annotator 2	27.1	28.3	35.5	9.0	0.0	442
Annotator 3	43.5	18.2	34.0	3.2	0.9	444
Annotator 4	16.7	42.1	39.1	1.7	0.0	299
Annotator 5	26.1	22.7	47.8	3.3	0.0	299
Annotator consensus	33.4	23.2	38.6*	4.7	-	742
Full annotator agreement	23.9	29.4	46.4*	0.3	-	366

Table 5: Percentage of emails annotated with specific sentiment. *indicates percentages including Irony cases which were re-labelled to Negative. Annotator distributions differ significantly (p<<0.01)

Above distributions are clearly not the same for each annotator. For example, Annotator 3 has much more 'None' classifications than 'Pos' classifications. While for Annotator 4 it is exactly the other way around. A χ^2 test results in p-value <<0.01 supporting the differences in distribution.

Emotion annotation distribution

From Table 6 it can be seen that there is quite some spread between annotators with regards to the total percentage of emails annotated with a specific emotion. Since an email can have multiple emotions in a single email, it is a multilabel instance. Therefore, the percentages do not add up to 100 percent. Full agreement on across all emotion categories occurs in 28.6% of all emails.

	Anger	**Disgust**	**Fear**	**Joy**	**Sadness**	**Support**
IBM NLU	26.3	0.5	0.4	23.3	13.9	742
NRC lexicon	20.6	16.4	22.2	38.7	33.4	742
Annotator 1	32.3	13.5	8.6	24.4	20.1	742
Annotator 2	25.6	9.5	6.8	18.1	38.7	442
Annotator 3	9.7	28.2	9.0	19.8	12.8	444
Annotator 4	17.4	5.4	18.4	18.7	24.7	299
Annotator 5	33.4	0.0	1.0	25.4	42.1	299
Annotator consensus	22.9	11.7	4.9	19.3	23.7	742
Full annotator agreement	15.4	2.0	2.0	16.0	11.8	
(Support full agreement)	(526)	(568)	(612)	(594)	(447)	

Table 6: Percentage of emails annotated with emotion. Email can contain multiple emotions therefore no 100% sum. Annotator distributions differ significantly for each emotion (p<0.05). Full agreement has substantial impact on available records. Fear has very low consensus.

An indication of the limited agreement between annotators can be seen in the difference between the percentages of consensus and full agreement. Especially for Fear, the percentage of emails is low. Since agreement is low across annotator pairs, this is reflected in the percentage of emails where all three annotators of an email fully agree. Although the highest agreement is found on Fear 82.5%, this is almost fully due to Fear not being annotated in emails. Best agreement can be found in Joy. In this case in 32.7% of the cases at least one annotator annotated emotion while in 12.8% all three annotators agreed on Joy being present.

A χ^2-test on the numbers per emotion shows that the differences are significant with p<0.05 (for details see Appendix D Table 26). Apparently, each annotator has its own personal bias towards certain emotions. It should be noted that annotators did not all annotate the same sets of emails and therefore the conclusion only holds under the assumption that each set of 150 emails contains approximately the same emotions.

Emotions versus emotions

Table 7 shows the co-occurrence for the five emotions. While Joy and Sadness often are the only emotion in an email, Anger and Disgust co-occur most often. Anger and Disgust hardly co-occur with Joy.

	Anger	Disgust	Fear	Joy	Sadness	No other
Anger	-	75	4	1	51	63
Disgust	75	-	1	2	28	5
Fear	4	1	-	0	10	24
Joy	1	2	0	-	12	130
Sadness	51	28	10	12	-	99
No other	63	5	24	130	99	-

Table 7: Number of co-occurrences of emotions in set of 742 emails. Anger-Disgust co-occur often while they hardly co-occur with Joy. Joy and Sadness is often the only emotion in the email

Using Spearman rank-order correlation coefficient to test for correlation shows that most emotions in emails are significantly correlated (Appendix D Table 27). Although most relations are found to be (very) weak. The Anger-Disgust relation may be classified as moderate with a correlation of 0.55. The emotions Anger, Disgust, Fear, and Sadness are usually related to negative sentiment while Joy is related to positive sentiment. This relation is confirmed with negative emotions having negative correlation to the positive emotion Joy.

Sentiment versus emotions

As stated above, four emotions are associated with a negative sentiment: Anger, Disgust, Fear, and Sadness. While Joy is associated with positive sentiment. Putting annotated sentiment and emotion in a crosstab, one would expect zero Positive - Anger/Disgust/Fear/Sadness and Negative - Joy combinations. While this is mostly the case for Anger, Disgust and Joy, it certainly is not for Fear and Sadness.

	Emotions				
Sentiment	**Anger**	**Disgust**	**Fear**	**Joy**	**Sadness**
Pos	4	2	42	400	60
Neg	507	264	86	3	372
Mix	26	15	23	63	91
None	11	2	41	15	54

Table 8: Crosstab Sentiment vs. Emotion. Negative emotions Fear and Sadness often co-occur with Positive sentiment. Incidentally co-occurrence can be seen for Joy -Negative and Anger and Disgust - Positive

Emails with Positive-Anger or Negative-Joy combinations have been presented to the original annotators for reassessment. In most cases this resulted in updating either sentiment or emotion which brought sentiment and emotion in line again. Any exceptions were caused by 'angry' emails which did have a pleasant or at least a polite tone of voice or in some cases a polite valediction which were part of the email template (e.g. "If you have any questions or you need any help, feel free to contact me and I will be happy to help. Please be assured that we endeavour to reply to your enquiries as quickly as possible.”)

6.1.2 Annotator agreement

The kappa of a pair-wise comparison of NRC, IBM and five annotators per emotion category and sentiment can be found in Appendix D Table 28. It shows that NRC and IBM have lower agreement then any of the annotators mutually have. While NRC has the most agreement for Anger, still the average kappa score is only 0.3, indicating fair agreement at most. The performance of IBM on Disgust, Fear and Sadness is bad. However, on Joy the average kappa score is 0.42, which demonstrates moderate agreement. Nevertheless, NRC and IBM are outperformed by annotators on every front. Though, there are major differences between various emotions and between various annotator pairs. For example, kappa for Anger ranges from 0.15 to 0.76. The category with most agreement is Joy, where the kappa is in the range of 0.5 to 0.71.

Table 9 shows that, Disgust, Fear and Sadness have low average kappa across all annotators. It shows that human annotation is hard for some emotions. The low agreement level likely renders the three categories useless for machine learning. Anger and Sentiment have kappa of at least 0.45, while Joy even has 0.61.

	Sentiment	Emotions				
		Anger	Disgust	Fear	Joy	Sadness
Avg annotator kappa	0.45	0.49	0.21	0.24	0.61	0.31

Table 9: Average kappa across all 5 annotators. Disgust, Fear and Joy have low kappa scores. Joy has the best score

6.2 Sentiment analysis results

Results presented in this section are the basis to answer the research questions: (1) what sentiment can be detected in customer emails, (2) does a domain specific sentiment detection machine learning model trained on a small set of emails, outperform a general model for sentiment analysis.

6.2.1 Model selection

Multiple machine learning models (Neural network, Naive Bayes, Support Vector Machines, Random Forest, Soft Voting ensemble, Label Power RAkEL) have been tested for different combinations of settings with regards to:

- Oversampling minority class yes / no (not applicable for multilabel)
- Combine multiple annotations based on majority vote or full agreement
  (not applicable for multilabel)

- Discriminate binary emotions base on 0.5 probability threshold or threshold optimized for maximum kappa (only applicable for single label emotions)

All model and setting combinations have been run 30 times with different train/test splits to reduce the influence of the random data split.

Sentiment model selection

Analysis results on sentiment are presented in Table 10. The best performing model is a soft voting ensemble of Neural Net, Random Forest and Support Vector Machines that uses oversampling of the minority classes and a majority vote for combining annotations. Setting a custom threshold is not applicable here since highest probability among classes is used as discriminator. For all models, oversampling outperforms no sampling. Also, majority vote outperforms full agreement for all models but SVM.

Model	Full agree. Oversampling	Majority vote Oversampling	Full agree No oversamp	Majority vote No oversamp
Neural Net (NN)	0.27	0.37	0.18	0.30
Naive Bayes (NB)	0.21	0.23	0.19	0.22
SVM	0.31	0.26	0.31	0.26
Random Forest (RF)	0.35	0.36	0.34	0.36
Voting NN+RF+NB+SVM	0.36	0.40	0.31	0.39
Voting NN+RF+NB	0.37	0.38	0.33	0.34
Voting NN+RF+SVM	0.34	**0.41**	0.22	0.35
Voting NN+RF	0.35	0.39	0.26	0.34

Table 10: Sentiment - Average kappa (30 runs) per scenario and ML model. Bold indicates the best kappa score

Emotion model selection

In total 340 different models or ensembles and settings have been tested to find the best performing model per emotion. Table 11 shows a selection of the model performance including the best performing model per emotion. Full test results can be found in Appendix C.

It can be seen in Table 11 that models with oversampling perform better than without oversampling. Balancing the unbalanced emotion classes helps performance. Furthermore, using only test cases with full annotation agreement reduces performance for most emotions. Only for Joy performance is a slightly better. Also, optimizing the threshold only works for Fear category, but kappa is still very low.

As significant correlation was found between emotions (Appendix D Table 27), and following results of Ye et al. (2012) using RAkEL, it would be expected that a multilabel approach gives best results. However, none of the multilabel approaches yield better results than the best dedicated single label model.

Kappa overall is not very high as was the case with manual annotation. Especially kappa for Fear and Sadness can be considered low. Taking kappa 0.4 as a cut-off point, emotions Fear and Sadness are discarded for further analysis with regards to feature selection. The soft voting ensemble with Neural Net and Random Forest is selected as model to be used at feature selection for Anger, Disgust and Joy.

Model	Anger	Disgust	Fear	Joy	Sadness	Remarks
Voting Neural Net + Random Forest	**0.48**	**0.46**	0.02	0.53	**0.36**	oversampling majority vote 0.5 threshold
Voting Neural Net + Random Forest	0.38	0.18	0.02	**0.54**	0.28	oversampling full agreement 0.5 threshold
Random Forest (single label)	0.40	0.36	**0.13**	0.46	0.35	oversampling majority vote optimized thresh.

Table 11: Average kappa (30 run). Selection of best performing emotion analysis models. Bold indicated best model for specific emotion. Each best model uses oversampling. Most models use majority vote and 0.5 threshold. Single label models outperform multilabel models. Fear and Sadness performance is poor (below 0.4).

6.2.2 Feature selection and feature importance

Available features

Each feature vector consists of 342 available features:

- 100 features related to **character**-based **tf-idf**
- 100 features related to **word**-based **tf-idf**
- 100 **Doc2Vec** features
- 2 features of one hot encoded **Case Origin**
- 10 features of one hot encoded **Contact Reason**
- 7 features of one hot encoded **dayOfWeek**
- 4 features of one hot encoded **partOfDay**
- 7 features related to **NRC** lexicon categories (countAnger, countPos etc)
- 12 **other features** such as threadItem, lengthMessage

While the available number of features for tf-idf is much larger than 100, the number of features is restricted to 100 most important ones. This keeps the overall number of features (342) well below the number of available data records (742) to limit overfitting when training the models.

Feature selection

The number of available features (342) is quite large compared to the number of records in the train set (593, being 80% of 742). This introduces the risk of fitting a model that doesn’t generalize well to new data. Reducing the number of features may help increase the performance as found in the initial stage when using all features on multiple models. Results of both forward and backward iterative feature selection are presented in Appendix D Table 29. Feature selection results in performance improvement compared to the baseline using all features. In case of Disgust the improvement is very small though.

In none of the cases the forward and backward approaches converge to the same feature set. In case of Anger, Joy and Sentiment, the forward approach gives best results. For Disgust, the backward iterative approach gives best results. In each case the backward approach produces quite some more selected features than in case of forward approach. For Joy the difference in number of features between forward and backward approach is the smallest 9 versus 15. Joy has the largest number of features in forward approach and has only marginal performance difference with the backward approach.

For Anger, Disgust and Joy, all features from forward approach are also included in the backward approach model. For Sentiment, this is the case too with one exception. The number of exclamation marks (countExclamation) is included in the forward model but not in the backward model.

Feature importance

After feature selection, the next step is determining feature importance. As calculating feature importance directly is not possible for the ensemble method used, a simple leave-one-out approach is used as described in paragraph 5.4.5. Table 12 shows the 50-run-average kappa difference where the identified feature is removed from the input features. Each feature is tested with the same data. Significance is tested using a paired t-test. While some difference can be seen between different emotions and sentiment, tf-idf and Doc2Vec are key features. Doc2Vec significantly improves performance for Anger (p<0.01), Disgust (p<0.05) and Sentiment (p<0.05). While tf-idf significantly improves Anger (char p<0.01, word p<0.01), Joy (char p<0.01, word p<0.01), Sentiment (char p<0.10). Joy has the only feature set where other features than Doc2Vec and tf-idf are significantly important; number of exclamation marks and number of words in NRC emotion lexicon.

	Anger	**Disgust**	**Joy**	**Sentiment**
Baseline, all features	0.51	0.44	0.61	0.42
char_Tfidf (100)	**0.029*****	0.015	**0.053*****	**0.013***
countExclamation			**0.019*****	-0.005
countNRC (7)		0.000	**0.038*****	-0.004
Doc2Vec (100)	**0.147*****	**0.037****		**0.023****
word_Tfidf (100)	**0.031*****	0.007	**0.074*****	0.002

Table 12: 50 run average difference in kappa compared to baseline when excluding feature. * Significant p<0.10; ** Significant p<0.05; *** Significant p<0.01. All significant features are indicated in bold. Depending on the emotion or sentiment, Doc2Vec or tf-idf are key features

A few features in Table 12 have negative performance impact. This impact is not significant though. The kappa difference of each of the features does not add up to the baseline kappa. Any performance improvement that is covered by more than one feature or improvements related feature interaction is not directly captures by the method used. Full feature importance results can be found in Appendix D Table 30.

6.2.3 Model evaluation

Now that the models and feature sets are selected based on kappa performance, a more extensive model evaluation with other metrics is appropriate. Using a different train (80%) /test (20%) data split each time, each model has been tested 100 times.

Figure 4 shows that for every model there is a large spread in kappa across multiple runs (spread between 0.36 and 0.47). Still, the inter quartile range (IQR) is not that big. The detailed descriptive for Figure 4 can be found in Appendix D Table 31. Joy has the best kappa performance of 0.61 which indicates substantial agreement, while Anger (0.51), Disgust (0.43) and Sentiment (0.43) have fair agreement.

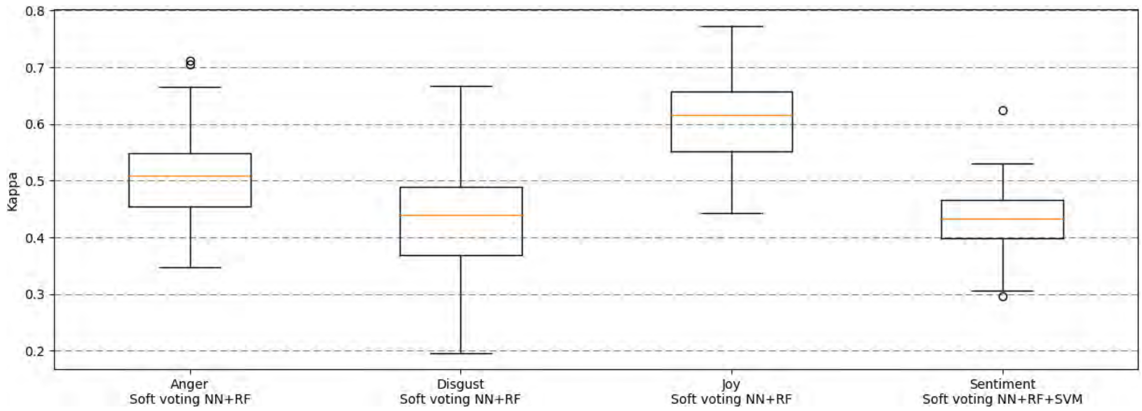


Figure 4: Kappa value of 100 runs of best model and feature selection. Each model uses oversampling and majority vote and 0.5 threshold (emotions only). All models have large spread in kappa values. Joy has substantial agreement

To get a better understanding of the performance of each of the models, the confusion matrix, precision, recall and F1-measure can be found in Appendix D Table 32 - Table 36.

Disgust has the highest average recall (0.89) and precision (0.89). However, this difference is caused by the imbalance in the dataset which is bigger for Disgust (7.8x) than for Anger (3.4x) or Joy (4.1x). The minority class precision, recall and F1 are a better measure for model performance. The performance for both Joy (F1: 0.68) and Anger (F1: 0.61) is reasonable while F1 for Disgust is only 0.49.

Sentiment analysis on categories None (F1: 0.58), Pos (F1: 0.63) and Neg (F1: 0.69) has reasonable performance (see Appendix D Table 35, Table 36). Category Mix (F1: 0.05) performance is not good. Since this category has only support of seven cases, the impact on overall precision (0.60), recall (0.60) and F1 (0.61) is not that big.

6.2.4 Comparison to benchmark

With the results of previous paragraphs the next step can be taken to come to a conclusion with regards to “(2) does a domain specific sentiment detection machine learning model trained on a small set of emails, outperform a general model for sentiment analysis”

Table 13 shows the results and comparison of the sentiment analysis using a domain specific machine learning versus the benchmarks using standard automated sentiment analysis. Furthermore, the machine learning results are compared to average annotator agreement. The domain specific model outperforms both NRC benchmark as well as IBM NLU benchmark significantly (p<0.01) on all categories. It should be noted that Fear and Sadness categories were discarded for feature selection due to insufficient performance, but still performance significantly outperforms the benchmark performance.

	Sentiment	Emotions				
		Anger	Disgust	Fear	Joy	Sadness
Domain specific model	0.43	**0.51**	**0.43**	0.13	**0.61**	**0.36**
Benchmark 1: NRC lexicon	0.09	0.33	0.31	0.02	0.06	0.27
Benchmark 2: IBM NLU	0.22	0.32	0.03	-0.01	0.46	0.14
Avg annotator agreement	**0.45**	0.49	0.21	**0.24**	**0.61**	0.31

Table 13: Comparison of Machine learning model kappa with manual annotator agreement and benchmarks for automated sentiment analysis. Domain specific model outperforms both NRC benchmark and IBM NLU benchmark significantly (p<0.01) on all categories. Bold indicates best performance.

Domain specific model and average annotator agreement have a different basis. The domain specific model compares performance against the majority vote of three annotators per email while average annotator agreement is an average across pairs of annotators. Therefore, the comparison may not be fully justified. Still, comparison between the domain specific model and human annotator agreement, shows that machine learning model performance on Sentiment, Anger, Joy and Sadness is comparable to human annotators. While on Disgust, machine learning outperforms human annotation.

6.3 Affect analysis results

Results presented in this section are the basis to answer the research question: (3) do CS response email features have predictive value for sentiment of a customer?

6.3.1 Affect data

A first step in modelling and understanding affect analysis in relation to customers, is to set up and understand the data involved. Affect analysis is done on records that are constructed from three subsequent emails in a client-customer support-client email conversation. The two client emails are enriched with annotated sentiment and emotions. In total 455 conversation records are available. While much more emails are available, the number was limited due to the limited number of annotated emails.

To see the dynamics in sentiment and emotions following customer support email, cross tabs are created displaying original sentiment and emotions versus sentiment and emotions after Customer Service email as displayed in the follow-up email of the customer.

Table 14 shows the sentiment of the originating customer email versus the sentiment of the second customer email. In case the originating sentiment is from categories Neg, None or Pos, the most likely category of the post CS response sentiment is the same as the originating category. Still, most customers do have different sentiment after CS response.

		Post CS response sentiment				
		Mix	Neg	None	Pos	Total
Originating Sentiment	**Mix**	0	5	4	13	**22**
	Neg	9	85	65	59	**218**
	None	7	31	63	51	**152**
	Pos	2	15	19	27	**63**
	Total	**18**	**136**	**151**	**150**	**455**

Table 14: Customer sentiment before and after CS response. Post CS response More customers have None or Pos sentiment.

From Table 14 it can be concluded that most customers (57%) starting with negative sentiment move away from negative sentiment. More customers starting with mixed sentiment or without sentiment, move towards positive sentiment rather than negative sentiment. Customers moving away from positive sentiment, more likely move towards neutral than negative sentiment.

Appendix D Table 37 through Table 41 show cross tables of the starting emotion versus the final emotion. Customers in general move away from the negative emotions Anger, Disgust, Fear, and Sadness. The total number of customers with these negative emotions is always lower post CS response. While the number of customers displaying Joy is increasing post CS response. More customers move from No Joy to Joy than vice versa.

Predicting customer sentiment and emotions following an email conversation could be easy if there is a very strong relation to the emotions of the customer as displayed in the starting email. However, supported by the cross tables, predicting customer affect is not as easy as just taking the starting sentiment and emotions of a customer.

6.3.2 Model selection

To perform affect analysis, the same selection of models was chosen as for sentiment analysis: Neural network, Naive Bayes, Support Vector Machines, Random Forest, Soft Voting ensemble, Label Power RAkEL (emotion only). In case of sentiment and single label emotion modelling (as opposed to multilabel), each of these models are executed using either data with or without oversampling of minority classes. Opposed to sentiment analysis, for affect analysis it was not possible to test optimized threshold or test combined annotation on full agreement. Testing with records with full annotator agreement only, would result in the loss of (too) much data. While testing for optimal threshold would require splitting the data set in a train, test and validation set, leaving little data for training machine learning models.

All model and setting combinations have been run 30 times with the same data to limit the influence of any specific results related to random stratified train / test data split. The overall number of features (681) is larger than the number of available data records (364, 80% of 455). To avoid overfitting when training the models, the word-based tf-idf and Doc2Vec features from both customer mail and CS email were excluded in the initial phase of testing the best model and setting. This reduces the number of features to 281, which brings it below the number of available training records.

Affect analysis results on sentiment are presented in Table 15. Performance is measured using accuracy metric. The best performing model is Random Forest without oversampling of the minority classes. For all but one model (Neural Net), no oversampling outperforms the models with oversampling.

Model	Oversampling	No oversamp
Neural Net (NN)	0.36	0.35
Naive Bayes (NB)	0.33	0.33
SVM	0.39	0.39
Random Forest (RF)	0.49	**0.50**
Voting RF+NB+SVM+NN	0.42	0.44
Voting RF+NB+SVM	0.39	0.40
Voting RF+NB	0.39	0.40
Voting RF+SVM	0.47	0.48

Table 15: Affect Sentiment - Average accuracy (30 runs) per scenario and ML model. Bold indicates the best accuracy score. In general, no oversampling works better than oversampling.

In total 20 different models or ensembles and settings have been tested on five different emotions to find the best performing model per emotion. Table 16 the model performance for shows a selection of the models using the F1-measure. Full results can be found in Appendix D Table 42.

Model	Anger	Disgust	Fear	Joy	Sadness
Single label without oversampling					
Naive Bayes	0.41	0.26	0.03	0.45	**0.29**
Single label with oversampling					
Neural Net	0.38	0.22	**0.04**	0.41	0.23
Naive Bayes	0.42	**0.26**	0.02	**0.46**	0.28
Voting NN+NB	**0.42**	0.26	0.02	0.45	0.28
Multilabel					
RAkEL – Random Forest (Label Power)	0.07	0.01	0.00	0.00	**0.30**

Table 16: Average F1-measure (30 runs). Bold indicates best performing model for specific emotion. Performance is very low for Fear. Best model for Sadness is multilabel with substandard performance for other emotions. Second best model is selected for Sadness.

Different emotions have very different best performing models. While Fear performs best with a Neural Net and Disgust and Joy perform best with Naive Bayes, Anger performs best with a voting ensemble of both methods. Sadness performs best with the multilabel model method RAkEL. However, RAkEL is not selected in the next steps since the multilabel model has substandard performance on all other emotions. The next best model for Sadness is a Naive

Bayes model which performs best without any oversampling. The difference with oversampling is small though. The best models for other emotions do you use oversampling. Performance for Fear is very low (max F1 0.039) for every model. Fear is therefore discarded for further affect analysis.

6.3.3 Feature selection and feature importance

Available features

Creating a conversation record by combining the incoming customer email and CS response email, results in many features. In total 681 input features are available:

- 100 (customer) + 100 (CS) features related to **character-based tf-idf**
- 100 (customer) + 100 (CS) features related to **word-based tf-idf**
- 100 (customer) + 100 (CS) **Doc2Vec** features
- 2 features of one hot encoded **Case Origin** (same for both customer and CS)
- 10 features of one hot encoded **Contact Reason** (same for both customer and CS)
- 7 (customer) + 7 (CS) features of one hot encoded **dayOfWeek**
- 4 (customer) + 4 (CS) features of one hot encoded **partOfDay**
- 7 (customer) + 7 (CS) features related to **NRC** lexicon categories (countAnger, countPos etc)
- **threadItem** (only CS threadItem taken since it is a sequence number)
- 11 (customer) + 11 (CS) **other features** such as lengthMessage, countWords
- 5 combined **annotated emotions** from originating customer email (Anger, Joy, Disgust, Fear, Sadness, Sentiment)
- 4 features of one hot encoded combined **annotated Sentiment** from customer email
- **Response time**, time between receive of customer email till response of CS rounded by days

The combined annotation labels of originating customer email which were used as a response variable for sentiment analysis, are now used as input feature for affect analysis. The Response Time variable is a new variable related to conversations. All other features are the same as used for sentiment analysis.

Feature selection

Using the models selected in the previous paragraph, a forward and backward iterative approach has been executed to further increase model performance and to determine which features are most important to model performance. The results of feature selection can be found in Appendix D Table 43. In most cases (Anger, Disgust, Sadness, Sentiment) the forward approach results in best performance. Only in case of Joy, the backward approach gives best results.

Since Joy's best model is a result from a backward approach, the number of features is much larger than the other models; 36 out of 40. For Anger, Disgust, Sadness, Sentiment respectively five, five, five and six features are selected. Since this thesis tries to answer the question if there is any relation between Customer Service emails and customer affect, the CS email features are of main interest. For sentiment and each emotion, at least two CS email features were included during the feature selection phase. Any other features come from: shared features, customer email features or conversation features. However, it should be noted that selected features may not be important since the method results in a set that's highly fit towards the data splits made.

Feature importance

Validation of the feature importance was done using 50 different stratified train (80%) and test (20%) data splits. Feature importance was determined using a paired t-test between the baseline containing all features and the model result which excluded the specific feature. The full results of the leave-one-out test can be found in Appendix D Table 44.Table 17 shows all noteworthy features and other the notable features per sentiment and emotion model.

As expected the baseline performance of the feature importance runs is lower than the (over)fitted performance of the feature selection runs. It is notable that in the previous phase multiple features have been selected that now have a negative performance effect. In the case of diversityRatio from both CS and customer email, the effect is even significant (p<0.10).

While the simple method used cannot precisely grasp full feature importance, it can give a good indication of the key features. The most important features for Anger and Disgust are respective originating customer Sentiment and Emotions. The second most important feature for both is the thread sequence number. From the CS email the features dayOfWeek (Anger) and lengthMessage (Disgust) are significant too. The most significant feature for the Joy and Sadness models is the word-based tf-idf of the CS email. While for the Sentiment model the character-based tf-idf from the CS email is most important. Surprisingly the Response Time feature is not important for any of the models, while one might expect that for example a large response time might spark Anger or Sadness and a short response time might result in Joy.

	Anger	**Disgust**	**Joy**	**Sadness**	**Sentiment**
Baseline. all features	0.45	0.31	0.50	0.31	0.49
Shared features					
threadItem	**0.042*****	**0.028*****	-0.006		
CS email features					
char_Tfidf (100)			0.004	-0.005	**0.059*****
dayOfWeek (7)	**0.020*****		-0.001		
diversityRatio	-0.005	0.004	**-0.011***		
lengthMessage		**0.016***	-0.001		
word_Tfidf (100)			**0.034*****	**0.026****	
Client email features					
char_Tfidf (100)			**-0.011***		
Emotions (5)		**0.035*****	-0.004		**0.013****
Sentiment (4)	**0.126*****		-0.005		**0.010***

Table 17: Added value of single feature to affect model performance. Selection of most notable features. *** significant p<0.01, ** significant p<0.05, * significant p<0.10. All significant features are indicated in bold. The backward feature selection has wrongfully also selected two features for joy with significant negative impact.

6.3.4 Model evaluation

To get a better understanding of the affect model performance as found in the previous phases, the models have been run 100 times each time with a different train (80%) and test (20%) stratified split. The overall results can be found in Figure 5 and Appendix D Table 45.

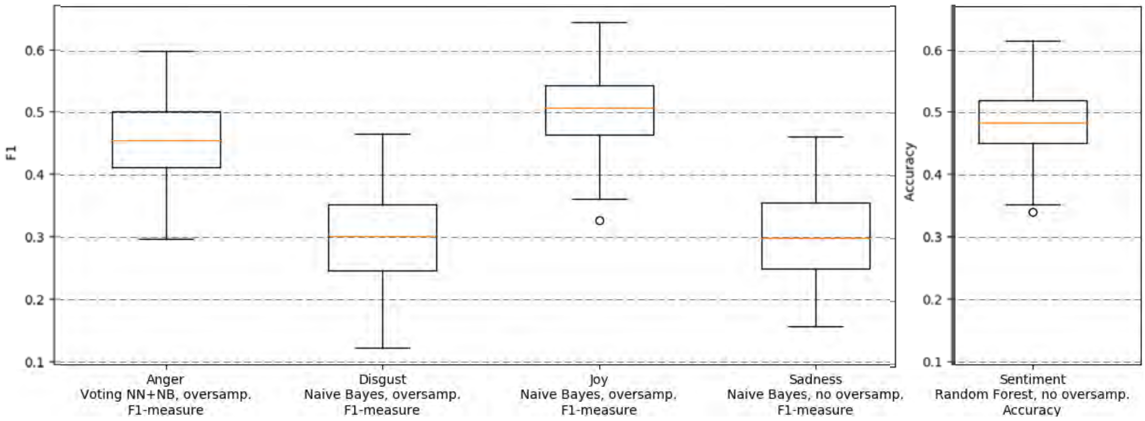


Figure 5: Average F1 or accuracy performs of 100 runs of the best affect analysis model and selected features. Overall the F1 values (emotions) and accuracy (sentiment) are on the low side.

The confusion matrix and precision, recall, F1 metrics for each of the modelled affect emotions in Appendix D Table 46 through Table 49 show that performance is low. Joy has the highest F1-score of 0.50. However, this was to be expected due to the nature of the imbalance of the classes. For Joy, the imbalance is the smaller (2.7x) than Anger (3.6x), Disgust (7.3x) and Sadness (5.1x). F1 measure for the other emotions is 0.45 (Anger), 0.30 (Sadness, Disgust).

With regards to the affect sentiment model, the performance is shown in Appendix D Table 50. The Mix category has a very small support; only 4 in the test set. None of the Mix records have been labelled correctly throughout each of the 100 runs, resulting in a recall of 0.00. The precision and recall of the other labels is about the same: Neg (0.47, 0.47), None (0.50, 0.55) and Pos (0.48, 0.50). This results in an accuracy of 0.49.

6.3.5 Comparison to benchmark

To assess the added value of the trained machine learning models, a comparison is made with two simple benchmarks. Together with the feature importance from paragraph 6.3.3, this forms the basis for answering the research question “(3) do CS response email features have predictive value for sentiment of a customer?”

The first benchmark is based on predicting a single class for all conversation records. For sentiment, the accuracy metric is determined with all records labelled with the majority category ‘None’. For emotions, the minority F1 measure is determined for all records labelled with all emotions being present. The second benchmark is based on predicting customer affect to be the same as the customer sentiment and emotion as annotated in originating email. The benchmark comparison can be found in Table 18.

	Sentiment	**Emotions, F1-measure**			
	accuracy	**Anger**	**Disgust**	**Joy**	**Sadness**
Affect model	0.49	0.45	0.30	0.50	0.30
Benchmark 1:					
single category	0.33	0.36	0.22	0.44	0.28
(p-value)	(<0.01)	(<0.01)	(<0.01)	(<0.01)	(0.01)
Benchmark 2:					
start emotion	0.38	0.44	0.29	0.20	0.28
(p-value)	(<0.01)	(0.13)	(0.07)	(<0.01)	(0.01)

Table 18: Comparison of machine learning based affect analysis versus benchmarks. In each case performance is better than both benchmarks. Only for Sentiment, Joy and Sadness it is significant. Performance difference compared to benchmark is small and may not be relevant.

The largest difference between benchmark and affect model performance can be seen with Sentiment. The difference is significant (p<0.01). The affect model for Joy also significantly (p<0.01) outperforms the benchmark. The Sadness model does perform better than the benchmark (p=0.01) but the difference is that small that it may not be relevant. Both Anger and Disgust models do perform better than benchmark 2 but not significantly with p>0.05.

7 Discussion

7.1 Discussion on Annotation

While the email sentiment annotation process was not the primary focus of this study, still it was an important part. The email sentiment annotation process revealed that humans find it difficult to agree on emotions displayed in email text. This was confirmed with the χ^2test on the distributions. Best agreement was achieved on Sentiment, Anger and Joy. Only for Joy the agreement could be described as substantial (average kappa 0.61) while for the other categories agreement was only fair or moderate at best.

For Disgust, Fear and Sadness the kappa values show that these categories pose issues. It is not surprising though that agreement is an issue. Even in day-to-day situations it is all too common that one misinterprets an email or any other text message. Still, agreement between annotators is better than an annotator and the NRC and IBM NLU benchmarks. It may be that NRC and IBM NLU annotators had different interpretation of the emotions compared to the annotators used for this thesis. This situation is even seen within this thesis with significant differences in annotator distributions. Another reason for lack of agreement with the IBM NLU benchmark could be the fact that IBM NLU has been trained on millions of news sources. This domain may not generalize very well to emails.

Agreement needs to be as high as possible to have reliable outcomes which can be used to train machine learning models. Due to the low average kappa of Disgust, Fear and Sadness, these emotions are not likely candidates. However, Disgust did prove to give reasonable results in the machine learning phase. As some annotators seldom or never annotated Disgust, kappa differs considerably between annotator pairs. Using the majority vote apparently filters out the deviant annotators. Bhowmick et al. (2010) partly tackled the issue by combining Anger and Disgust into a single category. Also, even if agreement lacks, a category may still prove useful if correlation exist between categories.

7.2 Discussion on Sentiment analysis

The best performing sentiment analysis model for emotions is a soft voting ensemble of Neural Net and Random Forest. For sentiment, the best model also includes SVM in the ensemble. A soft voting ensemble usually works best when you multiple well performing classifiers. In this case it works well despite individual classifier performance. The reason may be that the classifiers are very different and therefore complement each other.

From the annotation phase, lack of agreement on Disgust, Fear and Sadness was expected to cause poor results in sentiment analysis. Fear and Sadness indeed lacked satisfactory results. The machine learning performance was comparable to human agreement on Anger, Joy and Sentiment. On Disgust machine learning even outperformed human agreement. It shows that a model trained on a majority vote, works better than just comparing single annotators.

The size of the set of annotated emails (742) may have decreased model performance. To avoid overfitting, the number of features was limited accordingly. Tf-idf was an important feature in some models, but only 100 tf-idf features were used while the rest was discarded. With additional annotated data available, these tf-idf features may prove useful. Furthermore, only selecting data which had full agreement was not successful since it reduced the available data by 40%. Also, on the sentiment part, the number of mixed sentiment records was very low, which reflected in inferior performance on this class. Finally, due to the small test (149 records) model performance had a large spread depending on the data split. Increasing this set would be at the cost of the training set and would have had direct impact on training performance.

The emotion data is imbalanced. To resolve this issue oversampling turns out to be the best solution. Another option, which makes use of the correlation present between emotions, is the multilabel model RAkEL. This model gave good performance in a few studies on affect analysis (Bhowmick, 2009; Ye et al., 2012). In this thesis, the model was not the best performing model. This may be again caused by the amount of annotated data available. Since the data makes use of label power sets, the frequency of some labels would be low.

Using a simple forward and backward selection method for feature selection proved to have its flaws. After selection, still some features were present that have significant negative impact. Evidently feature interaction is complex to such an extent that more advanced feature selection method is more appropriate such as genetic algorithm (Yang and Honavar, 1998).

While this study showed that sentiment and emotions can be successfully identified using domain specific machine learning models, initial results were even better. These superior results were unfortunately the result of a bug in scikit-learn 0.18.1[4] with regards to stratified sampling of a multilabel data set which caused overlap in train and test set. A workaround resolved the issue.

Limitations

The best models easily outperform the benchmarks. However, it is not sure if the results can be generalized to other domains than email. Also, only 1 commercial tool has been tested (IBM NLU). Other commercial tools may perform better on the email domain.

Model performance is based on combined annotated sentiment. But, as we have seen that agreement is an issue, one can question if the combined annotated sentiment correctly reflects the email sentiment. This sparks the question to how reliable outcome of the model itself is when trained on this combined sentiment. However, we do know that with the majority vote method used, every annotated record has been agreed on by at least two out of three annotators.

It was found that the annotated data sample was not representative for the full set for feature day of the week, thread sequence number and two others (see Appendix D Table 25). Day of

[4] https://github.com/scikit-learn/scikit-learn/issues/9037

the week and thread sequence are however key features in affect analysis models of Anger respective Disgust. This means that the models might not generalize well.

7.3 Discussion on Affect analysis

As affect analysis is alike sentiment analysis with only difference being the subject of the sentiment, the discussion of sentiment analysis is applicable here too.

Model performance for Affect analysis, though significantly better than the benchmark in some cases, is low across the entire range of models. In this study, the highest accuracy is 0.70 for the Disgust model. Yang et al. (2009) found an accuracy up to 0.78 on predicting reader emotion. The results however may not be comparable due to the differences between the studies: news articles instead of emails, other emotion categories (awesome, heart-warming and six others). Furthermore, accuracy is heavily impacted by any class imbalance. This can work both ways; low Disgust model accuracy due to high class imbalance and higher Joy model performance due to moderate class imbalance.

The actual customer affect in this study was indirectly determined using a customer response email. This setup may result in a bias on certain emotions. Satisfied customers may be less likely to respond to customer support than angry customers. Furthermore, it could be the case that any emotion found in that email is in fact not the emotion that was triggered when reading the CS email response:

- negative mood from **earlier** unpleasant **events** may linger (Keltner et al., 1993) which in turn can be reflected in the customer email emotions
- A **time gap** between de CS response and the subsequent response by the customer allow for a period of calming down (Kendall et al., 1975) which will reduce the primary emotion
- emotion as annotated may **not** be the **true emotion** of the customer

Though not easy to realize, collecting actual customer affect through a more direct assessment can provide a better base to build a machine learning model on.

The affect data analysis shows that customers move away from negative sentiment and emotions and move towards positive sentiment and emotions. We can conclude that the CS department has a positive influence on customer affect.

8 Conclusions

In this study sentiment analysis and affect analysis was performed using machine learning models on an email dataset. This study aims to answer the main research question: Which aspects of a Customer Service email response affect customer sentiment? This research question is divided into the following sub-questions; (1) what sentiment can be detected in customer emails, (2) does a domain specific sentiment detection machine learning model trained on a small set of emails, outperform a general model for sentiment, (3) do CS response email features have predictive value for sentiment of a customer?

(1) What sentiment can be detected in customer emails?
The findings in this study show that voting ensembles of Neural Net, Random Forest and optionally Support Vector Machines can successfully identify **Sentiment** (kappa 0.43), **Anger** (kappa 0.51), **Disgust** (kappa 0.43) and **Joy** (kappa 0.61) in the domain they were trained for. Also, Sadness (kappa 0.36) and Fear (kappa 0.13) can be identified to a lesser extent, but the achieved performance makes these models less usable.

(2) Can a domain specific sentiment analysis outperform a general model?
Each of the **domain specific models significantly outperform** both IBM Natural Language Understanding (p<0.01) as well as NRC Emotion Lexicon (p<0.01) benchmarks. NRC Lexicon is however useful to generate word counts of affective words which are significant features to predict Joy.
It was a surprise that the domain specific models outperformed IBM NLU to the degree that they did. Apparently, while IBM NLU has been trained on millions of news articles, domain specific training on based on a small set (742) of annotated emails seems more effective. It should be noted that a larger set of 76k emails was used to generate Doc2Vec features, but still the amount of training data for the machine learning models was considerably less than used for IBM NLU.

(3) Do CS response email features have predictive value for sentiment of a customer?
The main research question regarding predicting customer affect and the role of Customer Service email features in the models, does not have a clear-cut answer. While the following models perform **significantly better than** the **benchmark:**

- **Sentiment** (p<0.01),
- **Joy** (p<0.01)
- **Sadness** (p=0.01)

Anger (p=0.13) and Disgust (p=0.07) model performance difference is not significant. Also, performance on Fear is very low (F1 0.04). In fact, the **overall performance** could be considered **low**.

The analysis of feature importance revealed that, depending of which model is analysed, the following **CS features** are important:

- day of the week (Anger),
- length of the message (Disgust),

- word-based tf-idf (Joy, Sadness),
- character-based tf-idf (Sentiment).

However, the machine learning models used (Random Forest (Sentiment); Naive Bayes (Disgust, Joy, Sadness); Voting ensemble Neural Net +Naive Bayes (Anger)), do not allow comprehending how these features exactly influence predictions.

Future work
In future research, increasing **annotator agreement** may increase model performance. Agreement can be increased by improving individual annotations results in any of the following ways:

- Improve annotation **instructions**. The instructions as can be found in Appendix A: Annotation guidelines, relied very much on the natural understanding of emotions of each annotator. The results however show that different annotators have different interpretation. Furthermore, some more direction could have been given on:
    - Only assess explicit language used versus more freely interpretation of emotion behind the email
    - Subject of the emotion. Is it directed towards Customer support or another party?
- Only use (near) **native speakers** for annotation.
  In this study only non-native speakers were available for the annotation task. While their level of English was more than adequate, it may still be that they miss some subtleties in the text that native speakers can better interpret. It should be noted that this may only benefit if customers are native speakers too.

Though above suggestions may easily improve the combined annotation result, unfortunately they all come at the cost of additional annotation effort. It is therefore a matter of weighing the pros and cons to determine the best approach in the situation at hand.

Other possibilities to improve the annotated email set are related to manner of selecting and combining annotations from different annotators:

- Only use emails where **full agreement** between annotators exists in combinations with annotating more emails. Many email records will not be used in case only emails with full agreement are used. Therefore, more annotated emails would be required to have sufficient data available for training machine learning models.
- Use **more annotators** per email. Making use of the wisdom of the crowd. While agreement may be low between annotators, still a good annotation result can be achieved by labelling an email with the majority label.
- **Discard annotators** that annotate significantly different from other annotators. When using multiple annotators, each taking care of annotating just a part of the emails, a comparison can be made of the annotation distributions. Any annotator results significantly deviating from the rest could be removed.

To improve sentiment analysis and affect **analysis performance,** future research may implement the following improvements:

- **Tune number** of **tf-idf feature**s and **Doc2Vec features**. For some models, the tf-idf and Doc2Vec feature proved to be important. Extending the number of features might bring additional performance, but this requires more data to avoid the number of features not being in balance with the number of data records.
- Perform **advanced feature selection**. The feature selection in this study was a simple one, changing the feature set with one feature at a time. This method quickly results in a reasonable feature set, but cannot cope with very complex multi feature interactions. To more broadly explore the full range of feature sets, one could resort to for example a genetic algorithm approach.
- Use **pre-trained Word2Vec** models. The Doc2Vec features are fully based on training on 76k domain specific emails. Pre-trained Word2Vec models might perform better than models trained domain specific emails.
- **Tune** sentiment analysis **model parameters**. Parameter tuning can be a very time-consuming activity. In this study, most of the time was spent on finding suitable machine learning models and selecting key features. Parameter tuning may improve performance of the currently selected models.
- Experiment with **other ensemble models**. Combining machine learning models in ensembles, often has proven to increase performance. Especially since correlation is present between different emotions in emails, combining dedicated emotion models into a single multilabel model may proof fruitful. However, training ensembles requires data not previously used in training the other models. With the limited annotated data available, this was not an option in this study.
- Extend train data set through **self-labelling**. The dataset with unlabelled emails is over 38 thousand emails. These emails could be utilized for supervised learning if labels would be present. This can be done by training a sentiment analysis model on the small available set and then label other emails using the prediction of the trained model. In this study, such an approach did not produce good preliminary results and was therefore not further investigated. However, when model quality is improved, self-labelling will automatically improve too.
- For sentiment use a **metric** that uses **class weights**. A misclassification of a sentiment class differs in severity. Classifying for example a Positive class as None has less impact than classifying it as Negative. A metric that takes this ordinality into account makes the metric a better indicator for model performance.

Final remarks

The practical applications of the results of this thesis can be found in multiple areas. The sentiment analysis approach can be applied not only to emails but also to other text based communication. Also, the generic approach is not limited to area of Customer Supports or sporting goods. If we stick to the investigated area, an application of the model could be to use it as a basis for routing emails, e.g. prioritizing angry emails to prevent even more anger. Or use it to select a specific template for a response email that does justice to the emotions felt by the customer. The affect analysis model could be used to assess the response email with regards to the customer affect. Using predicted probabilities of each emotion a customer support employee could opt to adjust the email to reduce certain emotion probabilities.

This study did not focus on the implementation of any of the models in production. Decisions around technical implementation, which data to use for training, how many often to re-train models, timing of model training and predictions are just a few open items before implementation could be realized.

To date, no studies have been published on affect analysis in the email domain. This study extents the knowledge in this area. While the study does have its limitations and areas of improvement, the models in some cases have significant better performance than the benchmark. This shows that the approach on affect analysis does have its potential.

9 References

Aman, S., & Szpakowicz, S. (2007). Identifying expressions of emotion in text. In *Text, speech and dialogue* (pp. 196-205). Springer Berlin/Heidelberg.

Artstein, R., & Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, *34*(4), 555-596.

Bennett, E. M., Alpert, R., & Goldstein, A. C. (1954). Communications through limited-response questioning. *Public Opinion Quarterly*, *18*(3), 303-308.

Bermingham, A., & Smeaton, A. F. (2009, July). A study of inter-annotator agreement for opinion retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (pp. 784-785). ACM.

Bhowmick, P. K. (2009). Reader perspective emotion analysis in text through ensemble based multi-label classification framework. *Computer and Information Science*, *2*(4), 64.

Bhowmick, P. K., Basu, A., & Mitra, P. (2010). Classifying emotion in news sentences: When machine classification meets human classification. *International Journal on Computer Science and Engineering*, *2*(1), 98-108.

Bosco, C., Patti, V., & Bolioli, A. (2013). Developing corpora for sentiment analysis: The case of irony and senti-tut. IEEE Intelligent Systems, 28(2), 55-63.

Bradley, M. M., & Lang, P. J. (1999). *Affective norms for English words (ANEW): Instruction manual and affective ratings* (pp. 1-45). Technical report C-1, the center for research in psychophysiology, University of Florida.

Byron, K. (2008). Carrying too heavy a load? The communication and miscommunication of emotion by email. *Academy of Management Review*, *33*(2), 309.

Calvo, R. A., & Mac Kim, S. (2013). Emotions in text: dimensional and categorical models. *Computational Intelligence*, *29*(3), 527-543.

Chang, Y. C., Chen, C. C., Hsieh, Y. L., Chen, C. C., & Hsu, W. L. (2015). Linguistic Template Extraction for Recognizing Reader-Emotion and Emotional Resonance Writing Assistance. In *ACL (2)* (pp. 775-780).

Charte, F., Rivera, A. J., del Jesus, M. J., & Herrera, F. (2015). Addressing imbalance in multilabel classification: Measures and random resampling algorithms. *Neurocomputing*, *163*, 3-16.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, *20*(1), 37-46.

Ekman, P. (1973). Cross-cultural studies of facial expression. *Darwin and facial expression: A century of research in review*, *169222*.

Goleman, D. (1996). Emotional Intelligence. Why It Can Matter More than IQ. *Learning*, *24*(6), 49-50.

Gupta, N., Gilbert, M., & Fabbrizio, G. D. (2013). Emotion detection in email customer care. *Computational Intelligence*, *29*(3), 489-505.

Ho, T. K. (1995, August). Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on* (Vol. 1, pp. 278-282). IEEE.

Keltner, D., Ellsworth, P. C., & Edwards, K. (1993). Beyond simple pessimism: effects of sadness and anger on social perception. *Journal of personality and social psychology*, *64*(5), 740.

Kendall, P. C., Nay, W. R., & Jeffers, J. (1975). Timeout duration and contrast effects: A systematic evaluation of a successive treatments design. *Behavior Therapy*, *6*(5), 609-615.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *biometrics*, 159-174.

Le, Q. V., & Mikolov, T. (2014, June). Distributed Representations of Sentences and Documents. In *ICML* (Vol. 14, pp. 1188-1196).

Lemaitre, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, *18*(17), 1-5.

Lin, K. H. Y., Yang, C., & Chen, H. H. (2008, December). Emotion classification of online news articles from the reader's perspective. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on* (Vol. 1, pp. 220-226). IEEE.

McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia medica*, *22*(3), 276-282.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mohammad, S. M. (2016). A practical guide to sentiment annotation: Challenges and solutions. In Proceedings of NAACL-HLT (pp. 174-179).

Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, *29*(3), 436-465.

Mohammad, S. M., & Yang, T. W. (2011, June). Tracking sentiment in mail: How genders differ on emotional axes. In *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis* (pp. 70-79). Association for Computational Linguistics.

Moraes, R., Valiati, J. F., & Neto, W. P. G. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*, *40*(2), 621-633.

Plutchik, R. (1980). *Emotion: A psychoevolutionary synthesis*. Harpercollins College Division.

Roberts, K., Roach, M. A., Johnson, J., Guthrie, J., & Harabagiu, S. M. (2012, May). EmpaTweet: Annotating and Detecting Emotions on Twitter. In *LREC* (pp. 3806-3813).

Scott, W. A. (1955). Reliability of content analysis: The case of nominal scale coding. *Public opinion quarterly*, 321-325.

Strapparava, C., & Valitutti, A. (2004, May). WordNet Affect: an Affective Extension of WordNet. In *LREC* (Vol. 4, pp. 1083-1086).

Tsoumakas, G., & Vlahavas, I. (2007). Random k-labelsets: An ensemble method for multilabel classification. *Machine learning: ECML 2007*, 406-417.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer science & business media.

Volkova, E. P., Mohler, B. J., Meurers, D., Gerdemann, D., & Bülthoff, H. H. (2010, June). Emotional perception of fairy tales: achieving agreement in emotion annotation of text. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*(pp. 98-106). Association for Computational Linguistics.

Wang, G., Sun, J., Ma, J., Xu, K., & Gu, J. (2014). Sentiment classification: The contribution of ensemble learning. *Decision support systems*, *57*, 77-93.

Wiebe, J., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language. Language resources and evaluation, 39(2), 165-210.

Yang, C., Lin, K. H. Y., & Chen, H. H. (2009, September). Writer meets reader: Emotion analysis of social media from both the writer's and reader's perspectives. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on* (Vol. 1, pp. 287-290). IEEE.

Yang, J., & Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems and their Applications*, *13*(2), 44-49.

Ye, L., Xu, R. F., & Xu, J. (2012, July). Emotion prediction of news articles from reader's perspective based on multi-label classification. In *Machine Learning and Cybernetics (ICMLC), 2012 International Conference on* (Vol. 5, pp. 2019-2024). IEEE.

Appendix A: Annotation guidelines

Instruction for Annotating Emotion and Sentiment in Text.

Introduction

Picture an A.I. chatbot automatically responding to a customer email regarding an information request or complaint.
The research goal is to automatically detect emotions and sentiment in text and identify which aspects from a (Customer Service) response email affect these emotions and sentiment.
In order to develop a computer system to do this, we need people to annotate (mark up) texts with relevant properties, such as whether the language expresses a certain emotion. Below are descriptions of the properties we want you to annotate. There are no formal criteria for identifying the properties. Please use your human knowledge and intuition to identify the information. The system will then look at your answers and try to figure out how it can make the same kinds of judgments itself.

In case of any questions following this instruction, during the annotation task or otherwise, please contact email address.

Annotation properties

Sentiment

To annotate “Sentiment”, please indicate what kind of language the speaker is using.
Note: only 1 category can be selected.

Pos the speaker is using positive language, for example, expressions of support, admiration, positive attitude, forgiveness, fostering, success, positive emotional state

Neg the speaker is using negative language, for example, expressions of criticism, judgment, negative attitude, questioning validity/competence, failure, negative emotion

Mix the speaker is using positive language in part and negative language in part

Irony the speaker is using expressions of sarcasm, ridicule, or mockery

None the speaker is neither using positive language nor using negative language

Emotions

To annotate “Emotions”, please indicate what kind of language the speaker is using.
Note: a writer can use language related to multiple emotions. Select for each emotion to what extent (None, Low, Medium, High) the language displays a certain emotion.

Anger the speaker is using language which expresses: animosity, annoyance, irritability, hostility, fury, outrage, resentment, wrath, exasperation, indignation, vexation, acrimony.

Disgust the speaker is using language which expresses: contempt, disdain, scorn, abhorrence, aversion, distaste, revulsion.

Fear the speaker is using language which expresses: anxiety, apprehension, nervousness, concern, consternation, misgiving, wariness, qualm, edginess, dread, fright, terror.

Joy the speaker is using language which expresses: happiness, enjoyment, relief, contentment, bliss, delight, amusement, pride, sensual pleasure, thrill, rapture, gratification, satisfaction, euphoria, whimsy, ecstasy.

Sadness the speaker is using language which expresses grief, sorrow, cheerlessness, gloom, melancholy, self-pity, loneliness, dejection, despair.

Notes:

- A good response to above questions is one that most people will agree with. For example, even if you think that sometimes the language can be considered negative, if you think most people will consider the language to be positive, then select the positive language option.
- Agreeing or disagreeing with the speaker’s views should not have a bearing on your response. You are to assess the language being used (not the views). For example, given the tweet, ‘Evolution makes no sense’, the correct answer is ‘the speaker is using negative language’ since the speaker’s words are criticizing or judging negatively something (in this case the theory of evolution). Note that the answer is not contingent on whether you believe in evolution or not.
- When you annotate, please try to be as consistent as you can be. In addition, it is essential that you interpret sentences and words with respect to the context in which they appear. Don’t take them out of context and think about what they could mean; judge them as they are being used in that particular sentence and document.

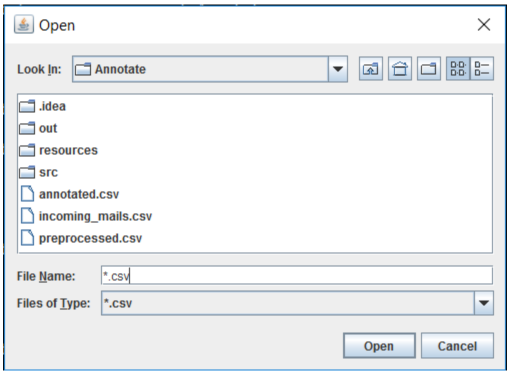
Using Annotate application

The annotation application can be started from command prompt:

```
java -jar Annotate.jar -t
```

NOTE: make sure you have write access to update the resulting ‘annotated.csv’.
In Windows execute the Command with “as Administrator”.
On Linux execute above command preceded by ‘`sudo`’ .

This opens a dialog where you can select the file for annotation



Look In: Annotate

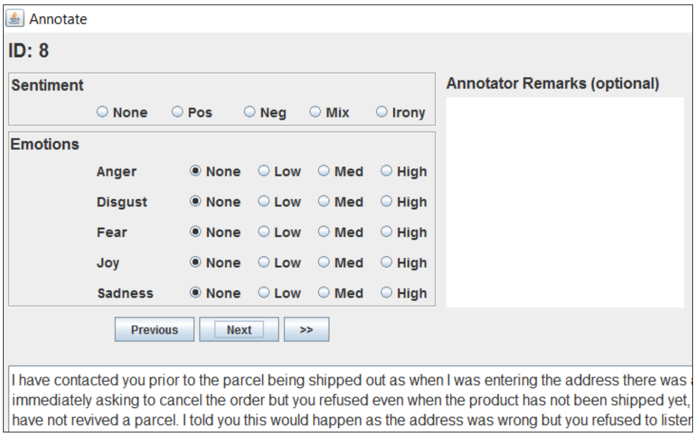
- .idea
- out
- resources
- src
- annotated.csv
- incoming_mails.csv
- preprocessed.csv

File Name: *.csv

Files of Type: *.csv

Open the file ‘annotated.csv’ (from the directory where you placed the file).

Next, the annotation screen opens.



ID: 8

Sentiment: ○ None ○ Pos ○ Neg ○ Mix ○ Irony

Emotions				
Anger	● None	○ Low	○ Med	○ High
Disgust	● None	○ Low	○ Med	○ High
Fear	● None	○ Low	○ Med	○ High
Joy	● None	○ Low	○ Med	○ High
Sadness	● None	○ Low	○ Med	○ High

I have contacted you prior to the parcel being shipped out as when I was entering the address there was
immediately asking to cancel the order but you refused even when the product has not been shipped yet,
have not revived a parcel. I told you this would happen as the address was wrong but you refused to lister

The application automatically opens the first email without annotation. You can stop annotating at any time to resume later. The results are saved upon closure of the application.

Please assess the text for each of the 6 categories (Sentiment, Emotions(5x)).
‘Annotator Remarks’ is optional and can be used to put in some remarks you might have when annotating a specific email.

Appendix B: Data descriptive incoming emails

Below statistics apply to the 742 emails which have been annotated with sentiment and emotions.

Numeric variables

Variable	Description	Mean	Median	Var	Min	Max
correctWordsRatio	Ratio of correctly spelled words / total number of words	0.90	0.93	0.01	0	1
countAllCaps	Number of words which are fully capitalized	0.41	0	3.24	0	36
countAnger	Count of words in NRC emotion word list with Anger label	0.32	0	0.66	0	9
countDisgust	Count of words in NRC emotion word list with Disgust label	0.23	0	0.39	0	6
countExclamation	Number of exclamation marks	0.23	0	1.29	0	26
countFear	Count of words in NRC emotion word list with Fear label	0.32	0	0.54	0	6
countJoy	Count of words in NRC emotion word list with Joy label	0.69	0	1.33	0	9
countNeg	Count of words in NRC emotion word list with Negative label	0.94	0	2.80	0	18
countPos	Count of words in NRC emotion word list with Positive label	2.32	1	7.63	0	21
countSadness	Count of words in NRC emotion word list with Sadness label	0.53	0	0.94	0	8
countSent	Number of sentences	3.56	3	6.85	1	28
countUniqueWords	Total number of unique words in the text body	39.14	33	755	1	208
countWords	Total number of words in the text body	51.54	38	2125	1	409
countWordSent	Number of words per sentence	15.09	13	103	1	110
diversityRatio	Ratio of unique words / total number of words	0.85	0.86	0.02	0	1
lengthMessage	Total length of the message in number of characters	230.45	167	42319	7	1937
lengthWord	Average length of a word	4.61	4.43	0.99	3.33	22
varLengthWord	Variance in word length	2.22	2.12	0.68	0	15.66
threadItem	thread sequence number (start 0)	3	2	12.16	0	21
Doc2Vec_xx	100 Doc2Vec features numbered from Doc2Vec_0 to Doc2Vec_99	-0.02	-0.01	1.10	-7.70	6.75

Categorical variables

Variable	Description	#Cat	Empty %	Top categories (%)
Case Origin	Originating channel of message	2	0%	Contact us form (79.5) Email (20.5)
Contact Reason	User supplied reason of contact	10	4%	Existing Orders / Purchased Product (73.2) Payment / Credit (6.2) My Account / Website / Email (5.5)
dayOfWeek	Day of the week from message time stamp	7	0%	Thursday (19.9) Tuesday (18.9)

				Monday (18.3)
partOfDay	Part of the day from message time stamp. Night 0:00-6:00, Morning 6:00-12:00, Afternoon 12:00-18:00, Evening 18:00-0:00	4	0%	Afternoon (46.9) Evening (28.2) Morning (19.3) Night (5.7)

Special variables

Variable	**Description**	**Top features**
xx_Tfidf	Top 100 tf-idf on text body for unigrams, bigrams and trigrams	thanks, order_no, code, thank, reply
xx_CharTfidf	Top 100 tf-idf on text body on character level for 2 to 5 characters	‘ep’, ‘y t’, ‘es’, ‘art’, ‘hanks’

Variable	**Description**	**#Cat**	**Remarks**
Message Date	Time stamp of the incoming message	50	Range: 1-8-2016 / 18-11-2016 Most common: 4-8-2016 (15.5%)

Appendix C: Detailed results sentiment analysis

Scenario	Agreement (A)	/ Majority vote (M)	A	A	M	M	A	A	M	M
	0.5 threshold (0.5)	/ Max kappa (Max)	0.5	Max	0.5	Max	0.5	Max	0.5	Max
	Oversample (Over)	/ No sampling (No)	Over	Over	Over	Over	No	No	No	No
Model	Neural Net (NN)		0.412	0.378	0.450	0.408	0.221	0.203	0.327	0.332
	Naïve Bayes (NB)		0.124	0.284	0.184	0.250	0.004	0.317	0.040	0.299
	SVM		0.324	0.308	0.282	0.327	0.295	0.314	0.290	0.332
	Random Forest (RF)		0.291	0.408	0.399	0.399	0.248	0.377	0.301	0.383
	Voting ensemble NN+RF+NB+SVM		0.388	0.342	0.446	0.440	0.228	0.233	0.388	0.405
	Voting ensemble NN+RF+NB		0.415	0.404	0.450	0.442	0.298	0.321	0.451	0.369
	Voting ensemble NN+RF+SVM		0.351	0.313	0.466	0.424	0.205	0.220	0.327	0.288
	Voting ensemble NN+RF		0.380	0.352	**0.483**	0.437	0.212	0.230	0.389	0.309

Table 19: Emotion Anger - Average Kappa (30 runs) per scenario and ML model. Bold indicates the best Kappa score

Scenario	Agreement (A)	/ Majority vote (M)	A	A	M	M	A	A	M	M
	0.5 threshold (0.5)	/ Max kappa (Max)	0.5	Max	0.5	Max	0.5	Max	0.5	Max
	Oversample (Over)	/ No sampling (No)	Over	Over	Over	Over	No	No	No	No
Model	Neural Net (NN)		0.242	0.362	0.424	0.377	0.052	0.169	0.244	0.248
	Naïve Bayes (NB)		0.000	0.260	0.075	0.227	0.000	0.270	0.000	0.271
	SVM		0.104	0.071	0.231	0.270	0.051	0.048	0.213	0.242
	Random Forest (RF)		0.000	0.337	0.149	0.356	0.000	0.257	0.111	0.291
	Voting ensemble NN+RF+NB+SVM		0.043	0.038	0.427	0.359	0.006	0.000	0.219	0.238
	Voting ensemble NN+RF+NB		0.065	0.052	0.441	0.403	0.002	0.003	0.353	0.234
	Voting ensemble NN+RF+SVM		0.000	0.000	0.196	0.174	0.003	0.000	0.177	0.122
	Voting ensemble NN+RF		0.182	0.144	**0.455**	0.363	0.076	0.038	0.288	0.263

Table 20: : Emotion Disgust - Average Kappa (30 runs) per scenario and ML model. Bold indicates the best Kappa score

Scenario	Agreement (A) / Majority vote (M)	A	A	M	M	A	A	M	M
	0.5 threshold (0.5) / Max kappa (Max)	0.5	Max	0.5	Max	0.5	Max	0.5	Max
	Oversample (Over) / No sampling (No)	Over	Over	Over	Over	No	No	No	No
Model	Neural Net (NN)	0.028	0.040	0.034	0.072	0.019	0.014	-0.001	0.003
	Naïve Bayes (NB)	-0.002	0.070	0.006	-0.001	0.000	0.060	0.000	0.017
	SVM	0.000	-0.018	0.017	-0.001	-0.005	-0.016	0.018	-0.003
	Random Forest (RF)	0.000	0.053	0.000	**0.133**	0.000	-0.005	0.000	0.012
	Voting ensemble NN+RF+NB+SVM	0.000	0.000	-0.001	-0.010	0.000	0.000	-0.005	-0.002
	Voting ensemble NN+RF+NB	-0.002	-0.001	-0.002	0.001	0.000	0.000	-0.008	-0.005
	Voting ensemble NN+RF+SVM	0.000	0.000	-0.002	-0.001	0.000	0.000	0.000	0.000
	Voting ensemble NN+RF	0.024	0.018	0.019	0.035	0.000	0.000	-0.006	-0.003

Table 21: Emotion Fear - Average Kappa (30 runs) per scenario and ML model. Bold indicates the best Kappa score

Scenario	Agreement (A) / Majority vote (M)	A	A	M	M	A	A	M	M
	0.5 threshold (0.5) / Max kappa (Max)	0.5	Max	0.5	Max	0.5	Max	0.5	Max
	Oversample (Over) / No sampling (No)	Over	Over	Over	Over	No	No	No	No
Model	Neural Net (NN)	0.500	0.411	0.498	0.452	0.168	0.293	0.301	0.354
	Naïve Bayes (NB)	0.287	0.270	0.266	0.249	0.021	0.140	0.050	0.112
	SVM	0.325	0.334	0.307	0.367	0.322	0.367	0.309	0.384
	Random Forest (RF)	0.454	0.482	0.435	0.459	0.419	0.513	0.393	0.481
	Voting ensemble NN+RF+NB+SVM	0.500	0.461	0.507	0.476	0.289	0.285	0.225	0.315
	Voting ensemble NN+RF+NB	0.524	0.499	0.505	0.480	0.452	0.456	0.472	0.432
	Voting ensemble NN+RF+SVM	0.464	0.406	0.495	0.443	0.130	0.112	0.351	0.233
	Voting ensemble NN+RF	**0.537**	0.462	0.528	0.507	0.261	0.208	0.322	0.312

Table 22: : Emotion Joy - Average Kappa (30 runs) per scenario and ML model. Bold indicates the best Kappa score

Scenario	Agreement (A) / Majority vote (M)	A	A	M	M	A	A	M	M
	0.5 threshold (0.5) / Max kappa (Max)	0.5	Max	0.5	Max	0.5	Max	0.5	Max
	Oversample (Over) / No sampling (No)	Over	Over	Over	Over	No	No	No	No
Model	Neural Net (NN)	0.257	0.296	0.328	0.314	0.094	0.185	0.220	0.239
	Naïve Bayes (NB)	0.152	0.298	0.239	0.274	0.027	0.298	0.144	0.284
	SVM	0.173	0.163	0.258	0.249	0.131	0.149	0.262	0.254
	Random Forest (RF)	0.183	0.353	0.332	0.351	0.081	0.346	0.177	0.361
	Voting ensemble NN+RF+NB+SVM	0.230	0.209	0.362	0.351	0.154	0.158	0.318	0.297
	Voting ensemble NN+RF+NB	0.256	0.230	0.352	0.336	0.127	0.146	0.319	0.301
	Voting ensemble NN+RF+SVM	0.217	0.228	0.323	0.297	0.088	0.108	0.234	0.189
	Voting ensemble NN+RF	0.277	0.241	**0.364**	0.319	0.110	0.095	0.210	0.222

Table 23: Emotion Sadness - Average Kappa (30 runs) per scenario and ML model. Bold indicates the best Kappa score

Scenario	Agreement (A) / Majority vote (M)	A	M	A	M
	Oversample (Over) / No sampling (No)	Over	Over	No	No
Model	Neural Net (NN)	0.273	0.369	0.182	0.295
	Naïve Bayes (NB)	0.206	0.234	0.185	0.222
	SVM	0.312	0.257	0.311	0.255
	Random Forest (RF)	0.346	0.364	0.338	0.361
	Voting ensemble NN+RF+NB+SVM	0.355	0.397	0.308	0.385
	Voting ensemble NN+RF+NB	0.365	0.381	0.328	0.338
	Voting ensemble NN+RF+SVM	0.339	**0.406**	0.218	0.354
	Voting ensemble NN+RF	0.345	0.385	0.259	0.337

Table 24: : Sentiment - Average Kappa (30 runs) per scenario and ML model. Bold indicates the best Kappa score

Appendix D: Detailed results

Variable	**Test type**	**p-value**
Case Origin	χ^2	**2.0E-04**
Contact Reason	χ^2	**8.8E-08**
Thread sequence number	KS	**0.04**
Total number of words in the text body	KS	0.84
Total number of unique words in the text body	KS	0.82
Diversity of words	KS	0.69
Ratio of correctly spelled words / total number of words.	KS	0.73
Number of exclamation marks	KS	1.00
Number of words which are fully capitalized	KS	0.52
Number of sentences	KS	0.84
Number of words per sentence	KS	0.60
Total length of the message in number of characters	KS	0.56
Average length of a word	KS	0.09
Variance in word length	KS	0.23
Day of the week (Monday. Tuesday.…)	χ^2	**0.03**
Part of the day (Night. Morning. Afternoon. Evening)	χ^2	0.21
Number of words with NRC-annotated Anger label	KS	1.00
Number of words with NRC-annotated Disgust label	KS	0.65
Number of words with NRC-annotated Fear label	KS	1.00
Number of words with NRC-annotated Joy label	KS	1.00
Number of words with NRC-annotated Sadness label	KS	0.99
Number of words with NRC-annotated Negative label	KS	1.00
Number of words with NRC-annotated Positive label	KS	0.86

Table 25: Comparison of sample distribution with remainder of email dataset. Boldface indicates significant. Only for Case Origin, Contact Reason, Thread sequence number and Day of the week it does not hold that distribution of the sample does not differ from the population with significance p<0.05. All other sample variables are representative for the population.

	p-value
Anger	2.1 e-20
Disgust	3.9 e-33
Fear	2.6 e-12
Joy	0.02
Sadness	1.9 e-27

Table 26: Results of chi-square test on human annotation distribution. For each emotion, there are significant (p<0.05) differences between annotators on the way they annotate.

	Anger	Disgust	Fear	Joy	Sadness
Anger	1	0.55***	-0.06*	-0.26***	0.08**
Disgust	0.55***	1	-0.06*	-0.16***	0.07*
Fear	-0.06*	-0.06*	1	-0.11***	0.02
Joy	-0.26***	-0.16***	-0.11***	1	-0.18***
Sadness	0.08**	0.07*	0.02	-0.18***	1

Table 27: Spearman's rho for correlation of emotions in emails. *** significance p<0.01, ** significance p<0.05, * significance p<0.10. Almost all relations are significant. Most relations are (very) weak. Only Anger-Disgust is moderate. The positive emotion Joy has a negative relation to all other (negative) emotions.

	Sentiment	Emotions					Support
		Anger	Disgust	Fear	Joy	Sadness	
NRC-IBM	0.24	0.22	0.01	0.02	0.08	0.13	742
NRC-An1	0.09	0.26	0.28	0.04	0.06	0.22	742
NRC-An2	0.10	0.45	0.27	0.09	0.08	0.28	442
NRC-An3	0.12	0.34	0.32	0.07	0.01	0.25	444
NRC-An4	0.14	0.32	0.18	0.07	0.09	0.23	299
NRC-An5	0.08	0.13	0.00	0.06	-0.01	0.28	299
IBM-An1	0.23	0.26	0.03	0.02	0.41	0.17	742
IBM-An2	0.29	0.36	-0.01	-0.01	0.46	0.16	442
IBM-An3	0.19	0.16	0.02	0.00	0.42	0.02	444
IBM-An4	0.22	0.23	-0.01	-0.01	0.47	0.12	299
IBM-An5	0.15	0.37	0.00	-0.01	0.35	0.01	299
An1-An2	0.45	0.62	0.45	0.32	0.64	0.32	442
An1-An3	0.44	0.31	0.49	0.22	0.51	0.34	444
An1-An4	0.38	0.51	0.16	0.28	0.70	0.20	299
An1-An5	0.47	0.55	0.00	0.12	0.60	0.25	299
An2-An3	0.51	0.54	0.35	0.39	0.50	0.23	144
An2-An4	0.38	0.76	0.41	0.21	0.70	0.48	149
An2-An5	0.44	0.66	0.00	0.17	0.55	0.53	149
An3-An4	0.40	0.32	0.07	0.26	0.71	0.25	150
An3-An5	0.58	0.15	0.00	0.18	0.57	0.16	150

Table 28: Pairwise Cohen kappa per emotion category and sentiment (gradient scale grey-white: worst to best). An4-An5 pair is absent since these annotators did not annotate a same set of emails. NRC has the most agreement for Anger but still only fair agreement. Performance of IBM on Disgust, Fear and Sadness is bad. While Joy has moderate agreement. There are major differences between various emotions and between various annotator pairs. E.g. Anger ranges from 0.15 to 0.76. NRC and IBM are outperformed by annotators on every front.

	Anger		**Disgust**		**Joy**		**Sentiment**	
Baseline, all features	0.48		0.46		0.53		0.41	
Selection method	Fwd	Bwd	Fwd	Bwd	Fwd	Bwd	Fwd	Bwd
Selected feat. performance	0.52	0.50	0.43	0.46	0.64	0.63	0.44	0.41
Selected features								
Case Origin (2)		X		X	X	X		X
char_Tfidf (100)	X	X		X	X	X	X	X
Contact Reason (10)		X		X	X	X		X
correctWordsRatio				X		X		X
countAllCaps		X		X				X
countNRC (7)		X	X	X	X	X	X	X
countExclamation		X			X	X	X	
countSent		X		X		X		X
countUniqueWords		X		X		X		X
countWords		X		X		X		X
countWordSent		X		X				X
dayOfWeek (7)		X		X	X	X		X
diversityRatio	X	X		X	X	X		X
Doc2Vec (100)	X	X	X	X			X	X
lengthMessage		X		X				X
lengthWord		X		X		X	X	X
partOfDay (4)		X		X		X		X
threadItem		X		X	X	X	X	X
varLengthWord		X		X				X
word_Tfidf (100)	X	X		X	X	X	X	X

Table 29: Selected features for best performing model. 'X' indicates selected feature. Fwd: Forward, Bwd: Backward. Feature selection results in improved performance compared the baseline using all features. Forward selection works best for Anger, Joy and Sentiment. Disgust has best results with backward selection. All features from forward selection are also present in backward selection except for countExclamation for Sentiment.

	Anger	Disgust	Joy	Sentiment
Baseline. all features	**0.51**	**0.44**	**0.61**	**0.42**
Case Origin (2)		0.007	-0.002	
char_Tfidf (100)	0.029**	0.015	0.053**	0.013
Contact Reason (10)		-0.004	0.000	
correctWordsRatio		0.003		
countAllCaps		-0.004		
countExclamation			0.019**	-0.005
countNRC (7)		0.000	0.038**	-0.004
countSent		0.007		
countUniqueWords		0.000		
countWords		-0.001		
countWordSent		0.005		
dayOfWeek (7)		0.002	0.005	
diversityRatio	0.005	0.006	0.004	
Doc2Vec (100)	0.147**	0.037*		0.023*
lengthMessage		-0.002		
lengthWord		-0.001		-0.007
partOfDay (4)		-0.009		
threadItem		0.012	0.008	0.001
varLengthWord		-0.007		
word_Tfidf (100)	0.031**	0.007	0.074**	0.002

Table 30: 50 run average difference in kappa compared to baseline when excluding feature. * Significant p<0.05; ** Significant p<0.01. Depending on the emotion or sentiment, Doc2Vec or tf-idf are key features

	Anger	Disgust	Joy	Sentiment
Mean	0.51	0.43	0.61	0.43
Median	0.51	0.44	0.62	0.43
Max	0.71	0.67	0.77	0.62
Min	0.35	0.20	0.44	0.30
IQR	0.09	0.12	0.11	0.07

Table 31: Descriptive for Figure 4. Kappa value of 100 runs of best model and feature selection. Each model uses oversampling, majority vote and 0.5 threshold (emotions only). The difference between Min and Max value is quite large. The IQR value is not that big.

		Predicted	
		No Anger	**Anger**
Actual	**No Anger**	106.6	8.4
	Anger	15.6	18.4

	Precision	Recall	F1-score	Support
No Anger	0.87	0.93	0.90	115
Anger	0.69	0.54	0.61	34
Avg micro	0.83	0.84	0.83	149
Avg macro	0.78	0.73	0.75	149

Table 32: Emotion Anger - Confusion matrix and key statistics of 100 runs of Voting ensemble Neural Net + Random Forest with feature selection based on majority vote, 0.5 threshold and oversampling minority class. No Anger class appears 3.4 more frequent than Anger. F1 of 0.61 is reasonable.

		Predicted	
		No Disgust	**Disgust**
Actual	**No Disgust**	125.3	6.7
	Disgust	9.2	7.8

	Precision	Recall	F1-score	Support
No Disgust	0.93	0.95	0.94	132
Disgust	0.54	0.46	0.49	17
Avg micro	0.89	0.89	0.89	149
Avg macro	0.73	0.70	0.72	149

Table 33: Emotion Disgust - Confusion matrix and key statistics of 100 runs of Voting ensemble Neural Net + Random Forest with feature selection based on majority vote, 0.5 threshold and oversampling minority class. No Disgust class appears 7.8 more frequent than Disgust. More imbalanced than Anger or Joy. F1 of 0.49 is lowest for emotions.

		Predicted	
		No Joy	**Joy**
Actual	**No Joy**	112.6	7.4
	Joy	10.3	18.7

	Precision	Recall	F1-score	Support
No Joy	0.92	0.94	0.93	120
Joy	0.72	0.65	0.68	29
Avg micro	0.88	0.88	0.88	149
Avg macro	0.82	0.79	0.80	149

Table 34: Emotion Joy - Confusion matrix and key statistics of 100 runs of Voting ensemble Neural Net + Random Forest with feature selection based on majority vote, 0.5 threshold and oversampling minority class. No Joy class appears 4.1 more frequent than Joy. F1 of 0.68 is reasonable.

		Predicted			
		Mix	**Neg**	**None**	**Pos**
Actual	**Mix**	0.2	3.7	2.0	1.1
	Neg	1.1	41.1	12.4	3.4
	None	0.8	11.8	29.4	8.1
	Pos	0.7	4.4	7.7	21.3

Table 35: Sentiment – Confusion matrix actual vs predicted best model Soft_NN_RF_SVM. Mix performance is not good.

	Precision	Recall	F1-score	Support
Mix	0.08	0.03	0.05	7
Neg	0.67	0.71	0.69	58
None	0.57	0.59	0.58	50
Pos	0.63	0.63	0.63	34
Avg micro	0.60	0.62	0.61	149
Avg macro	0.49	0.49	0.49	149

Table 36: Sentiment – Main classification metrics best model Soft_NN_RF_SVM. Performance on Mix category is not goof but has little impact on micro average because of low support (7). Performance of Neg, None and Pos is reasonable with F1 between 0.58 – 0.69.

		Post CS response emotion		
		No Anger	**Anger**	**Total**
Start emotion	**No Anger**	283	51	**334**
	Anger	72	49	**121**
	Total	**355**	**100**	**455**

Table 37: Customer Anger before and after CS response. More customers move from Anger to No Anger than vice versa.

		Post CS response emotion		
		No Disgust	**Disgust**	**Total**
Start emotion	**No Disgust**	355	40	**395**
	Disgust	43	17	**60**
	Total	**398**	**57**	**455**

Table 38: Customer Disgust before and after CS response. Distribution before and after CS response hardly changes.

		Post CS response emotion		
		No Fear	**Fear**	**Total**
Start emotion	**No Fear**	416	13	**429**
	Fear	25	1	**26**
	Total	**441**	**14**	**455**

Table 39: Customer Fear before and after CS response. Classes are highly imbalanced. Almost all customers move away from Fear after CS response.

		Post CS response emotion		
		No Joy	**Joy**	**Total**
Start emotion	**No Joy**	292	109	**401**
	Joy	36	18	**54**
	Total	**328**	**127**	**455**

Table 40: Customer Joy before and after CS response. There is a big shift in customers towards Joy after CS response.

		Post CS response emotion		
		No Sadness	**Sadness**	**Total**
Start emotion	**No Sadness**	280	46	**326**
	Sadness	100	29	**129**
	Total	**380**	**75**	**455**

Table 41: Customer Sadness before and after CS response. Overall number of customers with Sadness is reduced after CS response.

Model	Anger	Disgust	Fear	Joy	Sadness
Single label without oversampling					
Neural Net	0.16	0.06	0.00	0.16	0.05
SVM	0.00	0.00	0.00	0.01	0.00
Naive Bayes	0.41	0.26	0.03	0.45	**0.29**
Random Forest	0.07	0.00	0.00	0.18	0.00
Voting NN+SVM+NB+RF	0.11	0.06	0.00	0.20	0.05
Voting NN+NB+RF	0.23	0.08	0.00	0.28	0.07
Voting NN+NB	0.30	0.17	0.03	0.36	0.18
Voting NB+RF	0.41	0.25	0.02	0.45	0.29
Single label with oversampling					
Neural Net	0.38	0.22	**0.04**	0.41	0.23
SVM	0.01	0.00	0.00	0.07	0.01
Naive Bayes	0.42	**0.26**	0.02	**0.46**	0.28
Random Forest	0.15	0.00	0.00	0.35	0.01
Voting NN+SVM+NB+RF	0.37	0.18	0.00	0.41	0.16
Voting NN+NB+RF	0.41	0.25	0.00	0.45	0.27
Voting NN+NB	**0.42**	0.26	0.02	0.45	0.28
Voting NB+RF	0.41	0.25	0.03	0.45	0.28
Multilabel					
Neural Net (multilabel)	0.19	0.10	0.11	0.00	0.18
Random Forest (multilabel)	0.03	0.00	0.00	0.00	0.17
RAkEL – Neural Net (Label Power)	0.20	0.08	0.09	0.00	0.21
RAkEL – Random Forest (Label Power)	0.07	0.01	0.00	0.00	**0.30**

Table 42: Average F1-measure (30 runs). Bold indicates best performing model for specific emotion. Performance is very low for Fear. Best model for Sadness is multilabel with substandard performance for other emotions. Second best model is selected for Sadness.

	Anger		Disgust		Joy		Sadness		Sentiment	
Selection method	Fwd	Bwd	Fwd	Bwd	Fwd	Bwd	Fwd	Bwd	Fwd	Bwd
Selected feat. Perf.	0.47	0.41	0.34	0.29	0.51	0.52	0.32	0.31	0.52	0.48
Selected features										
Shared features										
Case Origin (2)		X		X		X		X		X
Contact Reason (10)		X				X		X		X
threadItem	X	X	X	X		X		X		X
CS email features										
char_Tfidf (100)		X		X		X	X	X	X	X
correctWordsRatio		X		X		X		X		X
countAllCaps		X		X		X		X		X
countExclamation		X		X		X		X		X
countNRC (7)		X		X		X	X	X	X	X
countSent		X		X		X		X		X
countUniqueWords		X		X		X		X		X
countWords		X		X				X		X
countWordSent		X		X		X		X		X
dayOfWeek (7)	X			X		X		X		X
diversityRatio	X	X	X	X		X		X		X
Doc2Vec (100)		X		X		X				X
lengthMessage		X	X	X		X		X		X
lengthWord		X		X		X		X		X
partOfDay (4)				X		X		X		X
varLengthWord		X		X		X		X		X
word_Tfidf (100)		X		X	X	X	X	X		X
Client email features										
char_Tfidf (100)		X		X		X				X
correctWordsRatio				X		X		X		X
countAllCaps		X		X		X		X		X
countExclamation		X		X		X	X	X		X
countNRC (7)		X		X		X		X	X	X
countSent		X		X		X		X		X
countUniqueWords		X		X		X		X		X
countWords		X		X		X		X		X
countWordSent		X		X				X		X
dayOfWeek (7)		X	X	X				X		X
diversityRatio	X	X		X		X				X
Doc2Vec (100)		X				X				X
Emotions (5)		X	X	X		X			X	X
lengthMessage		X		X		X		X		X
lengthWord		X		X		X		X		X
partOfDay (4)		X		X		X	X	X		X
Sentiment (4)	X	X		X	X	X		X	X	X
varLengthWord		X		X		X		X		X
word_Tfidf (100)		X							X	X

	Anger		Disgust		Joy		Sadness		Sentiment	
Selection method	Fwd	Bwd	Fwd	Bwd	Fwd	Bwd	Fwd	Bwd	Fwd	Bwd
Conversation features										
Response Time		X		X		X		X		X

Table 43: Selected features for best performing affect analysis model. 'X' indicates selected feature. Fwd: Forward, Bwd: Backward. Emotion performance metric: F1. Sentiment performance metric: accuracy. In all cases but Joy, forward selection gives best result. dayOfWeek is selected for Anger in forward selection but not in backward selection. In all other cases all forward selected features are also in backward selection.

	Anger	Disgust	Joy	Sadness	Sentiment
Baseline, all features	0.45	0.31	0.50	0.31	0.49
Shared features					
Case Origin (2)			-0.003		
Contact Reason (10)			-0.010		
threadItem	0.042***	0.028***	-0.006		
CS email features					
char_Tfidf (100)			0.004	-0.005	0.059***
correctWordsRatio			-0.002		
countAllCaps			-0.007		
countExclamation			0.008		
countNRC (7)			-0.005	-0.002	0.005
countSent			-0.001		
countUniqueWords			0.003		
countWords					
countWordSent			0.002		
dayOfWeek (7)	0.020***		-0.001		
diversityRatio	-0.005	0.004	-0.011*		
Doc2Vec (100)			0.001		
lengthMessage		0.016*	-0.001		
lengthWord			-0.005		
partOfDay (4)			-0.005		
varLengthWord			0.001		
word_Tfidf (100)			0.034***	0.026**	
Client email features					
char_Tfidf (100)			-0.011*		
correctWordsRatio			-0.001		
countAllCaps			-0.004		
countExclamation			-0.006	0.003	
countNRC (7)			0.007		-0.003
countSent			0.000		
countUniqueWords			-0.007		
countWords			0.004		
countWordSent			-0.006		
dayOfWeek (7)		0.014	0.001		

	Anger	Disgust	Joy	Sadness	Sentiment
diversityRatio	-0.008		0.006		
Doc2Vec (100)					
Emotions (5)		0.035***	-0.004		0.013**
lengthMessage			-0.005		
lengthWord			0.001		
partOfDay (4)				0.001	
Sentiment (4)	0.126***		-0.005		0.010*
varLengthWord			0.004		
word_Tfidf (100)					0.008
Conversation features					
Response Time			-0.002		

Table 44: Added value of single feature to affect model performance. *** significant p<0.01, ** significant p<0.05, * significant p<0.10. The backward feature selection for Joy has wrongfully selected some features with significant negative impact.

	Anger	Disgust	Joy	Sadness	Sentiment
Mean	0.45	0.30	0.50	0.30	0.49
Median	0.46	0.30	0.51	0.30	0.48
Max	0.60	0.47	0.65	0.46	0.62
Min	0.30	0.12	0.33	0.16	0.34
IQR	0.09	0.11	0.08	0.11	0.07

Table 45: Descriptive for Figure 5. Average F1 or accuracy performs of 100 runs of the best affect analysis model and selected features. Emotion performance metric: F1. Sentiment performance metric: accuracy

		Predicted	
		No Anger	**Anger**
Actual	**No Anger**	44.1	26.9
	Anger	6.4	13.6

	Precision	Recall	F1-score	Support
No Anger	0.87	0.62	0.73	71
Anger	0.34	0.68	0.45	20
Avg micro	0.76	0.63	0.67	91
Avg macro	0.60	0.65	0.59	91

Table 46: Confusion matrix and general metrics for affect analysis model wrt Anger. Model used: Voting Naive Bayes and Random Forest based on five selected features with oversampling of minority classes. Imbalance: 3.6x

		Predicted	
		No Disgust	**Disgust**
Actual	**No Disgust**	57.8	22.2
	Disgust	5.2	5.8

	Precision	Recall	F1-score	Support
No Disgust	0.92	0.72	0.81	80
Disgust	0.21	0.52	0.30	11
Avg micro	0.83	0.70	0.75	91
Avg macro	0.56	0.62	0.55	91

Table 47: Confusion matrix and general metrics for affect analysis model wrt Disgust. Model used: Naive Bayes based on five selected features with oversampling of minority classes. Imbalance: 7.3x

		Predicted	
		No Joy	Joy
Actual	No Joy	42.1	23.9
	Joy	8.7	16.3

	Precision	Recall	F1-score	Support
No Joy	0.83	0.64	0.72	66
Joy	0.41	0.65	0.50	25
Avg micro	0.71	0.64	0.66	91
Avg macro	0.62	0.65	0.61	91

Table 48: Confusion matrix and general metrics for affect analysis model wrt Joy. Model used: Naive Bayes based on 36 selected features with oversampling of minority classes. Imbalance: 2.7x

		Predicted	
		No Sadness	Sadness
Actual	No Sadness	51.6	24.5
	Sadness	8.0	7.0

	Precision	Recall	F1-score	Support
No Sadness	0.87	0.68	0.76	76
Sadness	0.22	0.47	0.30	15
Avg micro	0.76	0.64	0.68	91
Avg macro	0.54	0.57	0.53	91

Table 49: Confusion matrix and general metrics for affect analysis model wrt Sadness. Model used: Naive Bayes based on five selected features without oversampling of minority classes. Imbalance: 5.1x

		Predicted sentiment				
		Mix	Neg	None	Pos	Total
Actual sentiment	Mix	0.0	1.2	1.3	1.5	4
	Neg	0.1	12.6	6.9	7.4	27
	None	0.1	6.2	16.6	7.1	30
	Pos	0.1	6.6	8.3	15.0	30
	Total	0.3	26.6	33.1	31.1	91

	Precision	Recall	F1-score	Support
Mix	0.00	0.00	0.00	4
Neg	0.47	0.47	0.47	27
None	0.50	0.55	0.53	30
Pos	0.48	0.50	0.49	30
Avg micro	0.47	0.49	0.48	91
Avg macro	0.37	0.38	0.37	91

Table 50: Affect Sentiment – Main classification metrics best model RF without oversampling, five features. Only few case with Mix label are present which all have been misclassified.