Comparative Study of Techniques for Extracting Customer Needs from Hiking Backpack online Reviews

Dani Fallouh

Dublin Business School

Applied Research Project submitted in partial fulfilment of the requirements for the degree of Master of Science (MSc) in Data Analytics at Dublin Business School

Supervisor: Amit Sharma

January 2024

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Signed: Dani Fallouh

Student number: 10640663

January 2024

Abstract

Identifying customer needs has always been a significant factor that contributes to company's ongoing success (Wang and Ji, 2010, p. 173). However, the traditional ways of identifying customer are expensive and time-consuming in addition to many other problems (Timoshenko and Hauser, 2019, p. 4). This is why there is a new approach to identify customer needs using User Generated Content (UGC) as a source of data (Kühl et al., 2020, p. 2). The main challenge is about vectorizing the data and extracting only the relevant information. In this project, the steps of this new method will be explained using customer reviews of the product "Hiking Backpack" on Amazon, Etsy and Walmart. Also, twenty combinations of different vectorization methods and supervised machine learning classification algorithms will be compared regarding both accuracy and recall scores.

Table of Contents

List of Figures

List of Tables

1 Introduction

“The most important purpose of any product is to satisfy a customer’s need and not the entrepreneur’s need.” - Pooja Agnihotri

This citation shows the importance of the identifying customer needs in order to achieve a successful product (Wang and Ji, 2010, p. 174). Understanding and Identifying customer needs is a very hard, complex, and at the same time a very important part in product development. This affects the satisfaction of the customers and succuss of the product in a positive way. Accordingly, understanding customer needs is one of the most important aspects of customer satisfaction (Shane, 2008, p. 134).

Identifying customer needs has been always an important topic. Normally, companies use primarily on qualitative and quantitative methods such as surveys and interviews to understand customer needs and wants (Witell et al., 2011, p. 141). However, these traditional methods have many disadvantages. Surveys are normally time-consuming and require long time of planning and implementation. Interviews are expensive and also need long times to be implemented. In addition, this can be influenced by the type of question asked or by the interaction with the interviewer (Price et al., 2015, p. 236). Another main disadvantage is that these methods could only collect a limited amount of data, which makes it difficult to identify rare customer needs. Instead, only the general, main needs would be collected. For example, Timoshenko and Hauser (2019, p. 24) were able to find more than 8 needs through UGC, which were not found using the traditional method. These methods are not optimal for this purpose. Today, at a time when the Internet plays a central role in everyday life and business, many researchers try to find a new way to identify customer’s needs (Kühl et al., 2020, p. 2).

Internet users produce a huge amount of data daily. This data includes information about almost every aspect of our life. For example, personal blogs, social media posts, customer reviews and forum blogs etcetera. Only on Twitter, 500 Millions blogs are posted on a daily basis (Duncombe, 2019 cited in Rensing, 2022, p. 2). This user-generated content provides a deep insight into users' thoughts, opinions, likes and dislikes (Saura et al., 2019, p. 2). They could be a rich source for market research, product development and customer service. Companies can use this information to better understand what their customers really need. This method could be implemented much faster than the traditional methods (Timoshenko and Hauser, 2019, p. 4). Moreover, the cost of this method would not be as high as the old ways. Furthermore, the most important aspect is that it would be possible to collect huge amount of data, much more than from interviews or surveys. In this huge amount of data, there is surely data about customer needs, which could be used. However, due to the huge amount, it is not possible to manually all of the data. Much of the information available on the Internet is not directly related to customer needs, identifying relevant content a complex challenge (Liu et al., 2019, p. 1).

An important aspect in this method is to classify these massive amounts of data into two main categories – relevant and non-relevant to product development. For instance, online reviews like “The product is practical”, “I like the product” or “I bought this product as a Christmas gift for my daughter, and she loved it” would not be important at all for the product development. In contrast, Reviews like “I like that the backpack is waterproof” or “the is an extra small pocket for the phone and keys, and it is very practical” include important information for product development. As companies would know, that it is important to make the backpack waterproof and to add a small pocket for the phone and keys. This could influence the customer satisfaction about the product or service and make it more successful (Timoshenko and Hauser, 2019, pp. 3-4).

To solve this classifying problem, machine learning and deep learning algorithms offer promising approaches (Kühl et al., 2020). The crucial question is which algorithms are most effective and how the data can best be prepared to make optimal use of the algorithms. Another key aspect of using machine learning methods is the transformation of the text into a form that can be processed by the algorithms. This is done through text vectorization, which converts text data into numeric vectors. In this project, various vectorization methods - such as TF-IDF, Count Vectorization, FastText and Hashing - are examined and compared with each other. The goal is to identify the method that best helps efficiently use the Internet's immense data and provide deep insights into customer needs. In this project, this new method will be explained, and several machine learning algorithms will be compared. Also, each machine learning algorithm would be implemented using several vectorization techniques.

2 Literature Review

Due to the significant importance of this new approach, there have been lots of research in this area. One famous scientific paper is “Identifying Customer Needs from User-Generated Content” by Timoshenko and Hauser (2019). They have used 8.000 amazon reviews, in order to show this new method. They have chosen the oral care products as an example and used convolutional neural networks to classify the dataset. It is very interesting that they were able to compare their modern approach with the traditional approach by doing experiments. They found out, that the method is much cheaper and faster than experimenting. They also were able to compare the identified customer needs of both methods: There have been 83 Customer needs which have been identified by both methods. it is important to point out that the new method were able to identify 8 needs more, which have been not found in the experimental approach (Timoshenko and Hauser, 2019).

Another interesting scientific paper regarding this topic is “NEEDMINING: IDENTIFYING MICRO BLOG DATA CONTAINING CUSTOMER NEEDS” by (Kühl et al., 2020) The researchers collect 2400 blogs from twitter and try to identify customer needs. The researchers have compared several sampling methods, including Oversampling, Under sampling and Synthetic Minority Over-sampling TEchnique (SMOTE). Also, they have compared different classifiers including Bayes classifiers and Random Forest (Kühl et al., 2020). It is important to say that this new approach could be implemented to identify customer needs in any field and different languages. For example, (Luo et al., 2023) have used Chinese UGC in order to identify customer needs regarding the automobile industry. They have also used TF-IDF as a vectorization method before the classification algorithms.

There have been also research about using unsupervised algorithms to cluster sentences into different groups. George and Sumathy (2023) have proposed a new approach using Latent Dirichlet Allocation (LDA) and were able to classify large amount of text into groups with similar meanings. This could be also added to the approach of identifying customer needs, as the last step is to manually extract the needs from the relevant reviewed. It would be practical to cluster them automatically and without training classifiers first before manually extracting the data.

UGC could be used in different fields as a source of data and it could be work better than the traditional ways (Zhuang et al., 2023). An example for that is Farid and Mousa, (2023). The researchers have tried to investigate the quality of a hotel regarding specific aspects such as cleanness, location etc. They have used both booking reviews as a source of data and survey forms to compare the results. They have found out that there is no significant difference between the sentimental analysis of the UGC and the answers on the Surveys (Farid and Mousa, 2023). This could also be implemented in the topic of identifying customer needs in order to see the influence of the needs on the customer satisfaction and as a result identify the importance of each need.

3 Overview

The objective of this project is to show how it is possible to identify the customer needs from online reviews. Another objective is to compare the vectorization technique and classification algorithm works the best. This will be done by identifying the customer needs for the product hiking backpacks. Before starting with explaining each step, it is important to show the structure of the project:

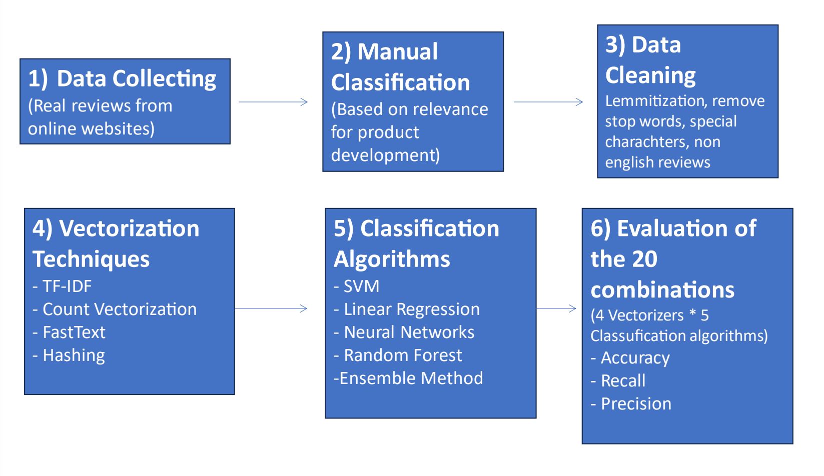

Figure 1 Strutcture of the Project

The first step is to collect the data from various online platforms, which are amazon, Etsy, and Walmart. After data collection, a manual classification follows, in which reviews are classified according to their relevance to product development. It is important to implement this step manually, in order to train and test the algorithms during the next steps. it will be classified into category (1) if the review includes information about customer needs. Other ways it would be classified into Category (0).

The third step is to clean and prepare the date. This step includes several important aspects such as removing stop words and special characters, deleting non-English reviews, write everything in lowercase, lemmatization, tokenization.

Then, in the fourth step, vectorization techniques such as TF-IDF, Count Vectorization, FastText and Hashing are used to convert the text data into numerical vectors, which are required for the application of machine learning algorithms.

In the fifth step, the cleaned and vectorized data are analyzed using various classification algorithms, including SVM, linear regression, neural networks, random forest and ensemble methods.

The sixth step involves evaluating the 20 combinations (4 Vectorization techniques * 5 classification algorithms based on metrics such as accuracy, recall and precision. This is to determine which method works the best for the current problem.

As a result, this project shows and discusses this innovative method to identify customer needs and to gain deeper insights from the large amount of online customer feedback and use this for targeted product development. The advantages of this new method will be shown during this project.

4 Main Body

In this chapter, all the steps of the Project will be presented and discussed. All of the following steps were important to achieve this project. The Results will be presented after that.

4.1 Step 1) Data Collection

Nowadays, due to the daily use of Internet, there are huge amounts of data online, that could be used for analytical purposes. Online platforms such as Twitter and Google Reviews are rich in user contributions; for example, millions of tweets are written every day and Google Reviews, Amazon and eBay contain countless reviews. These digital opinions could be a valuable resource for companies seeking in-depth market and customer analysis for many purposes such as identifying customer needs (Saura and Bennett, 2019, p. 2). As the data is already available, it is much faster and cheaper to collect this data, instead of generating new data by the traditional methods like surveys or interviews (Timoshenko and Hauser, 2019, p. 4).

It is extremely important to take the ethical and legal guidelines regarding data protection into consideration. As many companies do not allow users to use web scrap from their sites, it is very important to read the terms and conditions of the website before collecting the data. The accessibility of the data is facilitated by APIs, which are developed by companies and are offered for low costs. This is totally ethical, moral and legal, as it is developed by companies directly for many purposes, including extracting information for analytical objectives.

For this specific applied research project, APIs were not used for reasons of simplicity. It is not necessary for this project to collect and analyze millions of reviews. Instead, a few hundred reviews would be enough to compare the different vectorization techniques and machine learning algorithms and show the advantages of this new method. It is only important to get real reviews from famous websites such as amazon, Walmart and Etsy.

Instead of using APIs, as only a few hundred reviews from different websites have to be collected for this university project, data was extracted directly from the HTML source of the web pages. This was implemented using the Selenium library, which allows the website content to be imported. Then, the BeautifulSoup library was used to analyze and structure the data. So, using BeautifulSoup it was possible to save the reviews from the html text into a Pandas DataFrame. This approach allows automatic, precise and targeted collection of data. After building the function using Selenium and BeautifulSoup, it was only necessary to write the amazon link of the product on jupyter notebook. After running the function, all the reviews would be automatically saved into a pandas data frame.

	Rating	ReviewTitle	ReviewBody
10	4.0	Great for hiking	Plenty of storage. Belt too short to wrap around the waistline for bigger person. All in all very useful especially for hikers or travelers.
11	5.0	Honestly a very good backpack for price	Saw a lot of reviews on here before I bought that talked about breaking etc. Took this backpack on a camping trip in the mountains and it did everything I needed it for. Held up great and felt good when carrying, I'm 5,8" 180lbs and had no problem with staying comfortable. Also fit everything I ...
12	5.0	Do NOT buy a pack pack from a store... get this one....	My 9yo used this pack but my 12yo and 14yo have the same packs and they all worked great. We did a two night back packing trip through southern Utah and they did great. The seem to be of very good quality, they are light, comfortable and have plenty of usable space. My 9yo was a little small to ...
13	4.0	Nice	But smaller than I envisioned.
14	5.0	Great backpack! Perfect size with just the right amount of pockets; water bladder works great.	Perfect size with just the right amount of pockets; water bladder works great. I have tried several items similar to this one, but so far this is my favorite one. Would definetly recommend.
15	5.0	Calidad	Excelente producto
16	5.0	Hiking Bliss Unleashed! A Scottish Sassenach's Take on the Loowoko 50L Hiking Backpack	Ach, laddies, and lassies, let me tell ye about the marvel that is the Loowoko 50L Hiking Backpack! It's like a wee castle on your back, keeping your camping essentials safe and dry. This bonnie beast comes with a waterproof rain cover, so even the dreich Scottish weather can't dampen your adven...
[illegible]	[illegible]	[illegible]	Lightweight but spacious and strong. Holds a lot of clothes and stuffs. Easy to carry for 5 feet height person as well. Can

Figure 2 Example of the collected data

In figure 2 the first 15 lines of the pandas data frame are presented. Only the column Review Body is relevant for this project, but also the Rating and title were imported, because they could be used to find further insights.

4.2 Step 2) Manual classification

After the data has been successfully collected, it is important to manually classify the reviews. This phase is necessary for training and testing the vectorization techniques and machine learning algorithms in the next steps. This step is about manually reading each review to determine whether it contains relevant information about customer needs or not. For companies, this step would be important just to train a small fraction of the data, then the huge amount of the data could be analyzed automatically.

Classification is done by assigning labels, where a '1' represents a review that provides important insights into customer needs and preferences. For instance, space for a water bottle or special features or pockets etcetera. Those reviews would be valuable because it provides direct feedback about the functionality and appreciation of product features from the customer's perspective. However, reviews that do not contain such useful information are marked with a '0'.

In figure 3 below, it is possible to see some examples of reviews, which were manually classified as “1” (those are not the complete reviews, as it they could be too long, but it is possible to see that they include important information about customer needs. Figure 4 shows examples of reviews, which are manually classified as 0 (they do not include any customer need and would not be relevant for companies).

```
In [5]: df[df["Relevant:"]==1].head()
```

Out[5]:

	ReviewBody	Relevant:
0	There is nothing about this product I don't like. Very well built and large enough to put everyt...	1.0
1	Listed to have rain cover provided. No rain cover was delivered with my back pack. Zipper also a...	1.0
2	It didn't come with the waterproof rain cover. How can this be rectified? Other than that it see...	1.0
3	Straps are so uncomfortable!! Used for only a very easy hike and my back was sore!! Less than 4 ...	1.0
5	With all the terrible storms that we been having . You never know when you will have to grab and...	1.0

Figure 3 Example of the relevant reviews

197	Thankyou decathlon for such a great product..	0.0
221	great product fast delivery I'm pleased thank you	0.0
282	a good product. It has a large capacity. Excellent quality. I traveled to Europe for a month usi...	0.0
327	Excellent product that has plenty of space and matches the description exactly!	0.0
410	Was great product	0.0

Figure 4 Example of the Irrelevant reviews

4.3 Step 3) Data cleaning

In general, machine learning algorithms perform much better if the data is cleaned (Ilyas and Rekatsinas, 2022, p. 4). Especially when the analysis is for text data. The first step is to remove comments which are not written in English using the “langdetect” library. After that, the punctuations were removed using the lambda function, and all letters were written in lowercase. Tokenization was also in important aspect in data cleaning. The purpose of this step is to make the data in one format and simplify the analysis for the next steps.

After that, the stop words were removed, as they are not relevant for the analysis. Then follows the lemmatization step. This step is about replacing each word with its origin (Shah et al., 2020, p. 53). For instance, “better” would be transferred into “good“, „made” would be transferred into “make”. WordNetLemmatizer from nltk.stem did not work in an optimal way, so spacy library was used for this purpose.

After that, numeric values and short words (one or two characters) were also removed, as those make the analysis complicated and do not improve the content of sentences. Also, non-alphabetic values were removed. After that, spellchecker was implemented to correct words, which were spelled wrong. Figure () shows an example of comments, which were corrected (“carefull”) on the left side was corrected into “careful” (only one L). As vectorization techniques will be used, algorithms would perform worse, if same words are spelled each time differently.

[backpack, awesome, like, design, glass, pouch, neat, bladder, come, seem, cheap, bite, valve, hard, use, carefull, fill, cap, leak]	[backpack, awesome, like, design, glass, pouch, neat, bladder, come, seem, cheap, bite, valve, hard, use, careful, fill, cap, leak]

Figure 5 Data cleaning

Now the data is cleaned, and it is ready to start with the vectorization techniques and the machine learning algorithms.

4.4 Step 4) Vectorization Techniques

Vectorization is a crucial process in natural language processing (NLP). It plays a central role because it converts text into vectors, that can be processed by computers and machine learning algorithms. In general, Vectorization makes it possible to make text data usable for a variety of applications, such as text classification, sentiment analysis etcetera (Rani et al., 2022, p. 1).

so, vectorization allows text data to be processed and analyzed in ways that would not be possible without it. However, there are many vectorization techniques, which work differently and needs different processing time and memory (Rani et al., 2022, p. 5).

Each of the four vectorization techniques TF-IDF, Count Vectorization, FastText and Hashing will be implemented with different machine learning algorithms to compare and see how each one performs. In the following chapter, those will be only explained, then the results will be presented.

In order to explain how each of those techniques work, the explanation will be based on very simple examples.

4.4.1 Count vectorization

Count vectorization (or bag of words) is one of the most famous and simple vectorization techniques in text processing that converts text data into a numerical form to make it usable for machine learning and data analysis. The process begins with creating a vocabulary that includes all unique words from the entire text corpus. For each word in the corpus, the method counts how often each word from the vocabulary occurs in that text. These frequencies are noted in the corresponding positions of the vector. So the size of the vector is the number of the unique word number in the text (Rani et al., 2022, p. 3).

For instance, if we have only those three reviews:

```
sentences = [
    'good product',
    'good product love waterproof feature',
    'all good love the extra pocket, good feature'
]
```

The vectors would look like the following table:

	all	extra	feature	good	love	pocket	product	the	waterproc
good product	0	0	0	1	0	0	1	0	
good product love waterproof feature	0	0	1	1	1	0	1	0	
all good love the extra pocket, good feature	1	1	1	2	1	1	0	1	

Figure 6 Example of the Count vectorization

This method is often used for several advantages and disadvantages. First of all, this method is very simple. As a result, the vectors could be understood for the user. It is optimal and fast for small databases. However, there are also several disadvantages. The most important one is that it is only based on the frequency of the words without taking the context or the order of words into consideration. As a result, sentences which include some of the same words could be classified into the same category (even if they have opposite meanings). Also, as it is only based on the frequency of the words, that occurs very often, would have much more influence than rare words, so machine learning algorithms may not perform in an optimal way. Another disadvantage is the Memory Consumption, as for each sentence a vector will be built in the length of the unique words of all the sentences. This could lead to memory problem for huge datasets, and it could be harder to train the machine learning algorithms, if the vectors are huge (Kim et al., 2017, p.3).

4.4.2 TF-IDF

Term Frequency-Inverse Document Frequency" is also a vectorization technique in text processing that aims to convert text data into a numerical form, so that it can be processed by the machine leaning algorithms. As the name says, this method consists of term frequency (TF) and inverse document frequency (IDF) (Qaiser and Ali, 2018, p. 25). While TF counts the frequency of a word in a single document, IDF measures how unique or rare that word is in the entire corpus. The combination of these two measures results in a balanced evaluation of words (Rani et al., 2022, p. 3).

For the same example as above:

sentences = ['good product',

'good product love waterproof feature',

'all good love the extra pocket, good feature']

First part is the TF (same as count vectorization above):

	all	extra	feature	good	love	pocket	product	the	waterproof
good product	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
good product love waterproof feature	0.0	0.0	1.0	1.0	1.0	0.0	1.0	0.0	1.0
all good love the extra pocket, good feature	1.0	1.0	1.0	2.0	1.0	1.0	0.0	1.0	0.0

Figure 7 Example of the TF

The second part is the IDF, which gives rare words more importance than the words, that occurs very often. This is calculated using the formel (Rani et al., 2022, p. 4):

$$IDF = \text{natural log}\frac{Number\ of\ sentences+1}{Number\ of\ sentences\ which\ include\ the\ word+1} + 1$$

I have implemented this using python (by using TfidfVectorizer from sklearn library) in order to show the idf values for our example:

	all	extra	feature	good	love	pocket	product	the	waterproof
0	1.693147	1.693147	1.287682	1.0	1.287682	1.693147	1.287682	1.693147	1.693147

Figure 8 Example of the IDF

For example for "good" occurs in each sentence so it is

$IDF = natural\ \log\frac{3+1}{3+1} + 1 = 1$.

The word product appears only in one sentence, so it would be

$IDF = natural\ \log\frac{3+1}{1+1} + 1 = 1.693$

As a result, if a word occurs rarely, it will have more significance than other words and it would still influence the classification, even if it occurs only once or twice.

The last step would be to multiply the TF (frequency) with the IDF and the normalize the results.

An advantage of TF-IDF is that it is still a simple method and understandable for the users. The main advantage is that TF-IDF can help minimize the impact of over-frequent words (Qaiser and Ali, 2018, p. 25). This aspect is an improvement in comparison to count vectorization. However, TF-IDF also has some challenges. Like count vectorization, it does not take context or word order into account, which can cause problems in some applications (Qu et al., 2008, p. 80). Despite this disadvantage, TF-IDF remains a popular and effective method in the world of text analysis and machine learning.

4.4.3 Hashing vectorization

The hashing technique plays a crucial role in natural language processing when it is about vectorizing huge amounts of data text (Kanada, 1990, p. 1). These techniques use a hash function to convert words or phrases into numerical vectors with a specified and constant vector length. The main advantage of this method is its storage efficiency, as the size of the feature vector is constant and predefined, regardless of the vocabulary size of the unique words of the data (Tang et al., 2019, p.1). For count vectorization and TF-IDF the length of the vector is the number unique words in all the documents.

The hashing technique is particularly useful in situations where storage space is limited. Unlike other vectorization methods such as Count Vectorization or TF-IDF, which store a dictionary of unique words in the corpus, hashing technique does not require storage space for such a dictionary. As a result, it is much faster than than tf-idf and count vectorizations (Kanada, 1990, p. 1).

However, the hashing technique also has some challenges and limitations. The most important limitation is the irreversibility of the hash function: once converted into an index, the original word can no longer be reconstructed. This leads to a loss of interpretability of the features. There is also the risk of collisions where different words receive the same hash value. Such collisions can affect the accuracy of text representation, especially if the number of features is much smaller than the number of unique words in all the text (Kanada, 1990, p. 1).

For example, after using count vectorization and TF-IDF, I was able to know the influence of each word regarding the classification. In figure (9) below, it is possible to see the words and coefficients of relevant and not relevant words, when I used the TF-IDF with linear regression.

```
Relevant
          Feature  Coefficient
786        pocket     1.521491
590   lightweight     1.076787
739          pack     0.964879
245         cover     0.948441
835          rain     0.870783
1019        strap     0.859180
1195         well     0.829796
115       bladder     0.806387
105          belt     0.747052
1184        water     0.687797
Not relevant
        Feature  Coefficient
457       great    -0.828893
828     quality    -0.810476
810       price    -0.594118
1156        use    -0.567260
406       first    -0.543716
386   fantastic    -0.537763
148         buy    -0.460508
1163      value    -0.453630
453        good    -0.443673
```

Figure 9 Coefficient of the words

In summary, hashing techniques provide an efficient solution for processing large amounts of text in NLP, while could having some problems regarding accuracy and interpretability.

4.4.4 Fasttext

In contrast to the last three vectorization techniques, FastText is an advanced approach in the world of machine learning. This was developed by Facebook's AI Research Lab, to assign meanings to words in text in the form of vectors (Le et al., 2019, p. 4). This process is called "word embedding" and is particularly important for enabling computers to understand the text data (Athiwaratkun et al., 2018, p. 1).

In contrast to older methods such as TF-IDF and count vectorization, which are based only on whole words, FastText also looks at smaller units within the words, so-called "subwords" or letter combinations. This is comparable to not only looking at the word "vectorize" as a whole, but also looking at parts such as "vec", "tor" and "ize". This approach helps FastText better understand the meaning of words and even generate meaningful vectors for words. As a result, words which are misspelled would still have similar vectors as the correct spelling and not a completely different vector (Young and Rusli, 2019, p.2). For example, if a specific word was not in the training data set and occurs in the test set, Count vectorization or TF-idf, or hashing vectorization would not be able to understand the world and it would not have any effect on the sentence. However, fasttext would be able to use the vectors from other words (with the same subwords) in the training dataset and use it for the classification problem. FastText's learning process occurs through analyzing a large corpus of text. The model learns which vectors should be assigned to the different words and their subwords. The result is that words with similar meanings lie close to each other in what's called vector space, meaning they have similar vectors (Kal, 2021).

Fasttext has many advantages. First of all, this is particularly good at handling unfamiliar words (Ouchene and Bessou, 2023, p. 1). In contrast to older methods that only look at whole words, because FastText also analyzes subwords or letter combinations. The word vectors are often richer, resulting more accurate modeling of the meaning of words. FastText has also proven effective in various areas of text classification because it can capture the semantic meaning of words well (Pribadi et al., 2021, p. 382).

On the other hand, FastText also has some disadvantages. The main disadvantage is that it is very computationally intensive and requires longer times and computing power (especially on large amounts of text and with many subwords). In addition, the generated word vectors can require a lot of storage space. These aspects can be particularly problematic when processing very large amounts of data or when the processor of the computer is not very advanced. this was not a problem for my project, as the dataset needed for the objective of the project is not huge. However, I have noticed that it takes longer than the other vectorization techniques.

In summary, FastText provides a powerful way to deeply analyze and understand the meanings of words in texts, making it a valuable tool in language processing. However, the processing part is much more complicated than the traditional technique.

4.5 Step 5) Classification algorithms

In this chapter, the used machine learning algorithms will be presented. It will be shown what the main idea is and how they work. However, it would not be possible to cover every single aspect in detail. Instead, I will provide an overview of the most important categories and concepts.

The specific results and applications of these algorithms will be presented in the “Results” chapter. It is also important to mention that for all machine learning algorithms, the same train and test data has been used. So, the splitting into train and test datasets happens before trying the different vectorization methods and the classification algorithms. 80% of the data were used to train the models and 20% were used as a test dataset to evaluate the different approaches.

4.5.1 Logistic Regression

Logistic regression is a statistical model used mainly to solve classification problems. This model is very useful, especially when there are only two possible outcomes (for example yes or no, success or failure, presence or not etc.). This is why I have tried the logistic regression for our classification problem (whether a review is relevant or not).

Logistic regression calculates the probability of an outcome (for example "relevant") and then classify into the category with the highest probability. It uses the logistic function (sigmoid function) as presented below. This output of the function is a value between 0 and 1. This value is the probability and will determine to which category this would be classified (Coggeshall and Klinkenberg, 2014, pp. 6-7).

$$f(\boldsymbol{x}) = w_0 + \sum_{j=1}^{n} w_j x_j$$

$$P(\mathrm{y} = +1 | \mathbf{x} = \boldsymbol{x}) = \sigma(f(\boldsymbol{x})) = \frac{1}{1 + \exp(-f(\boldsymbol{x}))}$$

Logistic regression formula (Faouzi and Colliot, 2022)

As figure shows, it is easy to understand the output of logistic regression. After training the logistic regression function on my training dataset, it was possible to see the coefficients and understand the prediction for each output.

```
third: -0.1589517370518005
though: 0.5041905193090467
thought: -0.06605575983509188
thoughtfully: 0.0672612059985992
three: -0.20267136436626937
throw: 0.0840961826600233
tight: 0.30631396357420415
time: 0.25877960927597454
tiny: 0.3522186681158529
tip: 0.07921892072944597
tiptop: 0.07277346501704937
tired: 0.078053284430209
today: 0.09633268816853267
together: 0.05612026576525116
toiletry: 0.16876142538281008
toll: 0.09633268816853267
ton: -0.015145044197401725
toodefinitely: 0.08512355831727567
```

Figure 11 Example of the Coefficients in the Logistic Regression

Relevant

	Feature	Coefficient
786	pocket	1.521491
590	lightweight	1.076787
739	pack	0.964879
245	cover	0.948441
835	rain	0.870783
1019	strap	0.859180
1195	well	0.829796
115	bladder	0.806387
105	belt	0.747052
1184	water	0.687797

Not relevant

	Feature	Coefficient
457	great	-0.828893
828	quality	-0.810476
810	price	-0.594118
1156	use	-0.567260
406	first	-0.543716
386	fantastic	-0.537763
148	buy	-0.460508
1163	value	-0.453630
453	good	-0.443673

Figure 10 The highest and lowest coefficients

Figure (11) above shows a part of the words and their coefficients. Figure (10) on the right side shows the words in an ascending and descending order. It is interesting to see that relevant words are features, for example “pocket”, “lightweight”, “cover” etcetera. An irrelevant words are not relevant at all for product development such as “great”,”product”. For example, if I want to predict the review: “great”.

The probability would be using the formula in figure above:

$$\text{probability (of class 0)} = \frac{1}{1 + e^{-(x0+w1*x1+w2*x2\ etc.)}}$$

$$\text{probability (of class 0)} = \frac{1}{1+e^{-(-0.33188+1*-0.828893\ (coefficient\ of\ "great")}} = 0.2385$$

as 0.2385 <0.5, the review „great“ would be classified as 0 (non relevant). The Figures below shows the prediction manually using the logistic regression function, and once using the model.

```
In [31]:

fx = -0.33188318004059497 + (-0.828893 * 1)

probability = 1 / (1 + np.exp(-fx))
probability

Out[31]: 0.23852627780732896
```

```
In [30]:
new_review = ["great"]
predicted_probabilities = tfidf_model.predict_proba(new_review)
print("the probabilities are", predicted_probabilities)
predicted_labels = tfidf_model.predict(new_review)
print("classified into", predicted_labels)

the probabilities are [[0.76147365 0.23852635]]
classified into [0.]
```

Figure 12 Calculation of the logistic Regression Probabilities

Logistic regression has several advantages. First of all, it is easy to understand and implement in comparison to the other classification algorithms (Dineva and Atanasova, 2020, p.319). In addition, their results are easy to interpret because the weights provide information about the importance of individual characteristics (as shown above). Another advantage is the ability to provide probabilities for the classification results, which could be important in some cases (also as showing above) (Srinivas et al., 2023, p.5)

However, there are also disadvantages. Logistic regression assumes linear relationships between the input characteristics and the outcome, which is not always true (Dineva and Atanasova, 2020, p.319). It would not work optimally for very complex data structures because it can only model linear relationships. Additionally, with a high number of features and not enough data, there is a risk of overfitting.

Overall, logistic regression is a useful method for many classification problems, especially for simple datasets. However, more advanced methods could be needed for more complex tasks.

4.5.2 Support Vector Machines

The Support Vector Machine (SVM) is also one of the most famous and used classification algorithm in the field of machine learning (It could be also used for regression problems, but mainly for classification problems). One could say that SVM is known for its effectiveness when working with high-dimensional data and often provides very precise predictions.

The main idea of SVM is finding a decision boundary (hyperplane), that separates the different classes in an optimal way (Coggeshall and Klinkenberg, 2014, p. 12). In a binary classification, the goal is to find a hyperplane that has maximum distance to the closest data points of each class, which are called support vectors. The algorithm maximizes the margin, the distance between the support vectors and the hyperplane, to achieve a clear separation between classes. In situations where the data is not linearly separable, SVM uses the a "kernel trick". This makes it possible to transform the data into a higher dimensional space in which linear separation would be possible (Wu and Kumar, 2009, p. 42).

For my project, when I have used the tf-idf or the count vectorization technique, the length of the vector is only 1240. As it is not possible to visualize this, I have used the trancuatedSVC library, in order to transform it into a 2-dimensional dataset. The dataset looked like the following Figure. Yellow points are relevant reviews, whereas blue does not include relevant information.

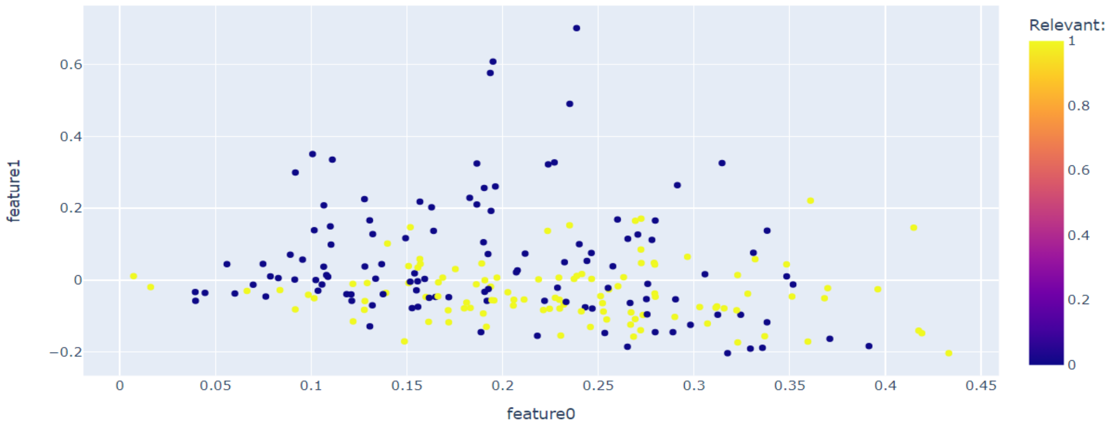

Figure 13 Reviews vectors in 2 Dimensions

It is possible to see that the points are not perfectly separated. However, it is clear that the relevant comments are grouped together in the lower part of the diagram.

I have trained the SVM function on the 2-dimensional dataset, and the output (of the training reviews) looked like in figure () on the left side below. The left side shows the Points in the red area were classified as relevant and the other parts were classified as not relevant. The exact results (Accuracy, Recall, precision) will be presented in the next chapter.

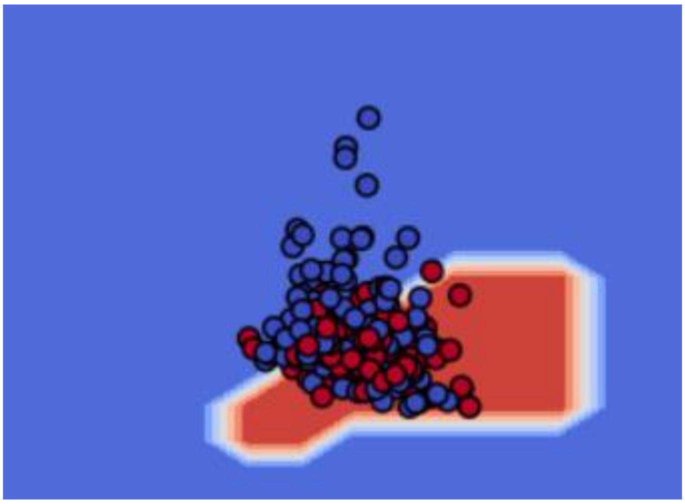

Figure 15 Training dataset

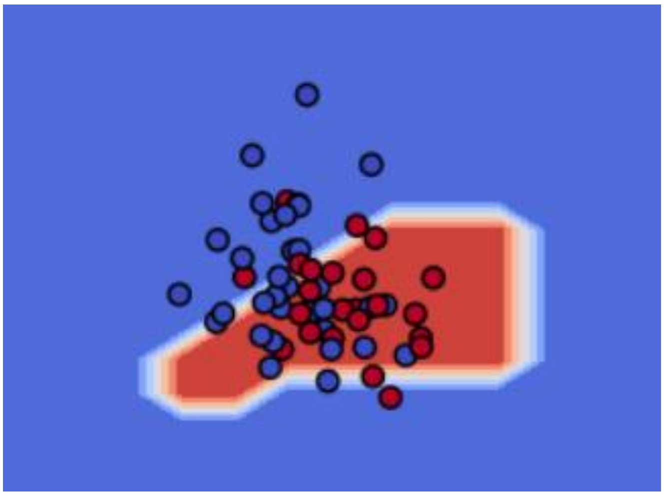

Figure 14 Test Dataset

There are many advantages for SVM. First of all, it is effective when the data set has many features, even if the number of features is greater than the number of observations. By using various kernel functions, SVM can be adapted to a variety of data sets (Dineva and Atanasova, 2020, p. 319). In addition, SVM is known for its high accuracy in complex classification problems. Another advantage is that SVMs mostly provide only one unique solution to divide the data and as a result, the classification would be more understandable for researchers (Awad and Khanna, 2015, pp. 42-43). Moreover, researchers have noticed in general that overfitting problems do not occur as much as when using other classifiers.

However, there are also disadvantages. SVM can be inefficient when dealing with huge data sets because computing the support vectors can be computationally intensive. Another disadvantage is that SVM only works on binary outcomes. In addition, when the points of the two classes overlap with a large dataset, it would not be easy to use the kernel concept to build the hyperplane. Also, it can be said the "process is time-consuming" (Dineva and Atanasova, 2020, p. 319).

In summary, SVM is a powerful tool for classification problems, especially for complex datasets with high dimensionality. The challenges when using SVM lie in choosing the right parameters and kernel function as well as dealing with huge data sets.

4.5.3 Random Forest

The Random Forest algorithm is a widely used and effective method in the field of machine learning. This could be used for both classification and regression tasks. It is one of the ensembles learning methods in which multiple models work together to achieve better results than individual models could (Coggeshall and Klinkenberg, 2014, p. 494).Like the name of the classifier implies, random forest uses several decision trees to process the data)

The main idea of the random forest is to use numerous decision trees. Each of these trees is built independently using a random sample of the training data (through the bagging procedure) and a random selection of features at each split in the tree. This randomness helps reduce the variance of the overall model and helps avoid overfitting, which is a common problem with single decision trees (Jordan and Mitchell, 2015 cited in Boyko et al., 2023, p. 2).

In classification, the final decision of the random forest is made by the majority vote of all trees' predictions. In regression problems, the average of the predictions of all trees is used. One of the biggest advantages of the random forest is the accuracy. It is typically less prone to overfitting than single decision trees and can handle unbalanced and incomplete data in an optimal way. Also, it would not be affected by outliers as much as other classification algorithms could be affected (Ali et al., 2012, p. 273)

The disadvantages include that building many decision trees can be computationally intensive, especially for large data sets. Additionally, random forest models are more difficult to interpret compared to individual decision trees (Boyko et al., 2023, p. 7). For example, when I have tried the logistic regression, I was able to see the effect of each word on the model. This could not be possible for random forests. Furthermore, when training a model for an extremely feature-rich data the performance of the random forest may also decrease.

In summary, Random Forest is a important and powerful algorithm in machine learning that is suitable for a wide range of applications. Its strengths lie in high prediction accuracy and avoiding overfitting, while the disadvantages are mainly the processing complexity and model interpretability.

4.5.4 Neural Networks

Neural networks are an important and famous technique in the field of machine learning. This is mostly used to solve classification problems, regression, picture recognition etcetera (Camburu, 2020). The idea of neural networks is similar to the way of the human brain, as it consists of a series of connected nodes in different layers as presented in the next figure (Uzair and Jamil, 2020, p1).

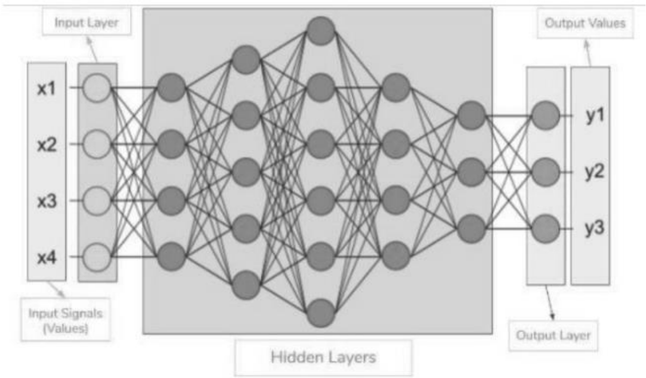

Figure 16 Structure of the Neural Networks (Uzair and Jamil, 2020, p1)

Figure (16) shows an example of the structure of neural networks. This consists of one input layer, at least one hidden layer and an output layer. Each neuron is connected to many neurons in the next layer. These neurons process input signals and transfer them to the next layer. For my project, I have used the following structure: MLPClassifier(hidden_layer_sizes=(1000,), max_iter=200, random_state=42). So I have used only one hidden layer with 1000 nodes. This is because my dataset is small, and "Overfitting occurs when the number of hidden layers become very large as compared to the complexity of the problem" (Uzair and Jamil, 2020, p2)

The learning process in neural networks works by adjusting the weights between neurons, which is typically achieved through a process called backpropagation (Wythoff, 1993, p. 119). Neural networks offer a high level of flexibility and can recognize complex, non-linear patterns in data, in contrast to logistic regression. They are particularly effective in areas such as image and speech recognition. In addition, they can independently learn to extract optimal features from the raw data, and they scale well with the amount of data and problem complexity (Uzair and Jamil, 2020, pp. 2-4).

However, neural networks also bring with them some challenges. The training process can be very computationally and memory intensive, especially for large networks. They may also be prone to overfitting, which can affect their ability to generalize (Shafi et al. 2007 cited in Uzair and Jamil, 2020, p4). Additionally, they are often difficult to interpret and explain, and choosing the right network architecture can be complex (Panchal and Panchal, 2014 cited in Uzair and Jamil, 2020, p4).

Overall, neural networks have proven to be extremely powerful in many application areas, for example, image recognition, language processing, complex classification processes. Its main strength is the ability to learn from large amounts of data and detect patterns that are difficult for other algorithms to access (non-linear patterns). The challenges are mainly the complexity of the networks, the high resource requirements and the difficult interpretability.

4.5.5 Ensemble Method

The Voting Classifier is a special type of ensemble model in the machine learning, I have imported this from the scikit-learn library. It combines the predictions of different classification models and makes a final decision based on the majority of these predictions. I have combined all the above algorithms (Logistic regression, support vector machines, random forest, neural networks) in the voting classifier. In the VotingClassifier there are two main types of voting, this is either “hard” or “soft” voting. In hard voting, the final prediction of the ensemble model is made by a simple majority vote. Each base model in the ensemble makes a prediction (e.g. class A or class B), and the class that receives the most votes is selected as the final prediction of the ensemble model. This approach is simple and straightforward (“sklearn.ensemble.VotingClassifier,” n.d.). According to many scientific papers, for example , this “hard voting” method could work in an optimal when all base models are about the same strength. The other way is soft voting, the prediction is based on the probabilities that each base model calculates for each class. Instead of just choosing the class with the most votes, the probabilities for each class are averaged across all models and the class with the highest average probability is selected. Soft voting can be more effective than hard voting because it takes more information into account - not just the final class prediction of each model, but also the certainty of those predictions (Ahmed, 2023).

The main advantage of the VotingClassifier is the improvement in prediction performance. By combining different models, the overall system can often achieve better results than each individual model on its own (Mohammed and Kora, 2023, p. 770). This is because the weaknesses of one model can be offset by the strengths of another. Additionally, combining different models can help reduce the overall variance (Coggeshall and Klinkenberg, 2014), which is particularly useful for minimizing the risk of overfitting. The VotingClassifier also offers flexibility through the ability to use both hard voting and soft voting, which makes it usable for different cases.

However, using the VotingClassifier has also some disadvantages. Using multiple models increases computing power and storage requirements and can extend training and prediction times. For my project only a small dataset was required, but it still took much longer time than the other individual models. The voting logic, particularly the choice between hard and soft voting, could be complex and requires careful consideration. Furthermore, combining different models can lead to unexpected and results, which could not be explained or interpreted. In addition, ensemble methods such as the VotingClassifier can be more difficult to interpret than individual models, which can be problematic in certain application areas where transparency is required.

Overall, the VotingClassifier offers a powerful opportunity to take advantage of different classification approaches but requires long time, computing power and storage requirements.

5 Results

After implementing the vectorization methods and training the classification algorithms, those methods have been evaluated on the test data (which is surely new data and was not part of the training data set). The evaluation is based on both the accuracy of the methods and also on the recall percentage of the relevant reviews. This very important, as there could be methods with high accuracy, and low recall percentage or other ways. In the following two tables, the results will be presented (once sorted based on the recall rate, and the other on the accuracy rate).

Table 1 Evaluation of the 20 different categorizations (sorted by Recall)

Vectorization technique	Classification algorithm	Overall Accuracy	Recall of the relevant Re-views
Fasttext	Neural Networks	53%	97%
Fasttext	Support vector machines	55%	96%
TF-IDF	Neural Networks	77%	87%
TF-IDF	Support vector machines	75%	83%
Count Vectorization	Support vector machines	75%	83%
Fasttext	Logistic Regression	66%	83%
Fasttext	Random Forest	66%	83%
Fasttext	Ensemble method	64%	83%
hashing vectorizer	Random Forest	79%	78%
TF-IDF	Ensemble method	71%	78%
TF-IDF	Random Forest	80%	74%
Count Vectorization	Ensemble method	80%	74%
Count Vectorization	Logistic Regression	79%	74%
Count Vectorization	Random Forest	79%	74%
TF-IDF	Logistic Regression	73%	74%
hashing vectorizer	Neural Networks	64%	70%
hashing vectorizer	Support vector machines	73%	65%
Count Vectorization	Neural Networks	73%	65%
hashing vectorizer	Ensemble method	68%	65%
hashing vectorizer	Logistic Regression	71%	61%

Table 2 Evaluation of the 20 different categorizations (sorted by Accuracy)

Vectorization technique	Classification algorithm	Overall Accuracy	Recall of the relevant Re-views
TF-IDF	Random Forest	80%	74%
Count Vectorization	Ensemble method	80%	74%
hashing vectorizer	Random Forest	79%	78%
Count Vectorization	Logistic Regression	79%	74%
Count Vectorization	Random Forest	79%	74%
TF-IDF	Neural Networks	77%	87%
TF-IDF	Support vector machines	75%	83%

Count Vectorization	Support vector machines	75%	83%
TF-IDF	Logistic Regression	73%	74%
hashing vectorizer	Support vector machines	73%	65%
Count Vectorization	Neural Networks	73%	65%
TF-IDF	Ensemble method	71%	78%
hashing vectorizer	Logistic Regression	71%	61%
hashing vectorizer	Ensemble method	68%	65%
Fasttext	Logistic Regression	66%	83%
Fasttext	Random Forest	66%	83%
Fasttext	Ensemble method	64%	83%
hashing vectorizer	Neural Networks	64%	70%
Fasttext	Support vector machines	55%	96%
Fasttext	Neural Networks	53%	97%

All in all, it can be said that this method is working and would save companies a lot of time, as the accuracy is high (80%). Also, because of the high recall rates of the relevant reviews, not much information will be lost during the process. The training was only on 300 reviews, so much better accuracies could be achieved, if the training set were larger.

The best method regarding only the accuracy was using the fasttext vectorization method, combined with the neural networks or Support vector machines (recall rate of 96-97% and an accuracy of 53-55%). However, this would not be beneficial for companies, because the accuracy is remarkably low. As a result, this would not save the companies much time with filtering the results.

The best method regarding only regarding the accuracy was the random forest classification algorithm and the ensemble method combined with TF-IDF vectorization. This method was able to achieve an accuracy of 80% and a Recall percentage of 74%.

In general, the Neural network classification algorithm with the TF-IDF vectorizing was the best method. This method achieved an accuracy of 77% with a recall percentage of 87%.

I have tried the same method (Neural network algorithm with the TF-IDF vectorizing) on the review column before cleaning, and the results were worse (71% instead of 77%). As a result, the step of data cleaning was very important to improve the results. This is especially important for the vectorization techniques, which were based only on the frequency of the words.

As results for the whole project, it can be said that this approach is not complicated and would be very beneficial for companies. As it is not as time consuming and costly as the traditional approaches and would be able to identify much more data in comparison to surveys or interviews. It is very hard to say which combination of vectorization technique and classification algorithms works the best, because each one of them has advantages and disadvantages. For example, whether the interpretation of the classification is needed or not. Another example is the time and space complexity of the classification algorithms. There is a need to do further researchers in order to improve this approach, so that it could replace the traditional methods and could be used for companies completely.

6 Improvements for further research

First of all, this project is based only on supervised methods. As a result, companies would need to first manually classify a small part of the data in order to be able to train the classification algorithms. It would be practical to test some of the unsupervised algorithms, that the progress of the classification would be completely automated. Secondly, the output of the applied method would be mostly relevant reviews which include customer needs. However, some of the reviews could be long. This is why it would be practical if there would be an algorithm, which gives only the relevant part of the relevant reviews. This would also save companies long times.

For this project, I had to find the link of a product on amazon. then the reviews would be automatically collected and saved. However, using APIs could make it possible to extract all the reviews from different products automatically without the need to find different links first. This is definitely possible on some websites like twitter (now X), in which it is possible to save all blogs, which contain specific keywords. There is only one scientific paper by Timoshenko and Hauser, (2019) where both of the new approach and traditional way are implemented and compared. It would be important to have more research about the results of this method in comparison to doing interviews.

7 Conclusion

Identifying customer needs has always been an important topic to companies and it influences the customer satisfaction (Wang and Ji, 2010, p. 174). Traditional ways of extracting customer needs are qualitative and quantitative approaches, which have many disadvantages, especially when it comes to the cost or time aspect. In addition, there are other disadvantages such as the complexity of identifying the rare needs or that competitor companies could know the purpose of the research (for example if the company is looking for participants for interviews) (Timoshenko and Hauser, 2019, p. 5). In this context, it can be concluded that extracting customer needs from User Generated Content is a useful method for this purpose. This new method solves the disadvantages of the traditional methods, as the process is less cost and time consuming among other advantages. For example, the UGC as a source of data is accessible effortlessly and fast for everybody. However, the challenge is to analyze the large amount of data in order to collect useful information, which includes specific customer needs.

This process is possible using Vectorization techniques, Deep and Machine Learning algorithms, as there are classifiers which automatically divide the data into two classes, namely informative and non-informative class regarding customer needs. This method is possible by firstly preprocessing the data by transforming the words into vectors. The next step is to apply the classifier algorithms to divide the data. Many algorithms have been compared in this project, but it is hard to decide which classifier is the best as this depends on the data set. The Random Forest algorithm with the TF-IDF was the best method for my dataset regarding recall and Accuracy. However, for some other cases, other vectorization techniques or algorithms could perform better. It is important to point out that I have also tried the classification step without cleaning the data first, and the results were significantly worse. This shows the importance of data cleaning before implementing the other methods.

To conclude, this thesis has shown that extracting customer needs from User Generated Content on a real dataset of customer reviews, and that it could be very beneficial for companies. This new method can improve the whole process of identifying customer needs and enhance the product planning practice. However, further researchers are still needed to replace the old methods.

References

Ahmed, I., 2023. What is Hard and Soft Voting in Machine Learning? Medium. URL https://ilyasbinsalih.medium.com/what-is-hard-and-soft-voting-in-machine-learning-2652676b6a32 (accessed 1.7.24).

Ali, J., Khan, R., Ahmad, N., Masqsood, I., 2012. Random Forests and Decision Trees - ProQuest, in: IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012 ISSN (Online): 1694-0814 Www.IJCSI.Org.

Athiwaratkun, B., Wilson, A.G., Anandkumar, A., 2018. Probabilistic FastText for Multi-Sense Word Embeddings. https://doi.org/10.48550/arXiv.1806.02901

Awad, M., Khanna, R., 2015. Support Vector Machines for Classification, in: Awad, M., Khanna, R. (Eds.), Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers. Apress, Berkeley, CA, pp. 39–66. https://doi.org/10.1007/978-1-4302-5990-9_3

Boyko, N., Omeliukh, R., Duliaba, N., 2023. The Random Forest Algorithm as an Element of Statistical Learning for Disease Prediction, in: Pickl, S.W., Lytvynenko, V., Zharikova, M., Sherstjuk, V. (Eds.), Proceedings of the 3rd International Workshop on Computational & Information Technologies for Risk-Informed Systems (CITRisk 2022), CEUR Workshop Proceedings. Presented at the Computational & Information Technologies for Risk-Informed Systems 2022, CEUR, Online Event, Neubiberg, pp. 59–73.

Camburu, O.-M., 2020. Explaining Deep Neural Networks. https://doi.org/10.48550/ARXIV.2010.01496

Coggeshall, S., Klinkenberg, R., 2014. Data Classification: Algorithms and Applications. https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Data%20Classification_%20Algorithms%20and%20Applications%20%5BAggarwal%202014-07-25%5D.pdf

Dineva, K., Atanasova, T., 2020. SYSTEMATIC LOOK AT MACHINE LEARNING ALGORITHMS - ADVANTAGES, DISADVANTAGES AND PRACTICAL APPLICATIONS. https://doi.org/10.5593/sgem2020/2.1/s07.041

Faouzi, J., Colliot, O., 2022. Classic machine learning algorithms, in: Colliot, O. (Ed.), Machine Learning for Brain Disorders. Springer.

Farid, ش.ط., Mousa, غ.م., 2023. A Comparative Study between User Generated Content (UGC) and online survey for measuring customer perception toward service quality in hotel industry: Case of Tolip Hotel in Egypt. المجلة العلمية للدراسات والبحوث المالية والتجارية 4, 135–182. https://doi.org/10.21608/cfdj.2023.258040

George, L., Sumathy, P., 2023. An integrated clustering and BERT framework for improved topic modeling. Int. j. inf. tecnol. 15, 2187–2195. https://doi.org/10.1007/s41870-023-01268-w

Ilyas, I.F., Rekatsinas, T., 2022. Machine Learning and Data Cleaning: Which Serves the Other? J. Data and Information Quality 14, 13:1-13:11. https://doi.org/10.1145/3506712

Jordan, M.I., Mitchell, T.M., 2015. Machine learning: Trends, perspectives, and prospects. Science 349, 255–260. https://doi.org/10.1126/science.aaa8415

Kal, 2021. Word Embedding Using FastText. Medium. URL https://medium.com/@93Kryptonian/word-embedding-using-fasttext-62beb0209db9 (accessed 1.6.24).

Kanada, Y., 1990. A vectorization technique of hashing and its application to several sorting algorithms, in: Proceedings. PARBASE-90: International Conference on Databases, Parallel Architectures, and Their Applications. Presented at the PARBASE-90: International Conference on Databases, Parallel Architectures, and Their Applications, IEEE Comput. Soc. Press, Miami Beach, FL, USA, pp. 147–151. https://doi.org/10.1109/PARBSE.1990.77135

Kim, H.K., Kim, H., Cho, S., 2017. Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. Neurocomputing 266, 336–352. https://doi.org/10.1016/j.neucom.2017.05.046

Kühl, N., Scheurenbrand, J., Satzger, G., 2020. Needmining: Identifying micro blog data containing customer needs. https://doi.org/10.48550/ARXIV.2003.05917

Le, N.Q.K., Yapp, E.K.Y., Nagasundaram, N., Yeh, H.-Y., 2019. Classifying Promoters by Interpreting the Hidden Information of DNA Sequences via Deep Learning and Combination of Continuous FastText N-Grams. Frontiers in Bioengineering and Biotechnology 7.

Liu, L., Zhang, L., Ye, P., Liu, Q., 2019. User Needs Mining Based on Topic Analysis of Online Reviews. Tehnički vjesnik 26, 230–235. https://doi.org/10.17559/TV-20181218012812

Luo, H., Song, W., Zhou, W., Lin, X., Yu, S., 2023. An Analysis Framework to Reveal Automobile Users' Preferences from Online User-Generated Content. Sustainability 15, 13336. https://doi.org/10.3390/su151813336

Mohammed, A., Kora, R., 2023. A comprehensive review on ensemble deep learning: Opportunities and challenges. Journal of King Saud University - Computer and Information Sciences 35, 757–774. https://doi.org/10.1016/j.jksuci.2023.01.014

Ouchene, L., Bessou, S., 2023. FastText Embedding and LSTM for Sentiment Analysis: An Empirical Study on Algerian Tweets, in: 2023 International Conference on Information Technology (ICIT). Presented at the 2023 International Conference on Information Technology (ICIT), pp. 51–55. https://doi.org/10.1109/ICIT58056.2023.10226060

Pribadi, N.H., Sarno, R., Ahmadiyah, A.S., Sungkono, K.R., 2021. Semantic Recommender System Based on Semantic Similarity Using FastText and Word Mover's Distance. | International Journal of Intelligent Engineering & Systems | EBSCOhost [WWW Document]. https://doi.org/10.22266/ijies2021.0430.34

Price, R.A., Wrigley, C., Straker, K., 2015. Not just what they want, but why they want it: Traditional market research to deep customer insights. Qualitative Market Research: An International Journal 18, 230–248. https://doi.org/10.1108/QMR-03-2014-0024

Qaiser, S., Ali, R., 2018. Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. International Journal of Computer Applications 181. https://doi.org/10.5120/ijca2018917395

Qu, S., Wang, S., Zou, Y., 2008. Improvement of Text Feature Selection Method Based on TFIDF, in: 2008 International Seminar on Future Information Technology and Management Engineering. Presented at the 2008 International Seminar on Future Information Technology and Management Engineering, pp. 79–81. https://doi.org/10.1109/FITME.2008.25

Rani, D., Kumar, R., Chauhan, N., 2022. Study and Comparision of Vectorization Techniques Used in Text Classification, in: 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT). Presented at the 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–6. https://doi.org/10.1109/ICCCNT54827.2022.9984608

Rensing, L., 2022. Twitter as a Data mine: Can user needs be derived from Twitter Data? - An example of Airbnb [WWW Document]. URL https://essay.utwente.nl/90936/ (accessed 1.6.24).

Saura, J.R., Bennett, D.R., 2019. A Three-Stage method for Data Text Mining: Using UGC in Business Intelligence Analysis. Symmetry 11, 519. https://doi.org/10.3390/sym11040519

Saura, J.R., Reyes-Menendez, A., Filipe, F., 2019. Comparing Data-Driven Methods for Extracting Knowledge from User Generated Content. Journal of Open Innovation: Technology, Market, and Complexity 5, 74. https://doi.org/10.3390/joitmc5040074

Shah, R., Aparajit, S., Chopdekar, R., Patil, R., 2020. Machine Learning based Approach for Detection of Cyberbullying Tweets. International Journal of Computer Applications 175, 51–56. https://doi.org/10.5120/ijca2020920946

Shane, S., 2008. The Handbook of Technology and Innovation Management. John Wiley & Sons.

sklearn.ensemble.VotingClassifier [WWW Document], n.d. . scikit-learn. URL https://scikit-learn/stable/modules/generated/sklearn.ensemble.VotingClassifier.html (accessed 1.7.24).

Srinivas, T.A., Thanmai, B., Donald, A., Thippanna, G., Sai, I., Srihith Indla, V., 2023. Battle of the Algorithms: An Exposé on Classification Techniques in Machine Learning. https://doi.org/10.5281/zenodo.8138139

Tang, X., Zhu, Y., Xuegang, H., Li, P., 2019. An Integrated Classification Model for Massive Short Texts with Few Words. Presented at the RSVT '19: Proceedings of the 2019 International Conference on Robotics Systems and Vehicle Technology, pp. 14–20. https://doi.org/10.1145/3366715.3366734

Timoshenko, A., Hauser, J.R., 2019. Identifying Customer Needs from User-Generated Content. Marketing Science 38, 1–20. https://doi.org/10.1287/mksc.2018.1123

Uzair, M., Jamil, N., 2020. Effects of Hidden Layers on the Efficiency of Neural networks, in: 2020 IEEE 23rd International Multitopic Conference (INMIC). Presented at the 2020 IEEE 23rd International Multitopic Conference (INMIC), pp. 1–6. https://doi.org/10.1109/INMIC50486.2020.9318195

Wang, T., Ji, P., 2010. Understanding customer needs through quantitative analysis of Kano's model. International Journal of Quality & Reliability Management 27, 173–184. https://doi.org/10.1108/02656711011014294

Witell, L., Kristensson, P., Gustafsson, A., Löfgren, M., 2011. Idea generation: customer co-creation versus traditional market research techniques. Journal of Service Management 22, 140–159. https://doi.org/10.1108/09564231111124190

Wu, X., Kumar, V., 2009. The Top Ten Algorithms in Data Mining. CRC Press.

Wythoff, B.J., 1993. Backpropagation neural networks: A tutorial. Chemometrics and Intelligent Laboratory Systems 18, 115–155. https://doi.org/10.1016/0169-7439(93)80052-J

Young, J.C., Rusli, A., 2019. Review and Visualization of Facebook's FastText Pretrained Word Vector Model, in: 2019 International Conference on Engineering, Science, and Industrial Applications (ICESI).

Presented at the 2019 International Conference on Engineering, Science, and Industrial Applications (ICESI), pp. 1–6. https://doi.org/10.1109/ICESI.2019.8863015

Zhuang, W., Zeng, Q., Zhang, Y., Liu, C., Fan, W., 2023. What makes user-generated content more helpful on social media platforms? Insights from creator interactivity perspective. Information Processing & Management 60, 103201. https://doi.org/10.1016/j.ipm.2022.103201