Text

Description automatically generated with low confidence

UNIVERSITY OF HERTFORDSHIRE

School of Computer Science

MSc Advanced Computer Science

7COM1039-0501-2020: Advanced Computer Science Masters Project

May 2020

Comparison between Naïve Bayes and Support Vector Machine for Sentiment Analysis on Movie Reviews.

Name: Nur Iman Binti Azmi

Student ID: 18058629

Supervisor: Mrs Renuga Jayakumar

# Abstract

The exponential growth of data, such as feedbacks and reviews, on the Internet has expanded not just valuable information but it also creates problem to some organisation in gathering those data for their business growth. The reason is because an individual could not afford to read and predict the data manually by himself and having to read many data can hinder the progression of their business, hence, this has encouraged organisations to use sentiment analysis in gathering and predicting the sentiment of the data faster. In this project, the author has proposed a methodology of comparison between two machine learning models, Naïve Bayes, and Support Vector Machines (SVM), for sentiment analysis on movie reviews. Sentiment analysis is used to classify text data, which is the movie reviews, into positive or negative. Prior to classification phase, the gathered data from a public dataset will undergoes a pre-processing phase where the raw data will be cleaned properly by using appropriate techniques. In addition, a feature extraction called Term Frequency – Inverse Document Frequency (TF-IDF) will also be used in the experiment to enhance the performance of the models. The results for both models will then be compared with each other and with other existing works for evaluation purposes.

# Acknowledgement

It has been a tough time for all of us during this pandemic, but I am eternally grateful to the Almighty God for gifting me a healthy body that I managed to complete the whole project without any hindrance, in terms of physically and mentally. I would also like to extend my gratitude to my family especially my parents for supporting me through it all in pursuing MSc in Advanced Computer Science in the University of Hertfordshire. Without them I would not be where I am right now. I would also like to take the opportunity to thank my supervisor, Mrs Renuga Jayakumar, for assisting me with the project throughout the semester. Not forgetting my friends at my hometown, I am thankful for them for trying to cheer me up and help me getting through the hardship as I complete the project.

# MSc Final Project Declaration

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Advanced Computer Science Masters Project at the University of Hertfordshire (UH).

It is my own work except where indicated in the report.

I did not use human participants in my MSc Project.

I hereby give the permission for the report to be made available on the university website provided the source is acknowledged.

Table of Contents

[Abstract 2](#_Toc89427345)

[Acknowledgement 3](#_Toc89427346)

[MSc Final Project Declaration 4](#_Toc89427347)

[Chapter 1: Introduction 6](#_Toc89427348)

[1.1: Problem statement 7](#_Toc89427349)

[1.2: Research question, aim, and objectives 7](#_Toc89427350)

[Chapter 2: Literature Review 9](#_Toc89427351)

[2.1: Machine Learning 9](#_Toc89427352)

[2.1: Sentiment Analysis 10](#_Toc89427353)

[2.2: Using machine learning models for Sentiment Analysis 12](#_Toc89427354)

[2.3: Steps in Sentiment Analysis 14](#_Toc89427355)

[2.4: Consideration of Ethical, Legal, Professional, and Social Issues. 17](#_Toc89427356)

[Chapter 3: Methodology 18](#_Toc89427357)

[Phase 1: Data collection 18](#_Toc89427358)

[Phase 2: Text pre-processing 18](#_Toc89427359)

[Phase 3: Classification models 23](#_Toc89427360)

[Phase 4: Performance analysis 26](#_Toc89427361)

[Chapter 4: Experiments 28](#_Toc89427362)

[4.1: Tools and techniques 28](#_Toc89427363)

[4.2: Data preparation 28](#_Toc89427364)

[4.3: Data pre-processing 29](#_Toc89427365)

[4.4: Classification algorithms 31](#_Toc89427366)

[4.5: Performance analysis 31](#_Toc89427367)

[Chapter 5: Results and Discussion 33](#_Toc89427368)

[Chapter 6: Evaluation 35](#_Toc89427369)

[Chapter 7: Future work 38](#_Toc89427370)

[Chapter 8: Conclusion 39](#_Toc89427371)

[Chapter 9: References 40](#_Toc89427372)

[Chapter 10: Bibliographies 43](#_Toc89427373)

[Chapter 11: Appendices 45](#_Toc89427374)

# Chapter 1: Introduction

Information technology has emerged since the 1980s and ever since then the technology has become more advanced to accommodate the people’s needs (Pang and Lee, 2008). Due to the advancement of technology, more data are being uploaded and consumed by users on the Internet and thus, exponentially increasing the data usage and storage needed to store the data (Sahu and Ahuja, 2016; Madasu and E, 2020).

Through the Internet also, organisations find the opportunity to sell their products and services to their customers since it can be easily accessible by many people (Agarwal, Aher and Sawant, 2018). This allows the organisation to keep track on what their customers’ views or opinions on their latest products or services based on their personal experiences. Not only that, there are other organisations that keep track on what their employees think about their current job and see if there are other recurring issues in the organisation Brown (2015). Hence, this is where sentiment analysis is being useful for.

According to Brown (2015), organisations like IBM, Intel and Twitter use sentiment-analysis software to gather their employee’s opinions on the current work environment in their organisation. The organisations have used machine learning models to analyse sentiment from text through their employee’s monthly feedback. In addition to that, there has been many research on sentiment analysis that revolves around the real-world situations and one of them is sentiment analysis on hotel reviews by Agarwal, Aher and Sawant (2018).

However, in this research project context, the domain involved would be the movie reviews domain, hence, the data discovered and needed mostly comes from users that express their opinions on movies on multiple platforms (Pang and Lee, 2008) but not limited to like Rotten Tomatoes, IMDb, Facebook, and Twitter (Nanda, Dua and Nanda, 2018). So, that is basically how the data keeps on increasing every second just from one domain. The reviews are either positive or negative, will be helpful to movie production companies as it will aid them in improving their production depending on the reviews given (Pang, Lee and Vaithyanathan, 2002). The reviews that can be seen publicly will also affect the other potential user’s decision including the film critics (Sahu and Ahuja, 2016).

In general, sentiment analysis is an approach where people analyse certain topics from texts, words, or documents which have structured, semi-structured or unstructured textual data to predict or assume the expression or emotions of the writer (Adetunmbi, Sarumi and Boyinbode, 2018).

In this project, sentiment analysis is used to analyse the expression and opinions from movie reviews and classify these reviews into two polarities which is positive and negative, which could be helpful to the movie production company in gathering reviews in a faster way and improve their business (Madasu and E, 2020). Most movie reviews on the Internet are written in informal, unstructured forms of grammar and sentences. This will become difficult for the machine learning models to capture the sentiment through informal words or sentences (Sahu and Ahuja, 2016). Hence, several steps needed to be done to process these raw texts properly.

The first step is vital which is the pre-processing step where all unstructured sentences from the movie reviews will be cleaned to become applicable for classification. The cleaning process will include important tasks such as tokenisation, stop word removal, lower case conversion, stemming, and removing numbers and special characters. Next step would be feature extraction, and lastly, evaluating machine learning models like Naïve Bayes and SVM, based on their accuracy when predicting the sentiment (Nagamma et al., 2015).

## 1.1: Problem statement

In general, once a product is released and being bought, most organisation would want to know their customers’ opinion on the said product but gathering a thousand of opinions from the customers to see whether the product is good or bad. This requires a lot of time and work to do and most of the time it is impossible for the organisation to manage manually. This is because there are too many incoming reviews that have been received everyday by the organisation and it is tedious for them to check the reviews manually one by one. Hence, this is where the sentiment analysis comes in handy for the organisation in finding the sentiment of their customers’ opinions faster by using suitable tools with appropriate techniques which obviously could save a lot of time.

However, in this project’s context, the problem acquired is how can the machine learning model determine the sentiment of customers’ reviews on a movie. The solution for this research project to run smoothly is to compare which machine learning model that is suitable for sentiment analysis in this movie reviews domain depending on its accuracy. Hence, to handle this problem, appropriate machine learning model that can classify the sentiment of movie reviews accurately might be a good approach for any organisation to use as this will also improve their production in the long run (Pang, Lee and Vaithyanathan, 2002).

## 1.2: Research question, aim, and objectives

In this research project, a research question that needs to be addressed is, can Naïve Bayes perform better than Support Vector Machine (SVM) in classifying sentiment on movie reviews? This leads to the author’s aim which is to determine the accuracy between two machine learning classifiers: Naïve Bayes and SVM, by comparing which algorithm will produce accurate result to the actual sentiment of the movie reviews. According to the result, it will help any organisation to use the appropriate classifier for future predictions.

To achieve the aim mentioned above, there are several objectives that needed to be done in this paper. Firstly, appropriate dataset needs to be acquired from an online site database and for this project, a public data that have been gathered by Maas *et al.* (2011) from Internet Movie Database (IMDb) website will be used where there will be abundant of reviews stored in the Excel file called IMDb Dataset. The software that will be used for this project is R Studio. Many free online courses can be found on the Internet and LinkedIn Learning will be chosen as a platform on how to program in R Studio. Then, more understanding will be needed for machine learning models involved. The dataset acquired will then be going through a pre-processing step before it will be split into a training set and testing set. Term Frequency – Inverse Frequency Document (TF-IDF) will be chosen as one of the feature extraction techniques for sentiment analysis. After that, a comparison will be made between Naïve Bayes and SVM by training them with the dataset that has been cleaned and evaluate the results achieved. The results will determine which model will be more accurate when classifying the sentiment on movie reviews. To evaluate the performance of each model, evaluation metrics will be used in this project, which is by calculating the Accuracy, Precision, Recall and F-measure. The results of the evaluation metrics will be displayed in a table for convenience.

More details will be explained and discussed thoroughly according to its relevant section. Literature review will be discussed heavily in Chapter 2. Followed by methodology in Chapter 3. Then the details of experimentation will take place in Chapter 4. In Chapter 5, discussion of the results will be made, followed up with evaluation of the system and the author’s development in completing the project in Chapter 6. In Chapter 7, the author will discuss about the future work that could be made for further improvement for the project. The author will conclude the whole research in Chapter 8. References will be in Chapter 9, followed by bibliographies in Chapter 10 and lastly Chapter 11 consists of the appendices.

# Chapter 2: Literature Review

## 2.1: Machine Learning

**Figure 1: Basics of machine learning (Lv and Tang, 2011).**

**Brief definition of machine learning**

Machine learning is a part of an Artificial Intelligence where the machine will be trained by human to learn from the past experiences and do what humans can do daily but faster. The machine will analyse the data and find matching patterns before it can be used to predict new data. Once the prediction has been made, the final results will be stored so that the machine can improve better as more incoming data are used to train (Lv and Tang, 2011; von Rueden *et al.*, 2021).

**Types of machine learning**

There are four ways in which the machine can learn data such as supervised, unsupervised, semi-supervised and reinforcement learning (Adetunmbi, Sarumi and Boyinbode, 2018). Supervised learning will be used only when the data is labelled, where it uses labelled data to train the model such as detecting fraud activity and predicting scores. The machine will know the features of the object and the labels associated with those features. Whereas unsupervised learning only uses the model to train unlabelled data. The model can only learn by searching the patterns. For instance, face recognition. Semi-supervised learning is in between supervised and unsupervised machine learning. It is when an abundant amount of data is involved but only a few data are labelled. Lastly, reinforcement learning is completely different because it is a learning system where the environment or surroundings will provide the training information and the best solution must be used to get the best results (Tyagi *et al.*, 2017). In another word, it is a reward-based learning where it works in the principle of feedback.

**Applications of machine learning**

There are various applications of machine learning that have been used widely to combat real-world problems such as health diagnostics, fraud detection, text classification, image classification (von Rueden *et al.*, 2021) and many more. Specifically in current days, machine learning has been used for classifying text which can also be known as sentiment analysis. Huge organisations find sentiment analysis helpful in predicting the sentiment of their customers’ feedback about the services that they provided (Agarwal, Aher and Sawant, 2018).

In this research project, the focus is on supervised learning only as the data that will be used is a labelled balanced data. Hence, appropriate machine learning algorithms will be needed to classify the sentiment specifically. More details regarding sentiment analysis and the algorithms will be discussed further in subsection 2.1 and 2.2 respectively.

## 2.1: Sentiment Analysis

**Brief definition of sentiment analysis**

Sentiment analysis is a research field that is branched out from natural language processing (NLP) where the process involves mining for opinions (Patil *et al.*, 2007). It is a process of identifying the expression, emotion and intention of an individual either on a product, a service, another individual, or an organisation, in text (Tripathy, 2015). For example, an employee’s honest review on their organisation’s work environment (Brown, 2015). The sentiment expressed by them can either be positive, neutral, or negative, which is also known as polarities. Usually, organisations will use sentiment analysis to comprehend the reviews given in a faster way so that they can improve better (Kalaivani and Shunmuganathan, 2013). This text data is not limited to reviews only, but it can also be emails, transcripts (Kalaivani and Shunmuganathan, 2013) and many more. Sentiment is a subjective matter as it is based on emotion of an individual with different preference, therefore, it is not facts. On additional note, traditional text mining is a technique of analysing facts meanwhile opinion mining revolves around the attitudes (Kalaivani and Shunmuganathan, 2013). Sentiment analysis is done by classifying their emotions whether it is positive or negative using specific machine learning models (Patil *et al.*, 2007).

**Why do we need to use sentiment analysis?**

The reason why sentiment analysis is highly used in the industry is because of the increase in the Internet usage exponentially day by day, especially during the pandemic where most people will stay at home or work from home (Patil *et al.*, 2007; Tyagi *et al.*, 2017). Nowadays it is easier for users to navigate through the Internet and look up things easily with just a few clicks on their device. The Internet has vast products and services that users can buy from (Patil *et al.*, 2007), hence, this has led to many people accessing social networking sites and blog websites which encourages others to leave their impression on certain things (Tyagi *et al.*, 2017; Adetunmbi, Sarumi and Boyinbode, 2018). Sometimes they will leave their own reviews regarding the products and services based on their experiences (Patil *et al.*, 2007). Reviews given by many users on certain things but not limited to product, company, brand, individual, forums and movies have helped other people in making their own judgement (Kalaivani and Shunmuganathan, 2013; Tyagi *et al.*, 2017). Therefore, sentiment analysis is very much needed to skim through all the reviews in an efficient and effective manner. Gathering such large volume of opinions and thoughts from the public has become a necessary and important task for an organisation to greatly improve their customer’s purchasing decisions while affecting greatly on their business (Adetunmbi, Sarumi and Boyinbode, 2018). This will also help a lot of organisations in improving their products and services after knowing the sentiment of the reviews from their customers (Patil *et al.*, 2007; Tripathy, 2015). In conclusion, sentiment analysis is important and necessary for real world applications (Madasu and E, 2020) such as business intelligence applications and recommender systems (Pang, Lee and Vaithyanathan, 2002).

**Problems regarding sentiment analysis**

There are some things that needed to be taken into consideration when analysing sentiment. That is, first, a word that may be denoted as positive in a certain context may have higher possibilities that it would be denoted as negative in another context (Mullen and Collier, 2004). Besides, tone and structure of words can be expressed differently by different people in various situations though it can be understood as the same meaning. For example, “this movie is bad” share the sentiment as “this movie is not that good” but within different degree and an individual would understand that. However, in sentiment analysis, the machine learning algorithm might produce different result if different inputs or data are being analysed. So, the algorithm will consider the sentence “this movie is bad” as negative meanwhile “this movie is not that good” as positive because the word ‘good’ in the sentence denotes positivity. In some cases, both sentences could either be positive or negative sentiment. This is due to the system’s nature in processing and analysing the sentence one at a time (Kalaivani and Shunmuganathan, 2013).

Turney (2001) has clarified that it is quite challenging for them to conduct a sentiment analysis on movie reviews because the reviews gathered have been written in informal words with ironic tone in it, therefore, it is difficult for the system to distinguish whether it is positive or not unless he has included some features that might aid the experiment (Kalaivani and Shunmuganathan, 2013).

**Types of sentiment analysis**

For many years, sentiment analysis has piqued so many interests of other researchers and there are different methods of classification have been used to classify sentiments of online reviews and one of those methods is machine learning-based sentiment classification method. (Adetunmbi, Sarumi and Boyinbode, 2018).

Typically, sentiment analysis is known to have two types which is supervised and unsupervised learning. Supervised learning is basically structured data that has been labelled and machine learning will be used to identify the sentiment (Adetunmbi, Sarumi and Boyinbode, 2018). Whereas unsupervised learning is unstructured data that has not been labelled and the way to identify the sentiment is to use lexicon technique which is rule-based (Kalaivani and Shunmuganathan, 2013). Unsupervised learning involves clustering; meanwhile supervised learning involves classification. The machine learning model used for classification is to analyse sentiment on existing data based on a target variable and predict the variable for new incoming data (Kalaivani and Shunmuganathan, 2013).

**Implementation of sentiment analysis**

There are three ways to implement sentiment analysis, either on document level, sentence level, word level (Bhoir and Kolte, 2015; Adetunmbi, Sarumi and Boyinbode, 2018), or aspect level (Patil *et al.*, 2007). On document level, as the name suggests, positive and negative parameters will be used to classify the whole document. Whereas on sentence level, positive and negative sentiments of each sentence will be used for classification while on word level, each word will be used instead (Adetunmbi, Sarumi and Boyinbode, 2018). As for the aspect level, it involves classifying the text based on the context of the sentences (Patil *et al.*, 2007). For this research project, only sentiment analysis on a word level will be used which focuses on individual word and its sentiment.

**Who uses sentiment analysis?**

Sentiment analysis is usually used by huge organisations to explore more strategies for business purposes that could help them advertise and promote their products and services more (Adetunmbi, Sarumi and Boyinbode, 2018). For example, Agarwal, Aher and Sawant (2018) has used sentiment analysis based on aspects to predict each hotel’s reviews. Then, evaluate whether the hotel is worth staying by using the rating system based on the whole context of the reviews. This will help the customers in finding their favourite hotel which piques their interest. Sentiment analysis is also useful in organisations that are involved in healthcare, finance, (Brown, 2015), banking (Bianchi, 2021), and many more. According to Bianchi (2021), sentiment analysis is used in healthcare department based in Jeddah to improve the quality of their services. This is done by analysing millions of annual reviews given by patients and in result, it has reduced the department’s expenditure on manual data processing and increased their efficiencies at work.

The main contribution of this research project is to conduct sentiment analysis on movie reviews by using appropriate machine learning models which could assist the movie production team in improving their business by taking the opportunity to gather feedbacks from their customers.

**How is it useful for my project?**

Based on the previous research and past experiments done by other researchers, it is a concrete reason to use sentiment analysis for this research project as it reduces a lot of time to determine the sentiment of abundant movie reviews by using machine learning models besides following the appropriate steps.

## 2.2: Using machine learning models for Sentiment Analysis

**Naïve Bayes and Support Vector Machine (SVM)**

Over the decades, many researchers have done experiments using different machine learning models for sentiment analysis with various chronological steps and they have evaluated multiple results. Previous research has greatly affected the development of this research project to achieve the aim.

Pang, Lee and Vaithyanathan (2002) have clarified that three machine learning models known as Naïve Bayes, Maximum Entropy (MaxEnt) and Support Vector Machines (SVM) did not perform well on text classification as compared on traditional categorisation which is topic-based. This is because some sentences needed more context before the sentiment can be concluded as compared to topic-based classification. They have experimented that Naïve Bayes performed optimally under certain condition with highly dependent features so this shows that Naïve Bayes has some limitation too. Although Naïve Bayes is known as one of the most used models to classify text (Madasu and E, 2020), but there are other models that might produce better results than Naïve Bayes under the same condition as stated by Pang and colleagues (2002).

As Pang, Lee and Vaithyanathan (2002) have mentioned in their paper, other models included MaxEnt and SVM where MaxEnt performed better than Naïve Bayes occasionally, not always. However, SVM had proven in their experiment to be the most effective model as compared to Naïve Bayes and MaxEnt. They have claimed that MaxEnt performed slowly during the training phase, especially when there is a huge number of folds used. So, they used lesser number of folds to combat the problem. They have also mentioned that MaxEnt model was expensive when it comes to training depending on the number of features including but not limited to n-grams, adjectives, and part-of-speech tags. In the experiment, they trained the data using these models without stemming and they did not remove a list of stop words either to preserve the context of the sentences. From their experiment results, it was proven that using SVM with various number of features gave the highest accuracy as compared to MaxEnt and Naïve Bayes although there were not many differences.

However, Pang and colleagues (2002) stood with their initial stance and still concluded that these three machine learning algorithms did not perform greatly on text classification as they expected because although human might understand the context of the whole sentence, but the algorithms might find it difficult to compute and analyse sentences that have vague and double meaning altogether in it. Hence, why they concluded that these algorithms needed more training with other features to possibly perform better.

The result from their experiment could be used as a reference and a baseline to this current research project as a comparison where another feature such as Term Frequency – Inverse Document Frequency (TF-IDF) would be included as well and see how the accuracy performs. MaxEnt would not be included in this research project since it requires an advanced software with certain amount of expenses just to train the data as what have been mentioned previously.

Even though Pang and colleagues (2002) have stated the three machine learning algorithms did not give satisfying results, however, in recent years, there are other researchers that have proven otherwise.

Kalaivani and Shunmuganathan (2013) have used sentiment classification of online reviews about movie for their experiment by including feature selection techniques. They have used three machine learning models which was Naïve Bayes, SVM, and k-Nearest Neighbour (kNN). Naïve Bayes model was used because it is widely known to calculate the probability on predicted classes. The outcome after using Naïve Bayes was mostly good along with benefits and the interpretation of the results was quite easy (Kalaivani and Shunmuganathan, 2013). SVM was used to classify linear data where the training data will be transformed into a higher dimension to find the best and optimal line that separates the hyperplane by using support vectors. Contradict to what Pang and colleagues have mentioned in their paper, Kalaivani and Shumuganathan (2013) stated that SVM is one of the models that is effective and efficient in classifying text.

Patil and colleagues (2007) have used SVM for sentiment analysis on comments and tweets on Twitter. Their aim is to categorise the comments and tweets gathered into positive and negative. Machine learning models like SVM will need to learn the input-output mapping. In text classification, the input would be the text documents such as comments and tweets, and the output are the predicted class, which is positive and negative sentiment. Another example, for email spam filter case study, the email would be the input and the target output would be 0 or 1, either spam or not spam (Patil *et al.*, 2007).

SVM has many benefits, as claimed by Patil and colleagues (2007). It is very useful and good at handling a huge number of features (words) since SVM has a protection for over fitting cases. This means that SVM is very useful with huge dataset because most of the time, text classification involves many features that are probably more than 1000 in amount (Patil *et al.*, 2007). The reason why SVM is used in this experiment is because it has been proven that this algorithm is very powerful and robust and is also good for text classification (Mullen and Collier, 2004; Patil *et al.*, 2007).

Meanwhile, Tyagi and colleagues (2017) have their own aim for their study which is to use SVM on product reviews by classifying the sentiments of the reviews into its respective classes, either positive or negative. These authors have used different datasets for the training set and testing set to predict the sentiments. The result showed that SVM produced higher accuracy compared to other algorithms and the performance could be enhanced and improved further by adding more sentence forms. Besides the accuracy performance, SVM has shown that it can behave well when conducting the experiment (Tyagi *et al.*, 2017).

## 2.3: Steps in Sentiment Analysis

Several steps were used for sentiment analysis throughout the years by different researchers and will be discussed in following paragraphs. These steps needed to be considered so that the experiment would work smoothly, and the result could be evaluated.

In Patil and colleagues’ (2007) study, they have mentioned the usual sequential ways of conducting the sentiment analysis. The authors have included part-of-speech indicator for sentiment analysis where it will detect the presence of adjectives. However, they have realised that by using only adjectives in their experiment affects the performance badly. These authors also have mentioned their usual steps when conducting the sentiment analysis. In a chronological order, they used the collection of comment database and then proceeds to pre-processing phase which includes tokenisation, stop words removal, and stemming. Patil and colleagues (2007) used tokenisation method where it is used to separate the whole sentence into its individual word called tokens. Next method would be removing stop words. Last method in this process would be stemming where the words will be reduced to its original or root form. Then, Patil and colleagues (2007) used Term Frequency – Inverse Document Frequency (TF-IDF) to calculate the weight of each word in the corpus. TF-IDF determines whether the word in a text document is occurring frequently and check whether it holds significance depending on its presence for further processing. They hypothesised that any reduced noise in the dataset will affect the classifier’s performance greatly besides speeding up the process of classification, thus, this will also speed up the sentiment analysis process in real-time. Another step is the feature selection, which is usually used to reduce the amount of analysis on the selected data besides refining other relevant features that can be considered in the classification process. Last step they did is the classification process where the authors used SVM to classify the data that have been cleaned.

Tyagi and other authors (2017) have proposed their work with several steps in this study. The author’s main purpose for this study is to explore the relationships between the online reviews for smart phone products and the performance’s revenue by training SVM on the data for sentiment analysis. The model will be applied on the reviews and the collection of the product will also be predicted based on the reviews taken. It will also analyse whether the reviews will affect the collection. The next collection will also be included referring to the predicted online reviews on the current day. The author collected a very detailed information of the smart phone which containing the following: brands, product date, rank of sale, and user’s review to conduct a prediction for high and low collection (Tyagi *et al.*, 2017). The author used Java programming language, WAMP server and SQL (Sequel Query Language) for the experiment. The sentiment analysis progress was completed according to the following steps: pre-processing, transformation, clustering, SVM classification, and evaluation (Tyagi *et al.*, 2017).

Tyagi and colleagues used part-of-speech tagging and removed stop words from their dataset. In part-of-speech, the text data which is the input reviews will be divided into tokens. Whereas stop words are a list of stop words that are frequently used in English text document, for example, “the”, “it”, “you”, and many more that are related. These stop words are also known as ‘functional words’ even though it does not signify any importance to the meaning. Therefore, this list of stop words will be removed so that it will be easier for classification and evaluation purposes. Then it is the transformation process where the TF-IDF will be involved. In this process, each sentence’s score in the document will be calculated (Tyagi *et al.*, 2017). The TF-IDF measurement is used for clustering of the document review. The measurement will point towards the cluster’s edge instead the centre of the cluster. It will find the centroid by choosing the cluster’s numbers (Tyagi *et al.*, 2017). After removing the outliers, SVM can finally be used for classification on the improved feature sets where positive and negative reviews will be classified (Tyagi *et al.*, 2017). To assess the performance of the sentiment analysis model, Tyagi and colleagues used performance metric such as Precision, Recall, F-measure, and Accuracy. A common measurement metric that can be used for classification performance is accuracy. It is generally a percentage of correctly predicted classes to the total of actual classes. Accuracy is useful with balanced dataset. These metrics can be defined in terms of the cells in the confusion matrix, like True Positives (TP) and False Negatives (FN) (Tyagi *et al.*, 2017). In their study, Java NetBeans IDE 7.2.1 was used for SVM classification. The product reviews were transferred to part-of-speech tagger (POS tagger) where each word will be assigned but not limited to noun, verb, and adjective. The author concluded that the experiment was successful as SVM achieved higher accuracy than other algorithms, which is 88.98%. It is obvious that SVM is better than others, but it can be further improved by adding more sentences or texts (Tyagi *et al.*, 2017).

Bhoir and Kolte (2015) and Sahu and Ahuja (2016) have experimented the sentiment analysis approach using movie reviews and their aim is to discover how likeable and unlikeable a movie is to some users at aspect level. At this level, Bhoir and Kolte (2015) claimed that it will be more accurate to discover what people really like and dislike based on the individual word as compared to other level such as document and sentence level since latter levels mostly focused on the positivity and negativity of the reviews as a whole document or whole sentences. Sahu and Ahuja (2016) claimed that it is difficult for a machine learning model to understand expressions made by humans at a phrase level. Besides that, it is also difficult for the models to detect sarcasm in sentences since the classifiers do not have the capability to determine the sentence basics such as subjective, objective, adverb, or adjective in a sentence captured from the movie reviews. Bhoir and Kolte (2015) have proposed SentiWordNet (SWN), a lexical approach, and Naïve Bayes, a machine learning-based approach in their paper. SWN is an approach which finds out the orientation of extracted opinion then classify them into a list of positive and negative words. This is a rule-based approach that does not require any machine learning algorithm to train data for sentiment analysis.

In the pre-processing step, Bhoir and Kolte (2015) replaced all slang and informal words into proper words such as changing “congrats” to “congratulations”. They also used part-of-speech tag on the pre-processed data because some words were hard to be analysed whether it is a verb or a noun. The authors used a dataset that contains 5000 subjective sentences and 5000 objective sentences, and they used SentiWordNet to assign each word with positive and negative scores. They used Naïve Bayes and SentiWordNet to conduct the subjectivity analysis and evaluate their performance based on four standard parameters: Precision, Recall, F1-measure, and accuracy. Their result showed that Naïve Bayes performed better than SentiWordNet and it has been claimed that subjectivity analysis is important in sentiment analysis because it improves the system’s efficiency and accuracy.

Untawale and Choudhari (2019) had mentioned that they utilised feature extraction in their experiment which included syntactic feature where word tags, patterns, phrases, and punctuations were used; semantic feature that worked well between words, signs, and symbols; and linguistic semantics where it is a feature to determine the human’s expression through language. The authors experimented sentiment analysis by classifying the data using linear classifiers such as Naïve Bayes and Random Forest. The data was acquired from reviews on websites such as Times of India and Rotten Tomatoes and will go through the similar pre-processing process before they were used for classification. Initial result produced by Naïve Bayes was classified into five categories, that is ‘strongly positive’, ‘strongly negative’, ‘weakly positive’, ‘weakly negative’ and ‘neutral review’. Then, they used the Random Forest classifier to analyse polarity count such as total count of positive and negative comments. They have implemented this process using Java with NetBeans tools and MySQL. The result produced was Random Forest took lesser time than Naïve Bayes to analyse the data because Naïve Bayes required huge memory and longer execution time.

Based on the research discussed previously, the author has decided to use Naïve Bayes and SVM for this research project because of several reasons. The author observed that Naïve Bayes and SVM are both popular to be used in text classification problems. Compared to other models, Naïve Bayes and SVM seem to be more robust and lenient when it comes to classifying text although the results and performance might vary for both models depending on the additional features included in the experiment.

Based on the previous research by different researchers, author decided not to include MaxEnt and used Naïve Bayes and SVM instead for her research project since it requires some additional expenses when training data using MaxEnt. From the experiment done by Bhoir and Kolte (2015), it is a good reason to use machine learning-based approach in this research project to conduct the sentiment analysis on the movie review dataset since SWN performed poorly as compared to Naïve Bayes. The author also used R language instead of Java or Python for this research project because the former can be used in R Studio which provides functionalities that are exclusive for classification.

Naïve Bayes and SVM will be compared to determine which one will give the satisfying result in terms of performance accuracy when classifying the sentiment of the movie reviews as mentioned in the aim. Author believed that the previous researchers’ experiments could aid this research project in terms of aim, techniques, and methodology, but deeper understanding could be achieved once the experiment and evaluation take place.

## 2.4: Consideration of Ethical, Legal, Professional, and Social Issues.

Sentiment analysis can be helpful in making a progress to an organisation, such as detecting fraud activity and improving healthcare, but it can also bring harm to an individual or a certain type of groups, such as voting manipulation. Thus, in this project, there are some issues regarding ethical, legal, professional, and social that need to be considered as well in the context of this project.

Sentiment analysis involves with gathering individual’s personal data and emotions. Continuously gathering those data without consent is already breaching an individual’s privacy. This activity could lead to an emotional distress to the individual and affect the society’s safety.

Sometimes, to achieve better predictions in sentiment analysis, an organisation could program their system to find more information about an individual. There should be a balance when collecting someone’s data without breaching their privacy such as preventing the violation of terms of services of an organisation by allowing them to scrap the data or redistribute the data.

Most people do not realise that their behaviour and expression are being observed once they posted or clicked something on the Internet. These people could be scrutinised by an organisation using sentiment analysis and it could be worse if they have a harmful intention towards the targeted group.

Transparency is vital when using sentiment analysis. For example, a news involving Facebook has caused an uproar among the public because they used sentiment analysis for the wrong purposes. Facebook altered the platform’s algorithms with frequent posts that has positivity and negativity on the user’s feeds just so that user’s emotions could be manipulated. Facebook did not request for consent from these users to be a part of the experiment which leads to people being stressed out because the lack of ethical and professional considerations from the organisation.

When dealing with data, anonymity should be considered as well, especially if the data is personal. Even though IMDb is a public platform, but some reviewers might not expect their movie reviews would be displayed to others on another platform. Attracting a large attention from strangers could be a potential risk to them if their identity is exposed. This could also invite cyber bullies to the platform if the reviews that they expressed were not accepted by this group of people. Hence, anonymity should be retained when conducting sentiment analysis to prevent cyber bullying.

Attending to these issues is significant to avoid any future problems. It would prevent individuals and their data from harm and future risks. These are the considerations that the author had spent learning when completing this project.

# Chapter 3: Methodology

Sentiment analysis on movie reviews will be done in four phases: data collection, pre-processing, classification, and performance analysis. Steps of sentiment analysis will be shown in Figure 2 below.

**Figure 2: Phases of sentiment analysis**

## Phase 1: Data collection

Movie review dataset that has been collected for this research project were based on the reference from Maas *et al.* (2011) research paper which has been labelled properly. They have collected 50,000 text documents which contained movie reviews from IMDb with half of the amount are positive reviews while another half are negative reviews. This means that this is a balanced data. They have released this dataset to the public for the benchmark purposes that can be used as a comparison in the future research as well.

The author only used 1,000 movie reviews for the initial experiment, with 500 movie reviews which are positive, and the rest are negative. 800 movie reviews were used for training the model while the rest will be used for testing to predict the actual sentiment. Later, additional movie reviews will be included to the project with 20000 movie reviews for the second experiment and 50000 movie reviews for the third experiment. This was done so that the author could observe the performance of the model in classifying sentiment.

## Phase 2: Text pre-processing

Since the dataset that have been collected contained messy data with insignificant information, there will be several steps need to be taken care of during this phase.

In definition, text pre-processing is the process of transforming the raw texts (raw natural language data) into the data that can be used for computational analysis. Text pre-processing is the most important step in sentiment analysis as this process involves cleaning and preparing the data for classification purposes in the later phase (Kalaivani and Shunmuganathan, 2013). There are several pre-processing techniques involved in this process. Some of these techniques are optional and some will have different settings that can be chosen from. In other words, there are no specific ways to pre-process text while there are different approaches with their own advantages and disadvantages.

Document term matrix (DTM) is by far the most common text representation used in text analysis. It is a matrix in which rows refer to documents, columns refer to terms, and the cells will give a weighted value for the presence of term in the document (Madasu and E, 2020). DTM is also called a bag-of-words representation as if it is like a bag full of words with no specific context (Pang, Lee and Vaithyanathan, 2002). Although it seems like there are some limitations as to what can be analysed with this representation, but it is informative. To create a good DTM, there are some basic but the most important techniques that need to be considered for this research project.

1. **Tokenisation**

To represent a text as bag-of-words, the text must be broken into individual words. This technique is called tokenisation (Tripathy, 2015). For reference, these tokens do not have to be necessarily words but for now the only focus is on the tokenisation into individual words.

Let us take a following sentence as an example.

Example of raw text: Let’s look at some basic pre-processing steps

After tokenisation: [Let’s] [look] [at] [some] [basic] [pre-processing] [steps]

There is no specific way as to how the text should be tokenised. In the example above, the word “let’s” is actually two words: “let us”. Depending on what tokeniser used and possibly what settings are used for the tokeniser, the single word could be single token, or two words could be taken as single token. So, this is an example of how tokenisation works in a pre-processing stage, the tools and choices of the researcher can affect the analysis. For example, “let us” as two tokens might be more accurate in terms of meaning of what is set but for some type of analysis, heuristic style of the researcher might also be relevant so that should also be considered whether they want to use “let’s” as a single token or “let us” as two tokens. So, after tokenisation, then only DTM can be created. However, this will not produce a good DTM. Consider the following example.

Document 1: Ducks are good!

Document 2: We need more ducks!

Document 3: This is a soliloquy about a duck.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ducks | are | good | ! | We | need | more | ducks | This | is | a | soliloquy | about | duck | . |
| Doc 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Doc 2 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Doc 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 1 | 1 |

**Table 1: DTM of raw tokens**

Table above shows a DTM of raw tokens without including other pre-processing techniques. However, this DTM is a bad representation and might affect the sentiment analysis poorly. This is because, in DTM, each unique token is given its own column but now if humans look at these tokens, they will see that some tokens have similar meaning. For example, there are three tokens refer to the animal duck. Now there is a trivial difference in the meaning of these tokens. The difference between the first column of “Ducks” and another “ducks” is just that the first one has capitalised “D”, but this simply happens because it is at the start of the sentence. The difference between “ducks” and “duck” makes more sense because the former is in a plural form while the latter is in singular form, but the meaning is still referring to the same animal duck. A human being can distinguish the similarity between Document 2 and Document 3 but in DTM, this similarity is completely ignored. Instead of making three different columns for “Ducks”, “ducks”, and “duck”, these words could be transformed into the same word, which is “duck”. So, only single column that recognises each of this document where the concept of a cat does occur will only be used.

1. **Lowercase and stemming**

Specifically, the techniques that are needed here are lowercasing and stemming. Lowercasing just means that the raw text will be replaced with lowercase version of it. The capitalised “D” in “Ducks” will be lowercase like “ducks.” Next is stemming, which is slightly complicated because it reduces those words to the root word (Parveen, Shrivastava and Tripathi, 2020). For example, a verb like “walk” can also be “walking,” “walks,” “walked” and so forth, but the general thing that is happening is the action is the same. Often, these different word forms would be reduced to the same single stem of the verb. Therefore, stemming would indeed reduce “ducks” to “duck” and would remove the suffixes such as “walking,” “walked,” and “walks” all to the same verb which is “walk.”

1. **Stop words removal**

Before applying lowercase and stemming to solve the issues and make a better DTM, another basic pre-processing techniques to the tool which is removing stop words. This simply means that there is a list of words which is called stop words where it can be removed from the DTM. Typically, these many text analysis packages have a list of stop words included in it which contain the most common words that do not have impactful meaning such as “the,” “it,” “is,” “me,” “myself,” and many more (Nagamma *et al.*, 2015). These words always occur in texts, but they are often irrelevant for text analysis technique. On a sidenote, stop words is a non-official category of words because it is just a list of words that people thought it would be irrelevant for analysis and whether a word is relevant for analysis is something that only the researcher can decide when they know what they are doing.

Once the pre-processing techniques have been applied, the result for DTM would occur just like the following. First, most columns in previous table have been removed because they were stop words and the three columns for “Ducks,” “ducks,” and “duck” have been removed and it has been replaced with just one column with a singular text “duck” in it. DTM also recognises that all three docs all mentioned the concepts of a duck.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document / Term | duck | good | need | soliloquy |
| Doc 1 | 1 | 1 | 0 | 0 |
| Doc 2 | 1 | 0 | 1 | 0 |
| Doc 3 | 1 | 0 | 0 | 1 |

**Table 2: DTM after pre-processing (lowercase, stemming and removing stop words)**

1. **TF-IDF**

Another technique to make a better DTM is to use TF-IDF. Literally, TF-IDF is a multiplication of two metrics. The first one called TF which refers to Term Frequency and the other is IDF which refers to Inverse Document Frequency. TF-IDF is one of the feature extractions that could be used to improve the performance of the analysis as it reduces the dimensionality of the features (Madasu and E, 2020). TF-IDF is used together to distinguish the importance of a word to the document in the dataset. A collection of documents is officially called as a corpus; therefore, corpus is a bunch of different documents usually related to the same subject area, in this case, the movie reviews. For example, there are 3 documents of movie reviews, and the author would like to know for any given term that might appear in any of these documents, the author would like to get a measure number that says how important is that term for one document relative to the entire corpus.

1. **Term Frequency**

TF refers to how many times a word would appear in a document. The frequency will increase as the occurrence of the words increase in the document (Rathi *et al.*, 2018; Untawale and Choudhari, 2019; Madasu and E, 2020). Supposedly, the word “duck” appears 10 times in Document 1 and the number of words overall in the document is 500 words. If 10 is divided by 500, the result would be 2%. 0.02 is the term frequency of the word “duck” in Document 1. The formula for TF is as follows:

TF:

**Equation 1: TF (Tripathy, 2015)**

TF: = 0.02

TF is not sufficient to use alone in solving the problem because certain words are bound to occur a lot in all documents such as words like “the”, “a”, “and” or any kind of casual words that does not have any specific meaning. That kind of words are going to have a noticeably big term frequencies no matter which document. Looking only at the terms that have the biggest term frequencies is not going to give the author this idea about which are the special words in each of these documents. That is when IDF comes in.

1. **Inverse Document Frequency**

IDF is a function of two things. The first one is a given term and the other one is the entire corpus. IDF is a measurement used to decrease the weight of a frequent word that occurs in a document and increases the weight of a less frequent word in the document (Rathi *et al.*, 2018; Untawale and Choudhari, 2019; Madasu and E, 2020). In a simpler term, it means how rare the word is across the whole document. The formula is log of large N, which is the number of documents in the corpus divided by the number of documents with T or that contain the term T (Madasu and E, 2020).

For example, if there are 3 documents and the term “and” is used, it is obvious that that term would appear in all documents, so the denominator for the number of documents with the term “and” would be 3 divided by 3 which would be log of 1 and the result would be 0. Since the result is 0, that means the word “and” is common to all documents which is not helpful in the analysis. However, if the word “good” appears in Document 1 but does not appear in either of the other documents, the IDF would be log of 3 which is the number of documents divided by 1 which is the number of documents that contain the word “good”. So, the number for IDF is bigger and that bigger number signals that the word “good” occurs in only one document therefore it has bigger weight because it would be more helpful to distinguish what is different about these documents.

The formula for IDF is as follows.

IDF: log

**Equation 2: IDF (Tripathy, 2015)**

IDF: log = 0.47

In conclusion, TF-IDF will be combined and calculated as below.

TF-IDF = TF result \* IDF result

TF-IDF = 0.02 \* 0.47 = 0.0094.

1. **Splitting dataset**

During the pre-processing phase, the dataset will be split into training set and testing set. The reason for that is to make it easier to evaluate the machine learning algorithms used, in this case, performance evaluation of Naïve Bayes and SVM. It will be better to split the data when a large dataset is involved. The appropriate number could be up to thousands and more than that. Small dataset is not suitable for training and testing set because the lack amount of data in the training set for the algorithm to learn in mapping the inputs to outputs. If the amount of data is not enough in the testing set, it would be difficult for the algorithm’s performance to be evaluated.

Training set will be used to train the machine learning algorithm on existing data then the testing set will determine the result by predicting on a new data. The ways to split the dataset is to assign a percentage to each set. For example, training set could have 80% of data and the rest would be left for testing set. There is no best percentage for the splitting dataset because it depends solely on the research’s objective by including other factor such as the computational cost.

## Phase 3: Classification models

After the pre-processing phase is done, then the data is finally able to be used as an input to the classification models. In Chapter 2, many researchers proposed different machine learning models for sentiment analysis but in this project, the author has decided to use only two, which is Naïve Bayes and SVM, to train the cleaned data. The reason is because Naïve Bayes is fast since it uses binary classification in sentiment analysis, and in this case, the author only has two labelled data which is positive and negative. Whereas SVM is known to be a robust model among most researchers when classifying text based on the end results of its accuracy.

**Naïve Bayes**

Naïve Bayes is one of the classification models under supervised learning method and it is suitable for training a large dataset. It calculates the probability of attributes that are associated with a specific class (Flores *et al.*, 2018). It is a machine learning model that is based on the properties of Bayes Theorem to assume the relationship between features with strong independence. Naïve Bayes has an advantage where only a small training data is required for it to predict the sentiment because the features calculated are independent. When the feature is independent of each other then it will be easier for the execution because it will reduce the time complexity (Parveen and Pandey, 2016). Also, the pre-processing phase that has been done previously will be helpful in training the data using the Naïve Bayes model.

P(A|B) =

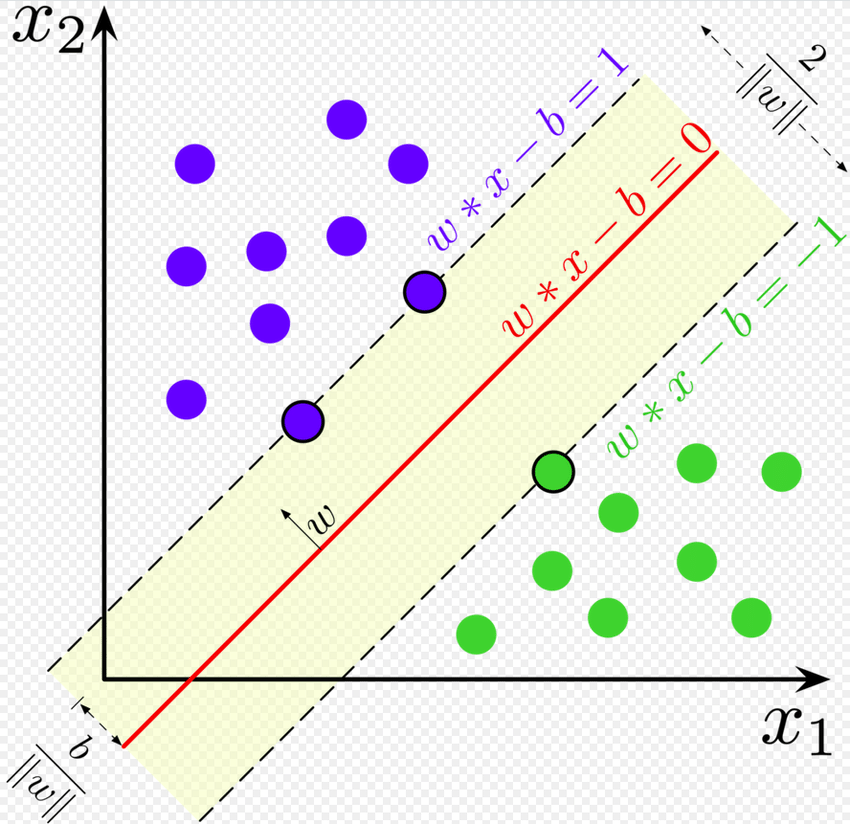
**Equation 3: Equation based on Bayes Theorem (Rana and Singh, 2016)**

Since Naïve Bayes is based on the Bayes Theorem, hence the equation above is adopted. It simply means that if **B** has happened then there is a probability that **A** will be happening as well, given that **B** would be the evidence and **A** would be the hypothesis.

**P(A|B)** is when event **B** has happened thenthere is a probability that event **A** might happening. Meanwhile, **P(B|A)** is vice versa. If event **A** has happened, then there is a probability that event **B** might happening. **P(A)** refers to the probability that event **A** might happening, and **P(B)** refers to the probability that event **B** might happening (Rana and Singh, 2016).

Relating to the project, based on the probabilities of a movie review to be positive or negative, Naïve Bayes model compute either of them as a result. If the movie reviews are considered as event **A** and the probability of them being positive or negative is event **B**, then the model applies the probabilities of event **A** and event **B** into the Bayes Theorem and predicts if the movie review is positive or negative. If the probability of **B** of **A** is more than 0.5, the model will assume it is positive, if it is less than **0.5**, then the model will assume it is negative.

**Support Vector Machine**



**Figure 3: Representation of SVM graph (Noyum et al., 2021)**

SVM is good at solving classification problems and in order to do that, it will create a hyperplane in between two classes that are plotted in n-dimensional space for classification provided that the distance between those two classes is maximum (Singh and Tripathi, 2021).

The hyperplane itself is also known as the decision boundary so that everything on one side gets classified as positive and another side gets classified as negative. The decision boundary must be far away from both classes so that it will maximise the margin. The margin means that the hyperplane to the positive class is the same distance as the hyperplane to the negative class. Some of the observations in the training data are sometimes called as support vectors because they are fully defining the margin (Noyum *et al.*, 2021).

According to Figure 3, assume that the purple symbols are the positive movie reviews, and the green symbols are the negative movie reviews. Consider those the two dotted lines and single red line as the training data but in a higher dimensional it will be known as hyperplanes. Upper dotted line does not seem great to be used as a hyperplane because it is too close to the positive class which means that if there are new data introduced to the model, there is a possibility that the new data will be classified on the wrong side of the class. Lower dotted line is also not an excellent choice to be used as a hyperplane because of the same reasons. Single red line is the most optimal fit because it is in the middle of these two classes so that when new data comes in, it will be in the correct side of the class.

As for the equation on how SVM finds the hyperplane is as shown in Equation 4 and the expanded version is shown in Equation 5.

**Equation 4: Hyperplane’s equation (Noyum *et al.*, 2021)**

**Equation 5: Expanded version of the hyperplane’s equation (Noyum *et al.*, 2021)**

This is simply the equation of the hyperplane and since only two dimensions (classes) are involved in here, then it is just going to be an equation of a line where **b** is the intercept, and is the coefficient or the weight vector.

This implies that each vector **x** is **p** dimensional so there are **p** predictors about the movie reviews. There are two other lines in Figure 3 which have similar equations, so the upper dotted line is given by and the lower dotted line is given by . It turns out that **w** and **b** are the only two coefficients that are important in this equation. If good values have been found for **w** and **b** then the calculation is completed, meaning the margin is maximised, as **w** is perpendicular to the hyperplane.

For example, there is a vector **x** on the decision boundary, then it must follow the equation of the decision boundary which is the hyperplane’s equation mentioned above. How many units needed to get to the w direction to get to this other blue line such that the equation is **?** Firstly, a unit vector known as **k**, must be created in the direction of **w** which is simply shown as the equation **.**

This is important since the size of the margin has been achieved because the margin is simply twice that amount which is over the magnitude of the vector **w,** which is the vector of weights. To maximise the margin, the quantity needs to be maximised which is essentially the same thing as minimising the denominator, so the magnitude of **w** needs to be minimised.

However, there are a couple of constraints that needed to be taken into consideration. It is needed so that the **w** and **b** that have chosen everything on one side of this margin is classified as positive and the other side of the margin is classified as negative. For instance, the movie reviews are classified as positive, and the negative reviews are classified as negative.

In mathematical equation, those constraints are expressed in Equation 6 and Equation 7.

**Equation 6: First constraint equation (Noyum *et al.*, 2021)**

**Equation 7: Second constraint equation (Noyum *et al.*, 2021)**

If , which means that if the movie reviews are positive, then must be greater than equal to 1. And if  **,** which means that if the movie reviews are negative then must be less than or equal to -1.

Combining those two equations together in a compact form will produce an equation such Equation 8.

**Equation 8: Two equations of constraints combined together into a compact form (Noyum *et al.*, 2021)**

In summary, to maximise the margin, which is what SVM is trying to do, the magnitude of **w** needs to be minimised, given that the constraint in Equation 8 is being obeyed. The support vectors would be the symbols in Figure 3 are situated exactly on the two dotted lines, which means that being any of those support vectors is either equal to 1, if it is in the positive class, or -1, if it is in the negative class. When the quantity is multiplied by its class label, the result would always be 1. Anything that is greater than 1 are not support vectors since those are the ones that are outside of the margin and those are the ones that do not affect the outcome of the problem even if they were shifted around the positive and negative class space. Hence, the support vectors can be expressed in Equation 9.

**Equation 9: Support vectors’ equation (Noyum *et al.*, 2021)**

## Phase 4: Performance analysis

Once the sentiment has been predicted, the performance of the models can be distinguished by using the performance metrics. During this phase, four metrics will be used to evaluate the performance of the classifiers, such as Accuracy, Precision, Recall and F1-score. A confusion matrix will be generated where the number of positive and negative reviews that have been predicted correctly and wrongly.

Accuracy refers to the ratio of when the method predicts the results correctly, meanwhile Precision refers to the ratio of when the method predicts the positive results correctly (Tyagi *et al.*, 2017). Recall is a measurement of sensitivity; the times when the system distinguishes a topic from an unstructured text over the total amount of times that topic actually appeared in the text. The higher the sensitivity, the lesser possibility for false negatives to happen and vice versa. As for the F1-score, the calculation includes average weight of Precision and Recall where false positives and false negatives are considered (Tripathy, 2015). These equations gathered from previous authors mentioned in Chapter 2 are shown in Equation 10, 11, 12, and 13.

Accuracy =

**Equation 10: To calculate the Accuracy (Tripathy, 2015)**

Precision =

**Equation 11: To calculate the Precision (Tripathy, 2015)**

Recall =

**Equation 12: To calculate the Recall (Tripathy, 2015)**

F1-score = 2\*

**Equation 13: To calculate the F1-score (Tripathy, 2015)**

The performance metrics will be generated in a table for each model and the results will be compared amongst each other based on which sentiment score give satisfying result for the movie review sentiment. The results obtained from the experiment will be compared with other previous results from some researchers that have been mentioned in Chapter 2.

# Chapter 4: Experiments

## 4.1: Tools and techniques

Software that will be used in this project is R Studio, a programming language for statistical computing and graphics. RStudio Desktop application on laptop will be used with these specifications: 16GB RAM, Intel Core i7 (9th Gen).

## 4.2: Data preparation

Firstly, relevant packages must be loaded so that the appropriate functionalities can be utilised when conducting the sentiment analysis.

Graphical user interface, text, application

Description automatically generated

**Figure 4: Preview of code when loading the libraries**

Then, the dataset will be loaded onto the R Studio global environment as shown in Figure 5. A public dataset from (Maas *et al.*, 2011) paper was used where they have provided a set of 25,000 movie reviews for training, and 25,000 for testing. In total, this dataset contains 50,000 text documents of labelled movie reviews with positive and negative sentiments. A preview of the raw movie reviews will be shown in Figure 6.

Graphical user interface, text, application

Description automatically generated

**Figure 5: Preview of code when reading the Excel file from the local file**

Graphical user interface, text, application

Description automatically generated

**Figure 6: Preview of the movie reviews with its labelled sentiment**

Initially, only 1,000 movie reviews containing both positive and negative sentiments were used for experiment as to see how the performance goes at first. Then, the remaining movie reviews were included up to 20,000 and 50,000 for comparison purposes after classification stage.

After loading the data onto the R Studio global environment, then the author needs to factorise the sentiment classes (dependent variable), which is the positive and negative, into categorical variable because the machine learning algorithm can only read specific format instead of sentences when conducting the classification later. The preview of code of factorisation is shown in Figure 7.

Graphical user interface, text, application

Description automatically generated

**Figure 7: Preview of code for factorising the dependent variable**

## 4.3: Data pre-processing

This stage is crucial in sentiment analysis because this is where many unnecessary information and noisy raw data will be removed completely since it does not help in the classification stage. It also helps to reduce the time complexity of executing the sentiment analysis because there will be lesser memory used to store the text data. Essentially, this is a cleaning phase to enhance model’s performance later in the classification stage.

The raw data of 1000 that have been acquired were split into two sets: where the training set will contain 800 raw data and the rest of the 200 raw data will be in the testing set. 501 movie reviews are positive, and 499 movie reviews are negative. The experiment will continue by using additional data into the training set and testing set for evaluation purposes. Different number of reviews have been used in every experiment with different numbers of training set and testing set just to evaluate and compare the performance. The training data consisted of 800, 600, 400, 200, and 100 movie reviews whereas the test data consisted of 200, 400, 600, 800, and 900 movie reviews. Then the experiment continued with two more datasets which consisted of 20000 and 50000 movie reviews where these reviews will be split up with the same manner: training and testing set. Figure 8 is the preview of the code used.

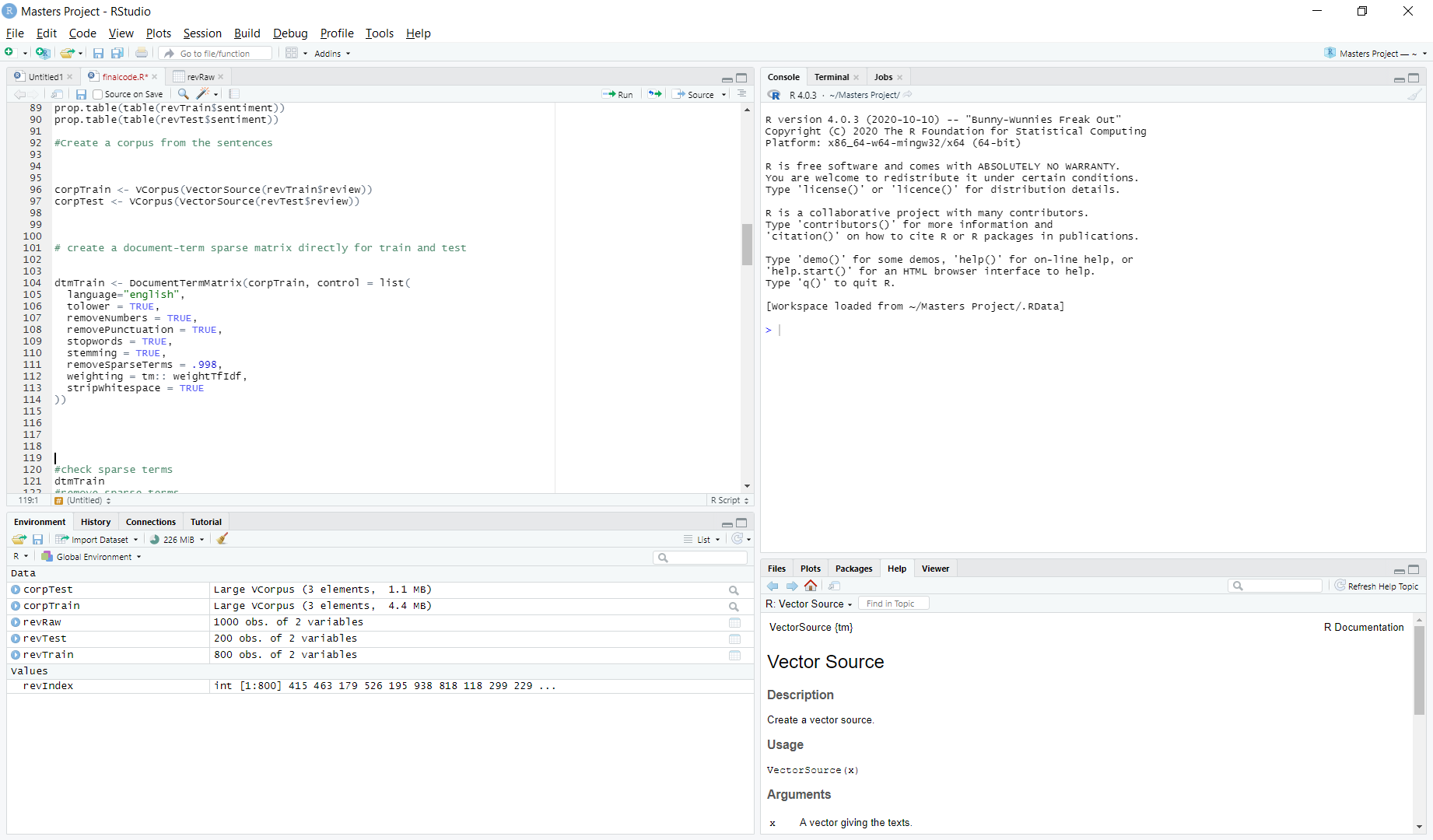
Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

**Figure 8: Preview of code in splitting the dataset into training and testing set**

In a separate set, the raw data have been cleaned by using some functionalities of R programming language in R Studio. The cleaning steps in this experiment included but not limited to tokenisation, stop words removal, numbers removal, special characters removal, white space removal, lowercase conversion, and stemming. A preview code of the cleaning process is shown in Figure 9.



**Figure 9: Preview of code for pre-processing the movie reviews**

|  |  |
| --- | --- |
| **Raw movie review** | **Pre-processed movie review** |
| One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. They are right, as this is exactly what happened with me. <br /><br />The first thing that struck me about Oz was its brutality and unflinching scenes of violence, which set in right from the word GO. Trust me, this is not a show for the faint hearted or timid. | one of the other reviewers has mentioned that after watching just oz episode youll be hooked they are right as this is exactly what happened with me the first thing that struck me about oz was its brutality and unflinching scenes of violence which set in right from the word go trust me this is not a show for the faint hearted or timid |

**Table 3: A side-by-side view of raw data taken from Maas et al., (2011) and the pre-processed data**

Table 3 shows a side-by-side view of a raw data that have been acquired from the original IMDb dataset collected by Maas and colleagues (2011) along with the data that have been pre-processed.

During this phase, feature extraction technique like TF-IDF will be executed as well, and the author will only extract unigram from the movie reviews where the machine learning algorithm will need to predict the sentiment based on each word in a document.

Figure 10 is the output of DTM where the sentences from the dataset have been pre-processed will be separated into individual words (terms) and the matrix will determine how often the terms appear in a sentence in one document.

Graphical user interface, text, application

Description automatically generated

**Figure 10: Output of DTM in R Studio**

## 4.4: Classification algorithms

At this stage, the author used the training set to train Naïve Bayes and SVM. Once the algorithms have been trained, then the models can be used to predict the new data which is the test set.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

**Figure 11: Preview of code for training the data and predicting the sentiment using Naive Bayes**

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

**Figure 12: Preview of code for training the data and predicting the sentiment using SVM**

The results of the predicted sentiment can be displayed using the table function against the actual sentiment in next section.

## 4.5: Performance analysis

To check the performance of Naïve Bayes, Confusion Matrix function is used, and it will automatically display the whole result of performance metrics such as Accuracy, Precision, Recall, and F-measure.

Graphical user interface, text, application

Description automatically generated

**Figure 13: Preview of code to show the results of performance metrics for Naive Bayes**

As for SVM, analytics function will be used instead to summarise the performance based on the same performance metrics.

Graphical user interface, text

Description automatically generated

**Figure 14: Preview of code to show the results of performance metrics for SVM**

Table 5, 6 and 7 show the results of Naïve Bayes and SVM based on their performance metric such as Accuracy. The results were achieved after conducting sentiment analysis on 1000, 20000 and 50000 movie reviews. The results for performance metric such as Precision, Recall, and F-measure will be displayed in Chapter 10: Appendices section on page 46 until 48.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment** | **No. of reviews in training dataset** | **No. of reviews in testing dataset** | **Accuracy 1, sparse term <100** | |
| **SVM (%)** | **Naïve Bayes (%)** |
| 1. | 800 | 200 | 77.0 | 76.0 |
| 2. | 600 | 400 | 77.0 | 75.0 |
| 3. | 400 | 600 | 77.0 | 71.0 |
| 4. | 200 | 800 | 77.0 | 70.0 |
| 5. | 100 | 900 | 77.0 | 66.0 |

**Table 5: Results of accuracy with different number of reviews in training and testing set for the total 1,000 reviews**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment** | **No. of reviews in training dataset** | **No. of reviews in testing dataset** | **Accuracy 2, sparse term <100** | |
| **SVM (%) sparse term <90** | **Naïve Bayes (%)** |
| 1. | 16,000 | 4,000 | 85.1 | 84.0 |
| 2. | 14,000 | 6,000 | 85.1 | 84.8 |
| 3. | 12,000 | 8,000 | 85.1 | 84.2 |
| 4. | 10,000 | 10,000 | 85.1 | 84.6 |
| 5. | 8,000 | 12,000 | 84.6 | 84.5 |

**Table 6: Results of accuracy with different number of reviews in training and testing set for the total 20,000 reviews**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment** | **No. of reviews in training dataset** | **No. of reviews in testing dataset** | **Accuracy 3, sparse term <100** | |
| **SVM (%)** | **Naïve Bayes (%)** |
| 1. | 40,000 | 10,000 | 85.6 | 84.4 |
| 2. | 35,000 | 15,000 | 85.6 | 84.6 |
| 3. | 30,000 | 20,000 | 85.6 | 84.5 |
| 4. | 25,000 | 25,000 | 85.6 | 84.5 |
| 5. | 20,000 | 30,000 | 85.6 | 84.4 |

**Table 7: Results of accuracy with different number of reviews in training and testing set for the total 50,000 reviews**

# Chapter 5: Results and Discussion

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Total number of movie reviews** | **Training set** | **Testing set** | **Classification accuracy (%)** | |
| **SVM** | **Naïve Bayes** |
| 1,000 | 800 | 200 | 77.0 | 76.0 |
| 20,000 | 16,000 | 4,000 | 85.1 | 84.0 |
| 50,000 | 40,000 | 10,000 | 85.6 | 84.4 |

**Table 8: Summary of classification accuracy for Naive Bayes and SVM**

Results in Table 8 shows the summary of classification accuracy for both models. In terms of classification accuracy, SVM has higher percentage as compared to Naïve Bayes. This means that SVM classification accuracy is slightly better than Naïve Bayes across three different dataset of movie reviews. At 1,000 movie reviews, the classification accuracy for SVM and Naïve Bayes is 77.0% and 76.0%, respectively. The accuracy increases as the total number of movie reviews increases. This shows that the increment of the percentage of the classification accuracy for both algorithms is affected by the increment of total number of movie reviews used. The results at 50,000 movie reviews seems to show that SVM achieves higher accuracy than Naïve Bayes with 85.6% and 84.4%, respectively.

From this experiment and as shown in the results, although SVM achieved higher accuracy than Naïve Bayes, but the time to achieve that accuracy is longer. As the total number of movie reviews increase, the longer the time taken for SVM to execute the analysis. Same goes to Naïve Bayes although the time taken is much lesser than SVM. This happens because SVM takes longer time to train the data with large number of reviews. SVM takes longer time to train 40,000 movie reviews as compared to 800 movie reviews. In general, algorithm’s complexity such as SVM will be calculated by using the Big-O notation which can be divided into two types, time and space complexity (Abdiansah and Wardoyo, 2015). In SVM, there are kernel methods will be introduced and most of them are expressed as quadratic programming (QP) problems. The reason is because the time complexity for SVM is O(n3) (Wang *et al.*, 2014) and the space complexity is O(n2) (Li, Wang and Hu, 2009), where **n** is the size of training set, which is not feasible for training large datasets because the algorithm depends on **n**.

Despite that, the aim, which is comparing two machine learning models in conducting sentiment analysis on movie reviews, and the objectives mentioned in Chapter 1 have been achieved in this research project. The answer to the research question addressed earlier has also been achieved. In terms of accuracy, SVM did give better results as compared to Naïve Bayes, however, in terms of time taken to produce those results, Naïve Bayes did compute faster than SVM.

However, another thing to note is that there is a possibility that the output result might change even after the experiment phase has completed. The results that the author achieved in this experiment could be totally different during the demonstration. This is due to the nature of the machine learning models which might have the tendency to give biased results whenever the same training data have been trained many times. This could also lead to the results being drastically good or bad.

This could affect badly the two models’ performances, especially SVM, because SVM model will become less effective when there are overlapping points (data). When this occurred, it will be difficult for the model to produce a hyperplane in classifying both classes.

Nevertheless, there are some future improvements that could be considered which can be discussed further in Chapter 7: Future work.

# Chapter 6: Evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| **Metrics** | **Author’s research project** | **Paper #1** (Tripathy, 2015) | **Paper #2**  (Pang, Lee and Vaithyanathan, 2002) |
| **Feature extraction methods** | TF-IDF | CountVectoriser, TF-IDF | Term frequency |
| **Feature types** | Unigram | Unigram | Unigram |
| **Machine learning algorithms** | SVM and NB | SVM and NB | SVM and NB |
| **Dataset** | Movie reviews | Movie reviews | Movie reviews |
| **Number of classifications** | Two | Two | Two |
| **Classification accuracy** | SVM: 85.6%  NB: 84.4% | SVM: 94%  NB: 89.5% | SVM: 72.8%  NB: 78.7% |
| **Number of features (words)** | 50,000 | 12,500 | 16,165 |
| **Pre-processing techniques** | Stemming, stop words; special characters; white space; and numbers removal. | Stemming, stop words; special characters; and white space removal | Removal of special characters, numbers, and white space. |

**Table 9: Comparative analysis of author’s proposed work with existing works**

Results in the table above shows the comparison of results between the author’s proposed methods and other researchers’ methods. The author’s proposed research models have performed better than Pang and colleagues’ result. The proposed models performed better because during the pre-processing phase, author has included multiple techniques that could clean the data including stemming, removal of stop words; special characters; white space; and numbers, and tokenisation. Whereas in Pang’s paper, they have stated that they did not include any stemming and they did not remove any stop words, so this could be one of factors as to why their proposed models did not perform well as compared to the author’s.

On a contrary to that, author’s proposed models performed poorly as compared to Tripathy’s results. One of the contributing factors to this is probably because author’s data has too many sparse terms due to large dataset that are probably useless which needs to undergo thorough cleaning process. Besides that, author did not include lemmatisation in her pre-processed phase which could affect the machine learning algorithms’ performances as well.

The different feature extraction methods used between three papers might also be the factor that could affect the performance of the algorithms. Author and Tripathy used TF-IDF meanwhile Pang, Lee and Vaithyanathanonly used Term Frequency. From the comparison of the results, it shows that using TF-IDF could contribute to producing better results rather than just using Term Frequency as the feature extraction method.

Across three different datasets with 50000, 20000, and 1000 movie reviews, the results show that SVM has slightly higher accuracy as compared to Naïve Bayes. At 50000 movie reviews, SVM predicted 85.6% accuracy of the sentiment to the target values, whereas Naïve Bayes achieved slightly lower which is 84.4%. Meanwhile at 1000 movie reviews, SVM predicted only 77.0% accuracy of the sentiment and Naïve Bayes achieved 76.0%. The lower the movie reviews used for sentiment analysis, the lower the percentage of accuracy in predicting the sentiment. This means that the most effective machine learning algorithm in this project is SVM even though there are some other things that need to be considered such as the computation of time and space complexity of the said algorithm. The difference of the percentage between SVM and Naïve Bayes is not that big either, therefore, Naïve Bayes can also be considered effective to be used for sentiment analysis purposes. However, in a real-life situation, most of the time the data gathered might not be linear and unbalanced, so at this point, Naïve Bayes would not be a good choice to solve the problem and SVM could be used instead.

**Management of the project**

Prior starting the project, the author did not have any knowledge on machine learning and sentiment analysis but as time goes by, she acquired a lot of useful information when she did thorough research regarding this project. At first, the author planned to use three models in this project including Maximum Entropy but decided to use only two because Maximum Entropy requires a different software to train the data which is locked behind a paywall. It is a learning process every day and through this project, the author has learned on how to visualise and classify data using R programming language in R Studio, although the project still has more room for improvement. While developing the project, the author has encountered a major problem which is related to coding. She could not use the same coding for both, Naïve Bayes and SVM. Initially, she tried to use the same coding for both, but the results were not fruitful. Hence, she decided to use two different libraries for two different models. Another problem that occurred during the experiment was the time taken to conduct sentiment analysis by using two machine learning models. Some researchers have claimed that SVM are fast at training large dataset, however, in this experiment SVM proved to be working slower than Naïve Bayes despite achieving higher accuracy. This have proved that there is more improvement could be made to the models so time complexity could be reduced. Final problem encountered was that SVM produced different results even after the experiment phase has completed. However, the author took the initial results for SVM because training the model once would give appropriate results since SVM tend to be less effective if there are overlapping data points if trained many times.

Initially, the time allocated for the author to complete the project was around 8 months, but the author managed to complete the project approximately within 7 months with the aim and objectives have been achieved. The author managed to observe the process and the results of two models before making a comparison. Deeper understanding regarding those two models has amplifies the need for the author to explore more within machine learning topic and related areas to improve her work further. Besides the technical skills, her soft skills have also improved as she virtually meets up with her supervisor every fortnight, increasing her productivity and improving her time management.

With the increment of data in real-life each day, automation systems or machineries are bound to be used and could be a necessary thing to have at some point in an organisation to ease the process. Hence, why sentiment analysis was introduced in the first place. It is to aid the organisation in getting the data, in this case, the reviews from their customers for their business growth. Therefore, the proposed models could be helpful to be used for real life industry in estimating or predicting sentiment of the movie reviews besides accelerating the process of human beings in gathering data to be analysed for future use.

# Chapter 7: Future work

There are many things that could be improved for this project which could be done in the future. Comparing the results achieved with other researchers’ results have encouraged the author to improve further in her research work by including more techniques in the pre-processing phase such as lemmatisation and bi-gram function. Author would like to explore more in machine learning topic by using different machine learning models for sentiment analysis. These models include reduced SVM (RSVM), k-means clustering, and LibLinear to see how their performance is when predicting the sentiment. Previously, the author encountered major problems where SVM took longer time in training the data as compared to Naïve Bayes although it provided better result than the latter. Therefore, the author would like to check the performance of SVM by using a high-performance computer as well as reducing the time and space complexity by adding more feature selection techniques besides TF-IDF. The author would also like to include lemmatisation in the pre-processing phase where the stemmed word will be transformed into a root word that is readable and meaningful. For example, ‘beauti’ that has been stemmed from ‘beautiful’ could transform to ‘beauty’ by using lemmatisation function in the future. Lastly, the author would like to explore more on other programming language that could work for sentiment analysis such as Python and WEKA and evaluate how the performance could be as compared to when using R programming language.

# Chapter 8: Conclusion

Due to the abundance of data on the daily has made sentiment analysis to become important and most sought for in classifying text. Sentiment analysis research has also been increasing to solve real-life problems. This project showed an experiment of sentiment analysis on movie reviews from IMDb by using supervised machine learning models, Naïve Bayes and SVM. The movie reviews used were from Maas and colleagues’ (2011) collection of datasets that they have set for public perusal. The original dataset contains 50000 movie reviews and the author decided to use that original dataset and split it into three datasets which contain 50000, 20000 and 1000 movie reviews each. The reason was to evaluate the performance of both algorithms with different number of movie reviews. Then, each dataset will be split into training and testing since the size of each dataset is large. Followed by the pre-processing phase, the author has used several techniques to clean the data such as tokenisation, stemming, lowercase conversion, stop words and special characters removal. In addition to that, TF-IDF was used to calculate the frequency of the terms and remove terms that occur frequently across the documents that do not contribute anything useful to the analysis.

By the end of the project, the author has achieved multiple results for both algorithms and the results show that SVM achieved higher accuracy than Naïve Bayes in predicting the sentiment of movie reviews for each dataset. The percentage is at 85.6% for SVM and 84.4% for Naïve Bayes, which is not much of a big difference. Then, the author’s proposed work was compared to other two existing works, which shows that the author’s algorithms performed poorly than the first existing work but performed better than the second existing work. This proves that the author’s algorithms could be further improved in the future. The main aim of the project has also been achieved after completing this project where the comparison between two machine learning algorithms, Naïve Bayes and SVM, has been made. This whole project was completed using R language in R Studio, which is a software that can be used for visualisation and classification. Naïve Bayes trained the data faster even though the accuracy might be less than SVM but in a real-life situation, the problems encountered might always need a non-linear classification. Therefore, SVM is still better to use for classification with further improvements required.

# Chapter 9: References

Abdiansah, A. and Wardoyo, R. (2015) ‘Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM’, *International Journal of Computer Applications*, 128(3), pp. 28–34. doi:10.5120/ijca2015906480.

Adetunmbi, A., Sarumi, O. and Boyinbode, O. (2018) ‘Machine Learning Approach to Sentiment Analysis of Users Movie Reviews’, in.

Agarwal, V., Aher, P. and Sawant, V. (2018) ‘Automated Aspect Extraction and Aspect Oriented Sentiment Analysis on Hotel Review Datasets’, in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India: IEEE, pp. 1–4. doi:10.1109/ICCUBEA.2018.8697364.

Bhoir, P. and Kolte, S. (2015) ‘Sentiment analysis of movie reviews using lexicon approach’, in *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Madurai, India: IEEE, pp. 1–6. doi:10.1109/ICCIC.2015.7435796.

Bianchi, N. (2021) *8 Business Examples of Sentiment Analysis in Action*. Available at: https://www.repustate.com/blog/sentiment-analysis-real-world-examples/ (Accessed: 2 December 2021).

Brown, J. (2015) *Companies using sentiment-analysis software to understand employee concerns*, *CIO Dive*. Available at: https://www.ciodive.com/news/companies-using-sentiment-analysis-software-to-understand-employee-concerns/407357/ (Accessed: 30 November 2021).

Flores, A.C. *et al.* (2018) ‘An Evaluation of SVM and Naive Bayes with SMOTE on Sentiment Analysis Data Set’, in *2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*. *2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*, Phuket: IEEE, pp. 1–4. doi:10.1109/ICEAST.2018.8434401.

Kalaivani, P. and Shunmuganathan, D.K.L. (2013) ‘SENTIMENT CLASSIFICATION OF MOVIE REVIEWS BY SUPERVISED MACHINE LEARNING APPROACHES’, 4, p. 8.

Li, B., Wang, Q. and Hu, J. (2009) ‘A fast SVM training method for very large datasets’, in *2009 International Joint Conference on Neural Networks*. *2009 International Joint Conference on Neural Networks (IJCNN 2009 - Atlanta)*, Atlanta, Ga, USA: IEEE, pp. 1784–1789. doi:10.1109/IJCNN.2009.5178618.

Lv, H. and Tang, H. (2011) ‘Machine Learning Methods and Their Application Research’, in *2011 2nd International Symposium on Intelligence Information Processing and Trusted Computing*. *2011 2nd International Symposium on Intelligence Information Processing and Trusted Computing (IPTC)*, Wuhan, China: IEEE, pp. 108–110. doi:10.1109/IPTC.2011.34.

Maas, A.L. *et al.* (2011) ‘Learning Word Vectors for Sentiment Analysis’, p. 9.

Madasu, A. and E, S. (2020) ‘Efficient Feature Selection techniques for Sentiment Analysis’, *arXiv:1911.00288 [cs]* [Preprint]. Available at: http://arxiv.org/abs/1911.00288 (Accessed: 3 August 2021).

Mullen, T. and Collier, N. (2004) ‘Sentiment analysis using support vector machines with diverse information sources’, p. 7.

Nagamma, P. *et al.* (2015) ‘An improved sentiment analysis of online movie reviews based on clustering for box-office prediction’, in *International Conference on Computing, Communication & Automation*. *2015 International Conference on Computing, Communication & Automation (ICCCA)*, Greater Noida, India: IEEE, pp. 933–937. doi:10.1109/CCAA.2015.7148530.

Noyum, V.D. *et al.* (2021) ‘Boosting the Predictive Accurary of Singer Identification Using Discrete Wavelet Transform For Feature Extraction’, *arXiv:2102.00550 [cs, eess]* [Preprint]. Available at: http://arxiv.org/abs/2102.00550 (Accessed: 1 December 2021).

Pang, B. and Lee, L. (no date) ‘Opinion mining and sentiment analysis’, p. 94.

Pang, B., Lee, L. and Vaithyanathan, S. (2002) ‘Thumbs up? Sentiment Classification using Machine Learning Techniques’, *arXiv:cs/0205070* [Preprint]. Available at: http://arxiv.org/abs/cs/0205070 (Accessed: 21 June 2021).

Parveen, H. and Pandey, S. (2016) ‘Sentiment analysis on Twitter Data-set using Naive Bayes algorithm’, in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*. *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, Bangalore, India: IEEE, pp. 416–419. doi:10.1109/ICATCCT.2016.7912034.

Parveen, R., Shrivastava, N. and Tripathi, P. (2020) ‘Sentiment Classification of Movie Reviews by Supervised Machine Learning Approaches Using Ensemble Learning & Voted Algorithm’, in *2nd International Conference on Data, Engineering and Applications (IDEA)*. *2020 2nd International Conference on Data, Engineering and Applications (IDEA)*, Bhopal, India: IEEE, pp. 1–6. doi:10.1109/IDEA49133.2020.9170684.

Patil, G. *et al.* (2007) ‘Sentiment Analysis Using Support Vector Machine’, 2(1), p. 6.

Rana, S. and Singh, A. (2016) ‘Comparative analysis of sentiment orientation using SVM and Naive Bayes techniques’, in *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*. *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, Dehradun, India: IEEE, pp. 106–111. doi:10.1109/NGCT.2016.7877399.

Rathi, M. *et al.* (2018) ‘Sentiment Analysis of Tweets Using Machine Learning Approach’, in *2018 Eleventh International Conference on Contemporary Computing (IC3)*. *2018 Eleventh International Conference on Contemporary Computing (IC3)*, Noida: IEEE, pp. 1–3. doi:10.1109/IC3.2018.8530517.

von Rueden, L. *et al.* (2021) ‘Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems’, *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1. doi:10.1109/TKDE.2021.3079836.

Singh, J. and Tripathi, P. (2021) ‘Sentiment analysis of Twitter data by making use of SVM, Random Forest and Decision Tree algorithm’, in *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*. *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, Bhopal, India: IEEE, pp. 193–198. doi:10.1109/CSNT51715.2021.9509679.

Tripathy, A. (2015) ‘Classification of Sentimental Reviews Using Machine Learning Techniques’, *Procedia Computer Science*, p. 9.

Tyagi, E. *et al.* (2017) ‘Sentiment Analysis of Product Reviews using Support Vector Machine Learning Algorithm’, *Indian Journal of Science and Technology*, 10(35), pp. 1–9. doi:10.17485/ijst/2017/v10i35/118965.

Untawale, T.M. and Choudhari, G. (2019) ‘Implementation of Sentiment Classification of Movie Reviews by Supervised Machine Learning Approaches’, in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India: IEEE, pp. 1197–1200. doi:10.1109/ICCMC.2019.8819800.

Wang, S. *et al.* (2014) ‘Training data reduction to speed up SVM training’, *Applied Intelligence*, 41(2), pp. 405–420. doi:10.1007/s10489-014-0524-2.

# Chapter 10: Bibliographies

*An Introduction to Text Processing and Analysis with R* (no date). Available at: https://m-clark.github.io/text-analysis-with-R/ (Accessed: 14 September 2021).

Bhalla, D. (no date) ‘R : Keep / Drop Columns from Data Frame’, *ListenData*. Available at: https://www.listendata.com/2015/06/r-keep-drop-columns-from-data-frame.html (Accessed: 4 September 2021).

Bhateja, V. *et al.* (2020) *Evolution in Computational Intelligence: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020), Volume 1*. Springer Nature.

Brownlee, J. (2017) ‘How to Prepare Movie Review Data for Sentiment Analysis (Text Classification)’, *Machine Learning Mastery*, 15 October. Available at: https://machinelearningmastery.com/prepare-movie-review-data-sentiment-analysis/ (Accessed: 1 September 2021).

Brownlee, J. (2020) ‘How to Perform Data Cleaning for Machine Learning with Python’, *Machine Learning Mastery*, 19 March. Available at: https://machinelearningmastery.com/basic-data-cleaning-for-machine-learning/ (Accessed: 1 September 2021).

Coombs, A. (2017) ‘Understanding Sentiment Analysis in Social Media Monitoring, Unamo Blog’, *Unamo Blog*, 12 July. Available at: https://unamo.com/blog/social/sentiment-analysis-social-media-monitoring (Accessed: 2 December 2021).

edureka! (2017) *Sentiment Analysis in R | Sentiment Analysis of Twitter Data | Data Science Training | Edureka*. Available at: https://www.youtube.com/watch?v=-JW6\_kcHDj4 (Accessed: 30 September 2021).

edureka! (2019) *Support Vector Machine Tutorial Using R | SVM Algorithm Explained | Data Science Training | Edureka*. Available at: https://www.youtube.com/watch?v=RKZoJVMr6CU (Accessed: 30 September 2021).

Gupta, S. (2018a) ‘Data Preprocessing in R’, *Medium*, 19 November. Available at: https://medium.com/@shubhanshugupta/data-preprocessing-in-r-2f0e25487bb (Accessed: 18 October 2021).

Gupta, S. (2018b) *Multi-Class Classification in Text using R*, *Medium*. Available at: https://towardsdatascience.com/multi-class-classification-in-text-using-r-e6cf72ef1da3 (Accessed: 18 October 2021).

Jivane, N. (2018) *Twitter Sentiment Analysis of Movie Reviews using Machine Learning Techniques.*, *Medium*. Available at: https://medium.datadriveninvestor.com/twitter-sentiment-analysis-of-movie-reviews-using-machine-learning-techniques-23d4724e7b05 (Accessed: 20 October 2021).

Sharma, M. (2020) *Sentiment Analysis (Introduction to Naive Bayes Algorithm)*, *Medium*. Available at: https://towardsdatascience.com/sentiment-analysis-introduction-to-naive-bayes-algorithm-96831d77ac91 (Accessed: 10 August 2021).

Sheik Abdullah, A. *et al.* (2021) ‘Sentiment Analysis of Movie Reviews Using Support Vector Machine Classifier with Linear Kernel Function’, in Bhateja, V. et al. (eds) *Evolution in Computational Intelligence*. Singapore: Springer Singapore (Advances in Intelligent Systems and Computing), pp. 345–354. doi:10.1007/978-981-15-5788-0\_34.

# Chapter 11: Appendices

**Results of different performance metrics with different number of reviews for SVM and Naïve Bayes.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment** | **No. of reviews in training dataset** | **No. of reviews in testing dataset** | **Precision 1, sparse term <100** | |
| **SVM (%)** | **Naïve Bayes (%)** |
| 1. | 800 | 200 | 83.0 | 80.0 |
| 2. | 600 | 400 | 78.0 | 72.0 |
| 3. | 400 | 600 | 77.5 | 67.0 |
| 4. | 200 | 800 | 72.0 | 67.0 |
| 5. | 100 | 900 | 66.5 | 62.0 |

**Table 10: Results of Precision with different number of reviews in training and testing set for the total of 1,000 movie reviews**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment** | **No. of reviews in training dataset** | **No. of reviews in testing dataset** | **Precision 2, sparse term <100** | |
| **SVM (%)** | **Naïve Bayes (%)** |
| 1. | 16,000 | 4,000 | 86.0 | 85.8 |
| 2. | 14,000 | 6,000 | 85.0 | 85.9 |
| 3. | 12,000 | 8,000 | 85.0 | 85.8 |
| 4. | 10,000 | 10,000 | 83.5 | 85.1 |
| 5. | 8,000 | 12,000 | 84.0 | 85.4 |

**Table 11: Results of Precision with different number of reviews in training and testing set for the total of 20,000 movie reviews**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment** | **No. of reviews in training dataset** | **No. of reviews in testing dataset** | **Precision 3, sparse term <100** | |
| **SVM (%)** | **Naïve Bayes (%)** |
| 1. | 40,000 | 10,000 | 86.0 | 85.6 |
| 2. | 35,000 | 15,000 | 86.0 | 85.5 |
| 3. | 30,000 | 20,000 | 85.5 | 85.5 |
| 4. | 25,000 | 25,000 | 85.5 | 85.4 |
| 5. | 20,000 | 30,000 | 85.0 | 85.6 |

**Table 12: Results of Precision with different number of reviews in training and testing set for the total of 50,000 movie reviews**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment** | **No. of reviews in training dataset** | **No. of reviews in testing dataset** | **Recall 1, sparse term <100** | |
| **SVM (%)** | **Naïve Bayes (%)** |
| 1. | 800 | 200 | 83.0 | 75 |
| 2. | 600 | 400 | 78.5 | 80 |
| 3. | 400 | 600 | 77.5 | 80 |
| 4. | 200 | 800 | 72.0 | 80 |
| 5. | 100 | 900 | 65.5 | 81 |

**Table 13: Results of Recall with different number of reviews in training and testing set for the total of 1,000 movie reviews**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment** | **No. of reviews in training dataset** | **No. of reviews in testing dataset** | **Recall 2, sparse term <100** | |
| **SVM (%) sparse term <90** | **Naïve Bayes (%)** |
| 1. | 16,000 | 4,000 | 86.0 | 83.6 |
| 2. | 14,000 | 6,000 | 85.5 | 83.8 |
| 3. | 12,000 | 8,000 | 85.0 | 82.5 |
| 4. | 10,000 | 10,000 | 84.0 | 84.1 |
| 5. | 8,000 | 12,000 | 84.0 | 83.4 |

**Table 14: Results of Recall with different number of reviews in training and testing set for the total of 20,000 movie reviews**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment** | **No. of reviews in training dataset** | **No. of reviews in testing dataset** | **Recall 3, sparse term <100** | |
| **SVM (%)** | **Naïve Bayes (%)** |
| 1. | 40,000 | 10,000 | 86.0 | 83.2 |
| 2. | 35,000 | 15,000 | 86.0 | 83.4 |
| 3. | 30,000 | 20,000 | 85.5 | 83.0 |
| 4. | 25,000 | 25,000 | 85.5 | 83.1 |
| 5. | 20,000 | 30,000 | 85.0 | 82.8 |

**Table 15: Results of Recall with different number of reviews in training and testing set for the total of 50,000 movie reviews**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment** | **No. of reviews in training dataset** | **No. of reviews in testing dataset** | **F-measure (f-1) 1, sparse term <100** | |
| **SVM (%)** | **Naïve Bayes (%)** |
| 1. | 800 | 200 | 82.5 | 78 |
| 2. | 600 | 400 | 78.0 | 76 |
| 3. | 400 | 600 | 77.0 | 73 |
| 4. | 200 | 800 | 72.0 | 73 |
| 5. | 100 | 900 | 64.5 | 70 |

**Table 16: Results of F-measure with different number of reviews in training and testing set for the total of 1000 reviews.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment** | **No. of reviews in training dataset** | **No. of reviews in testing dataset** | **F-measure (f-1) 2, sparse term <100** | |
| **SVM (%) sparse term <90** | **Naïve Bayes (%)** |
| 1. | 16,000 | 4,000 | 86.0 | 84.7 |
| 2. | 14,000 | 6,000 | 85.0 | 84.8 |
| 3. | 12,000 | 8,000 | 85.0 | 84.1 |
| 4. | 10,000 | 10,000 | 83.5 | 84.6 |
| 5. | 8,000 | 12,000 | 84.0 | 84.4 |

**Table 17: Results of F-measure with different number of reviews in training and testing set for the total of 20,000 reviews.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment** | **No. of reviews in training dataset** | **No. of reviews in testing dataset** | **F-measure (f-1) 3, sparse term <100** | |
| **SVM (%) sparse term <90** | **Naïve Bayes (%)** |
| 1. | 40,000 | 10,000 | 86.0 | 84.4 |
| 2. | 35,000 | 15,000 | 86.0 | 84.5 |
| 3. | 30,000 | 20,000 | 85.0 | 84.3 |
| 4. | 25,000 | 25,000 | 85.0 | 84.2 |
| 5. | 20,000 | 30,000 | 85.0 | 84.2 |

**Table 18: Results of f-measure with different number of reviews in training and testing set for the total of 50,000 reviews.**

**Final code used in this research project**

# -------------------------------------------------------------------------

################## USING NAIVE BAYES CLASSIFIER ###################

# -------------------------------------------------------------------------

# step 1: load library ----------------------------------------------------

library(readxl)

library(tm)

library(e1071)

library(RTextTools)

library(caret)

library(gmodels)

library(pillar)

library(quanteda)

library(tidyverse)

# -------------------------------------------------------------------------

# STEP 1: DATA PREPARATION

# -------------------------------------------------------------------------

# load up the excel data and explore data in R studio----------------------

#1K reviews

revRaw <- read\_excel("IMDBDataset\_altered/IMDBDataset.xlsx")

#20K reviews

revRaw <- read\_excel("IMDBDataset\_altered/IMDB20K.xlsx")

#50K reviews

revRaw <- read\_excel("IMDBDataset\_altered/IMDB50K.xlsx")

#shows the raw data of movie reviews

view(revRaw)

#check data to see if there are missing values

length(which(!complete.cases(revRaw)))

# factorise your dependent variable into categorical object/function/variable

revRaw$sentiment <- factor(revRaw$sentiment)

# Check the counts of positive and negative scores

table(revRaw$sentiment)

#in percentage

prop.table(table(revRaw$sentiment))

# -------------------------------------------------------------------------

# STEP 2: DATA PRE-PROCESSING

# -------------------------------------------------------------------------

# sampling

# Create random samples // #1000,800 / 20k, 16k / 50k, 40k

set.seed(123)

#1k

revIndex <- sample(1000, 800)

#20k

revIndex <- sample(20000, 16000)

#50k

revIndex <- sample(50000, 40000)

#split to training and testing set

revTrain <- revRaw[revIndex, ]

revTest <- revRaw[-revIndex, ]

# review the proportion of class variable

prop.table(table(revTrain$sentiment))

prop.table(table(revTest$sentiment))

#Create a corpus from the sentences

corpTrain <- VCorpus(VectorSource(revTrain$review))

corpTest <- VCorpus(VectorSource(revTest$review))

# create a document-term sparse matrix directly for training set

dtmTrain <- DocumentTermMatrix(corpTrain, control = list(

language="english",

tolower = TRUE,

removeNumbers = TRUE,

removePunctuation = TRUE,

stopwords = TRUE,

stemming = TRUE,

removeSparseTerms = .998,

weighting = tm:: weightTfIdf,

stripWhitespace = TRUE

))

#check sparse terms

dtmTrain

#remove sparse terms

dtmTrain <- removeSparseTerms(dtmTrain, 0.99)

#summary of train dtm

glimpse(dtmTrain)

# create a document-term sparse matrix directly for test set

dtmTest <- DocumentTermMatrix(corpTest, control = list(

language="english",

tolower = TRUE,

removeNumbers = TRUE,

removePunctuation = TRUE,

stopwords = TRUE,

stemming = TRUE,

removeSparseTerms = .998,

weighting = tm:: weightTfIdf,

stripWhitespace = TRUE

))

#check sparse terms

dtmTest

#remove sparse terms

dtmTest <- removeSparseTerms(dtmTest, 0.98)

#summary of test dtm

glimpse(dtmTest)

# create function to convert counts to a factor

# ifelse(condition, value if condition is true, value if condition is false)

convertCount <- function(x) { x <- ifelse( x> 0, "Positive", "Negative") }

# apply() convert\_counts() to columns of train/test data

binaryTrain <- apply(dtmTrain, MARGIN = 2, convertCount)

binaryTest <- apply(dtmTest, MARGIN = 2, convertCount)

#check term one by one

binaryTrain

binaryTest

#check summary

str(binaryTrain)

str(binaryTest)

# -------------------------------------------------------------------------

# STEP 3: CLASSIFICATION & EVALUATION

# -------------------------------------------------------------------------

# train model on a data -------------------------------------------

# naive bayes -------------------------------------------------------------

#train on training data

revClassifier <- naiveBayes(as.matrix(binaryTrain), revTrain$sentiment)

# test on new data (which is the test data) -------------------------------

revPredict <- predict(revClassifier, as.matrix(binaryTest))

result <- confusionMatrix(revPredict, revTest$sentiment,

dnn = c('pred', 'real'))

#to check whole result including accuracy, precision, recall and f-1

result

# using precision aka pos pred value

result[["byClass"]]["Pos Pred Value"]

# using recall aka sensitivity

result[["byClass"]]["Sensitivity"]

# using f-measure aka F1

result[["byClass"]]["F1"]

# -------------------------------------------------------------------------

################## USING SVM CLASSIFIER ###################

# -------------------------------------------------------------------------

# -------------------------------------------------------------------------

# STEP 1: DATA PREPARATION

# -------------------------------------------------------------------------

#load the package

#set the seed and load/read the data

set.seed(123)

#1k

reviewLabel <- read\_excel("IMDBDataset\_altered/IMDBDataset.xlsx")

#20K reviews

reviewLabel <- read\_excel("IMDBDataset\_altered/IMDB20K.xlsx")

#50K reviews

reviewLabel <- read\_excel("IMDBDataset\_altered/IMDB50K.xlsx")

#factorise dependent var of chr into a factor

reviewLabel$sentiment <- factor(reviewLabel$sentiment)

#change to corpus from sentences

reviewCorpus <- VCorpus(VectorSource(reviewLabel$review))

# -------------------------------------------------------------------------

# STEP 2: DATA PRE-PROCESSING

# -------------------------------------------------------------------------

#document term matrix to separate the sentences into individual word

dtMatrix <- DocumentTermMatrix(reviewCorpus, control = list(

language="english",

tolower = TRUE,

removeNumbers = TRUE,

removePunctuation = TRUE,

stopwords = TRUE,

stemming = TRUE,

removeSparseTerms = .998,

weighting = tm::weightTfIdf,

stripWhitespace = TRUE

))

#check sparsity

dtMatrix

#remove sparse terms

dtMatrix <- removeSparseTerms(dtMatrix, 0.99)

#summary sparse

glimpse(dtMatrix)

#create a container to split the data into train and test set. #1000,800 / 20k, 16k / 50k, 40k

svmContainer <- create\_container(dtMatrix, as.numeric(as.factor(reviewLabel$sentiment[])),

trainSize=1:799,

testSize = 800:1000,

virgin = FALSE)

# -------------------------------------------------------------------------

# STEP 3: CLASSIFICATION & EVALUATION

# -------------------------------------------------------------------------

#train the model

svmTrain = train\_model(svmContainer, "SVM")

#predict the model

svmPredict = classify\_model(svmContainer, svmTrain)

# model summary: precision, recall, fmeasure

svmAnalytics = create\_analytics(svmContainer, svmPredict)

summary(svmAnalytics)

head(svmAnalytics@document\_summary)

svmAnalytics@ensemble\_summary

N=4

set.seed(123)

#cross validate for svm to check accuracy with 4 fold

crossVal <- cross\_validate(svmContainer,N,"SVM")

crossVal