Programme:

M.Sc. Data Analytics

Module:

B9DA107: Applied Research Project

Project Title:

Comparative Study of Sentiment Analysis of Movie Reviews using Machine Learning and Specialized Deep RNN

Submitted To:

Prof. Terri Hoare

Submitted By:

Bhakti Bidwalkar

10541969

Declaration

I hereby declare that this Applied Research Project which has been submitted for the award of Master of Science in Data Analytics to Dublin Business School is of my own investigations, except where otherwise stated, where it is acknowledged by references. Furthermore, this research is fully compliant with Dublin Business School's academic honesty policy and has not been submitted for any other degree.

Signed: Bhakti Bidwalkar

Student Number: 10541969

Date: 11th January 2021

Acknowledgment

I would like to express my sincere gratitude to my thesis guide Prof. Terri Hoare for guiding me throughout my research and providing a fertile ground at the start of my project. Her valuable and constructive suggestions helped me shape my research in a productive way. It has been a wonderful and learning experience for me to work under her tutelage which has helped me in achieving the target set initially. This research would not have been possible without RapidMiner and its resources and also H2O.ai for the general booklets.

I am also grateful for Dublin Business School for providing me opportunities to work on my desired field of study and carrying out this research.

Table of Contents

List of Figures:

List of Tables:

Abstract

Sentiment Analysis is the process of analyzing the texts and emotional tone behind any sequence of words. It is identifying the emotions laid in the opinions or feedbacks of viewers. This research introduces the problem of Sentiment Analysis of large movie reviews, consisting of multiple sentences. When these reviews consist of multiple sentences, different lengths of documents, sarcasm and irony, it is difficult to handle them using machine learning techniques such as SVM, Naive Bayes and Decision Tree.

The research compares traditional machine learning algorithms for TF-IDF, Term Occurrences and Term Frequency vectors. A new architecture that overcomes the challenges in the machine learning algorithms uses Recurrent Neural Network with Gated Recurrent Units (GRUs) for better results. This is the study of GRU's and application of its implementation to provide insights to companies and a better understanding of reviews. Movie Reviews need to be analyzed efficiently to make the best out of them for future references.

Chapter 1: Introduction

Entertainment is backbone of the society; it is the way of finding sanity in the hasty practical lives of humans. It is thus that one wants to choose the movie that he/she wants to watch specifically based on what others had to think about it. In the digital world that we have essentially become a part of, generation of data has become more common. Each time one uses the internet it is most likely that the various activities even opening or liking something generates data in one way or another. The most relevant activity is looking for someone's review for a particular service or product. IMDb is the largest online database which holds all information related to movies and their reviews. The generation of data is not enough, what makes it valuable is understanding the polarity of a given review that whether it can be classified as a positive or negative review. This classification is what at layman level known as Sentiment Analysis. Sentiment analysis is the process of extracting the attitude of the reviewer towards the movie in the text that they have written. Sentiment analysis is done at both at phrase level and paragraph level. In the present scenarios, the freedom of speech has drastically increased the importance of customer reviews and various platforms are available for customers to write. Sentiment analysis is applied to such reviews to discover the viewers' opinion on that movie; this is of important value to companies, stocks and politics. The classification of these reviews has a contribution to the revenue generation and summarizes the market response. Automation of this classification is helpful as it provides valuable information for the business.

1.1 Business Problem

The reviews generated by worldwide users are nothing but a bunch of unlabelled text until they undergo a smart classification. The purpose is to read, analyze and report the data. Machine Learning techniques are a fast and efficient way to automate this task. Sentiment analysis is a unique blend of artificial intelligence and machine learning, which helps organizations deploy

advanced tools contributing to attract consumers towards the product. This not only helps in knowing a consumer's interest but also helps in retaining them.

Wide range of techniques have been proposed and tested to retrieve information for a long time now. Starting with text-mining and data mining, the interest in analyzing non-topical text has increased and sentiment analysis is the end result of the study. The general problem with textual review classification is that, it creates a dilemma within the consumers with the contrast in the reviews. Thus understanding of emotions and the attitude that an individual holds towards a movie can be found only after identifying the underlying viewpoint. This leads to a need of specialized algorithm which accurately predicts the sentiment of a public review.

1.2 Research Problem

The advent of deep learning approaches has given rise to techniques which handle sentiment analysis to a great extent and accuracy. The availability of large unlabelled textual data can be used to learn the meanings of words and the structure of sentence formation. This has been attempted earlier by word2vec wherein it learns word embedding from unlabelled text samples. It works on using both approaches, predicting the word given its surrounding words (CBOW) and SKIP-GRAM which is predicting the surrounding words from a given word. More approaches are found capturing sentence level representations like recursive neural tensor network (RNTN). The accuracy of sentimental analysis completely depends on the base words level. Information retrieval methods are of huge significance as they are the representation of different mathematical models used to filter the data and calculate the similarity between documents. This is a classic research problem of automatic categorization and organizing data. This research compares the results of machine learning algorithms for different vector creations like TF-IDF, Term Frequency and Term

Occurrences. The later stage of the research implements a specialised deep learning algorithm- Gated Recurrent Units which outperforms the traditional machine learning algorithms.

Research Question:

Comparison of traditional machine learning algorithms and specialized deep neural network algorithm for sentiment analysis of Movie Reviews

Aim:

Automate accurate classification of various movie reviews as positive or negative

Objective:

To compare different machine learning algorithms on various information retrieval models such as TF-IDF score, word frequency and word occurrences; after obtaining insignificant results for Naïve Bayes, Deep Learning and Decision Tree, a specialized deep learning algorithm was implemented which results in a relevant classification accuracy.

Hypothesis:

Specialized deep learning algorithm Gated Recurrent Units outperforms the traditional machine learning algorithms for Sentiment Analysis of Movie Reviews. The models are compared for different vector creations such as TF-IDF, Term Occurences, and Term Frequency. The performance varies for traditional algorithms and specialized algorithms, as memorization from the previous text makes classification faster and more accurate.

1.3 Scope

The scope of the research covers the range of algorithms for general vectorization of data and understanding the different factors which yield better results. The gating mechanism used to memorize data before classification is less complex and computes faster.

1.4 Limitations

The deployment of the project on any cloud environment was limited due to hardware requirement and GPU constraint.

1.5 Dissertation Roadmap

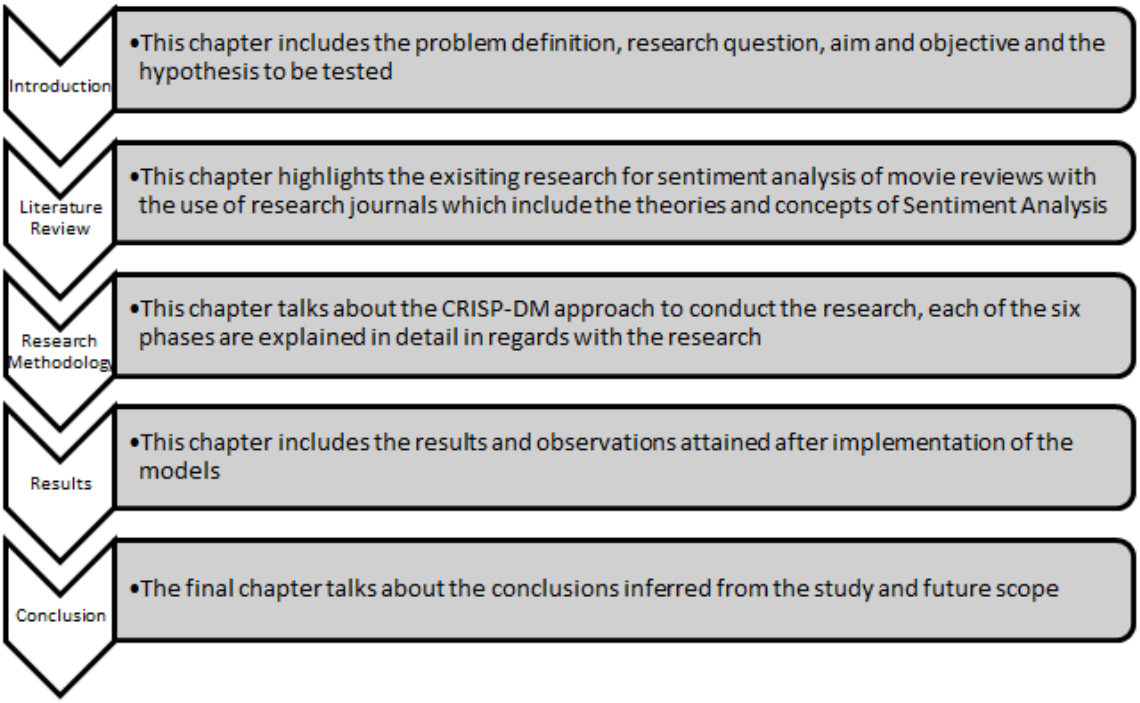

Figure 1.1 Dissertation Roadmap

Chapter 2: Literature Review

The research for sentiment analysis of movie reviews was mainly focused on the study of classic machine learning algorithms for various information retrieval models and specialized deep learning algorithms. The task was to understand which factors lead to a better classification of the sentiment. To grip on the knowledge on various theories, concepts, models and evaluation parameters the following Books and Research Journals were referred.

(Ateinza,2015) is a thoroughly curated Advanced Deep Learning book with TensorFlow 2 and Kera's. It explores the study of Multi-Layer Perceptron, CNN's and RNN's and more advanced techniques. It focuses on how to implement variants of RNN's and how they are used in machine translation and question answering problems. It simplifies how GRU's addresses the issue of long-term dependency and presents output remembering the past information. The book explains the functioning of different gating mechanisms used and explains the cell state and hidden states. Configuration of RNN's is thoroughly explored in the book. It compares classification by using different machine learning algorithms such as Naïve Bayes, Maximum Entropy, Stochastic Gradient Descent, and Support Vector Machine on social media textual data which belongs to movie review domain.

(Chollet,2017), this book on Deep learning with Python talks in deep about pre processing of text data into useful sequences of words or characters for further analysis. It explains the text sequence processing using the IMDB dataset.The book gives insights about breaking of text into tokens, all text-vectorization processes after applying tokenization schemes and then associating the numeric vectors with generated tokens. It normalizes use of e RNNs for timeseries regression ("predicting the future"), timeseries classification, anomaly detection in timeseries, and sequence labeling. The discussion of word embedding explains the various techniques to go from raw text files to a Numpy tensor that can be fed to a Keras network. . It works and engages in better working deep-learning

models. It introduces GRU layers to over the vanishing gradient problem of general RNN's. It also explains padding and truncating of text sequences for data preparation before it is fed to the Neural Networks.

(Luo, L.X., 2019.) Public reviews fall into two categories: perception based on the semantic characteristics of sentiment words and perception based on the statistical processing of functions. With the rapid development of the Internet, online media is recognized as a major medium that reflects social problems. He studies that the use of some words may not be sufficient to describe the text, and the text cannot be effectively distinguished. The text should be displayed in the semantic space. The most basic skill of this presentation is latent semantic analysis. This method splits the text matrix into individual values to represent it as a hidden concept in a low-dimensional semantic space. It has been shown that applying this method effectively increases the efficiency of text classification of texts by Latent Dirichlet Localization (LDA). Combine LDA with GRU-CNN for best results. In the proposed study, each subject is represented by a hidden space theme instead of a word space, and the CNN's long-term sequence processing power is enhanced by combining GRUs to produce a highly accurate classification.

(Dey, R. and Salemt, F.M., 2017) The three versions of the Gated Recurrent Unit (GRU) in recurrent neural networks (RNNs) are tested by preserving the configuration and consistently reducing the parameters of the GRU1, GRU2, and GRU3 RNNs in the upgrade and reset gates. It contrasts the MNIST and IMDB datasets of the three version GRU models and reveals that these GRU-RNN design models perform as well as the initial GRU RNN model, while reducing the cost of computing. The performance of Gated RNNs is largely attributed to the signaling of the gating network that controls how the new input and previous memory are used to change the current activation and create the current state. In the learning process, these gates have their own sets of weights that are adaptively modified. Using three constant base learning frequencies of 1e-3, 1e-4

and 1e-5 across 100 epochs, the IMDB data collection was educated on all 4 GRU versions. A 128-dimensional GRU RNN version was used in the training and a batch size of 32 was introduced. The usage of stochastic gradient descent often indirectly carries network state knowledge.

(Zhang, L., Wang, S. and Liu, B., 2018.) Sentiment research has been researched in data processing, online mining, text mining, and data recovery. Aspect-level emotion classification with a deep memory network has been added. Applying deep learning to emotion analysis has recently become a common subject of study. It implemented diverse frameworks for deep learning and their implementations in emotion analysis. For emotion analysis tasks, several of these deep learning approaches have shown state-of-the-art performance. It speaks to multiple levels of emotion analysis.

(Diaz et al in 2017) proposed a process to improve the accuracy of the decision support systems in sentiment analysis using the basis of the soft computing technique and probabilistic classifier called Naïve Bayes classifier. The author proposed a 'DSocial' platform to automate the process of information obtained from social networks. The author proposed a sentiment analysis based recommender system using probabilistic classifier such as Naïve Bayes classifier to identify subjective information in DSocial platform. The approach uses 2-class polarity sentiment training set of tweets from movie review domain which results in 3-class polarity sentiment classification. An appraisal was shown by the author to equate the classifier's findings with the results using 'DSocial' and professional information. For the identification of the message as 3-class polarisation, this technique may be used. The final training precision score was reported as 0.789.

(Paoli et al., 2019) concludes that sequential data could be any data that is dependent on the previous version of it. For example, text data in communication would require an understanding of a topic in a sequential manner. Another good example is sound data, as we need to remember what

someone said earlier to understand the context of the current discussion. These data recruit RNN's to perform on them. These models are generally dependent on the data sequence and changes in these affect the accuracy. RNN has the limitation of memorization, it cannot remember the context. So as a solution to this Gated Recurrent Unit (GRU) was presented by the researchers. It has a memory cell unit to remember the context of previous sequences. The study concludes that Long Short Term Memory is the best representation of Sequential models for applications that needs to understand context of the data. The study compares the variants of RNN's on their complexity and computational speed.

(Rui et al in 2015) proposed a dual sentiment analysis framework which is a novel data expansion technique to create a sentiment-reverse review for training and test data reviews. This framework will classify two-class and three-class polarity classification. The main contributions to propose this technique uses dual training and dual prediction to perform sentiment analysis. In the study duality means the original reviews with sentiment creates a dictionary and same reviews with reversed sentiment create an antonym dictionary. For sentiment reversed reviews, the author proposed a data expansion technique. This is a different technique where original and reversed reviews are constructed in one-one correspondence. They expanded this technique to training and testing stage. By making use of training set reviews sentiment, the test set reviews sentiment was predicted.

For feature extraction, author proposes the Bag of Words method and for classification purpose mainly uses logistic regression. For conducting experiment evaluation uses data sets from a different domain such as DVD, electronics and kitchen, reviews from Amazon. Even though the proposed method shows performance accuracy upto 84 percent, the experiment evaluation was not done on real-time social media or real-time data.

(Abinash et al in 2016) The suggested n-gram solution to the classifying of emotions by utilising various machine learning algorithms such as Naïve Bayes, Maximum Entropy, Stochastic Gradient Descent, and Vector Machine help for textual knowledge in the field of social networking that forms part of the field of film analysis. Better effects on the usage of unigram, bi-gram, tri-gram, and the mixture are seen in the assessment of results of the suggested solution. 88.94 percent indicates the efficiency accuracy of the planned solution utilising the unigram, bi-gram and tri-gram mix. For better outcomes, this comparison study operates on the pre-processing of data.

(Nowak and Scherer, R., 2017) This research shows that recursive neural networks are much better solutions, especially in both the form of Long Short Term Memory and BLSTM. In contrast with previous RNN's, their recurrent network worked quite well. More effectively, the other LSTM modifications performed, especially where the Gated Recurrent Unit (GRU) was used. The classification outcome is precise for the RNN network after consecutive terms in the list. The outcome returned by the network is defined by each thread. The series can be divided into three distinct groups in this situation. Proper interpretation by the reader of written texts is often problematic if they are not explicitly articulated by the speaker. Perhaps more difficult is machine text comprehension. The paper shows that recursive neural networks, especially in both LSTM and BLSTM type, are far better solutions in comparison to general algorithms. For more detailed outcomes, the potential focus of the analysis was oriented towards the usage of GRU.

(Zainuddin, N. and Selamat, A in 2014) proposed a sentiment analysis task using Support Vector Machine. For feature extraction, this proposed approach uses N-grams, different weighting schemes and also explores Chi-Square weight features. This Chi-Square weight feature provides a significant improvement in classification accuracy. The evaluation of the results in movie-related domain shows prominent results. AUC on unigram approach is 0.917, whereas AUC on bi-gram approach is 0.728. Text classification uses training and testing data-set from Pang Corpus, which has movie reviews collected from IMDb.

The literature survey on sentiment analysis concludes movie domain related data set provides improved performance results. Evaluation of SVM approach sentiment analysis technique, on movie domain, shows highest performance accuracy i.e. 0.917 on unigram model, whereas 0.728 on bi-gram model. If the experiment evaluation was performed on either social media or on real-time data, there could be a chance of change in the performance results. It then proposed a composite model for n-gram sentiment analysis showing performance accuracy up to 0.88 on both unigram and bi-gram model. The results of the model were not evaluated on any real-time data and it can perform only 3-class classification.

(Wang, X., Jiang, W. and Luo, Z., 2016) The combined CNN paradigm for Sentimental Research suggested that the analysis was focused on the previously established fact that the mood depends often on the qualitative details which provides a need for background storage. This model uses CNN's ground-grained local elements, and connects them to RNN's learning dependencies. The model range displayed a precision of 89.95%. It takes the term embedding as the input to the portion of the CNN; the RNN takes its phase in after convolution and pooling. RNN is capable of studying the long term dependencies and feeding encrypted function map to a completely connected network.

(Maas, A and Potts, 2011), The IMDb dataset is the initial journal. Stanford University researchers used unsupervised learning methods for the study of emotion. In this analysis, with near semantics, clusters of terms were created. The additional word vectors were constructed from these clusters. To split the polarity of movie ratings, they ran numerous classification strategies on these vectors. This research provided productive findings under situations where the knowledge should be rich in emotions. It also depended on the terms' semantic affinity and was focused on their true meanings. This was a constraint if a non-word accompanied the rich sentiment adjectives. Their study

explored the usage of Random Forest classifier and SVM's for the classification of the analysis. In the analysis, different function extraction techniques were also used and contrasted.

The literature review is suggestive that the traditional machine learning algorithms like Naïve Bayes, Decision Tree, SVM provide a general classification result for the movie reviews under certain limitations. Sentiment Analysis at different levels provides different results. The features on which the data is retrieved also hold a significant role in the accuracy of prediction. Thus a comparison for these features is necessary. The Bag of Words techniques also hold a major factor, as the mathematics behind them affects the data preparation. The data preparation further facilitates the classification. As the advent of deep learning methods continue, it is also studied that generating long-term dependency is much needed for accurate classification. Researchers compared the two variants of RNN's on the basis of complexity and evaluation speed. Gated Recurrent Units outperformed the Long Short Term Memory. Based on the study of existed research, this study performs a comparison of Naïve Bayes, Deep Learning and Decision Tree models for different features and finally applying GRU for significant accuracy.

Chapter 3: Research Methodology and Methods

The study will be focused on the methodology of CRISP DM, the Cross Industry Standard Process for Data Mining, which is a transparent, scalable, well-proven and versatile standard model. The CRISP-DM approach organizes the analysis into six stages, helping to further explain the procedure and offering a guide to be taken while preparing and implementing the study. The diagram below indicates the CRISP-DM model stages.

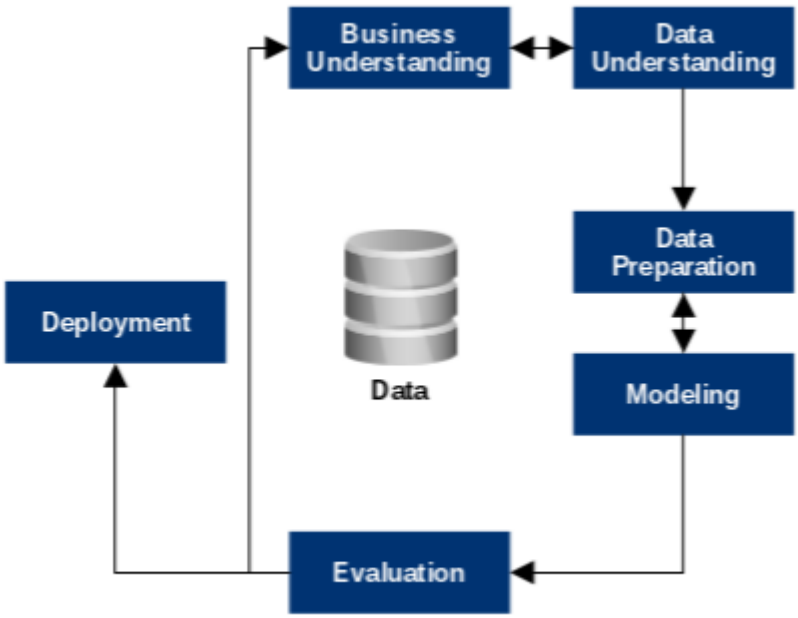

Figure 3.1 CRISP-DM Methodology

The arrows expose the phase flow and the regular dependencies between the levels. The CRISP-DM technique steps are as follows:

1. Business Understanding
2. Data Understanding
3. Data Preparation
4. Modelling
5. Evaluation
6. Deployment

3.1 Business Understanding

This process is one of the most crucial and original stages of any project for data mining. In this step, the key aim is to consider the project goals from a market point of view and this information is translated into a description of the issue of data mining and eventually create a strategy to meet the project goals. This process has taken big measures, such as identifying market priorities, evaluating the condition, determining the aims of data mining, and producing the project proposal. (2000, p. 14, Shearer). Understanding the market point of view is the essence of this process. The Internet is an immense repository of knowledge in the modern age. Companies are constantly on the hunt for reports about rivals, vendors and input from consumers. This is a cyclical chain, and consumers are still involved in other customers' opinions of the items they are shopping for. Researchers have made an excessive attempt to recognize the influence of this approach on consumer insight, patterns and the financial world. In various fields such as sentiment analysis of product ratings, financial news and healthcare, several uses of sentiment analysis came up. The electronic classification of feedback would help attract buyers.

Sentiment Analysis:

Analysis of sentiment is often called mining of opinion. Opinion mining is a text mining and natural language processing related methodology. The perception can be conveyed as an optimistic or negative feeling. The study of emotion offers the views on the data such that judgments in multiple domains can be rendered simpler. Sentiment analysis is one of the common methods in the processing of natural language, which extracts subjective data from the data provided and classifies opinions. In three stages of research dependent on the available data- Text, Sentence and Aspect levels, the method of classifying sentiment polarity occurs.

Sentiment analysis tasks include different types of strategies which are classified into mainly two types of approaches such as:

Machine learning approach: The solution to machine learning has been divided into two methods: one is controlled learning and the other is unsupervised learning. The polarity of the goal results or test data can be predicted by the supervised learning method centered on the training dataset with a finite range of groups such as positive and negative. Whereas unsupervised learning strategies are suggested when there is no chance of supplying mine data with a prior training dataset.

Lexicon-based approach: This approach is considered an unsupervised learning strategy. This approach explicitly acts on the reciprocal polarity of a statement. In a statement representing the positive or bad sense of the sentence, attach a rating to the set of words.

This review finds that all of them are a mixed operation. The accuracy of the machine learning method and the speed of the lexical approach was shown.

Under this phase, the goal of the study is to be understood and what must be the planning to attain them. The study is carried out in two different stages. The first stage involves using automodel feature of the RapidMiner tool. It generates the comparison of different models for different vector creations. The second stage involves implementation of GRU for review classification. This study holds the comparison of various information retrieval systems and the accuracies they produce for different machine learning algorithms. The input raw text-words need to be converted into tokens which are the integer values. This is required to put them through a neural network. This deep learning model is selected based on the literature review. The objective is to find which type of vector creation supports machine learning algorithms. To generate an effective accuracy recurrent neural network is implemented. As raw text cannot be fed to the neural network, it undergoes the preprocessing of tokenization and embedding into vectors. The fig. provides a general planning of how the deep learning technique will be implemented.

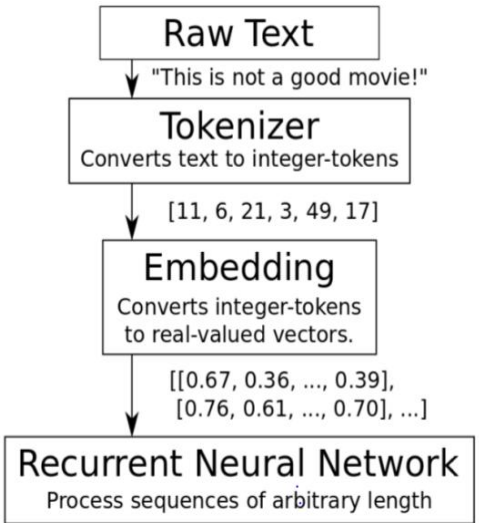

Figure 3.2 Planning of Deep Learning Technique

3.2 Data Understanding

The second phase begins by the collection of the data, which is the primary resource to begin the study. Data Understanding phase depicts the quality of the data and also generates insights about it. The dataset considered for prediction of the sentiment of movie reviews is a large movie review dataset V1.0 by Andrew L. Maas. This dataset has a total of 50,000 IMDB movie reviews divided into positive and negative data. The dataset contains the text review by the viewer and the second column contains the sentiment of the review.

3.3 Data Preparation

The third phase of the methodology involves steps to clean and transform the data for modeling phase and to generate the final dataset for the project. There are various sub steps in Data Preparation. Data selection is done to choose what data can be included in the study which will eventually contribute to the objective. The next step is to increase the quality of data. For the two stages in this study, different data preparation has been implemented.

Stage 1:

The first step is using the Auto Model feature of RapidMiner, which accelerates the process of building and validating predictive models. It can be used for clustering, classification and also for detecting outliers. This process gives an overview of which traditional models are close to predict the accurate sentiment for given text. For the dataset to undergo the Automodeling the initial step is to balance the dataset. The original dataset is retrieved from the local machine. The nominal to Text operator changes the attributes to text or string. Meta data is attached to the data for the input as attributes specification is in the metadata. The input needs a data table or an ExampleSet. The next Multiply operator creates different copies of the RapidMiner Object. The ExampleSet from the previous operator is the input for the next operator. The Filter Examples operator chooses which

Examples from the Datatable are taken and removes the others. The examples can be filtered using various conditions in parameter attribute filter. It gives three outputs. The example set output – the examples which satisfy the condition, the original- unchanged input data table and the unmatched example set- examples not satisfying the condition. As the to be predicted column is sentiment for the chosen data table, two Filter Examples operators are used. The first one filters examples for sentiment equals to positive. The second operator filters negative sentiments.

The Sample operator automatically generates a sample from the data table by choosing randomly. The numbers of data examples in sample are specified on absolute, relative or probability basis. Two samples are created for two different Filtered examples. After the sampling of two example sets the Append operator joins the two forms a combined dataset. The merging of sets is necessary for further work. The next operator is the Set Role. This operator changes the role of the attribute. For this example set, the sentiment is classified as label. After the label is set, the Write CSV operator generated the balanced CSV file. This dataset is used for further analysis.

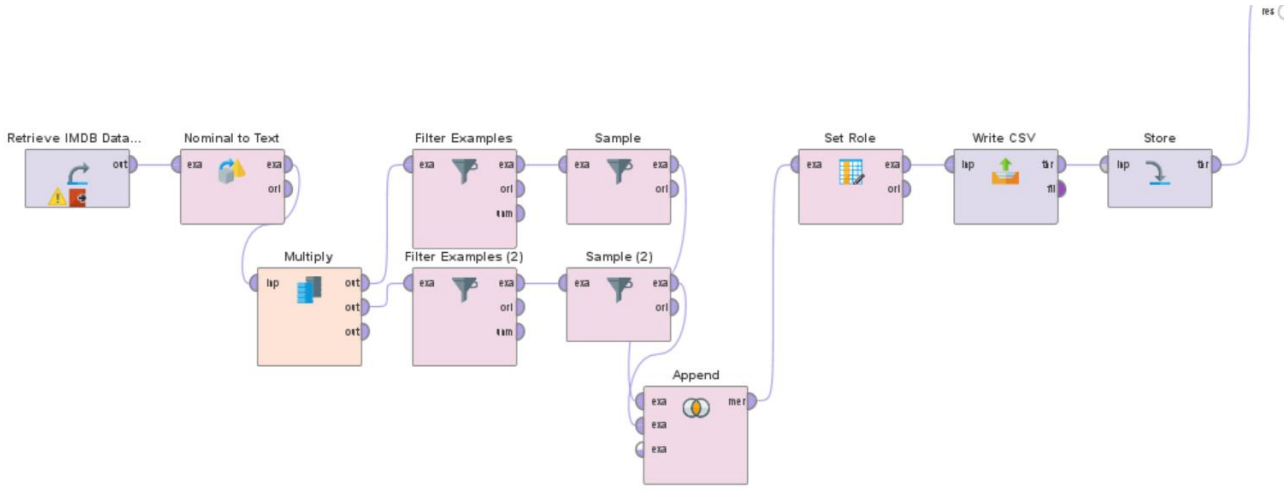

Figure 3.3 Preprocessing in RapidMiner

There are three different ways of calculating or computing the term waiting scheme for text documents. The estimation of the average term occurrences of terms in documents is centred on

both of the types which also utilises a discriminatory approach to exclude less relevant weights from the documents based on the text centroid vector. The term-weighting method is a key component of an information retrieval (IR) process utilising the vector space model.

Bag of Words (BoW):

Bag of Words is an algorithm that records the occasions in a document or a text file that a phrase appears. It is important because to enable quest, classification and modelling, it helps us to compare documents and produce comparisons. BoW is a conventional technique which is used in natural language processing and text mining to produce text representation. Each document is described in this model as a bag of words. A document is translated into a numeric function vector with a fixed duration centred on the Bag of Words model, any aspect of the document may be the word incidence, word frequency, or TF-IDF score. Its scale correlates to the size of the vocabulary. A single cell is the table where the vectorized terms and documents are saved, words become rows; documents become columns, and word count. Normally, a text vector from the Bag of Words model is a high-dimensional sparse vector in which early neural networks follow a feature representation environment. For the classification mission, this representation is further exploited. The word order is neglected in the Bag of Words paradigm, which is because the two documents will share the same word order for the same representation. The semantics of words can be coded by the Bag of Words model. The word order suffers from sparsity and high dimensionality in a short sense, even though the extension of the Bag of Words paradigm is considered such as Bag of N-Grams.

Term Frequency

The word frequency, as the name implies, is the number of occasions in a text that a term appears. The phrase applies to terms or sentences in the sense of this report. In Text Mining, Machine Learning, and Knowledge Retrieval activities, the word frequency is widely used. For various text lengths, the presence of a term in lengthy documents is more common than in shorter ones. This may result in a challenge to deciding that a certain word in a longer document is more important than in a shorter one. It is separated by the overall number of words in a paper to decrease this bias. This is known as normalization.

TF (term) = (No. of times term appears in a document) / (Total no of terms in the document).

Term Frequency is important to characterize documents. Language determiners, connectors and conjunctions dominate the term frequency.

TF-IDF or (Term Frequency (TF) — Inverse Dense Frequency (IDF))

The method of Bag of Words means that all words are broken into count and frequency, with no direct preference for a single phrase. The generic frequency is retained by all the terms, which may often not lead to the appropriate classification. Another drawback is that an insignificant term has higher frequency often and becomes more significant in the creation of features, and as it is less replicated, the important word is skipped. Sees restrictions are solved by TF-IDF. TF-IDF is a weighted ranking that is maintained as a combination of the use of a term in an expression to its use in the whole text. In an entire text, it gives the significance of a term. Word frequency (tf) is how much a term appears in a text and Reciprocal Document Frequency (idf) lowers the weight of frequently used terms, increasing the weight of less used words. Their brand offers a weight that measures how relevant a term is in the whole text. The fundamental implementation of TF-IDF can be interpreted by how it excludes the meaning provided to terms such as "the" and "a."

Term Occurrences

This is the most basic word token vectorization in RapidMiner. It simply counts the word token occurrences in each text to create a term occurrence vector.

The three Process Documents choices are the various vector analysis available in Rapid Miner for comparison of results. Auto model for different algorithms is generated on the balanced dataset.

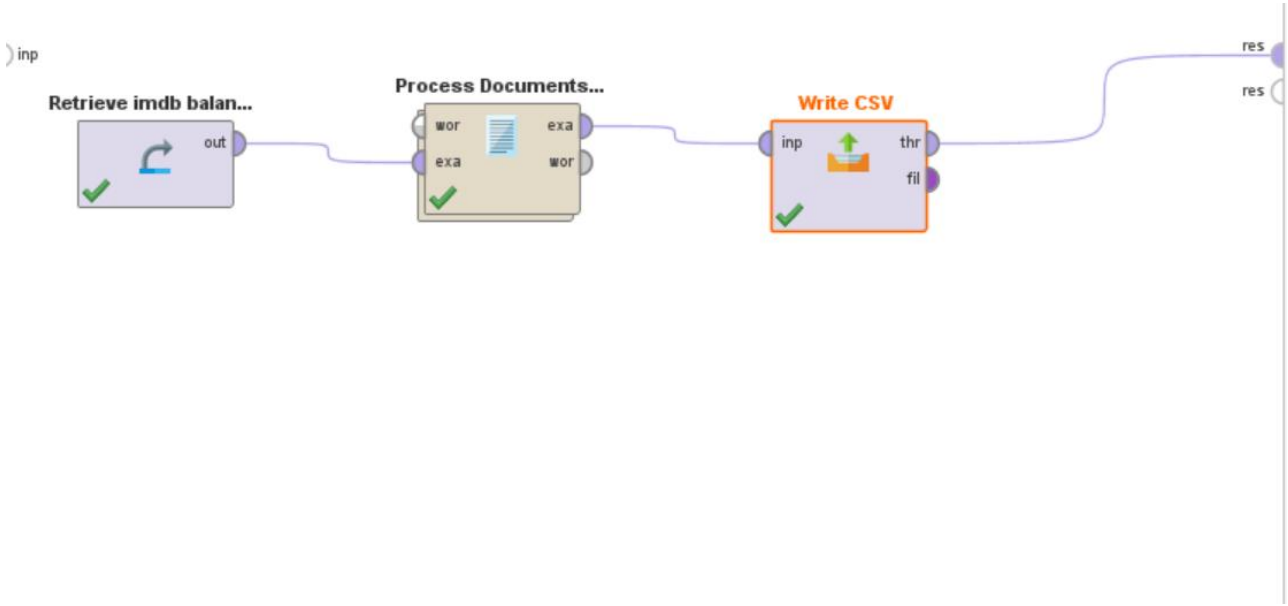

Figure 3.4 Automodel for different Algorithms

Stage 2:

Data Preparation for recurrent neural network covers functions, Tokenizer, word cloud and word embeddings. The general text reviews in the dataset included some with emoticons and various special characters which need to be removed. Also all the digits or numbers from the reviews are removed.

Stemming and Lemmatization

Stemming is the mechanism by which the term is normalised into a foundation or a single shape. It eliminates and distinguishes prefixes and suffixes. A basic stemming algorithm, for instance, would normalise playing, played into play. Lemmatization, on the other hand, is a sophisticated type of stemming technique. By depending on the central definition of the term, lemmatization groups terms. To evaluate the central concept of terms, it uses grammatical data and often uses context environments. If a term 'meeting' is taken and can either be used as a verb or noun, so stemming will generate meeting while lemmatization will generate meeting in the case of noun for verb and meeting. Lemmatization is commonly used by PorterStemmer and LancasterStemmer. A big distinction between these two algorithms is that, as opposed to Porter stemmer, Lancaster stemming requires the usage of more vocabulary of distinct sentiment. Outstanding lemmatizers involve both nltk and spacy sets. NLTK lemmatizer will be used in this analysis.

A wordcloud is also generated to see which word gets repeated the highest number of times in the first 100 reviews.

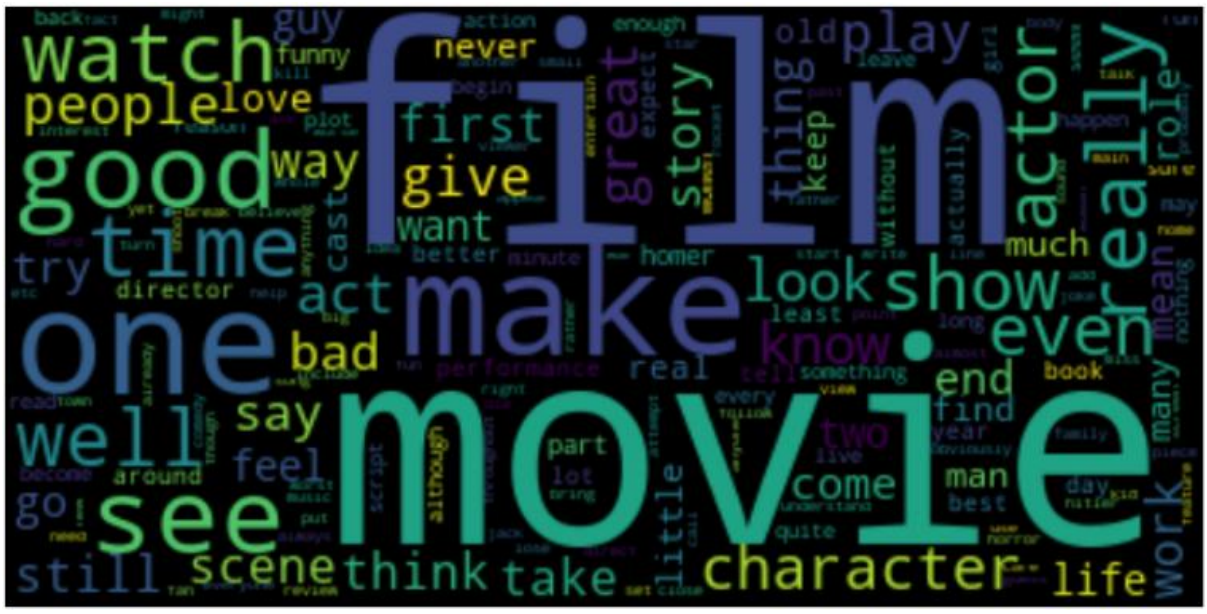

Figure 2 3.5 Word Cloud

Tokenizer

In this conversion, there are two stages, the first step is called the "tokenizer" which converts words to integers and is performed on the data set before the neural network inputs. An embedded component of the neural network itself is the second stage and is named the "embedding" layer. The tokenizer is directed to use the data-15,000 set's most common terms to construct a vocabulary.

Result: Number of unique words: 22423

Result: TOP 50 words generated by tokenizer

```
{'film': 1,
'movie': 2,
'one': 3,
'make': 4,
'like': 5,
'see': 6,
'get': 7,
'time': 8,
'good': 9,
'character': 10,
'watch': 11,
'even': 12,
'would': 13,
'think': 14,
'story': 15,
'really': 16,
'well': 17,
'show': 18,
'look': 19,
'much': 20,
'say': 21,
'end': 22,
'know': 23,
'people': 24,
'bad': 25,
'also': 26,
'first': 27,
'great': 28,
'give': 29,
'go': 30,
'act': 31,
'take': 32,
'play': 33,
```

```
'love': 34,
'come': 35,
'find': 36,
'way': 37,
'could': 38,
'movies': 39,
'seem': 40,
'work': 41,
'plot': 42,
'two': 43,
'many': 44,
'life': 45,
'want': 46,
'never': 47,
'little': 48,
'best': 49,
'try': 50,}
```

Padding and Truncating Data

Neural Networks can take arbitrary length of sequences, but to obtain a result on the whole batch of data the text sequences need to have an equal length. This is done in the study by writing a custom data generator which ensures similar length. This further leads to truncating of longer sequences and padding of shorter sequences.

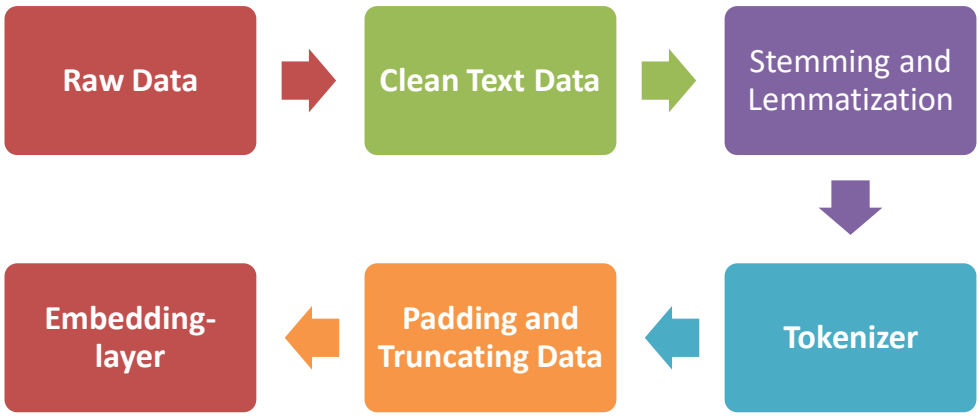

Figure 3.6 Pipeline of Data Preparation

3.4 Modelling

This phase of the methodology comprises of the selection and application of different machine learning algorithms in the research. This phase includes the building and assessment of models.

For Stage 1 modelling, the balanced dataset from the previous phase was added to the 'TurboPrep' tab of RapidMiner and began with the process of auto modeling. The auto model was run for Naïve Bayes, Deep Learning and Decision Tree.

Naïve Bayes

Naïve Bayes algorithm is exerted from statistics and probability theory. It uses the probabilistic relationship between the attributes and the class label. Based on "naïve" assumption, the naïve bayes classifier assumes that each feature being classified is independent of each other. In some cases the assumption holds true, but sometimes the simplicity and robustness of the algorithm backfires.

For X $(X1, X2, \dots)$ being the attribute set and Y being the label class. The bayes equation explaining the theorem is:

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

$$P(Y|X) = P(X1|Y) * P(X2|Y) * \dots\dots * P(Xn|Y) * P(Y)$$

$P(Y|X)$ is the conditional or the posterior probability of label given the predictor

$P(X|Y)$ is the class conditional or the likelihood of the given predictor class

$P(Y)$ is the prior probability of label class

$P(X)$ is prior probability of predictor

For the automodel the multinomial naïve bayes classifier is implemented as it is generally useful for document classification. It gives information which document belongs to which category or class, in this cases positive or negative sentiment. The predictors used in the classifier are the TF-IDF, Term Frequency and Term Occurences present in the text.

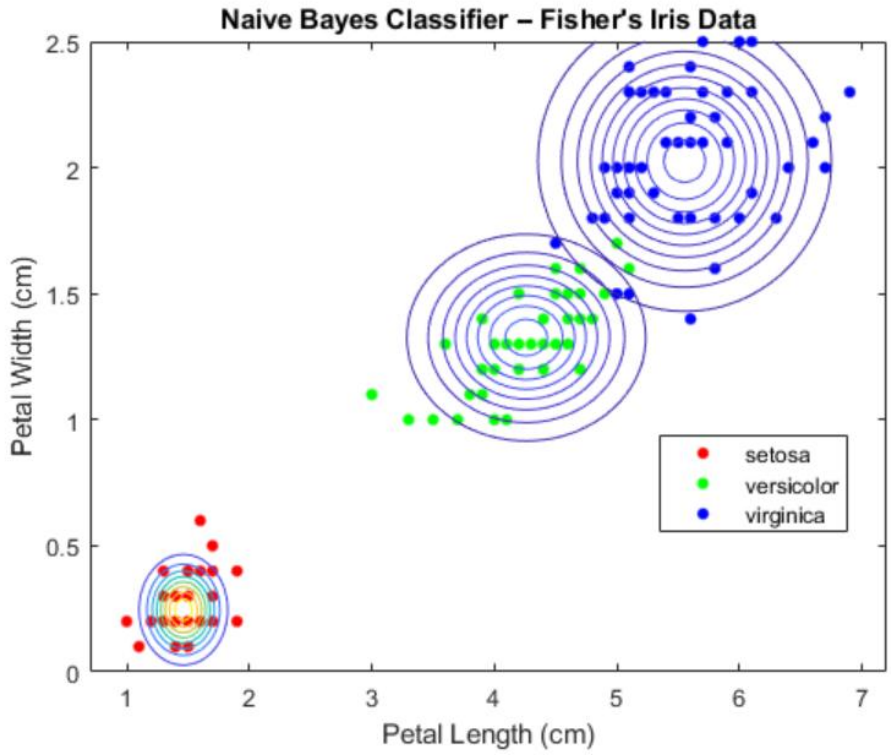

Figure 3.7 Multimodal NBC Sample (www.mathworks.com, n.d.)

Decision Tree

The key theme of the decision tree is used to explain the tree layout characterization strategy, where each node denotes a test on a feature attribute and each branch speaks to a test outcome. The leaves of the tree talk to the classes. As part of the experiment, the tree of option evaluated from our testing dataset was used. This displays the associations contained in the dataset for training. This approach is fast unless the information for planning is significant. It does not offer any

suspicion about the distribution of the likelihood of that specific data. The procedure of building the tree is called induction.

Building a Decision Tree

The decision tree algorithm is a greedy top-down algorithm that implies generating a tree with leaves that are as homogeneous as might possibly be predicted. The real phase in the algorithm is to preserve the separation of non-homogeneous leaves into leaves that are as homogeneous as might be predicted under the conditions until no further division is conceivable.. The algorithm is:

1. In the case where a majority of the features are constantly valued, classifications can be discretized.

2. If the training dataset occurs in same class, then the event will stop.

3. By choosing an attribute from the individual attributes that better partitions the articles in the node into subsets, the corresponding node is partitioned and the decision tree is made.

4. Part the node as indicated by the selected attribute chosen in the step 3. Stop if any of the accompanying conditions meets, generally proceed with step 3

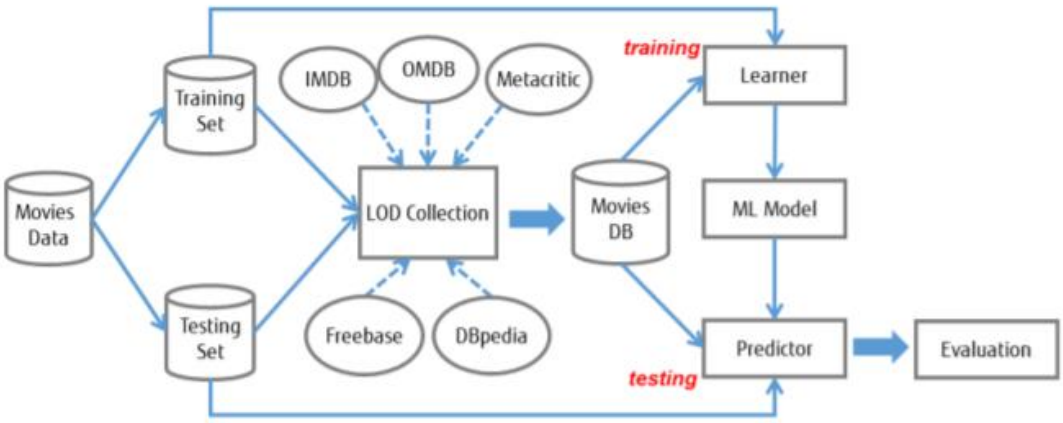

Figure 3.8 Decision Tree on Reviews (Analytics Vidhya,2019)

Deep Learning

Deep learning is the usage of different layers of artificial neural networks for the task of learning. The neural networks in deep learning was influenced by the dynamics of the human brain. A significant number of processing units named neurons have deep learning models. These neurons conduct separate functions, such as classification and representation of text. In natural language processing tasks, such as sentiment classification tasks, including text and sentence classification, recent deep learning models display remarkable results. To learn sophisticated characteristics from the sample, deep learning frameworks are often used. Deep learning is the state in which data is processed across its secret layers. Text or feature representation plays an important role in the sentiment analysis task, which represents the original knowledge expressed in a text through terms or phrases.

Stage 2

Word Embedding: For word representation, this technique is used to create a dense vector. The Term Embedding model will encode terms with semantic and syntactic properties. The system of Word Embedding takes the input of a document's terms and portrays the document as a dense vector. It is possible to derive a dense vector from the input words of a text by utilizing neural networks. Sentiment recognition can be achieved using a number of neural network models following the conventional supervised learning environment where documents are appropriately described. In certain instances, it is only feasible to use neural networks to derive text characteristics or text representations, where certain characteristics are fed into some other simple neural classifiers or neural classifiers.

Embedding-layer: The first layer in the RNN is a so-called Embedding-layer which converts each integer-token into a vector of values. This is necessary because the integer-tokens may take on

values between 0 and 1500 for a vocabulary of 1500 words. The embedding-layer also needs to know the number of words in the vocabulary (num_words) and the length of the padded token-sequences (max_tokens_length). The max_tokens_length is 282 in this case.

```
max_tokens_length = np.mean(num_tokens) + 2 * np.std(num_tokens)
max_tokens_length = int(max_tokens_length)
max_tokens_length
```

```
282
```

Keras Embedding Layer

In neural networks, the embedding layer is used for text. The embedding layer transforms vectors to positive integers. This includes encoded integer data as an input in order to provide a unique integer for each word. This stage of data preparation can be carried out using the Keras Tokenizer API. With random weights, an embedding layer is initialised and then embedding is trained for all terms in the testing dataset. During model planning, this layer must always be the first layer and three obligatory arguments are needed.

• Input_dim: It's the vocabulary size in the text info. Suppose our data is encoded in an integer range of 0-10, so 11 will be the vocabulary size. We measure this value in our case at runtime.

• Output_dim: It is the size of the vector from this layer that we will get as the output for each term. This value could be 100, 64, 32 or even greater, so for a given problem, try this for different values.

• Input_length: it’s the length of sequences as input.

Convolutional Neural Network (CNN)

Convolutional neural networks are the neural networks used for the processing of grid-known data such as topology. Initially used in the area of computer vision, CNN is a special feedforward neural network. CNN consists of many convolutional layers in the visual cortex that serve the work of the cells. The pooling layers are preceded by the convolutionary layer, which is commonly used to increasingly reduce the network's number of functions and computational complexity. Using CNN, natural language processing tasks leverage one dimension(1D) structure of text knowledge for precise prediction. Convolution layers perform the function of feature extraction in CNN, which is a helpful feature for text classification. Architectures are added to text at CNN convolutional and pooling. One-dimensional sequence convolutions are specifically associated with the classification mission. CNN has the potential to incorporate pre-trained word-embedding effectively. Convolutional Neural Networks model architecture for sentence classification was shown in figure

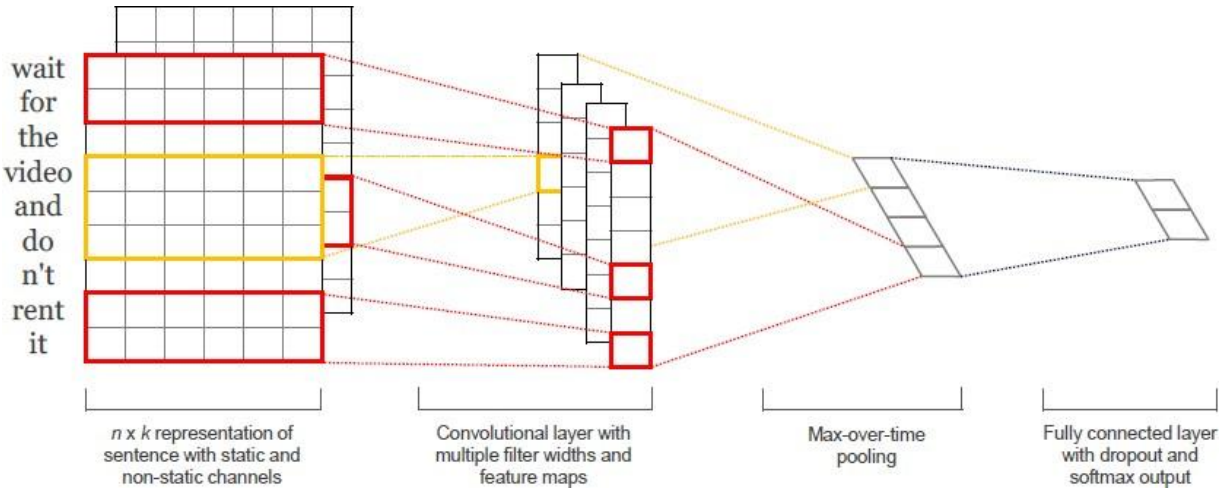

Figure 3.9 CNN Model Architecture

Recurrent Neural Networks: Recurrent neural network models are a form of neural networks that, without depending on the size of the window, are used to function for natural language processing tasks. The sentence sequences can be processed by most recurrent networks of variable duration. In a different context, recurrent networks exchange parameters. Each output member is a component of the previous output member. Each output member is generated using the same update rule that was applied to previous outputs. The exchange of parameters across a very deep computational graph results in this recurrent formulation. The simple formula behind the forward propagation in Recurrent Neural Network is, where H is the hidden state and coefficients u, v, w are weight parameters.

$$H = uX + wH'$$

$$Output = vH$$

The three-step recurrent neural network model is seen in the diagram.

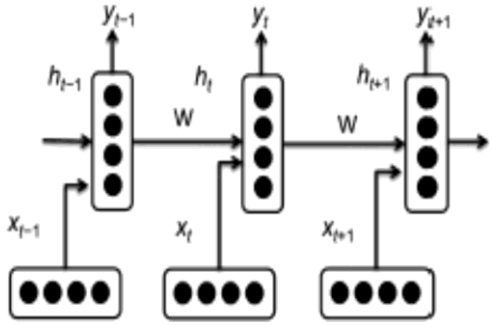

Figure 3 3.10 RNN Model Architecture

Gated Recurrent Unit

In this work, a popular variant of RNN architecture is used, which is called a Gated Recurrent Unit (GRU). There exist alternative cell architectures such as Long-Short Term Memory (LSTM) and variations of it. However, our work lies outside of specific cell architecture choice. Both GRU and LSTM cell architectures use a variant of a gating mechanism that addresses a well-known vanishing gradient problem of RNNs.

GRU incorporates the same recurrent principle of RNN, but in a gated manner. In particular, at every sequence processing step, each cell has an opportunity to either (i) shunt its previous hidden state using a reset gate r if it wants to selectively consider information from the past when computing (i.e. detecting) relevant signal q, or (ii) use an update gate s to choose how much of the previous hidden state to copy directly and how much to update with newly detected signal. From this formulation, it is notable that there is a way for the GRU cell to propagate previous hidden state without modifying it at all if that is helpful to the task, which directly addresses the vanishing gradient problem. Formally, given an external input xk at step k and the previous hidden state $hk-1$, the GRU layer is updated as follows; where Wr, Wq , Ur, Uq are weight parameters.

(1) Determine reset gate settings

$$rk \leftarrow \sigma(Wrxk + Urhk - 1 + br)$$

(2) Detect relevant event signals:

$$qk = \tanh(Wqxk + Uq(rk * hk - 1) + bq)$$

(3) Determine update gate settings

$$sk \leftarrow \sigma(Woxk + Uohk - 1 + bo)$$

(4) Update hidden state

$$hk \leftarrow sk * hk - 1 + (1 - sk) * qk.$$

These mathematical solutions make GRU's the highly specialized algorithm for classification of Sentiment Analysis.

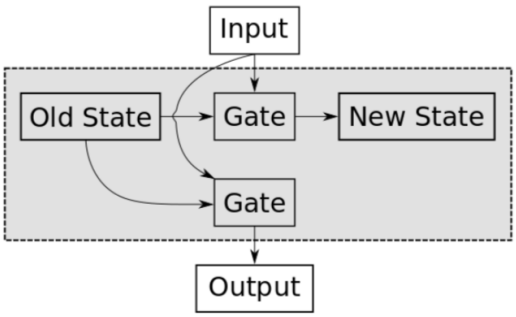

Figure 3.11 GRU Cell

After implementing GRU for the dataset, Adam optimizer is used with the given learning rate of 1e-3. Adam optimizer is used to replace the classic stochastic gradient procedure to update network. The model is then ready to be trained.

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 282, 8)            120000
_________________________________________________________________
gru_12 (GRU)                 (None, 282, 16)           1248
_________________________________________________________________
dropout_8 (Dropout)          (None, 282, 16)           0
_________________________________________________________________
gru_13 (GRU)                 (None, 282, 8)            624
_________________________________________________________________
dropout_9 (Dropout)          (None, 282, 8)            0
_________________________________________________________________
gru_14 (GRU)                 (None, 4)                 168
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 5
=================================================================
Total params: 122,045
Trainable params: 122,045
Non-trainable params: 0
_________________________________________________________________
```

Figure 3.12 Trained Model

3.5 Evaluation

The next phase involves the evaluation of the models thus reviewing if it achieves the business objectives. The research evaluates all the models based on the accuracy, precision and recall. These parameters are calculated on the basis of if the sentiment classified is true or false. If the model classifies a positive sentiment as positive it is counted as true positive(TP), while if it says negative it is inferred as true negative (TN). On the same ground, if a negative sentiment is classified as negative , it is false positive(FP) and negative classified as positive is said false negative(FN).

Accuracy

The most frequently used metric for analyzing the performance of different models is accuracy. It is sum of correct predictions by the number of total predictions made. In simple terms, it is the measure of the number of the times the model was absolutely correct.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall

Recall holds the measure of how much the model correctly identifies True Positives. In this study it is the number of true positive reviews in all the positive reviews. It is calculated by dividing the true positives with the sum of true positives and false negatives.

$$Recall = \frac{TP}{TN+FN}$$

Precision

Precision is the ratio between the True Positives and all the Positives. Precision is a measure which comes in handy when the false positives are in the priority. It suggests the calculation of when the model classified a review as positive but it was not positive in real.

$$Precision = \frac{TP}{TP+FP}$$

3.6 Deployment

The final phase of the methodology is deployment. After the models are built, modelled and evaluated they need to be presented in a way that will aid the decision making. This phase is the complete understanding of the model built and ready to be used. The deployment of this study was done on the local machine with the required experimental setup.

The experimental setup is to generate and test models for text review classification GRU Recurrent neural networks. All the experiments are carried on identical hardware and operating system.

Operating System: Windows 10 Education 64-bit (10.0, Build 18363)

System Manufacturer: HP Inc.

Processor: Intel(R) Core (TM) i5-8265U CPU @ 1.60 GHz ~1.8GHz

Memory: 8192 MB RAM (8 GB)

The Python packages used to deploy the neural network with their version were:

Table 1 Packages Used

Package name	Version
Keras	2.3.1
matplotlib	3.1.3
nltk	3.4.5
numpy	1.18.3
pickle	4.0
pandas	1.0.1
re	2.2.1
sklearn	0.20.3
tensorflow	1.14.0

Chapter 4: Results and Discussion

The evaluation metrices will be used to compare the results among the different models built and deployed. Stage 1: For the automodeling stage of Rapid Miner, the three traditional machine learning algorithms; Naïve Bayes, Deep Learning and Decision Tree were assessed on the basis of various evaluation metrices such as classification error, accuracy, precision, recall and more. These evaluation metrices are different for the three different vector creations of Process documents.

Results for TF-IDF score. Tf-Idf score is a simple is a linear transformation of frequency. After comparison of all the metrics Deep Learning was the highest accuracy classifier.

Table 2 Results for TFIDF

Metric	Naïve Bayes	Deep Learning	Decision Tree
Accuracy	59.7 ± 1.0	71.1 ± 1.8	61.5 ± 1.5
Classification Error	40.3 ± 1.0	28.9 ± 1.8	38.5 ± 1.5
AUC	64.9 ± 1.0	82 ± 1.8	61.7 ± 1.6
Precision	85.4 ± 6.0	78.2 ± 1.1	57.4 ± 1.0
Recall	23.8 ± 1.9	58.7 ± 3.4	89.7 ± 2.5
F Measure	37.1 ± 2.1	67.0 ± 2.6	70.0 ± 1.0
Sensitivity	23.8 ± 1.9	58.7 ± 3.4	89.7 ± 2.5
Specificity	95.8 ± 2.0	83.6 ± 0.8	33.3 ± 3.4
Train(1000)	267ms	6s	802ms
Score(1000)	55s	53s	46s
Total(1000)	1min44s	1min40s	1min35s

When using the 'Process documents' from data operator, a wordlist is gained which gives overview of different words in the document or the exampleset given. The occurrences in the label set shows the values as different categories for which term occurrence is calculated. Results for Term Occurrences in vector creation. It was the same result in this Data Mining technique as well.

Table 3 Results for Term Occurences

Metric	Naïve Bayes	Deep Learning	Decision Tree
Accuracy	51.0 ± 1.1	71.8 ± 1.7	56.6 ± 1.4
Classification Error	49 ± 1.1	28.2 ± 1.7	43.4 ± 1.4
AUC	56.0 ± 2.3	82.1 ± 3.4	56.7 ± 1.5
Precision	69.3 ± 13.2	82.2 ± 2.5	53.6 ± 0.8
Recall	53.5 ± 1.7	55.6 ± 3.6	97.9 ± 2.4
F Measure	66.6 ± 3.1	66.3 ± 2.5	69.3 ± 1.1
Sensitivity	53.5 ± 1.7	55.6 ± 3.6	97.9 ± 2.4
Specificity	98.6 ± 0.5	87.9 ± 2.4	15.4 ± 2.1
Train(1000)	135ms	8s	413ms
Score(1000)	50s	56s	47s
Total(1000)	49s	2min1s	1min25s

- Results for Term Frequency.

Table 4 Results for Term Frequency

Metric	Naïve Bayes	Deep Learning	Decision tree
Accuracy	61.7 ± 1.2	72.0 ± 2.3	50.0 ± 0.2
Classification Error	38.3 ± 1.2	28.0 ± 2.3	50.0 ± 0.2
AUC	65.8 ± 0.4	81.4 ± 2.1	61.6 ± 1.9
Precision	82.6 ± 2.6	78.3 ± 2.7	52.5 ± 1.2
Recall	29.8 ± 2.1	60.7 ± 4.4	82.9 ± 2.2
F Measure	43.8 ± 2.4	68.4 ± 3.2	61.9 ± 1.6
Sensitivity	29.8 ± 2.1	60.7 ± 4.4	86.4 ± 2.5
Specificity	93.7 ± 1.0	83.2 ± 2.4	89.7 ± 1.4
Train(1000)	162s	8s	861ms
Score(1000)	52s	55s	1min1s
Total(1000)	50s	1min52s	1min1s

When studied about the Deep Learning algorithm in the RapidMiner it was found that it was too good to be true. The general state of the art algorithm used is a multi-layer, feedforward neural network consisting of many layers. This algorithm is too general with default parameters and cannot produce accurate result for complex data; but it needed a specialization for accurate results.

After evaluating the results for the three traditional algorithms it was quite clear that deep learning takes more time but rather gives highest accuracy in all three cases. This led to an implementation of specialized deep learning algorithm for Stage 2 of the study.

The specialised deep learning model implemented- Gated Recurrent Unit.

The results achieved by GRU outperformed any of the model implementation and also in the variants of neural networks as suggested by the Literature review. The time taken to train the model is high, but it leads a satisfactory accuracy in the end making it best performing. 10% of data was split for validation, thus leading to a comparison of validation and testing accuracy. It is found that GRU cells should show higher specificity as they do not have their own memory and therefore tend to learn more like an exclusion principle. The difference between Validation and Testing Accuracy is around 0.36. This could be understood by considering the size of the data.

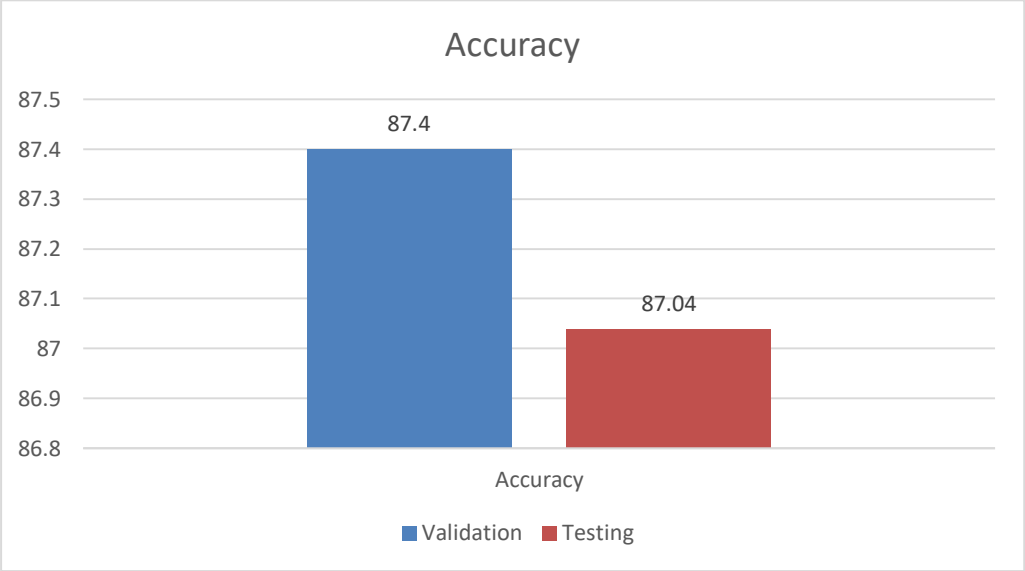

Figure 4.1 Comparative Validation and Testing Accuracy

In Machine Learning field for statistical classification, a confusion matrix is a table which allows visualization of the performance of an algorithm. It calculates the instances of actual and predicted values. The rows of matrix represent the instances of predicted class while columns represent the instances in actual class. The confusion matrix the GRU implemented was

```
Confusion matrix:
array([[1145,  186],
       [ 138, 1031]])
```

Figure 4.2 Confusion Matrix

Out of the 2500 texts used, 138+ 186 = 324 were incorrectly classified.

The results mentioned in table 1,2 and 3 are of Naïve Bayes, Decision Tree and Deep Learning. First as decided in the evaluation, the models will be compared on the basis of their recall accuracy and precision as our main objective here is to find a model with higher accuracy. It is clearly visible that Deep Learning has a better accuracy than DT and NB in all the three cases. When compared to the Test Accuracy of the GRU, the GRU outperformed all the traditional algorithms. The final test accuracy is Test Accuracy: 87.04%.

```
print("Test Accuracy: {0:.2%}".format(test_result[1]))
```

```
Test Accuracy: 87.04%
```

Limitations of the research

The major limitation of RNN's in general is the hardware complexity. GRU is difficult to train for higher epochs as it requires memory-bandwidth-bound computations, which limits the real-life applications of neural network solutions. The output and hidden state at any given state are the same in the case of GRU. This may give LSTM an edge over GRU. Temporal Convolutional Networks (TCN) are more likely to achieve the hierarchical neural encoder limitations. It has the limitation that human brain sometimes interprets character and not just read sequentially. This rolls out the faster processing. GRU exposed the complete hidden content without any control. If the computational power could be increased a more significant research is possible on different specialized neural networks.

Chapter 5: Conclusion

The key emphasis of this study is the classification of the reviews in order to have its emotions in the context of positive or negative polarity. Comparison of the findings presented in this analysis with literature results indicates that the proposed results have been stronger than some. The used dataset is the IMDB movie review dataset to see the performance of Naïve Bayes, Deep Learning and Decision Tree algorithms for the task of sentiment analysis. The neural network based approach outperformed all the other approached in Binary classification as evident in the previous research.

The research began with studying various Data Mining techniques for such complex text analysis. Various Bag of words techniques were studied and their differences were understood. Vector Analysis does not bring a drastic change but Term frequency gave the highest accuracy in Deep Learning of 72%. The Deep Learning implementation by H2o.ai is generally applied and works with default parameters. The technique of Word2Vec pre-trained word embedding to acquire the contextual semantics of terms from the text can be effectively categorised for the production of documents. On the basis of experimental observation, it is found that RNNs models would easily collect valuable knowledge from a vast array of sequential data and the better option in terms of precision, in a repeating network.

When compare with the traditional machine learning algorithms like Naïve Bayes and Decision Tree, it was evident specialized RNN performed better. Naïve Bayes belongs to a category of models called generative. During training of model NB focuses to find how the data was generated. It studies the underlying distribution of the examples fed to the model. While RNN is a discriminative model which figures out the differences between the label class.

Since the Deep Learning implemented in Auto model runs too general with default parameters, a more specified algorithm with tuned hyperparameters was to be studied and implemented. The two variants that lay ground for the same were LSTM and GRU. As suggested by previous research, GRU has less complexity and fast computation. Adam optimizer is used to update network weights iterative based in training data. Since looking at the outcome of the implementation, it could be concluded that GRU obtained the exactness of 87.04 percent independent of the pre-processing techniques used. A training loss of 9.49% was achieved and Epoch-5/5 and 633 iterations with a validation loss of 4.45%.

The accuracy attained can be understood with the working of the model. The GRU generates an embedding layer of the required dimension. Then the terms in the expression, while preserving their order, are fed into the first layer as 1 of K encoding. The embedding layer will learn to use a real valued vector of size equal to the fixed dimension to reflect each term. The weights between the embedding layer and the concealed layer on top of it are these quantities. The concealed layer is composed of periodic gated modules. They are not only linked below and above the layer, but often linked to units within their own layer. Even though stacking them is feasible, only a single layered GRU is used. A description of the whole series is inferred at the end of the secret layer that this can be used as an input to the linear model or classifier. A general trend was found which improved results when there were more dimensions in word embedding and gated units. It was a general observation that because of their ability to recall long time dependencies, GRU's are productive in the task of sentiment analysis.

Being a recurrent network it can effectively capture long sequence data required for natural language understanding. They perform better than traditional bag of features models which disregard the order of the features. They eliminate the problem of exploding and diminishing gradient problem. In this research we learned how to handle basic dictionary approach for

sentiment analysis with various words formed in the serial formation of a dictionary. Next we analysed IMDB dataset within keras.

After a thorough review, it is concluded that GRU has advanced in forecasting all groups for each pre-processing technique. Another significant factor is the expense of computing. After Word tokenizer & GRU were connected with recurrent neural network, there was a significant reduction in the training time for each point. This makes GRU relatively more costly than other processes. One thing that must be noted is that the data used to train and evaluate the models was generated by human beings. The class labels are taken from the ratings given by the reviewers. There might be some events including two independent reviews that share the same emotion, but they have a separate ranking and vice versa. This may create discrepancies and affect the data for which the model is trained. Improving the epoch number will definitely increase the accuracy and reduce the loss. It was limited due to hardware. GRU cells tend to learn content that is rarely found in the data and overcome the constraint of prevalence. It is often true for categories as well, but this study had only two categories so it was not a point of significance. The sensitivity of GRU cells is also less than all variants of RNN.

In future work, there are many ways to extend this work. Future research can be dedicated to the proposed approach using multiple sentiment lexicons and much powerful ranking approaches to enhance the sentiment classification performance and also reduce the computational complexity of the proposed model.

Future Scope

The future scope of the research will be extending more algorithms for more accurate results on the dataset. A comparative study of the specialized neural network may result in better results. A softmax activation applied to lexical layer can be considered as an attention mechanism for word feature, which can make full use of the emotional semantic and lexical information. The middle layer of the auxiliary labels not only can well constrain the embedding layer of learning, but also can serve as effective feature for predicting the final sentiment labels.

It can also be explored for other word embeddings' effect on learning features and also using chi-squared method to remove irrelevant features that do not affect the orientation of the text.

It is known that certain people sometimes repeat the last character of the phrase a lot of times, such as " Greatttt, Fineee " in order to emphasise a word. Typically, these terms do not have a proper meaning; however they may be taken into consideration and further processed to describe emotions connected with the sentence as a whole. Stemming and Lemmatization has been implemented but more accurate techniques could be studied.

More specialized Deep learning approach can be used for the classification of sentiment reviews to check whether the deep learning approach shows a better accuracy result in compare to that of traditional methods. The performance of the proposed approach is checked by using confusion matrix. But, in future, the performance must be check by rigorous statistical test.

References

Adikari et al (2019) Online incremental machine learning platform for big data-driven smart traffic management. *IEEE Transactions on Intelligent Transportation Systems*, *20*(12), pp.4679-4690.

Agrawal, A. et al (2016) Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, *57*, pp.117-126.

Aizawa, A., (2003) An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, *39*(1), pp.45-65.

Albon, C., (2018) *Machine learning with python cookbook: Practical solutions from preprocessing to deep learning*. " O'Reilly Media, Inc.".

Atienza, R. (2018) *Advanced Deep Learning with Keras: Apply deep learning techniques, autoencoders, GANs, variational autoencoders, deep reinforcement learning, policy gradients, and more*. Packt Publishing Ltd.

Chanona-Hernández. et al (2014) Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, *41*(3), pp.853-860.

Chollet, F. (2018) *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG.

Chung, J., et al (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Diaz, L et al (2017) Sentiment analysis of review datasets using naive bayes and k-nn classifier. *arXiv preprint arXiv:1610.09982*.

Dey, R. and Salemt, F.M. (August 2017) Gate-variants of gated recurrent unit (GRU) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)* (pp. 1597-1600). IEEE.

Hourrane, O. and Idrissi, N. (October 2019) An Empirical Study of Deep Neural Networks Models for Sentiment Classification on Movie Reviews. In *2019 1st International Conference on Smart Systems and Data Science (ICSSD)* (pp. 1-6). IEEE.

Ibrahim, O.A.S. and Landa-Silva, D (2016) Term frequency with average term occurrences for textual information retrieval. *Soft Computing*, *20*(8), pp.3045-3061.

Kharde, V. and Sonawane, P. (2016) Sentiment analysis of twitter data: a survey of techniques. *arXiv preprint arXiv:1601.06971*.

Li, L., Goh, T.T. and Jin, D (2020) How textual quality of online reviews affect classification performance: a case of deep learning sentiment analysis. *Neural Computing and Applications*, *32*(9), pp.4387-4415.

Luo, L.X. (2019) Network text sentiment analysis method combining LDA text representation and GRU-CNN. *Personal and Ubiquitous Computing*, *23*(3-4), pp.405-412.

Maas and Potts, C., (2011) Multi-dimensional sentiment analysis with learned representations. *Stanford University. Zugriff am*, *9*, p.2014.

Nowak, J. et al, (June 2017) LSTM recurrent neural networks for short text and sentiment classification. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 553-562). Springer, Cham.

Oweis, N.E. et al (2015) A survey on big data, mining:(tools, techniques, applications and notable uses). In *Intelligent Data Analysis and Applications* (pp. 109-119). Springer, Cham.

Paoli, C. et al (2020) Time series forecasting on multivariate solar radiation data using deep learning (LSTM). *Turkish Journal of Electrical Engineering & Computer Sciences*, *28*(1), pp.211-223.

Prathipati et al (no date) Data Conditioning and Comparison of Data Cleansing tools.

Uijlings, et al (2009) July. Real-time bag of words, approximately. In *Proceedings of the ACM international Conference on Image and Video Retrieval* (pp. 1-8).

Wang, L. and Xia, R., (September 2017) Sentiment lexicon construction with representation learning based on hierarchical sentiment supervision. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 502-510).

Wang, M. et al (July 2014) Microblog sentiment analysis based on cross-media bag-of-words model. In *Proceedings of international conference on internet multimedia computing and service* (pp. 76-80).

Wang, X.et al , (October 2019) OGRU: An Optimized Gated Recurrent Unit Neural Network. In *Journal of Physics: Conference Series* (Vol. 1325, No. 1, p. 012089). IOP Publishing.

Xu, F., Pan, Z. and Xia, R., (2020) E-commerce product review sentiment classification based on a naïve Bayes continuous learning framework. *Information Processing & Management*, p.102221.

Yang, S. et al (2020) LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. In *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)* (pp. 98-101). IEEE.

Zainuddin, N. and Selamat, A. (2014) September. Sentiment analysis using support vector machine. In *2014 international conference on computer, communications, and control technology (I4CT)* (pp. 333-337). IEEE.

Zhang, L and Liu, B., (2018) Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), p.e1253.

Zhang, W. et al (2011) A comparative study of TF* IDF, LSI and multi-words for text classification. *Expert Systems with Applications*, *38*(3), pp.2758-2765.

Zheng, X., and Ordieres-Meré, J., (2018) Comparison of data preprocessing approaches for applying deep learning to human activity recognition in the context of industry 4.0. *Sensors*, *18*(7), p.2146.

Appendices

The more detailed materials of the research which could not be involved in the report but are necessary to guide through the study are included in this section.

Contents of Artefacts

Datasets:

- IMDB Dataset.csv: The primary dataset of the research
- imdb balanced.csv: pre-processed dataset for automodel
- imdb tdm.csv: cleaned dataset

Model Results: RapidMiner

- imdbpreprocessing.rmp- process for preprocessing which results in imdb balanced.csv
- imdb1.rmp- process for automodeling different models
- DecisionTreeTO.rmp –Automodel process for Decision Tree
- DeepLearnigTO.rmp – Automodel process for Deep Learning
- NaiveBayesTO.rmp –Automodel proces for Naïve Bayes

Python File

- imdb_gru.py – python file implementing GRU

List of Abbreviations

BoW- Bag of Words

CNN- Convolutional Neural Network

CRISP-DM – Cross Industry Standard Process for Data Mining

GRU- Gated Recurrent Units

IMDB- Internet Movie Database

IR- Information Retrieval

LSTM- Long Short Term Memory

NLTK- Natural Language Toolkit

RNN- Recurrent Neural Network

SVM- Support Vector Machine

TF-IDF- Term Frequency – Inverse Document Frequency