Higher Diploma in Science in Data Analytics

Sentiment Analysis of IMDb Reviews using Machine Learning Models

Final Report

Rajan Ramesh Prajapati
10385441
10385441@mydbs.ie

Supervisor: Prof. John Rowley

Date:

6th March 2020

Abstract

The aim of this study is to compare the word embeddings or vectorization techniques for IMDb reviews dataset to reveal the sentiment of the user. Word embedding can be explained as a real number, vector representation of a word. Using Natural Language Processing techniques, the textual reviews are cleaned to make sure any unwanted and useless data is not used for the machine learning models. Techniques like sub-setting, stemming, stop-word filtering, tokenization are used to create a clean corpus. The vectorization techniques taken into consideration are count vectorization, term frequency and term frequency – inverse document frequency. Multiple supervised machine learning algorithms are modelled for each vectorization technique and then compared based on performance and accuracy. On comparing all the vectorization techniques for all the models, Logistic Regression on TF-IDF outperformed other machine learning algorithms.

Acknowledgment

I would like to acknowledge and thank the following people for their support and motivation during this project.

I would like to express my sincere gratitude and appreciation towards my supervisor **Prof. John Rowley**, without whose guidance and mentoring I would not have been able to complete this project. His words of constant encouragement and meticulous reviews have helped me improve the work and its quality in many aspects. Short and succinct meetings with him has played a key role in this project and has helped me stay on top of the schedule.

I would also like to thank **Dr. Shazia A Afzal**, the project head for her support for her guidance and suggestion that led to the completion of the project.

I would also like to thank everyone in Dublin Business School, for their kind support. All my lecturers and academic staffs have been a tremendous help. This journey would have not been possible if it wasn't for your support.

Finally, I would like to thank my friends for their constant motivation and my family for their full-fledged support, without which pursuing this degree wouldn't have been possible.

Table of Contents

Chapter 1: Introduction

The study about opinions of people, emotions or attitude towards an event or entity is called Sentiment Analysis or Opinion mining. These sentiments can help to understand the feeling and impact of an event or entity on the audience. In this era of internet and social media everyone is extremely vocal about their thoughts and emotions. There are multiple ways to express their concerns like in form of reviews or comments and ratings. Reviews and comments are textual in nature whereas ratings are numeric form of data. Since, reviews give more details it is more efficient to understand the sentiment of the user or customer along with the underlying reason. These revelation of sentiment and opinion of users and customers can help businesses, government and other customers as well. The reviews will act as a feedback for the government and businesses to improve or modify their products and services. It also helps to understand the impact of their work on the masses. New customers can be influenced based on the reviews, opinion and sentiment projected by old customers.

The aim of this project is to create and compare models using machine learning to analyse the sentiments of IMDb users based on their reviews. IMDb dataset is a collection of reviews and sentiments where the sentiment is an emotion identified from the review which can be either positive or negative. The scope of this project is to apply data pre-processing techniques to clean the data.

Chapter 2: Background/Literature Review

In a paper by Joscha, Stefan and Helmut [1] they developed and performed a comparative study on different techniques to improve the performance of sentiment analysis like Bag of words model, for using semantic information used n-grams. The Bag of words technique does not consider any association between sentences or documents parts semantically.

Using various techniques like objectivity or subjectivity analysis, feature extraction and summarizing the reviews were designed and discussed by Ahmad Kamal on opinion mining framework. [2] For subjectivity and objectivity, supervised machine learning approach was used for review classification. Other supervised machine learning algorithm used are Naïve Bayes, Decision Tree, Multilayer Perceptron and bagging.

Pang et.al. [3] classified sentiments based on categorization, as either positive or negative sentiment. Using n-gram technique multiple machine learning algorithms were used to model the experiment, i.e., Support Vector Machine, Naïve Bayes Classification and Maximum Entropy Classification.

In a paper by Turney [4], thumbs up and thumbs down are used to classify as recommended or not recommended using unsupervised algorithms. Parts of Speech (POS) tagger works as a phrase identifier which contains adverbs and adjectives.

Structured reviews are used to train and test, feature identification and scoring techniques to determine whether the reviews are positive or negative by Dave in this paper. [5] Data fetched from web search engines by querying and applying product name as search conditions uses classifier to classify.

In a paper by Pang and Lee [6], they used subjective and objective as labels for the documents. To prevent the useless and misleading data affecting the polarity of subjectivity group, machine learning classifier were applied. The paper also discusses about the various extraction methods which were explored based on minimum cut formula.

Whitelaw et.al. [7] discussed a classification method for sentiments by analysing and extracting appraisal groups. Appraisal groups describes a groups of attribute values in task independent semantic taxonomies.

To solve the issue of unbalanced set of labelled data, a semi-supervised method for sentiment classification was proposed by Li in this paper [8]. Using under sampling technique they dealt with the problem of imbalanced sentiment classification.

One of the feature reduction technique based on variance mean was proposed by Wang and Wang [9] is used for filtering the features and reducing the number of dimensions for representational phrase of text classification. [9] This feature filtering method proved to be of great importance as it took only the best features into consideration, improved the overall model performance and reduced the computation time as the incoming text was classified automatically.

Chapter 3: Requirements Specification and Design

In this project, a dataset is taken from source[10]. It contains 50,000 movie reviews with sentiment as positive or negative. The dataset is balanced with 25K reviews of each sentiment polarity. For the ease of processing we have randomly selected 10K reviews with balanced set of 5K reviews of each sentiment polarity.

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. . .The filming...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Petter Mattei's "Love in the Time of Money" is...	positive
4	Probably my all-time favorite movie, a story o...	positive
5	I sure would like to see a resurrection of a u...	positive
6	If you like original gut wrenching laughter yo...	positive
7	This a fantastic movie of three prisoners who ...	positive
8	Some films just simply should not be remade. T...	positive
9	I remember this film,it was the first film i h...	positive
10	After the success of Die Hard and it's sequels...	positive

Figure 1: Snapshot of dataset

The dataset consists of 2 columns, i.e., review – which contains the textual opinion of the user and sentiment – which represents the corresponding review's emotion as either positive or negative.

Figure 2: Donut chart of distribution of reviews (50-50)

The donut chart displays that the number of positive and negative reviews are equal.

The high-level design of the proposed approach is as follows:

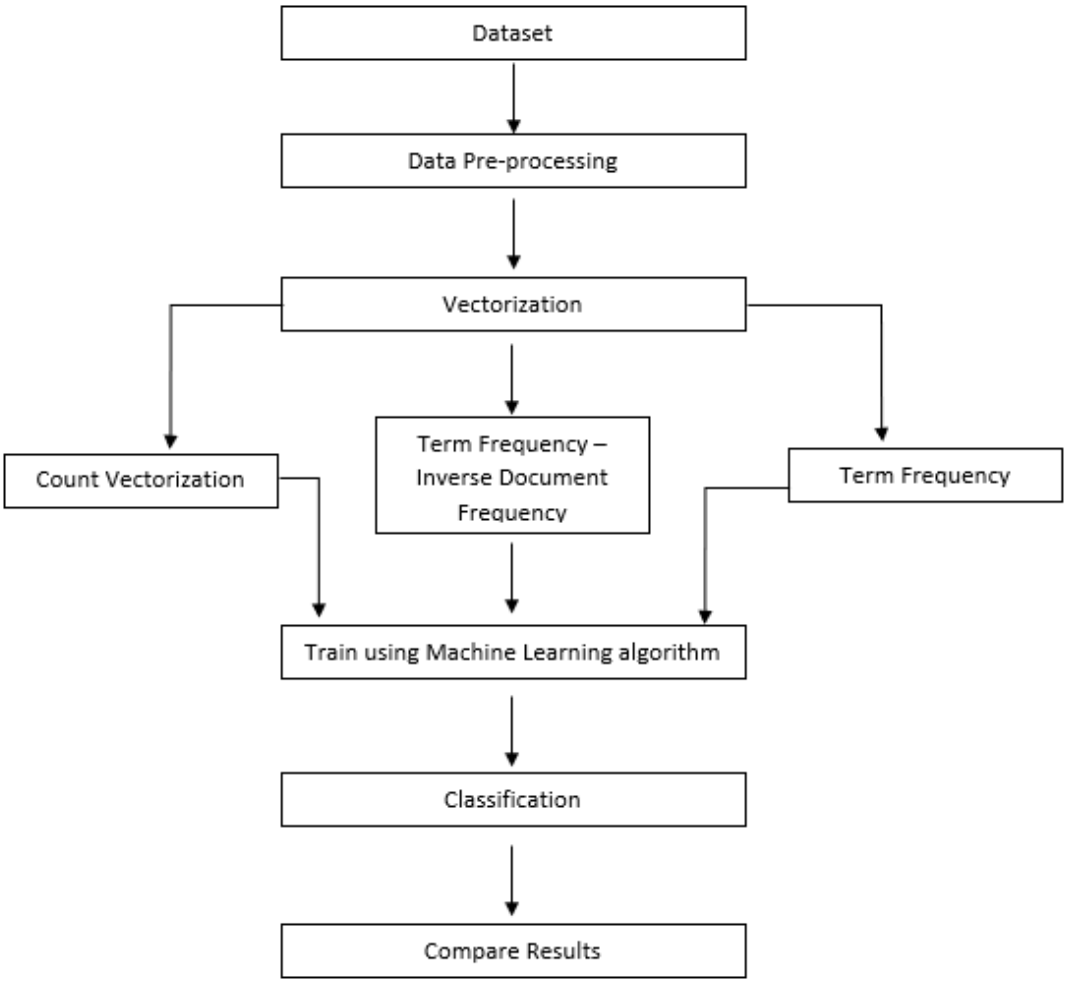

Figure: Diagrammatic view of proposed approach

To help the machine or computer understand the human's language, Natural Language Processing (NLP) is used.[11] Textual reviews needs to be processed using a series of data process techniques.

There are number of techniques for data pre-processing as listed below:

Lemmatization: Reducing various words into single terms.

Word segmentation: Dividing large sentences into smaller units.

Part-of-speech tagging: Identifying part-of-speech from words.

Parsing: Analysis of grammar of given sentences

Stemming: Getting the root term from the word

Lower-case: Converting all words to lower case for uniformity

Stop-word filtering: To remove words which are redundant and add no value

After processing data, a corpus is created which contains clean data, no unwanted values are present in this corpus.

Once the data is clean, we need to go ahead and create our machine learning models. But in case of textual reviews we need to go through a process of creating word embedding or vectorization. Word embedding can be explained as a real number, vector representation of a word. Word embedding could capture any relationship in that space like meaning, context, morphology or any other relationship. [12] The reason to use word embedding is to make the computer or machine understand the data being fed. Words are not compatible for machines. Using word embeddings makes the task of machine learning easier.

The following word-embedding or vectorization techniques are used in this project:

1. Count Vectorization

This method is also called as One-Hot Encoding. It is the simplest technique of representing words into numbers. The idea here is to create a vector which consists of all unique words present in the document. Each unique word has unique dimension and will be identified by a 1 in that position and 0s in all other positions. It does not hold any relationship information.

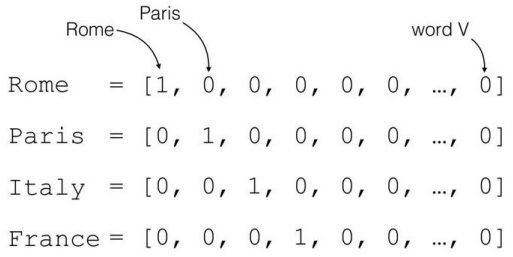

Figure 3: Count Vectorization Example

2. Term Frequency

This method helps to count the frequency of each term in given document. This technique is highly dependent on the length of the document and on how common the word is. Here we find the probability of the term appearing in the document by dividing the number of occurrences by the total number of words in the document. This helps to understand how impactful the word is in each document.

Term frequency is at document-level; hence we can formulate it as follows:

```
TF (t) = (Number of times term t appears in a document) / (Total
number of terms in the document) [13]
```

3. Term Frequency – Inverse Document Frequency (TF-IDF)

TF-IDF is related to one-hot encoding. In this technique, instead of counting features to know if they are present or absent, this method represents words by their term frequency multiplied by inverse document frequency. It means, words that appear everywhere are given less importance and less weightage. Example of such words are ‘the’, ‘and’ in case of English language. They do not add any large amount of value. However, if a term appears less frequently then the significance of those words is higher.

This method helps to find how common or rare a term is in each text or document.

```
tf(t,d) = count of t in d / number of words in d

df(t) = occurrence of t in documents
tf-idf(t, d) = tf(t, d) * log(N/(df + 1))[14]
```

The dataset available here consists of target variable as labelled data - positive or negative hence supervised machine learning algorithms can be used to classify the sentiment for a review. Then for each vectorization technique we will apply various supervised machine learning algorithms to check for performance and compare if one is better than another. The various algorithms taken into consideration are as follows:

- Random Forest
- Multinomial Naïve Bayes Classification
- Bernoulli Naïve Bayes Classification
- Decision Tree
- Neural Network
- Logistic Regression

Support Vector Machine is very time consuming hence it was not taken into consideration.

The project problem here is to identify a machine learning model which works well to identify the sentiment and give most efficient result. Along with the machine learning model we also need to find the best vectorization method which helps the machine learning model to give the most accurate results. The information required to find from this project is the best vectorization technique and the performance of the algorithm.

The functional requirement to achieve the target is knowledge of Natural Language Processing and machine learning models. Using Python programming language and Jupyter notebook to code we can achieve this.

Chapter 4: Implementation

The implementation of this project requires some basic system (hardware and software) requirements to be fulfilled.

Recommended System Requirements:

- Processors: Intel® Core™ i5 processor 4300M at 2.60 GHz or 2.59 GHz (1 socket, 2 cores, 2 threads per core), 8 GB of DRAMIntel® Xeon® processor E5-2698 v3 at 2.30 GHz (2 sockets, 16 cores each, 1 thread per core), 64 GB of DRAMIntel® Xeon Phi™ processor 7210 at 1.30 GHz (1 socket, 64 cores, 4 threads per core), 32 GB of DRAM, 16 GB of MCDRAM (flat mode enabled)
- Disk space: 2 to 3 GB
- Operating systems: Windows® 10, macOS*, and Linux*
- Python* versions: 2.7.X, 3.6.X.

Steps :

1. **Importing libraries :**

   For the purpose of Sentiment Analysis and Natural Language Processing various libraries are required. All these libraries need to be imported in the current code before using them. The following are the libraries used so far:

   i) numpy
   ii) panda
   iii) re – Regular Expression used for cleaning data
   iv) nltk – Natural Language Toolkit for language processing
   v) nltk.corpus.stopwords – Stop words to be removed from the data
   vi) nltk.stem.porter. PorterStemmer – Stemming is removing morphological affixes from words
   vii) sklearn. feature_extraction.text. CountVectorizer – Helps to create token counts from the given collection of documents
   viii) pickle – Used for object serialization

2. **Importing dataset :**

   The dataset IMDb reviews is selected and loaded into Python environment. The dataset contains 2 columns – ‘review’ and ‘sentiment’. The ‘review’ column contains 10K records of textual reviews each corresponds to a sentiment as either ‘positive’ or ‘negative’.

3. **Pre-processing textual data:**

   Data may contain various characters which are either unimportant or misleading for the model. We need to create bag-of-words which will contain all the features from the text document. With the help of these features we can create feature vectors which in turn can be used to train the machine learning algorithm.

   In this case, we have used the following data cleaning techniques:

   i) Discard unwanted characters : Using Regular Expression library, we substring the data which contains only alphabets and no special characters or numbers.
   review = re.sub('[^a-zA-Z]', ' ', dataset['review'][i])

ii) Convert to Lowercase : All the text needs to be converted to lowercase to normalize the data.
review = review.lower()

iii) Tokenization : The process of separating sentences into individual words is called tokenization. The default separator is space (‘ ’)
review = review.split()

iv) Stemming: The process of reducing the word to their root form[2]
ps = PorterStemmer()

v) Stopword removal: Using *nltk.download(‘stopwords’)* a dictionary of all the possible stopwords is downloaded. This dictionary contains stopwords in various languages, we can select the language based on the data available and remove stopwords from the content. In the current scenario, language used is English, hence we select
set(stopwords.words('english'))
The data created after stemming is reviewed to check for stopwords and removed if found.

```
review = [ps.stem(word) for word in review
            if not word in set(stopwords.words('english'))]
```

vi) Create corpus: Once all the necessary text processing is done, we join the string array to create string and add it to the corpus. Now the corpus contains all clean, stemmed, lowercase, text data.

4. Create Word embeddings or Vectorization :

The clean corpus is used to select top features which are basically the most significant words in the corpus. The various techniques discussed previously are used here to create dimensions or matrices which consist of numerical representation of the word.
The feature selection is limited to 15K words.

Count vectorization:

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 15000)
```

Term Frequency (TF):

```
from sklearn.feature_extraction.text import TfidfTransformer
tf = TfidfTransformer(norm='l2', use_idf=False, smooth_idf=True, sublinear_tf=False)
Xft = tf.fit_transform(X).toarray()
```

Term Frequency – Inverse Document Frequency (TF-IDF):

```
from sklearn.feature_extraction.text import TfidfTransformer
tf = TfidfTransformer(norm='l2', use_idf=True, smooth_idf=True, sublinear_tf=False)
Xtf = tf.fit_transform(X).toarray()
```

5. **Convert sentiment values into numerical values:**

   This is important as few machine learning algorithms do not accept categorical target variable.

   ```
   sentiment = {'positive': 1,'negative': 0}
   y = [sentiment[item] for item in y]
   ```

6. **Save files using Pickle library:**

   Using pickle library, the generated vectorized corpus is exported in a file. It saves time later to use the dataset as the serialized version is already available.

   ```
   pickle_out = open("X_rr_cv.pickle","wb")
   pickle.dump(X, pickle_out)
   pickle_out.close()

   pickle_out = open("y_rr_cv.pickle","wb")
   pickle.dump(y, pickle_out)
   pickle_out.close()
   ```

7. **Model building:**

   To build the machine learning model we need to divide the dataset into two parts – training set and test set. The train set is used for making the model learn patterns and the test set acts as unseen data and is used to test the model performance and validate the model accuracy. The train and test set can be divided in any ratio as per the developer. In this case, we divide the train and test set in the ratio of 80:20.

   To validate the model, we use the test set where the target variable - labelled data is present. The significance of output variable being available is that it could be compared with the predicted value which will help to evaluate the model performance like accuracy of the model. Accuracy is the percentage of correctly predicted values. Because of this accuracy value it can be checked if the model is performing well or if not if it is underfitting or overfitting.

   Underfitting refers to a model that fails to model the training data and is not useful to generalize the new data. Underfit model is not suitable and will have poor performance on the training data.

   We cannot work with overfitting or underfitting models. Hence, to make sure the model is well built we perform evaluation of the machine learning model. There are various cross validation techniques which can be adopted to scrutinize the model performance of a supervised machine learning algorithm, like k-fold cross validation, which is used in this project.

   The k-fold cross-validation technique used here is with k = 10. This technique helps to split the data in different combinations to make sure no pattern is missed out and the model is not mugging up data.

   Other performance evaluation techniques apart from cross validation comes from confusion matrix.

Confusion matrix is a tabular representation of the performance of the classifier. The matrix consists of the relationship between all the correct and incorrect predictions.

	Correct Labels	
	Positive	Negative
Positive	True Positive(TP)	False Positive(FP)
Negative	False Negative(FN)	True Negative(TN)

Figure 4: Confusion Matrix

Using this confusion matrix multiple performance evaluation parameters can be calculated. The parameters like precision, recall, F-measure and accuracy.

Precision: This parameter defines the exactness of the classifier. It is the ratio of correctly predicted positive (TP) to total number of positives (TP+FP)

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

Recall: This parameter defines the completeness of the classifier. It is the ratio of correctly predicted positive (TP) to actual number of positives (TP + FN)

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

F-measure: This parameter is the weighted average of Precision and Recall. The best value for this parameter is 1 and the worst being 0.

$$F - Measure = \frac{2*Precision*Recall}{Precision+Recall}$$

Accuracy: It is the most common performance evaluation technique. It is the ratio of correctly predicted values to total number of values available.

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

Chapter 5: Testing and Results

The textual reviews are processed to identify and understand the words in the reviews. Using Word cloud visualisation technique, we can find the top 'n' number of words and the based on their relative size we can understand the frequency of the words. The following word cloud consists of the top 100 words in the dataset and it is clear that the terms *movi, film, one, time, make, good* have much higher frequency compared to words like *watch, love, seem, end, first, plot.*

Figure 5: Word cloud of top 100 words

The vectorization techniques were used to create different matrices, each of those were used to create machine learning models.

The following charts are comparison between the algorithms using the similar vectorization techniques.

1. **Term Frequency:**

The following chart shows that the algorithm accuracy ranges from 70% to 86%. The highest accuracy being 86.3% for Logistic Regression. Followed by Random Forest and Multinomial Naïve Bayes with 85.75% and 85.65% respectively.

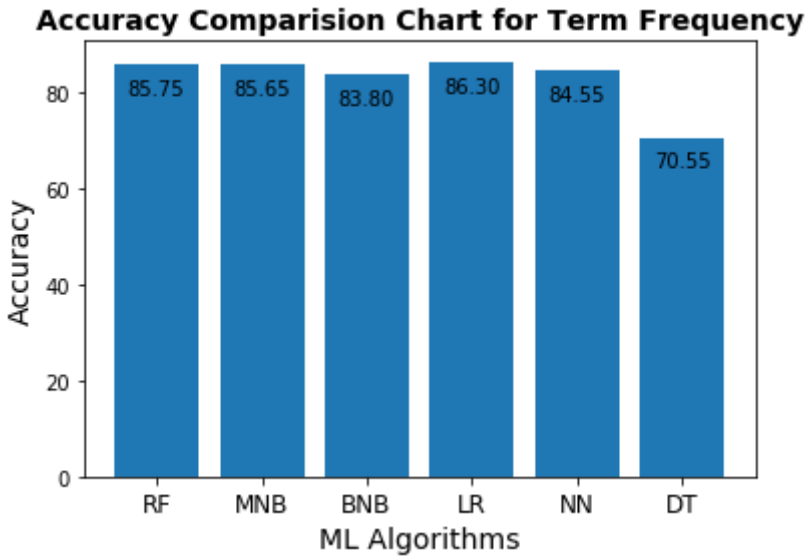

Figure 6: Term Frequency – Accuracy

To make sure the model is neither underfitting nor overfitting, k-fold Cross Validation evaluation technique is used. The comparison of model accuracy with and without cross validation is depicted in the chart below. From the following grouped bar-chart it is visible that all the models are good neither underfit nor overfit.

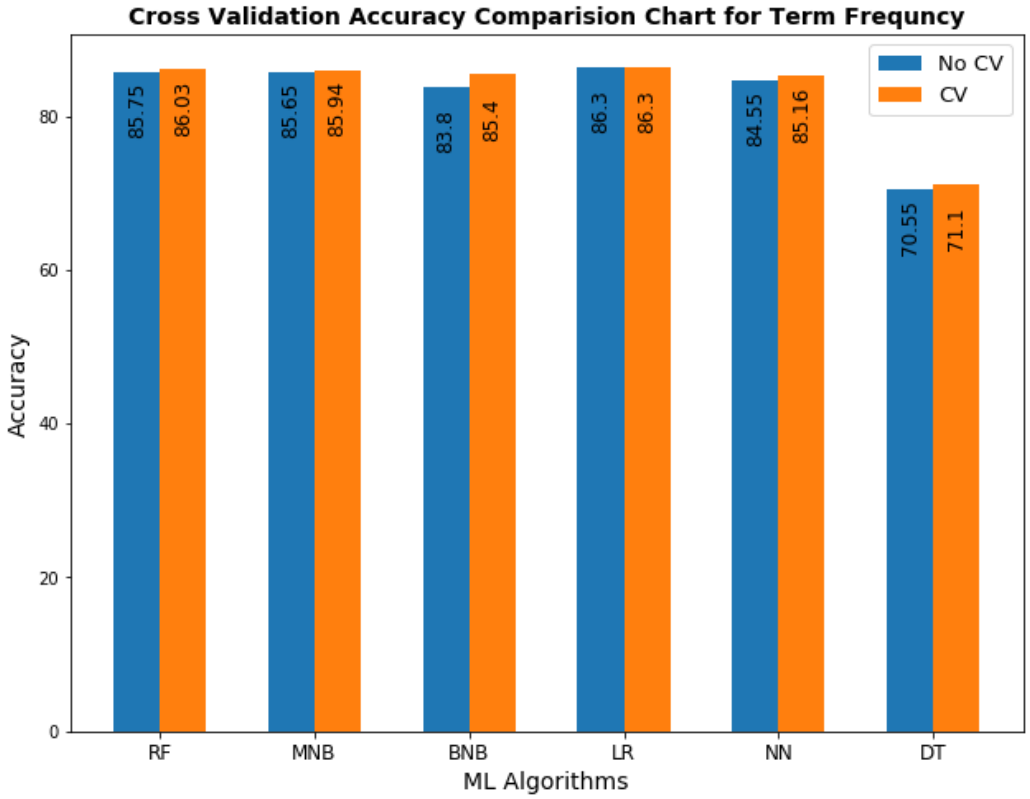

Figure 7: Term Frequency – Cross Validation Accuracy

2. Count Vectorization:

The following chart shows that the algorithm accuracy ranges from 69% to 86%. The highest accuracy being 86.35% for Logistic Regression. Followed by Random Forest with 85.85%. Decision tree having least accuracy of 69.5%.

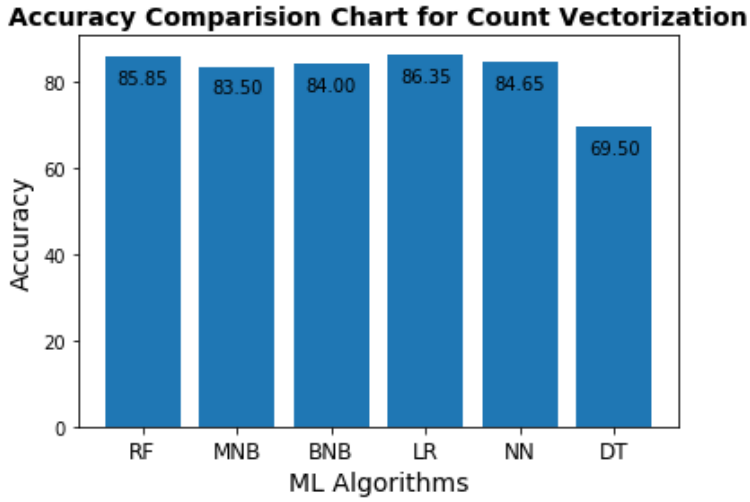

Figure 8: Count Vectorization - Accuracy

The cross-validation accuracy reveals the accuracy levels are pretty good. No model is an overfit.

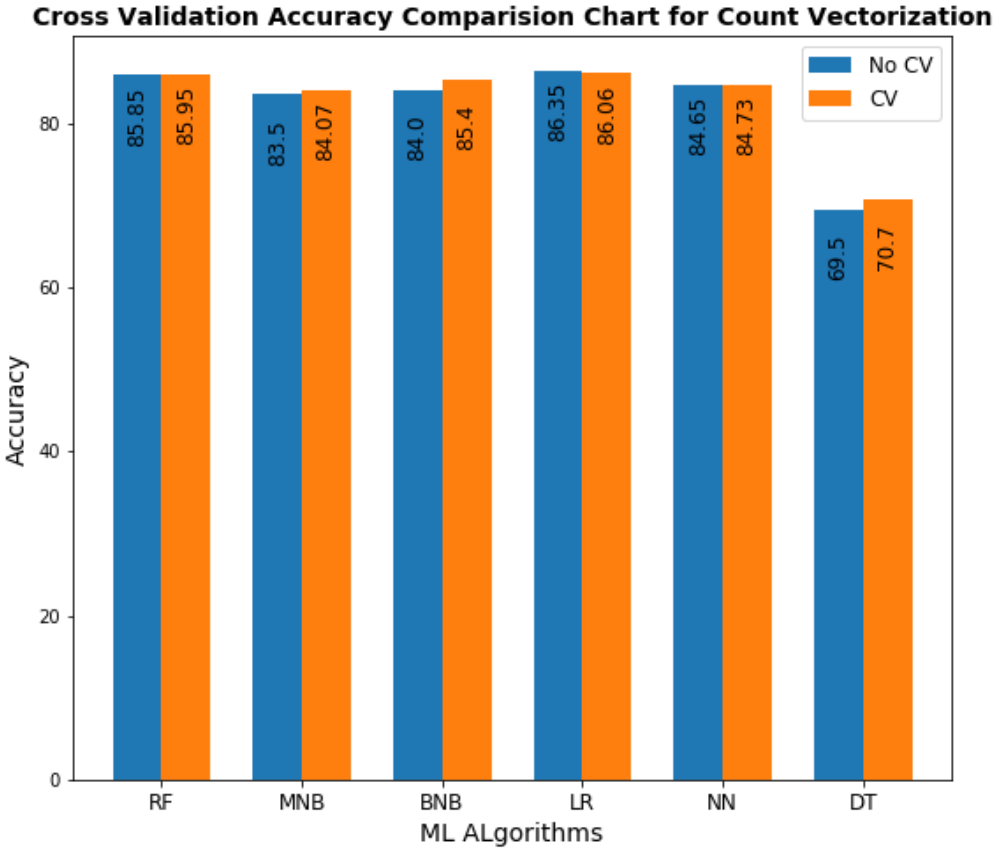

Figure 9: Count Vectorization – Cross Validation Accuracy

3. Term Frequency - Inverse Document Frequency:

The following chart shows that the algorithm accuracy ranges from 70% to 87%. The highest accuracy being 87.6% for Logistic Regression. Followed by Neural Network and Random Forest with 86.45% and 86.15% respectively. Decision tree having least accuracy of 70.8%.

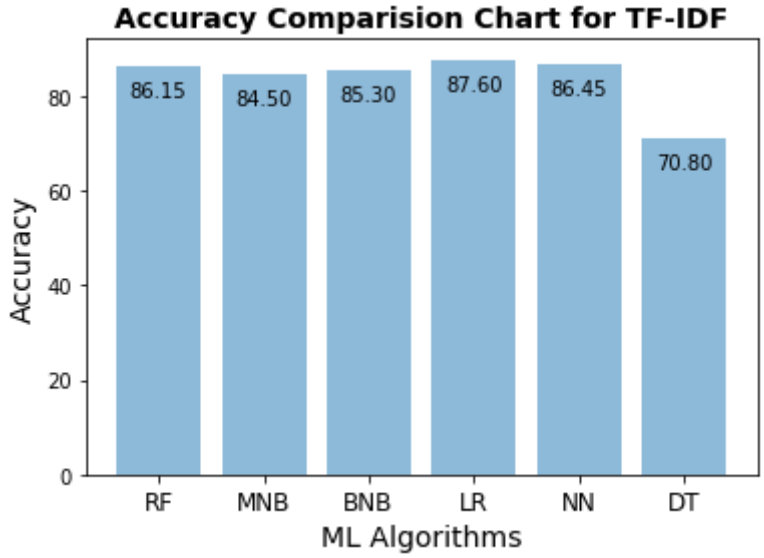

Figure 10: Term Frequency-Inverse Document Frequency - Accuracy

The cross validation in case of TF-IDF gives decent results. The models are well fit.

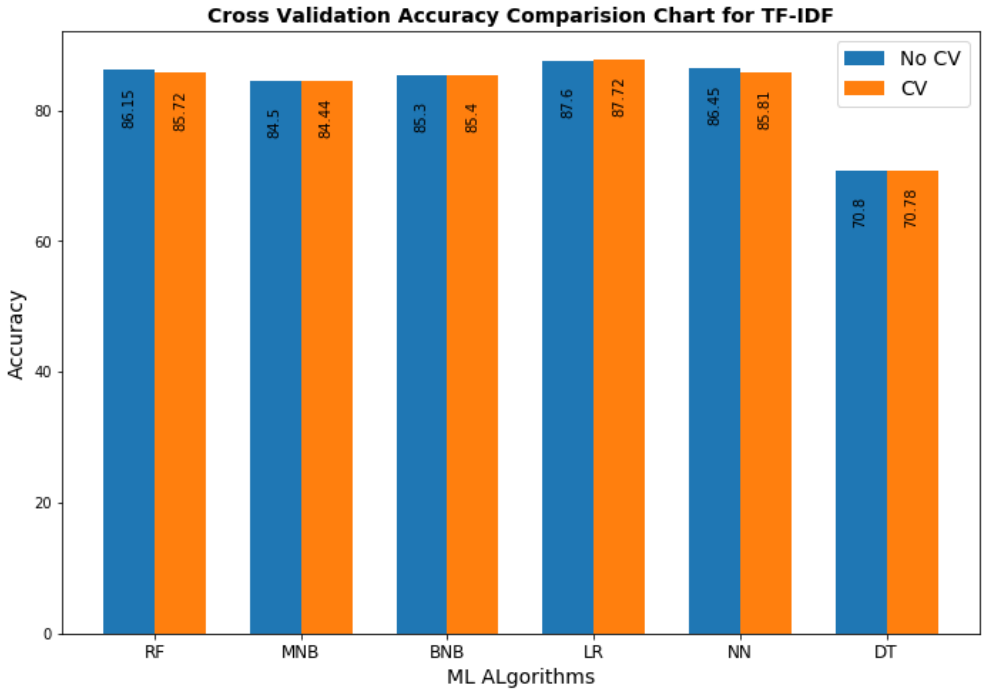

Figure 11: Term Frequency-Inverse Document Frequency - Accuracy

The following chart compares all the models for accuracy.

Logistic Regression on TF-IDF outperformed other machine learning algorithms.

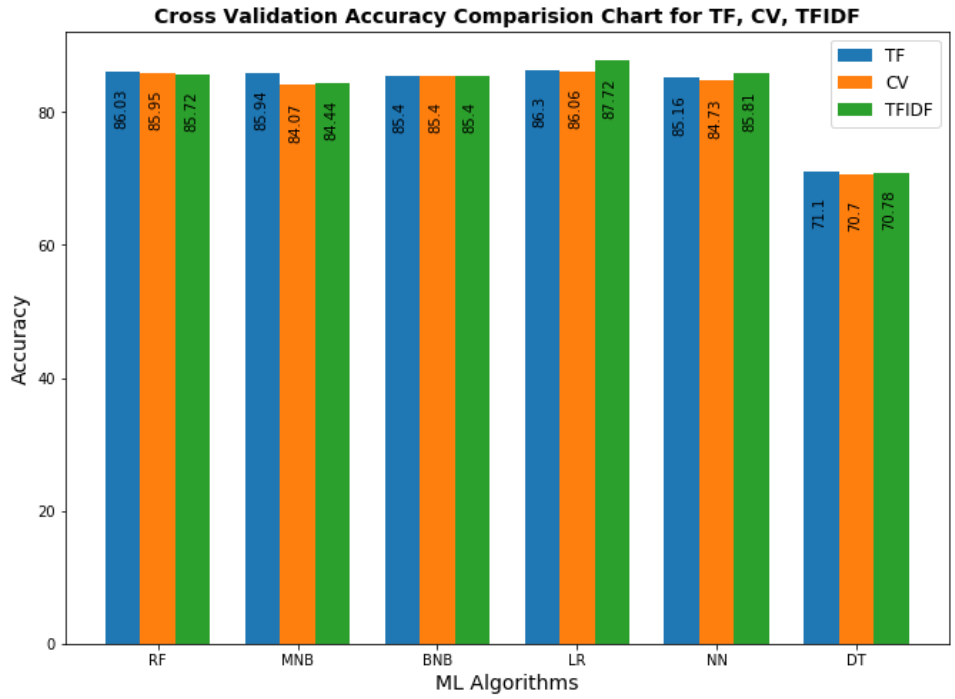

Figure 12: Term Frequency-Inverse Document Frequency - Accuracy

Chapter 6: Conclusions and Future Work

This paper analyses sentiments of movie reviews. A comparative study is conducted on various machine learning algorithms which are modelled on top of various vectorization techniques to find which works best in terms of performance and accuracy. Sentiment analysis using Natural Language Processing by implementing various machine learning techniques has been studied in this paper by assessing movie reviews.

The comparative study of the vectorization technique shows that Decision tree performed consistently poor across all vectorization techniques. Bernoulli Naïve Bayes was unaffected by the input, as it performed equally well with all three techniques. Other algorithms had a slight difference in accuracy but still close. The one which outperformed every technique with highest accuracy score is Logistic Regression with Term Frequency – Inverse Document Frequency.

The future scope of this project is to fine tune the models and implement the best sentiment analysis technique to find the emotion from the review. It could be implemented in the website itself to visualize and give an overall view of the reviews. As many people do not have time to check the reviews the implementation of the sentiment analysis and visualization techniques would ease the process for both, the users and the business.

In future, other models could be built and compared which would consider the intent of the review to make the analysis more reliable and sophisticated.

References

1. Joscha Markle-Huß, Stefan Feuerriegel, Helmut Prendinger. 2017 Improving Sentiment Analysis with Document-Level Semantic Relationships from Rhetoric Discourse Structures, Proceedings of the 50th Hawaii International Conference on System Sciences.
2. Kamal A., 2015, Review Mining for Feature Based Opinion Summarization and Visualization.
3. B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002, pp. 79–86.
4. P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002, pp. 417–424.
5. K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in Proceedings of the 12th international conference on World Wide Web. ACM, 2003, pp. 519–528.
6. B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in Proceedings of the 42nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2004, p. 271.
7. C. Whitelaw, N. Garg, and S. Argamon, "Using appraisal groups for sentiment analysis," in Proceedings of the 14th ACM international conference on Information and knowledge management. ACM, 2005, pp. 625–631.
8. S. Li, Z. Wang, G. Zhou, and S. Y. M. Lee, "Semi-supervised learning for imbalanced sentiment classification," in IJCAI Proceedings- International Joint Conference on Artificial Intelligence, vol. 22, no. 3, 2011, p. 1826.
9. Y. Wang and X.-J. Wang, "A new approach to feature selection in text classification," in Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on, vol. 6. IEEE, 2005, pp. 3814–3819.
10. http://ai.stanford.edu/~amaas/data/sentiment/
11. https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32
12. https://towardsdatascience.com/introduction-to-word-embeddings-4cf857b12edc
13. https://www.opinosis-analytics.com/knowledge-base/term-frequency-explained/
14. https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089

Appendix

YouTube Link – Interim Presentation and Demonstration:

https://www.youtube.com/watch?v=6pWRGoFrEYQ&feature=youtu.be

Python Code file – GitHub link:

https://github.com/Rajanprajapati1308/Sentiment-Analysis