Principles of Statistical Modelling

Project Report
On
Sentiment Analysis

By Sanyukta Suman

June 14, 2020

Abstract

The report presents the multi-class classification of movie reviews. The task is to make predictions of sentiment using different machine learning techniques. The first component of the report presents the description of the data acquisition process and the description of data. The second component is pre processing of text data into its numeric format. Due to the high number of features in the dataset, the feature extraction step is performed by analyzing the data distribution and correlation between the features. After the feature extraction phase, different types of Machine learning algorithm are used- they are Logistic regression, decision tree and random forest obtaining an average of 85% accuracy. Finally, conclusions are presented, opening the space for various approaches in the future.

Introduction

With the growth of web text data such as: online review data posted by users for hotel booking, e-commerce website and movie reviews, can be of great help to understand the business and the need of the user plays an important role in making decisions for companies [2]. The objective of this project is to use multi-class classification,instead of binary class classification (positive/negative) to predict the Phrases from the sentences of the movie review given user in the sentiment scale 0 to 4, where 0 is the lowest sentiment (negative) and 4 is the highest sentiment(positive). This project first introduces the description of data in mathematical form and also the description of the features of the dataset. It then describes one of the major tasks in sentiment analysis which is preprocessing text data into numeric data. Next, it focuses on analysis and distribution of the feature which helps in the next step which is feature extraction. Furthermore, it also introduces several machine learning methods such as logistic regression, decision tree and random forest used for classifying sentiments. Finally, the result of the machine learning is presented with comparison and suggests future direction for this project.

Data Description

The dataset is a collection of movie reviews from the website “www.rottentomatoes.com”. The dataset was provided by the website “www.kaggle.com”, originally collected by Pang and Lee. The dataset consists of Tab Separated files (tsv), which consist of phrases from the Rotten Tomatoes dataset. Here, each phrase has its phrase Id and each sentence has a sentence Id. Phrases which are repeated are only included one in the dataset. The source of the dataset is https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews/data.

Description and format

Description of dataset in mathematical correct formalism

1. **Universe Ω** = {Website (Rotten Tomatoes), User who is writing a review, Internet}
2. **Elementary Events ω=** The possibility of the user writing the review in the comment section.
3. **Measurable Function (RV-function)=** procedure of reading reviews given by the users and measuring the reviews according to the sentiment.
4. **Data Value Space=** {PhraseId, SentenceId, Phrase, Sentiment}

Format of the dataset

The dataset is divided into training and test data, represented by “train.csv” and “test.csv” files respectively. The RV-function of the dataset is a procedure of reading reviews given by the users and measuring the reviews according to the sentiment. Starting with the training dataset file, whose first line identifies the feature names followed by feature values. The feature name or the Data Value space (DVS) of the training dataset are PhraseId, SentenceId, Phrase and Sentiment. Table 1 shows a version of the data for the train.tsv.

PhraseId	SentenceId	Phrase	Sentiment
0	1	A series of escapades demonstrating the adage ...	1
1	2	A series of escapades demonstrating the adage ...	2
2	3	A series	2
3	4	A	2

Table 1: Excerpt of train.csv

Similarly, the test.tsv file is formatted using the same structure except for the Sentiment column, which is unknown. The purpose of this project is to predict the sentiment of the phrases from the model trained with the help of train.tsv where sentiment is known. Table 2 shows a lightweight version of the test.tsv.

PhraseId	SentenceId	Phrase
0	156061	An intermittently pleasing but mostly routine ...
1	156062	An intermittently pleasing but mostly routine ...
3	156063	An
3	156064	intermittently pleasing but mostly routine effort

Table 2: Excerpt of test.csv

The columns have the following meaning:

1. **PhraseId**: The ID of the Phrase.
2. **SentenceId**: The ID of the sentence, which helps to track the phrases taken from sentences.
3. **Phrase**: The phrases from the sentences written by the user in Rotten Tomatoes.
4. **Sentiment**: It is a label given to the phrases to convey sentiments. The sentiments range from 0-4. The sentiment labels are:

Labels	Sentiment
0	Negative
1	Somewhat negative
2	Neutral
3	Somewhat positive
4	Positive

Data Pre-processing

For the purpose of this project the data taken from train.tsv and test.tsv is of a shape of *100X4* and *100x3* respectively. The dataset is fairly clean with no missing values. For each phraseId there is a phrase, sentenceId and sentiment mapped to it in traiv.tsv file. Similarly, for test.tsv for each phraseId there is a phrase, sentenceId mapped to it.

Before preprocessing the data I used several statistical methods to understand the data. The number of each sentiment in the train.tsv file was visualized using a barplot. Figure 1 shows the barplot of the division of the phrase according to their sentiments.

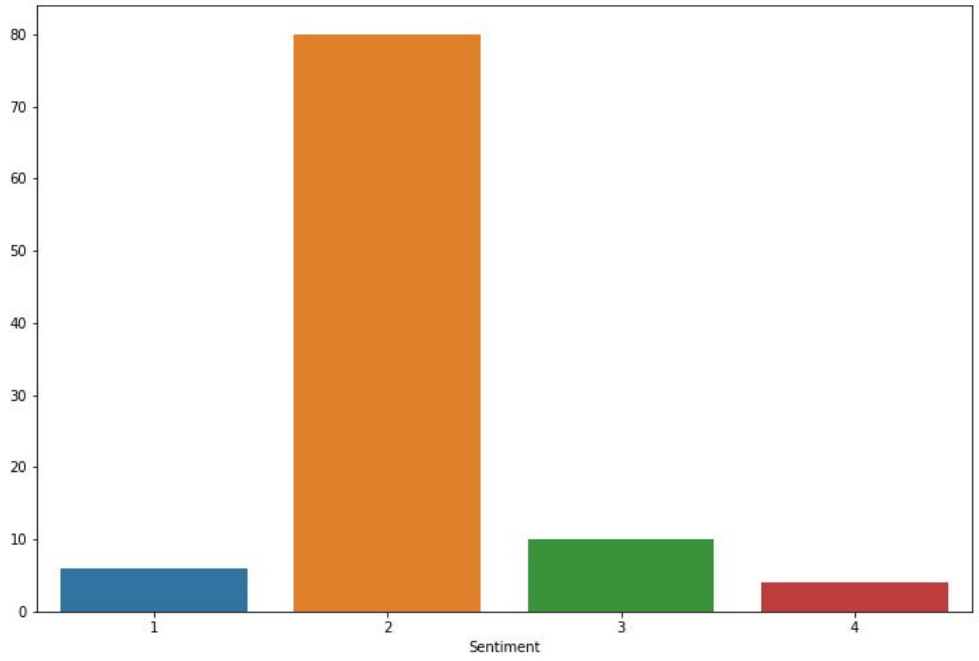

*Figure 1:*Barplot for sentiment count

According to the barplot, sentiment class seems to be following a normal distribution, with most of the frequently distributed class sentiment labelled 2 - which represent neutral from the range given.

One of the features in the dataset is “Phrase”, this feature stores data in the form of words. These words need to be tokenized into numeric format. Figure 2 shows the example of a phrase from the dataset.

```
Out[15]: 'A series of escapades demonstrating the adage that what is good for the goose is also good for the gander , some of which o
         ccasionally amuses but none of which amounts to much of a story .'
```

Figure 2: One of the phrase from the dataset

To begin with, in order to change the word to a numeric format, I used the Word2vec method. The word2vec method takes the corpus of text as its input and converts the text into a vector space with several dimensions. Words which are common in context in the corpus are located close to one another in a vector space. For example “Have a nice day.” and “Have a great day.” Here *great* and *good* will be placed closer in the vector space because they convey similar meaning in this context. Figure 3 shows the conversion of words into a vector space.

```
Out[7]: 0    [a, series, of, escapades, demonstrating, the,...
        1    [a, series, of, escapades, demonstrating, the,...
        2                                        [a, series]
        3                                                [a]
        4                                           [series]
        Name: Words, dtype: object
```

```
Out[10]: 0    [22, 34, 33, 47, 36, 29, 4, 40, 26, 1, 31, 3, ...
         1    [22, 34, 33, 47, 36, 29, 4, 40, 26, 1, 31, 3, ...
         2                                          [22, 34]
         3                                              [22]
         4                                              [34]
         Name: Tokens, dtype: object
```

Figure 3: From word to a vector conversion using word2vec.

The frequency of the words present in the phrase column in the train.tsv is shown in figure 4.

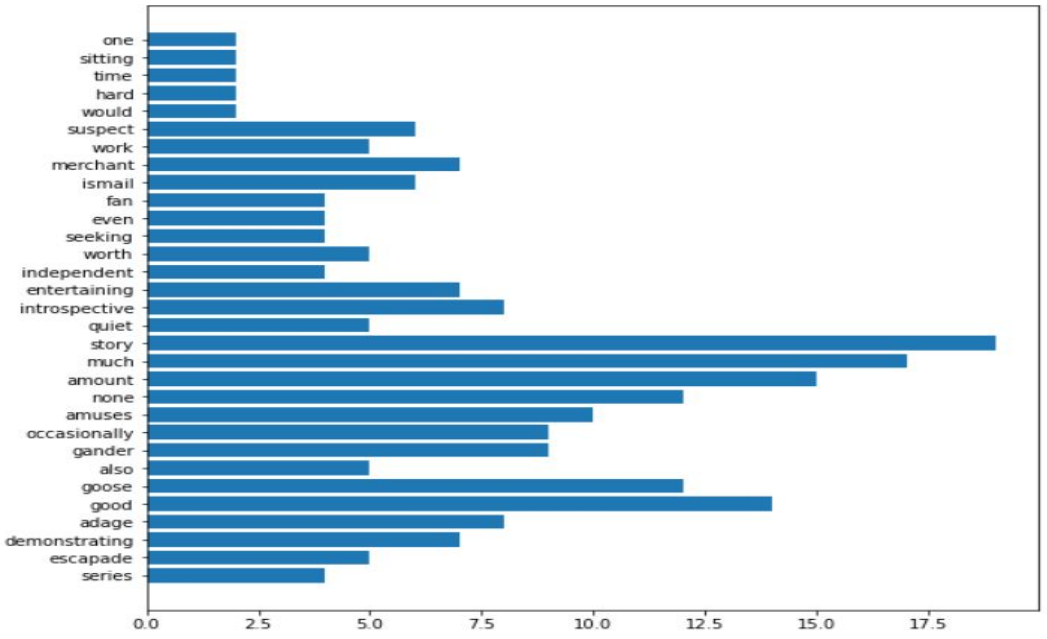

Figure4: Word frequency of training dataset

Similarly, The frequency of the words present in the phrase column in the test.tsv is shown in figure 5.

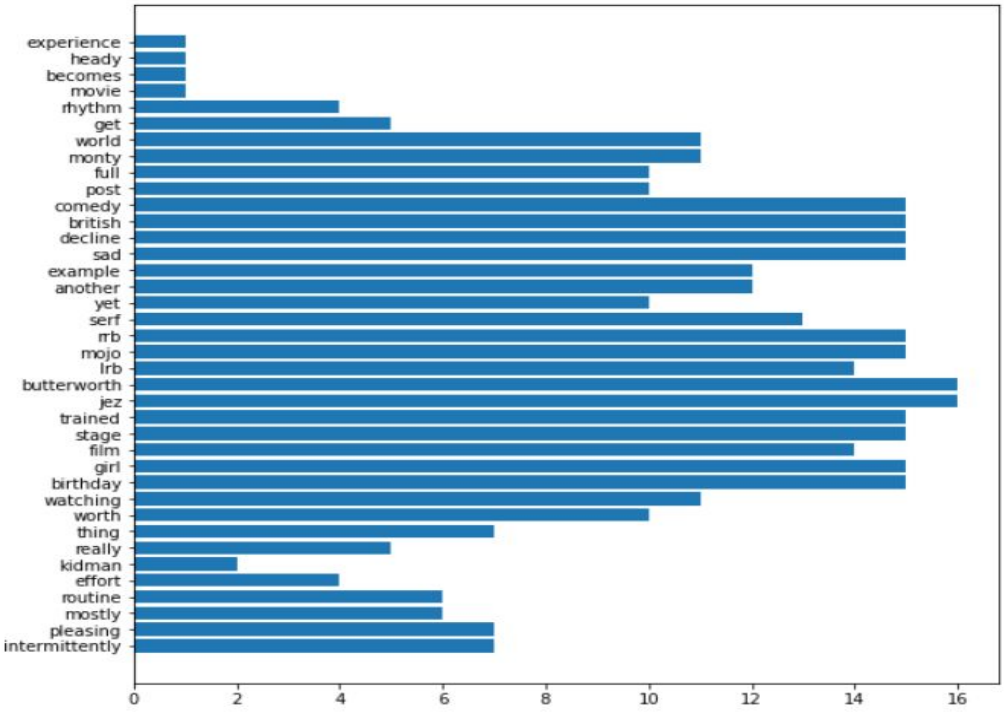

Figure 5: Word frequency for testing dataset

At this point we can visualize the frequency of the words in the phrase. However, we still do not know the sentiment of the phrases, since the sentiment of the phrases, also the number of features after converting word into its numeric format has increased drastically. Therefore, to understand the relationships between the features I analyzed the correlation between words. Figure 6 shows the graph for correlation of words with each other.

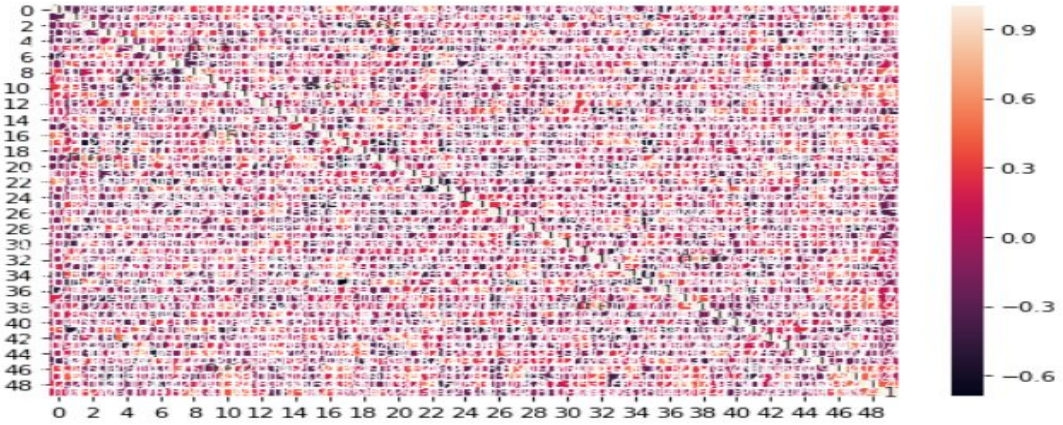

Figure 6: Correlation between words

We see that in Figure 6 the correlation between words are difficult to interpret, also it will affect the machine learning models's performance. Therefore, the next step is to reduce the dimension. Here, in this project to understand the data better, I used an algorithm called t-SNE, which is an effective algorithm suitable for dimension reduction for word embedding and also sued for visualization of high dimensional datasets and also visualization of the similar words clustered together in the graph which will give us an idea about the sentiment

of the phrase profoundly. Figure 7 shows the t-SNE visualization of a word *"Good"* and the words which are closer to this word.

Figure 7: t-SNE visualization for Good.

Machine Learning

Logistic regression approach

Logistic regression is a simple classification technique, it is a common and useful regression method for solving binary classification problems [3].Here, I fit the model on the training dataset and performed prediction on the test set, the accuracy of this model was 83%. Figure 8 shows the plot for the predicted result from the model.

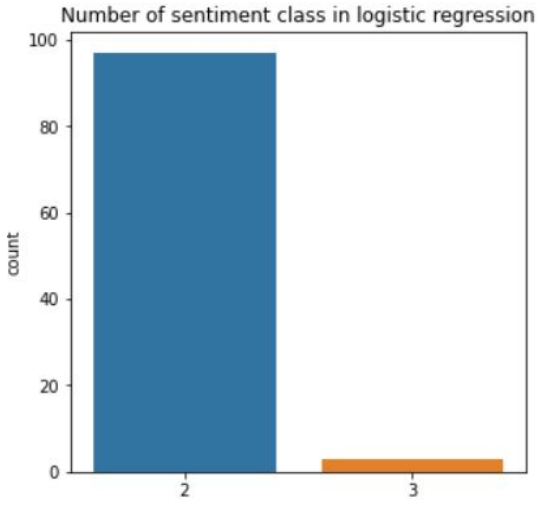

Figure 8: Prediction result for Logistic regression

Decision tree model

Decision tree model is another model for classification and is capable of both binary and multiple class classification. The goal of using decision trees is to create a model that predicts the value of sentiment for the test dataset by learning simple decision rules inferred from the training dataset [4]. The accuracy of this model was 99%. Figure 9 shows the plot for the predicted result from the model.

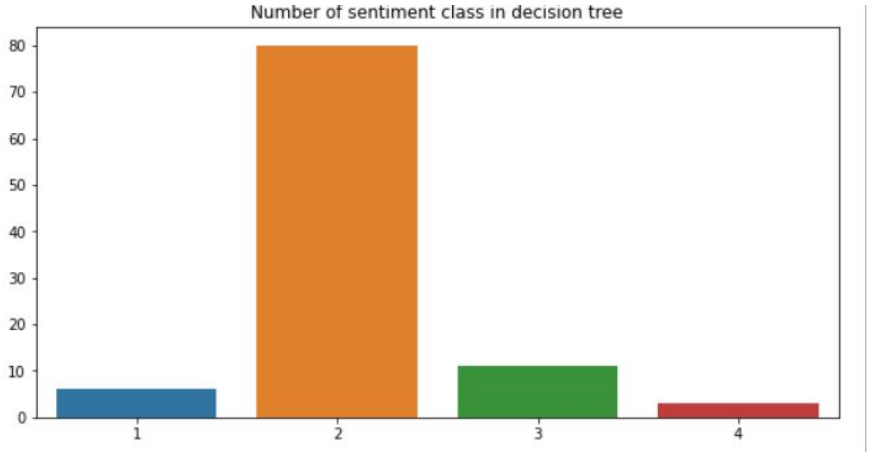

Figure 9: Plot result for Decision tree

Random Forest Approach

Random forest consists of a large number of decision trees that operate on ensembles. In this model each individual tree runs its class prediction and the class with most common votes becomes the prediction of the model [5]. In our dataset based on the number of classes in the training dataset yields accuracy of 98%. Figure 10 shows the prediction result for the model.

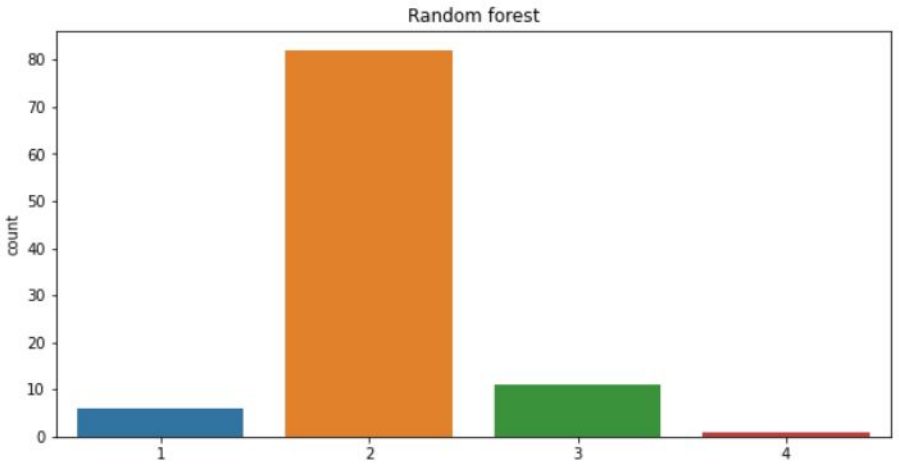

*Figure 10:*Prediction result for decision tree

Lastly, by comparing the result of three different approaches : Logistic Regression, Decision tree and random forest. By training a data set for 100 rows, we see that the majority of the prediction shows that phrase has sentiment class 2, which represents “Somewhat negative” according to the labels given to the sentiments. Figure 11 shows overall prediction for each model.

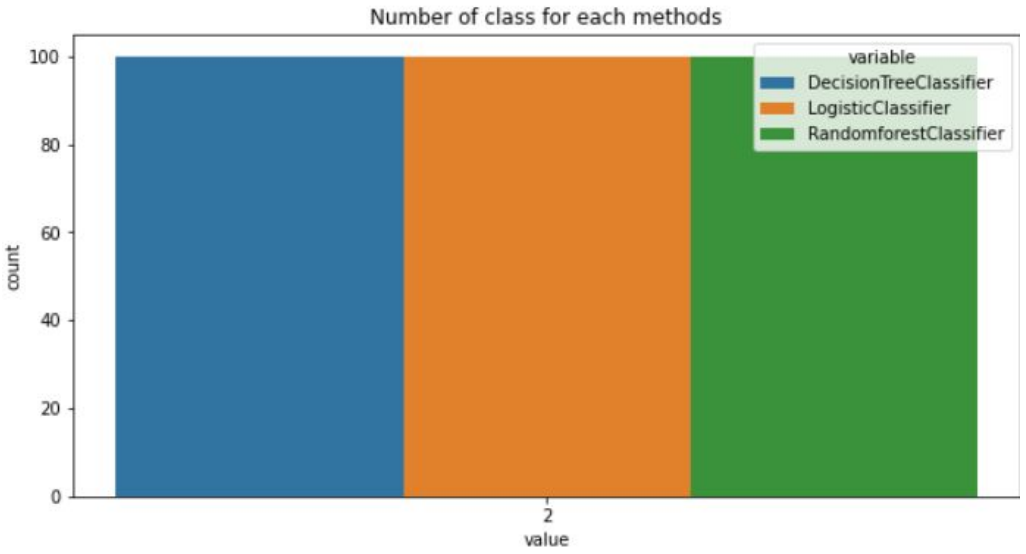

Figure 11: Result for each method

Conclusion

This report concludes by encompassing the basic steps of statistical learning, such as collecting data, cleaning the data, preprocessing data which could be fit for the model, analyzing data distribution and finally using machine learning algorithms to make better prediction. Defining data samples in the form universe, event, RV-function and data value space helped to understand the fundamentals of the dataset and then by analyzing data distribution, frequency of the word and correlation among the features helped to understand the data in a deeper and meaningful way.

Specifically, data preprocessing step where words had to be converted into numeric format using word2vec method played an important role in classification of sentiment class. Using logistic regression, decision tree and random forest as classification problems can prove to be beneficial for text analysis and sentiment analysis.

Finally, the model accuracy was around 80-90% in all three models. In both training and testing dataset, the variation in the sentiment was not diverse, which led the models to overfit the prediction.

Reference

[1] Data source- https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews/data

[2] Zhou, Li-zhu, Yu-kai He, and Jian-yong Wang. "Survey on research of sentiment analysis." *Journal of Computer Applications* 28.11 (2008): 2725-2728.

[3] Kleinbaum, David G., et al. *Logistic regression*. New York: Springer-Verlag, 2002.

[4] Kothari, R. A. V. I., and M. I. N. G. Dong. "Decision trees for classification: A review and some new results." *Pattern recognition: from classical to modern approaches*. 2001. 169-184.

[5] Biau, GÃŠrard. "Analysis of a random forests model." Journal of Machine Learning Research 13.Apr (2012): 1063-1095

Code Repository

All the code implemented and utilized during the execution of the process described in this report is available at the GitHub repository:
https://github.com/sumansanyukta/statistical-learning