Mendel
University
in Brno

FACULTY OF BUSINESS AND ECONOMICS

# PREDICTING THE RATING OF A MOVIE BASED ON ITS REVIEW

BY:
EUGENE ADU-GYAMFI

Supervised by: Roman Valovic

04/06/2021

## Table of Contents

# ABSTRACT

The project investigates reviews of movie goers whether positive or negative while considering sentiments in the data as well. Three classifiers were chosen and pegged against a baseline classifier for performance analysis as well as overall best classifier in such application.

**Keywords:** Sentiment Analysis, Stopwords, Lemmatization, Support Vector Machine, Random Forest , Naïve Bayes Classifier, Dummy Classifier, Confusion Matrix

## 2. INTRODUCTION

Text mining is the process of deriving relevant and high-quality information by analyzing a heap of text. Since the advent of the internet, text has become an integral means of us contributing to data, either by us communicating with our loved ones, giving out recommendations or reviews, tweeting, amongst others. Over time through text analysis we have been able to systematically pull out a worth of information from this supposedly scattered source of data. The movie industry is not excluded from this either. Reviews are an integral part of this industry as well (Gemser et al., 2007), as written by  critics/ fanatics and it can play an important role in consumer decision making in general and, film choice in particular. Most of these reviews are the opinion of people mostly based on feelings, termed as sentiment analysis or opinion mining.

## 3. OBJECTIVE

The aim of this study is to investigate the rating of a movie based on the review of a movie goer/critic.

## 4. METHODOLOGY

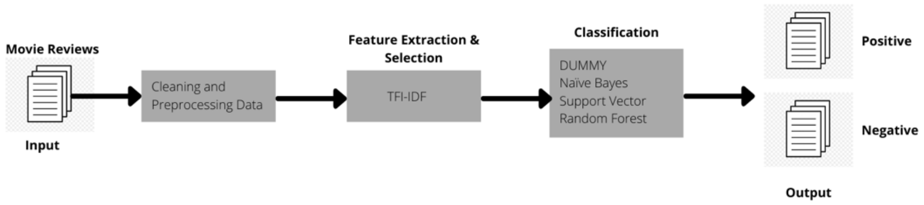

Fig 1. Proposed Methodology

### 4.1 Data Description

The dataset used was sourced from Kaggle website, it is comprised of movie reviews written by people and their general sentiments whether positive or negative. The dataset consists of 50000 rows and two columns (review and sentiments). It is a balanced with an equal distribution of 25000 positive and 25000 negative reviews. For this project, the balanced distribution is effective for predictive modeling.

## 4.2 Installing and Importing Dependencies, Data

The data was downloaded from Kaggle and hosted on Amazon S3 for easy importation for analysis. The various dependencies such as NLTK library, Pandas library, NumPy library amongst others were installed and imported as relevant for the case.

## 4.3 Cleaning Data

For the first step in cleaning the data, the following steps were done;

### 4.3.1 Conversion to Lower Case

The data is converted to lower case as the model has to treat words like “film” and “Film” as same. (As a result, the lines of code below transform all of the words to lowercase.)

```
X =”Movie-Review”
x = str(x).lower().replace('\\', '').replace('_', ' ')
movie review
```

### 4.3.2 Removing Email & HTML Tags

All potential forms of html tags such as <br/> <br/> and email forms such as abc@xyz.com were removed.

```
x = ps.remove_emails(x)
x = ps.remove_html_tags(x)
```

### 4.3.3 Removing Special & Accented Characters

Also, all forms of special and accented characters were removed to make prediction smooth. Characters such as “§”, “ÿ”, “ñ”, “ð”, “Æ”, “ㅈ”, “\”, “⍰” are removed.

```
x = ps.remove_accented_chars(x)
x = ps.remove_special_chars(x)
```

### 4.3.4 Spelling Correction

Words in the dataset is checked and spellings corrected. For instance,

```
x = 'niiiiceee moooovie'
x = re.sub("(.)\\1{2,}", "\\1", x)
print(x)
```

```
nice movie
```

## 4.4 Preprocessing Data

### 4.4.1 Tokenization

Tokenization is a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. In the concept of NLP, tokenization is needed to break word down in a way that the computer can understand. The words in the corpus are tokenized.

```
tokens = nltk.word_tokenize(review)
```

### 4.4.2 Stopwords Removal

Stopwords are normally common words in any natural language. In natural language processing such words can be termed as useless words(data) as it does not add any significant feature for the computation. Stop words such as “the”, “is”, “in”, “for” etc. are removed. However, the word “not” is kept as it gives a sentimental value in the review. For instance, a critic can use “not good’ for a “bad” movie review.

```
stopwords = stopwords.words('english')
stopwords.remove('not')
review = [word for word in tokens if word not in stopwords]
```

4.4.3 Lemmatization

Lemmatization considers the context and converts the word to its root form. It is the process of combining a word's inflected forms so that they can be examined as a single entity. For instance, words like “watching”, “watched” and “watches” are reduced to their root form, “watch”. Lemmatization was chosen as it is necessary to get valid words for the predictive modeling. Unlike stemming where the process requires the reduction of inflected words to their root form, lemmatization reduces words properly ensuring the root word belongs to the language (Hafsa Jabeen, 2018). The downside was that lemmatizer requires more time to compute as compared to stemmers, this however was not a problem as the words per review is at a reasonable mean value of 231 words.

## 5. FEATURE EXTRACTION AND SELECTION

### 5.1 Document Term Matrix

A document-term matrix, also known as a term-document matrix, is a mathematical matrix that represents the frequency of terms found in a set of documents. In a document-term matrix, rows correspond to documents and columns to terms. There are several methods for determining what value each matrix element should take. The TF-IDF method will be used in this instance. They are helpful in the natural language processing. TFIDF is the often-weighting method used in the Vector Space Model, particularly in Information Retrieval (IR) domain including text mining. It is a statistical method to measure the important of a word in the document to the whole corpus. (Amir Sjarif et al., 2019)

$$\text{TF-IDF} = \frac{TERM\ FREQUENCY}{DOCUMENT\ FREQUENCY}$$

Where document corresponds to the reviews per each row and the column corresponds to the terms/words. The term frequency indicates how relevant it is if the term occurs frequently in the document and the inverse document frequency refers to how rare a word appears in the entire document set.

## 6. CLASSIFICATION

Classification is a process of categorizing a given set of data into classes, it can be performed on both structured or unstructured data. It requires the use of a machine learning algorithm that know how to assign a class label to examples from the problem domain (Jason Brownlee, 2020). A classifier uses training data to learn how given input variables relate to the class. For this analysis, known positive and negative reviews have to be used as the training data. When the classifier has been properly trained, it can be used to predict the rating of a movie.

### 6.1 **Confusion Matrix**

Confusion matrix is a performance measurement for machine learning classification. It is extremely useful for measuring recall, precision, accuracy etc. It provides insight into the predictions1` therefore; it was used in evaluation of each classifier.

### 6.2 Dummy Classifier as A Baseline

A dummy classifier is a type of classifier which does not generate any insight about the data and classifies the given data using only simple rules. It is useful as a simple baseline to compare with other classifiers. The baseline was used with the understanding that any analytic approach for a classification problem should be better than this random guessing approach.

#### 6.2.1 Confusion Matrix

| N=15000 | | 0 | 1 |
|---|---|---|---|
| True Label | 0 | 3752 | 3659 |
| | 1 | 3828 | 3761 |
| | | Predicted Label | |

The accuracy of the classifier per the confusion matrix is calculated by the true value of a positive review plus the true value of a negative review rightly predicted divided by the total size.

i.e $\frac{3752+3761}{15000} = 0.50\%$

## 6.3 Naïve Bayes Classifier

Naive Bayes is a probabilistic classifier inspired by the Bayes theorem that operates on a single assumption: the characteristics are conditionally independent. It is based on the premise that the predictor variables in a Machine Learning model are independent of each other; indicating the outcome of a model is precedented on a group of independent variables that are unrelated to one another. It is used in cases such as document classification, spam filters, sentiment analysis etc.

### 6.3.1 Confusion Matrix

| N=15000 | | 0 | 1 |
|---|---|---|---|
| True Label | 0 | 5392 | 1893 |
| | 1 | 2062 | 5546 |
| | | Predicted Label | |

The accuracy of the classifier per the confusion matrix is calculated by the true value of a positive review plus the true value of a negative review rightly predicted divided by the total size.

i.e $\frac{5392+5546}{15000} = 0.73\%$

## 6.4 Support Vector Machine

It is a supervised learning machine classification algorithm that is capable of performing classification, regression and outlier detection. The SVM classifier represents the training data as points in space separated into categories by a gap as wide as possible. SVM are used for image classification, handwriting recognition, text categorization amongst others. In this study, the linear support vector classifier (LinearSVC) was used due to the runtime error. The LinearSVC is a faster implementation of the SVC for the case of a linear kernel.

6.4.1 Confusion Matrix

| N=15000 | | 0 | 1 |
|---|---|---|---|
| True Label | 0 | 5889 | 1522 |
| | 1 | 1353 | 6236 |
| | | Predicted Label | |

The accuracy of the classifier per the confusion matrix is calculated by the true value of a positive review plus the true value of a negative review rightly predicted divided by the total size.

i.e $\frac{5889+6236}{15000} = 0.81\%$

## 6.5 Random Forest

Random decision trees, often known as random forest, are an ensemble learning approach for classification, regression, and other tasks. It works by creating a large number of decision trees during training and then outputs the class that is the mode of the classes or classification or mean prediction (regression) of the individual trees. Random forest classifier can be used in the case of sentimental analysis and various forms of classifications.

6.5.1 Confusion Matrix

| N=15000 | | 0 | 1 |
|---|---|---|---|
| True Label | 0 | 5518 | 1893 |
| | 1 | 2062 | 5527 |
| | | Predicted Label | |

The accuracy of the classifier per the confusion matrix is calculated by the true value of a positive review plus the true value of a negative review rightly predicted divided by the total size.

i.e $\frac{5518+5527}{15000} = 0.74\%$

## 7. RESULTS AND DISCUSSION

The table below compare the performance of algorithms in the experiment. The performance evaluation measures such as accuracy, precision and f-measure.

| Algorithm | Accuracy | Precision | F-measure |
|---|---|---|---|
| TF-IDF + Dummy Classifier | 0.50 | 0.49 | 0.50 |
| TF-IDF + Naïve Bayes Classifier | 0.73 | 0.73 | 0.73 |
| TF-IDF + Support Vector Machine | 0.81 | 0.80 | 0.81 |
| TF-IDF + Random Forest | 0.74 | 0.74 | 0.74 |

NB: when the codes are re-run, there is a possibility of a standard deviation around 0.01 of the values above

The baseline classifier has an accuracy of 50%, which comparatively is lower as all the other classifiers are significantly above it. The Naïve Bayes and Random Forest Classifiers are relatively close with a 1% difference between them, i.e 73% and 74% respectively. However, the Support Vector Classifier has a higher accuracy score of 81%.

## 8. CONCLUSION

The Support Vector Machine Classifier (Linear) outperform all the other classifiers but the result is not conclusive as only few sampled classifiers were used. Also, there is a possibility of different preprocessing techniques affecting the results of this classifications. For this project, in comparison to the classifiers used and the relatively good precision and f-measure scores, the support vector classifier stood out in properly categorizing a positive and negative movie reviews

## 9. Python Code link

https://colab.research.google.com/drive/1UktwOo4cWD7FzwaL2Nt5zz2JRR6lwqEK?usp=sharing

## 10. REFERENCE

Amir Sjarif, N. N., Mohd Azmi, N. F., Chuprat, S., Sarkan, H. M., Yahya, Y., & Sam, S. M. (2019). SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm. *Procedia Computer Science*, *161*, 509–515. https://doi.org/https://doi.org/10.1016/j.procs.2019.11.150

Gemser, G., Van Oostrum, M., & Leenders, M. A. A. M. (2007). The impact of film reviews on the box office performance of art house versus mainstream motion pictures. *Journal of Cultural Economics*, *31*(1), 43–63. http://www.jstor.org/stable/41810940

Hafsa Jabeen. (2018). *Stemming and Lemmatization in Python*. Data Camp. https://www.datacamp.com/community/tutorials/stemming-lemmatization-python

Jason Brownlee. (2020). *4 Types of Classification Tasks in Machine Learning*. Machine Learning Mastery. https://machinelearningmastery.com/types-of-classification-in-machine-learning/

Narkhede, S. (2018). *Understanding Confusion Matrix*. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Jason Brownlee. (2020). *4 Types of Classification Tasks in Machine Learning*. Machine Learning Mastery. https://machinelearningmastery.com/types-of-classification-in-machine-learning/

*ML | Dummy classifiers using sklearn*. (2019). https://www.geeksforgeeks.org/ml-dummy-classifiers-using-sklearn/

Waseem, M. (2020). *How To Implement Classification In Machine Learning?* Data Science with Python. https://www.edureka.co/blog/classification-in-machine-learning/

Asiri, S. (n.d.). *Machine Learning Classifiers*. Towards Data Science. Retrieved June 3, 2021, from https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623

Muñoz, E. (2020). *Getting started with NLP: Tokenization, Document-Term Matrix, TF-IDF*. Analytics Vidhya. https://medium.com/analytics-vidhya/getting-started-with-nlp-tokenization-document-term-matrix-tf-idf-2ea7d01f1942