**Movie Data Analysis**

**Group #8**

Alekha Polavarapu 811189207

Priyanka Yakkateela 811190147

Vanaja Karpurapu 811191666

Sithamshu Teegala 811170902

**Introduction:**

The most practical kind of enjoyment for most people is watching movies. However, only a small number of films are well-liked and profitable. There are many of ratings websites that will guide movie enthusiasts in choosing which films to view and which to skip. The most popular of those are websites like IMDB, Rotten Tomatoes, etc. By awarding a movie a score out of 10 based on the stars that the viewers have given it, these websites measure the success of the film. However, there is no technique that can offer a prediction based on movie reviews. Sentiment analysis is thus used to assess a film's popularity based on its reviews.

The interpretation and classification of emotions found in text data using text analysis methods is known as sentiment analysis. Businesses can determine customer sentiment via online conversations and reviews about their products, brands, or services by using sentiment analysis. Sentiment analysis models place a strong emphasis on polarity (positive, negative, and neutral), as well as on sentiments and feelings (angry, joyful, sad, etc.), and even on intents (e.g. interested v. not interested). Sentiment analysis has gained popularity, and many major corporations are spending resources in it to forecast the outcomes for their company.

Tokenization, word filtering, stemming, and classifications are all part of the sentiment analysis process. Text must be divided into tokens, such as words, numbers, or punctuation, in order to tokenize it. The process of stripping a word of its prefixes and affixes in order to reveal its stem is called stemming. Following pre-processing, we do classification on the dataset using Naive Bayes, Support Vector Machine, and Logistic Regression. Here, accuracy is used to choose the best model. As a result, we evaluate and research the elements that have an impact on the ratings of our review text before classifying the movie as good or negative.

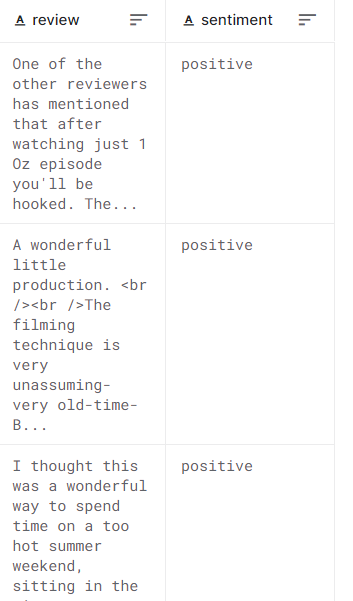
**Project Description:**

To forecast how viewers will evaluate a film, we are using IMDB reviews. Movies and forecast whether or not they will receive favourable or unfavourable reviews. We put out a model with a variety of sentiment analysis techniques to assist us extract information from the data and forecast, based on accuracy, which classifier would be most suited for this particular area. Models such as Logistic Regressions, Support Vector Machines, and Naive Bayes were used.

**Dataset:**

We have data which comprises a dataset with 50,000 IMDB reviews that are equally split into 25,000 for training and 20,000 for testing.

As reviews for the same movie sometimes have associated ratings, there are only 30 reviews per movie. Additionally, because the train and test sets comprise different movies, it would be pointless to memorize a specific movie's terms and their related labels. A review is given a score of <=4 if it is unfavourable, >=7 if it is favourable, and between >4 and 7 if it is neutral.



**Problem Description:**

The ratings on different websites determine the success of the movie by giving it a score out of 10 based on the stars given by the viewers. But, there isn’t any method that can provide the prediction based on movie reviews. Due to the lack of strong grammatical formats in movie reviews, we might have large amount of noise text, special characters, stop words. Presence of these huge unwanted words in the reviews makes the analysis harder and also takes large amount of time for the completion of our analysis. Hence, there is huge need to eliminate, normalize noise text, stop words, special characters from the reviews.

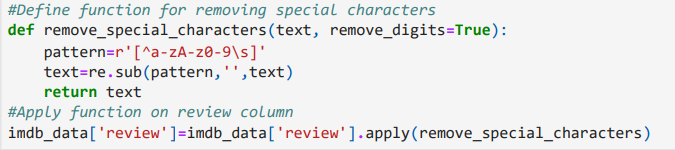
**Proposed Techniques:**

Data cleaning, data pre-processing, applying classifiers to the data, and lastly comparing the outcomes from the various classification models we employed are all included in the framework we suggest for our approach.

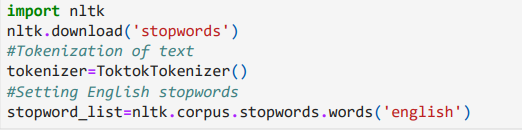
1. **Data Pre-Processing:**

We have made some adjustments to the data that we have gathered in order to improve the performance of our model. We have eliminated extraneous noise from the data to aid our model's classification. It includes the subsequent process:

* Lemmatization: The method of lemmatization involves changing the given word to its base word. Lemmatization's primary goal is to correctly interpret a word's morphological meaning by using the library's integrated dictionary. We made use of porter stemmer lemmatization and wordNet.
* Removing Stop words and special characters: Stop words like "a," "the," "this," "that," etc. make up the data. These mostly irrelevant terms are found in numerous reviews. Hence we need to remove them as they are unnecessary for our analysis process.



* Tokenization of text: It is often known as text splitting into a list of tokens, is the process of tokenizing or dividing a string of text. A token can be thought of as a component, much like a word in a sentence or a sentence in a paragraph. The NLTK library was used for tokenization. It includes numerous NLTK-trained languages, including German, English, Spanish, and French. The word tokenization function in NLTK isolates all punctuation apart from periods and makes use of treebank tokenization.



1. **Extraction of Features:**

The process of transforming a word into a matrix form is called feature extraction. The following techniques have been applied by us for feature extraction:

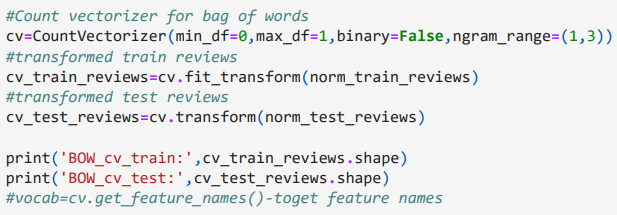
**Bag Of Words:** By counting how many times each word appears, the bag-of-words (BOW) model converts any text into fixed-length vectors. Vectorization is a common term used to describe this procedure. We utilized the vectorization technique described below:

**Count Vectorizer:** The procedure combines tokenizing a group of datasets' worth of

documents with creating a vocabulary for those tokens. This yields a vocabulary word

length and an integer value for each word based on how frequently it appears. Sparse

words are those that are rare and may have zero values.



**Classification Models:**

We used Naive Bayes, Logistic Regression, and Support Vector Machine in our experiment. In order to predict whether a movie would be good or bad, we have trained our model using the classifiers mentioned above.

1. **Naive Bayes:** It is a text classification technique that predominantly utilizes high dimensional training data sets. For instance, sentiment analysis and spam screening. This algorithm determines the likelihood that an object with specific characteristics belongs to a particular class. A probabilistic classifier is used. Because it makes the naive assumption that the occurrence of particular traits is independent of one another, which is false, this algorithm is known as Naive Bayes.

P(A/B) = P(B/A) P(A) / P(B)

B is referred to as the evidence, and A is known as the proposition.

Prior probability of the claim is denoted by P(A), while prior probability of the evidence is denoted by P(B). We refer to the posterior as P(A/B). We refer to the likelihood as P(B/A).

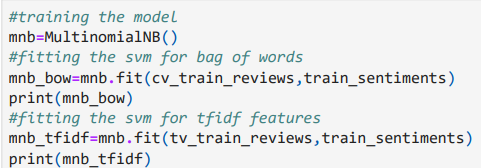
P(A/B) is the likelihood that event A will occur given that event B has already happened.

P(A) = Chance that event A will occur

P(B) = Chance that event B will occur.

P(B/A) is the likelihood that event B will occur given that event A has already happened.

With count vectorizer, our accuracy is 85.48% for Naive Bayes.

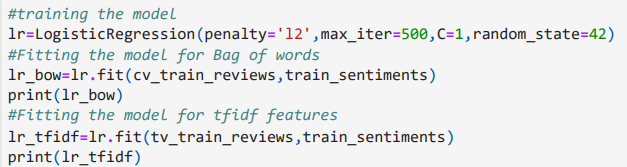


1. **Logistic Regression:** In contrast to linear regression, which fits data to a line, logistic regression attempts to fit a "S"-shaped logistic function to the data (Sigmoid Function). Despite the word "regression" being in the name, this method is actually utilized for classification. Its ability to categorize data using both continuous and discrete metrics makes logistic regressions a well-liked machine learning algorithm. Maximum Likelihood is a technique used in logistic regression to fit data. It may be used to classify samples and can classify samples using many types of data. It can be used to evaluate which variables are helpful for categorizing samples.

Sigmoid function

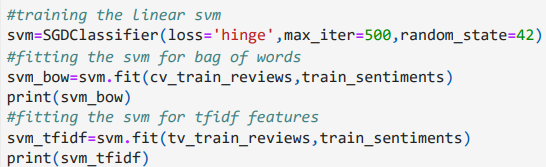
Y= e (b0 + b1 \* x) / (1 + e (b0 + b1 \*x) )

Here b0 is the bias and b1 is the coefficient for the value x and y. In logistic Regression with Count Vectorizer we get an accuracy of 86.89 %



1. **Support Vector Machine:** SVM is a classification and regression algorithm. To do the classifications, it creates a hyperplane or collection of hyperplanes in infinite dimensional space. It examines the data set's extremes and delineates a decision boundary (Hyperplane). SVM is renowned for its effective operation. It determines the separation between the two provided observations, after which it searches for a decision boundary to determine the separation between the nearest individuals in each class.

SVM are resilient when the model is overfit. SVM from the scikit-learn toolkit was utilized in this case. Both dense and sparse sample vectors are acceptable inputs for scikit-support learn's vector machines. With Countvectorizer in SVM, we achieve an accuracy rating of 85.29%.



**Experimental Evaluation:**

Based on accuracy, we have contrasted the results of all the applied classification models.

Accuracy: It is merely the proportion of observations that were successfully predicted to all observations. We might say that the model is better the higher the accuracy.

Accuracy is provided by

Accuracy = TP + TN / TP + TN + FP + FN.

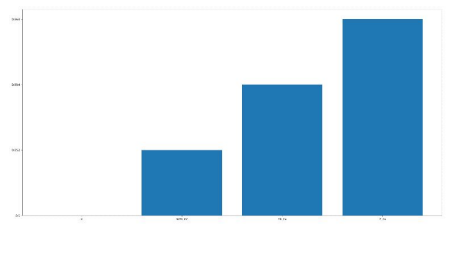
The analysis of accuracies of all the models is given below:

nb\_cv = Naïve Bayes with Count Vectorizer = 85.48

svm\_cv = SVM with Count Vectorizer = 85.29

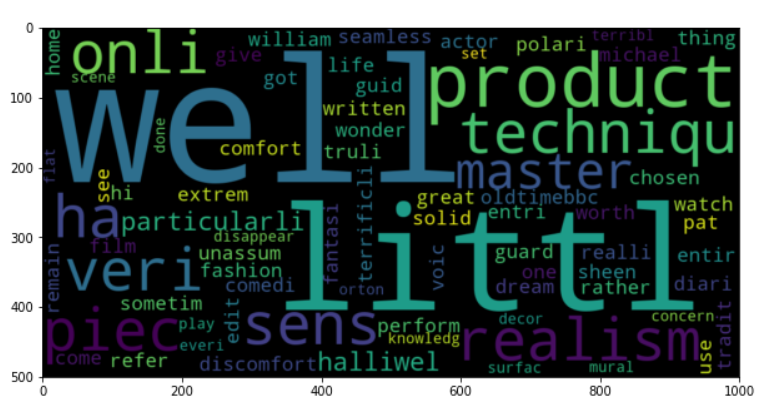
lr\_cv = Logistic Regression with Count Vectorizer = 86.89

Accuracy Graph:

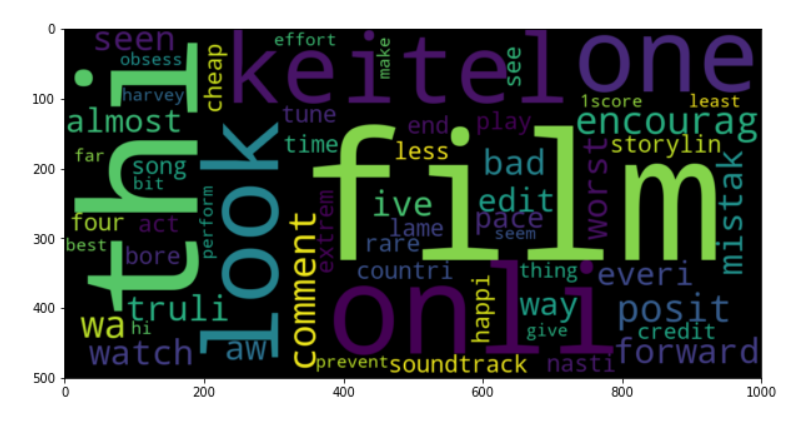


We have also visualized the Word cloud for positive review words and Negative review words.

**Word cloud for positive review words:**



**Word cloud for negative review words:**

****

**Future Work and Conclusion:**

The primary goal of this project was to build a sentiment analysis model that would enable us to comprehend the movie reviews we have gathered better. The output of the three classifiers, Naive Bayes, Logistic Regression, and Support Vector Machine, was compared (SVM). For evaluation, we looked at the accuracy that each model offered. By comparing the models, we discovered that Logistic Regression had the greatest accuracy rating, coming in at 86.89%.

In future we can still improve the accuracy of the models by preprocessing data and by using lexicon models like Textblob.

**References:**

* Nisha Rathee, Nikita Joshi, Jaspreet Kaur.“Sentiment Analysis Using Machine Learning Techniques on Python”. 978-1-5386-2842-3

<https://ieeexplore.ieee.org/document/8663224>

* MaisYasen, Sara Tedmori. “Movies Reviews Sentiment Analysis and Classification”. IEEE Jordon International Joint Conference on Electrical Engineering and Information Technology (JEEIT). 978-1-5386-7942-5.
* Sagar Chavan, Akash Morwal,Shivam Patanwala,Prachi Janrao, “Sentiment Analysis of Movie Rating System”, IOSR Journal of Computer Engineering (IOSR-JCE), eISSN: 2278-0661, p- ISSN: 2278-8727 , PP 43-47, 2017.
* Asiri Wijesinghe, “Sentiment Analysis on Movie Reviews”, Australian National University Technical Report – RESEARCH GATE, October 2015.
* Sourav Mehra, Tanupriya Choudhary. “Sentiment Analysis of User Entered Text”. International Conference of Computational Techniques, Electronics and Mechanical Systems (CTEMS). 978-1-5386-7709-4.