****

**ASSIGNMENT 4**

**AIDI 1003 – CAPSTONE TERM 1**

**AMIT MARAJ**

AHNCH BALA 100424062

SONAKSHI KARKERA 100720763

SURBHI THAKUR 100732335

ARUN KALAESWARAN 100771700

**October 25, 2019**

1. **Competition**- Word2Vec – Bag of Words Meets Bags of Popcorn

This competition uses word to vector sentimental analysis for movie reviews. The dataset provides us with labeledTrainData.csv, testData.csv and unlabeledTrainData.csv

labeledTrainData.tsv file which includes id, sentiment and reviews. The sentiment 1 classifies the positive review and 0 classifies the negative review.

testData.tsv and unlabeledTrainData.csv includes and id and reviews.

We will be working on labelled Training data and doing the cleaning and preprocessing of the reviews and then training it in the model. Afterwards we will be predicting our result using test data file.

Final Output- For the competition it will generate the csv file which includes the id and the sentiment of the reviews and we will submit it in the competition.

Reason for choosing this competition-

We will be working on the NLP and sentimental analysis on Yelp dataset to predict the closure of restaurant. So, this competition will help us getting hands on the similar situation.

1. **Description of Algorithm we choose and why-**

We choose random forest algorithm to implement this problem. Random forest is a decision tree model that is split on a subset of features.

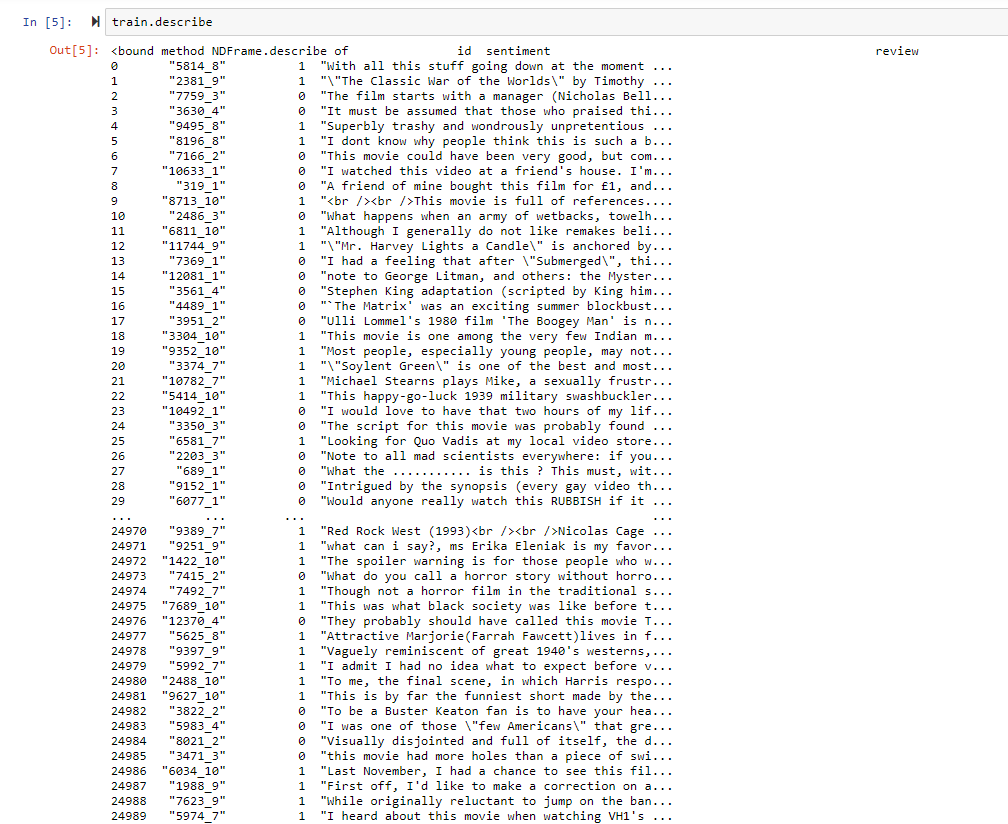
Reason for using Random Forest-

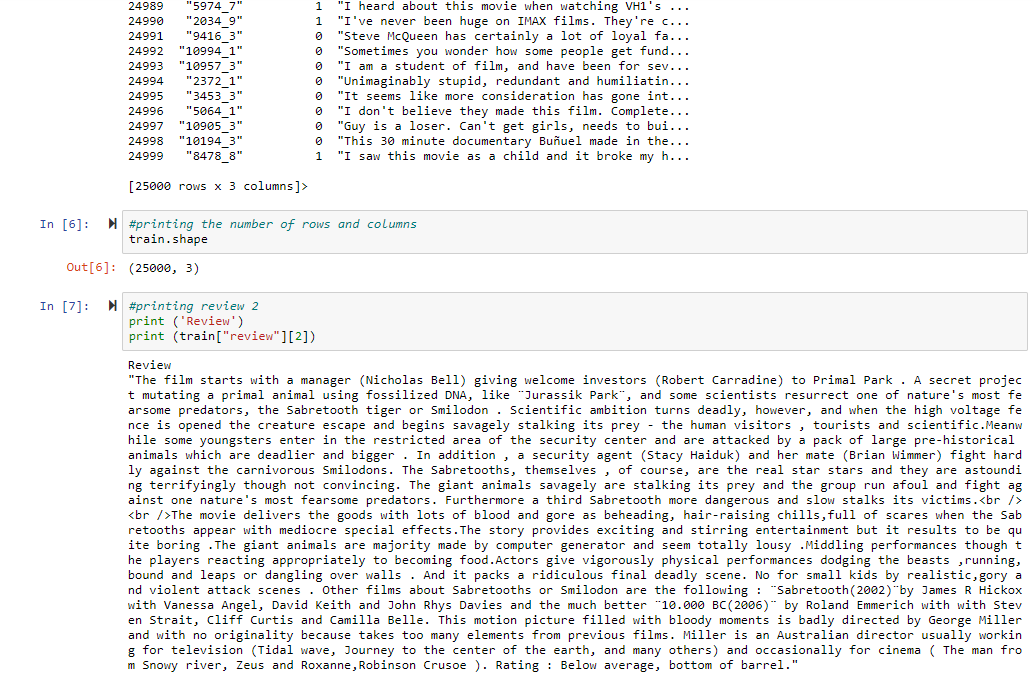
Random forest can effectively handle binary and numerical features as we will be handling the numerical ID’S and the sentiment values i.e 0 and 1

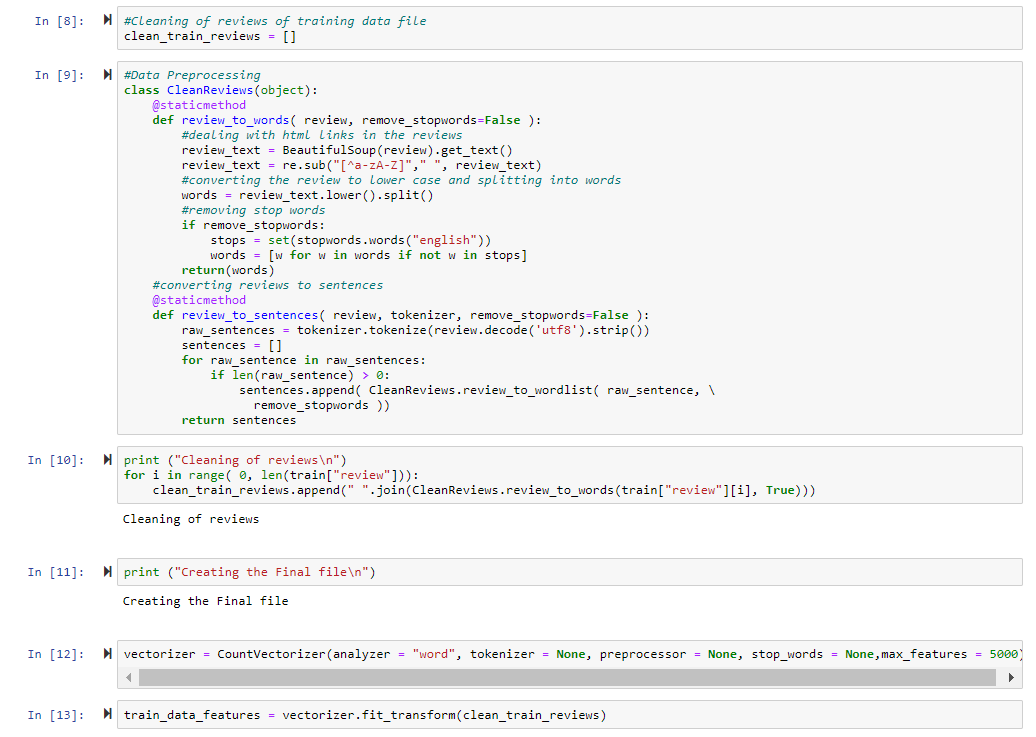
Random forest algorithm trains faster as we can split the process to multiple machines to run. This results in faster computation time. Random forest provides quick predictions.

1. CODE-



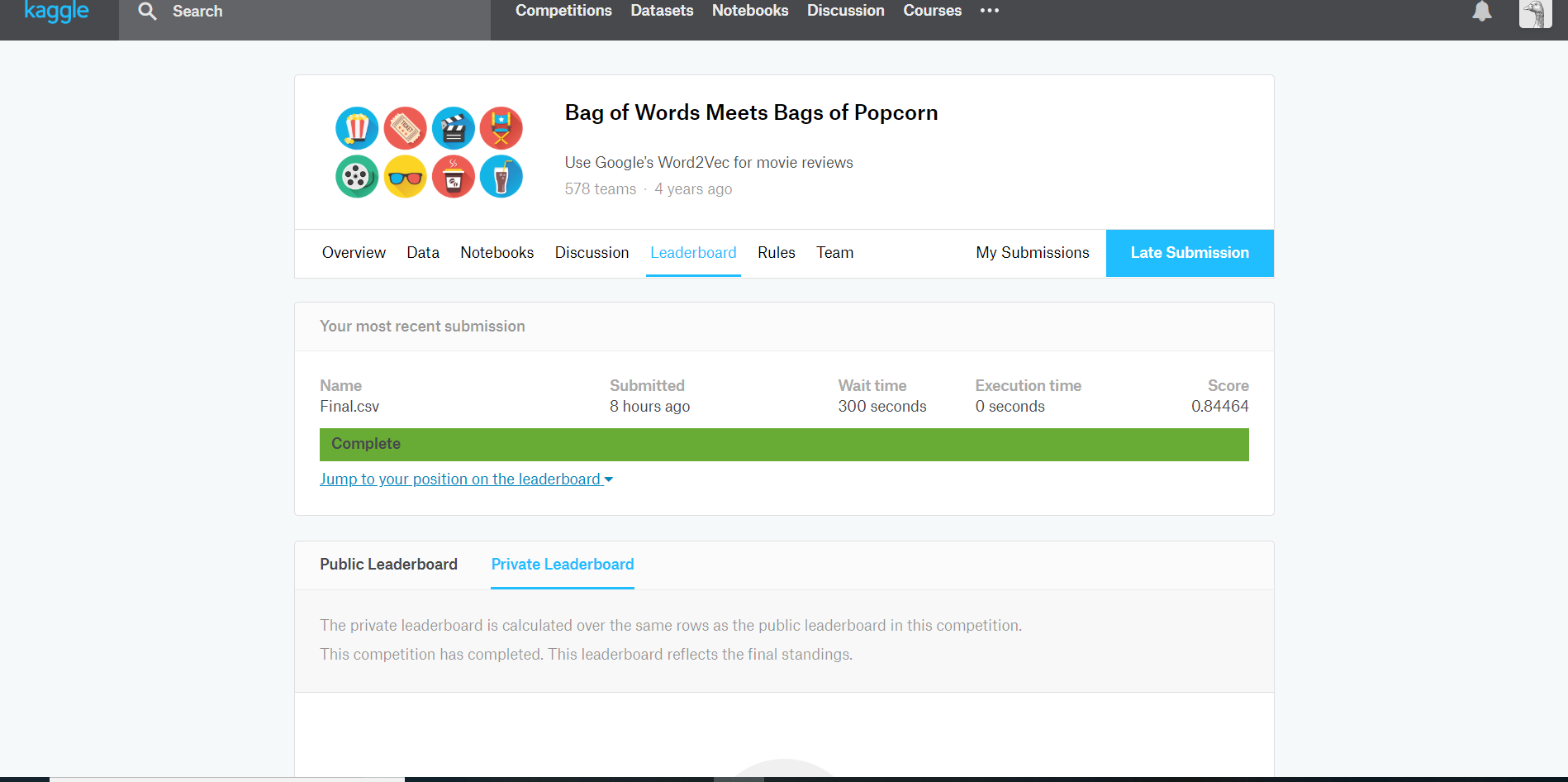


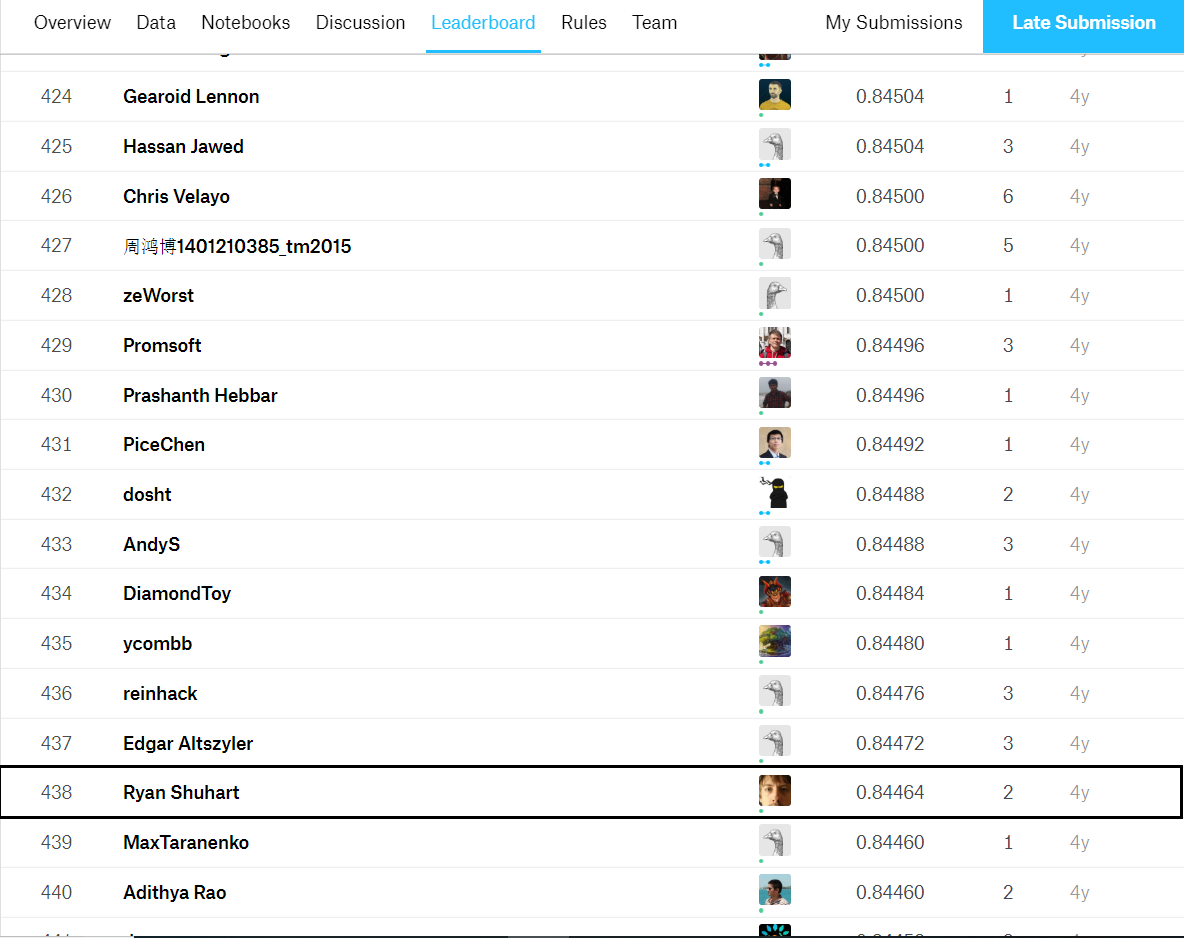






4.





5.

Summary of how you did it?

First we load the input file LabelTrainData.csv. Then we proceed with cleaning and preprocessing of reviews by removing HTML links, converting the reviews to lower case, splitting into words and removing stop words. After we do data preprocessing, we convert the words into vectors using vectors fit transformer. Then we train the random forest model to predict the sentiments of the testData.tsv file.

Finally we store the result in the csv file which at the end will be submitted to the competition.

The limitation in this competition is that the dataset is imbalanced and is getting overfitted. If given more time then we can deal with the imbalance dataset using SMOTE technique which can lead to higher accuracy.

**1. Competition 2- Sentiment Analysis of Movie Review**

This competition uses a dataset provided by Rotten Tomatoes. This dataset has many movie reviews that will be used with sentimental analysis to determine if a movie review was negative, somewhat negative, neutral, somewhat positive or positive.

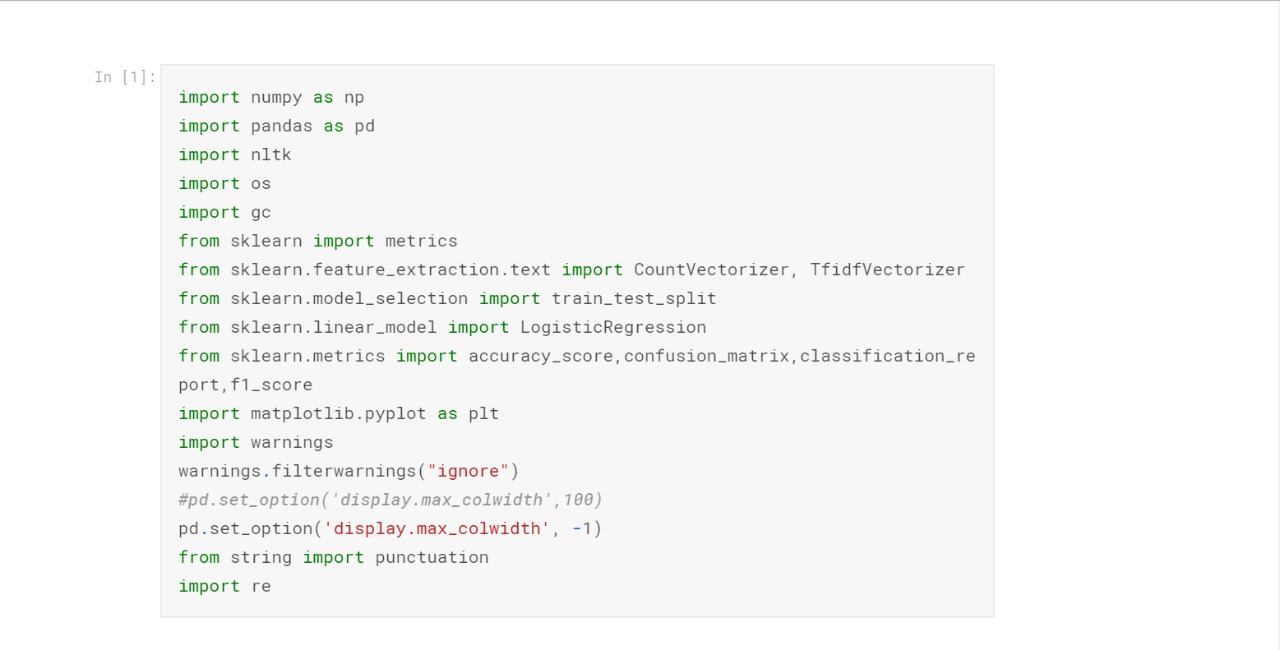
The competition provides you with 2 datasets: test.tsv, train.tsv. The final submission will be a .csv file. The training dataset has PhraseID, SentenceID, Phrase Sentiment. The test dataset has PhraseID, SentenceID, Phrase.

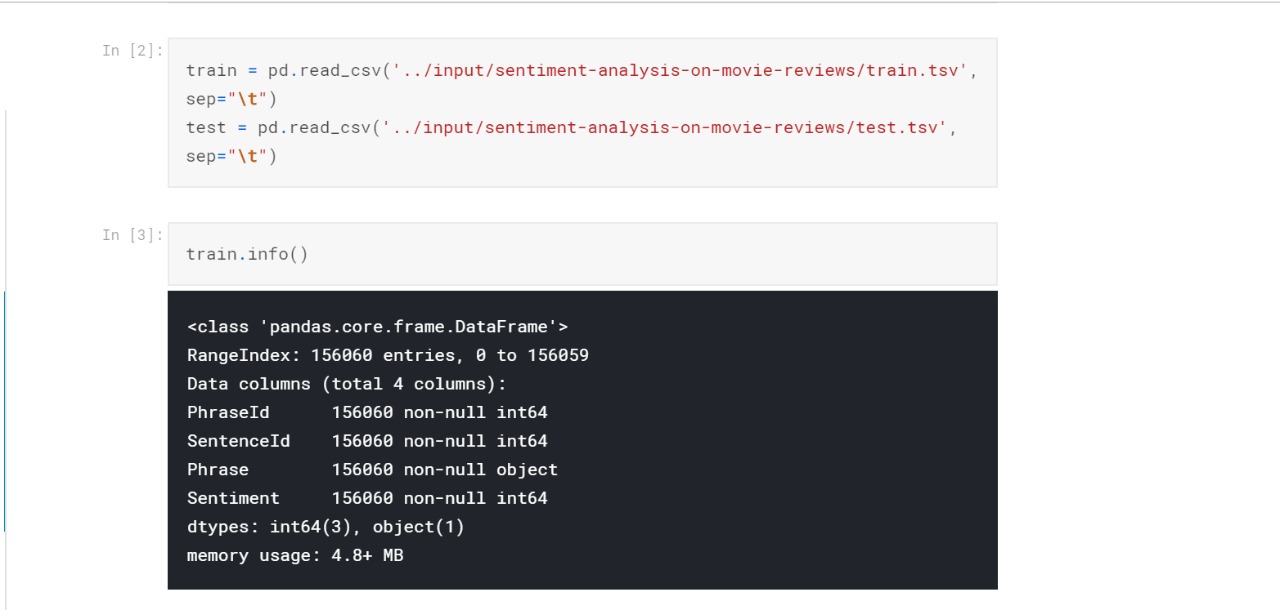
We chose this dataset because we wanted to see how logistic regression can be used with sentimental analysis. If it does well we can use it for our capstone final project which heavily relies on sentimental analysis.

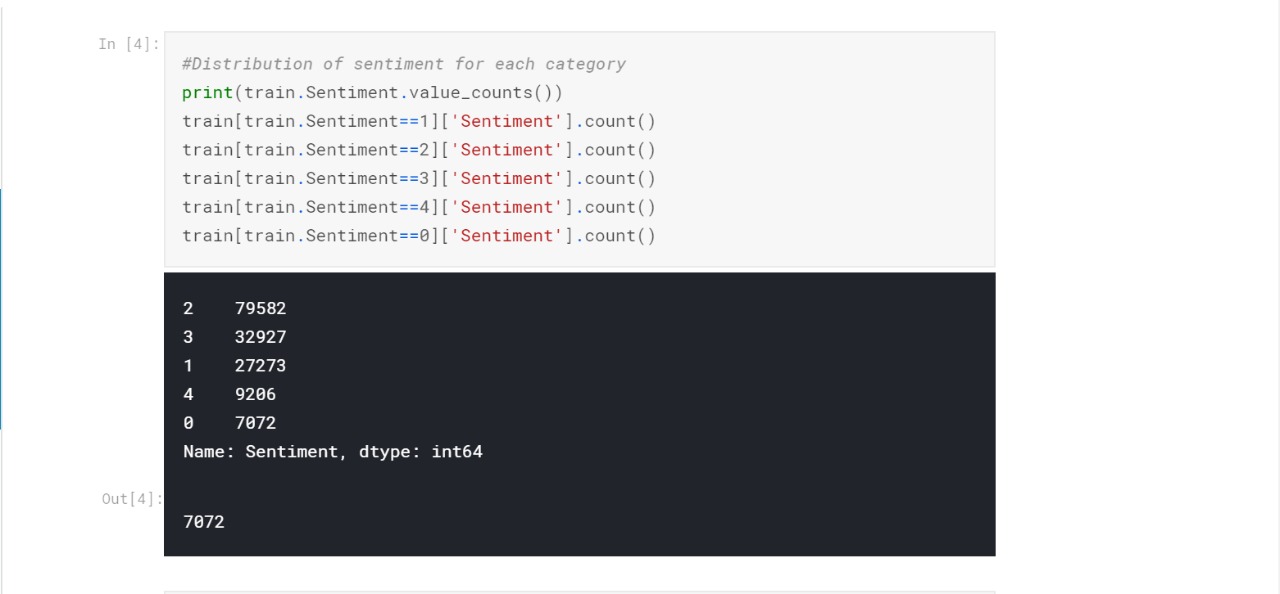
**2. Description of Algorithm**

For this competition we chose to utilize logistic regression. Logistic regression would be the ideal algorithm to use for this competition as it can efficiently classify the reviews into 1 of the 5 categories. It is also significantly easy to implement compared to other algorithms.

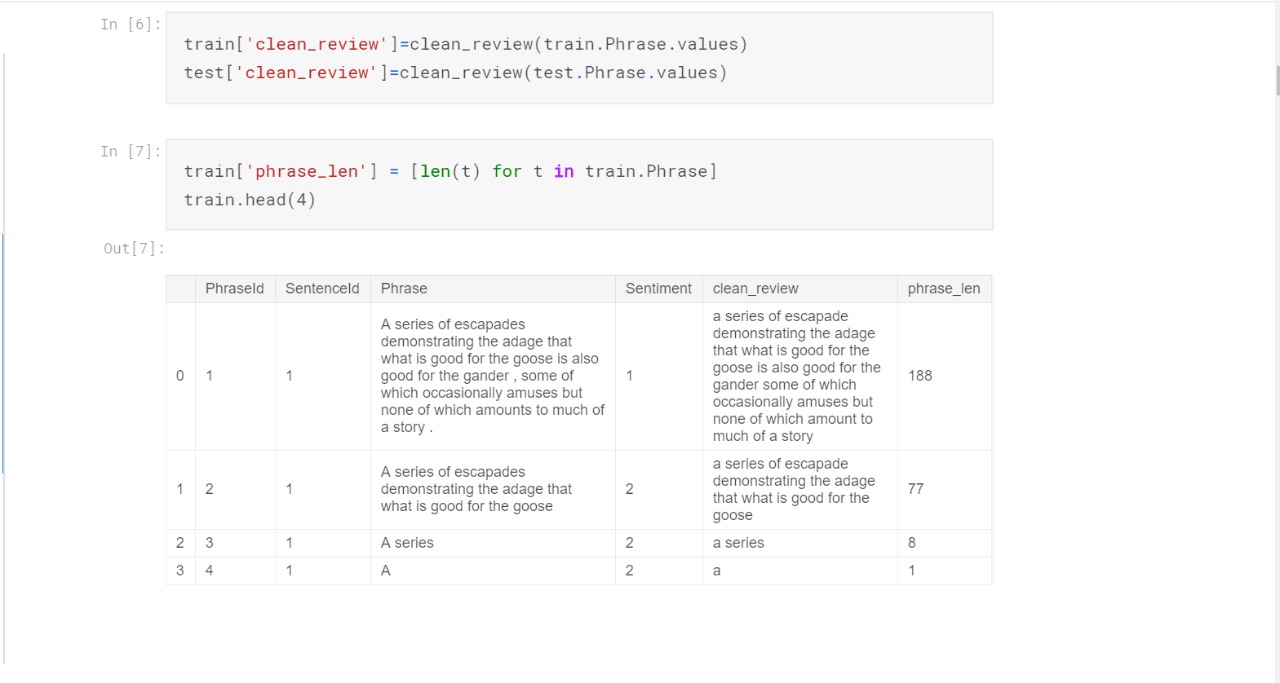
**3.Code Snippet**

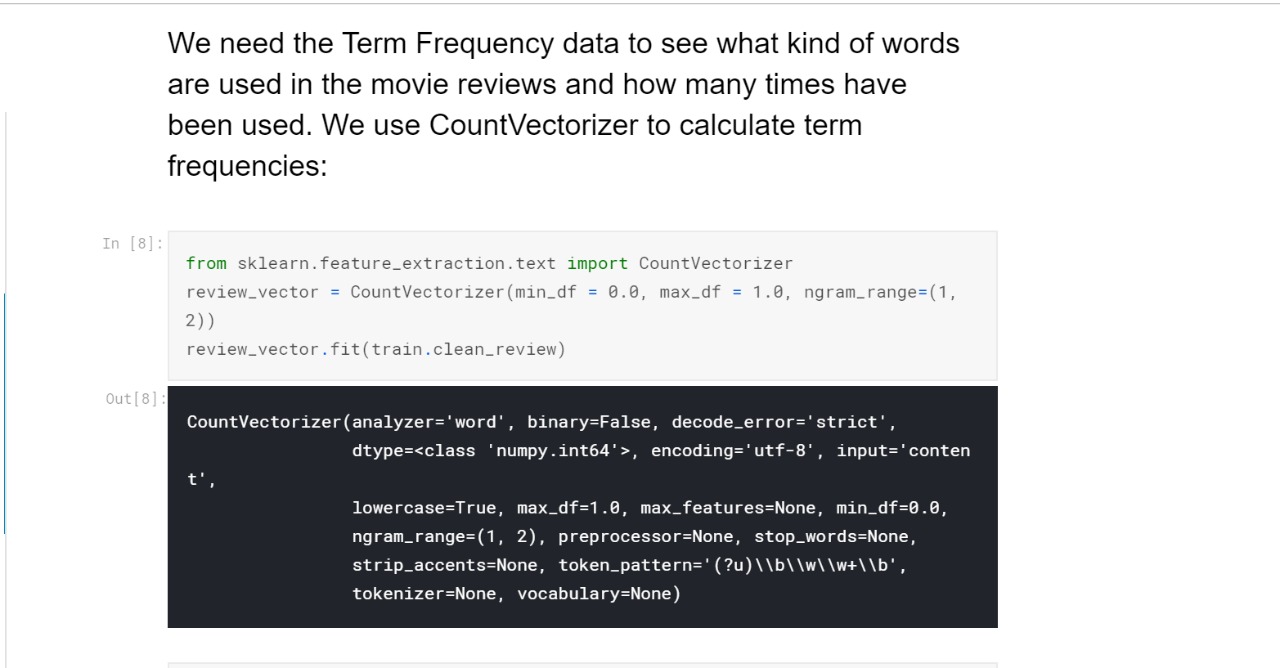


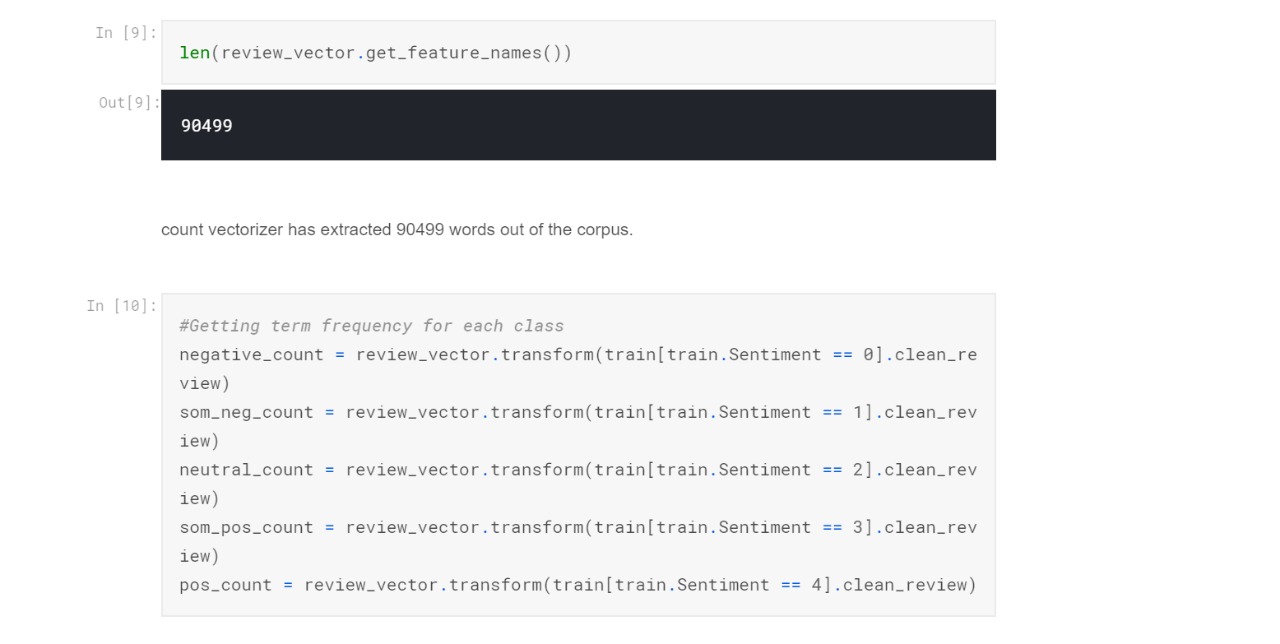








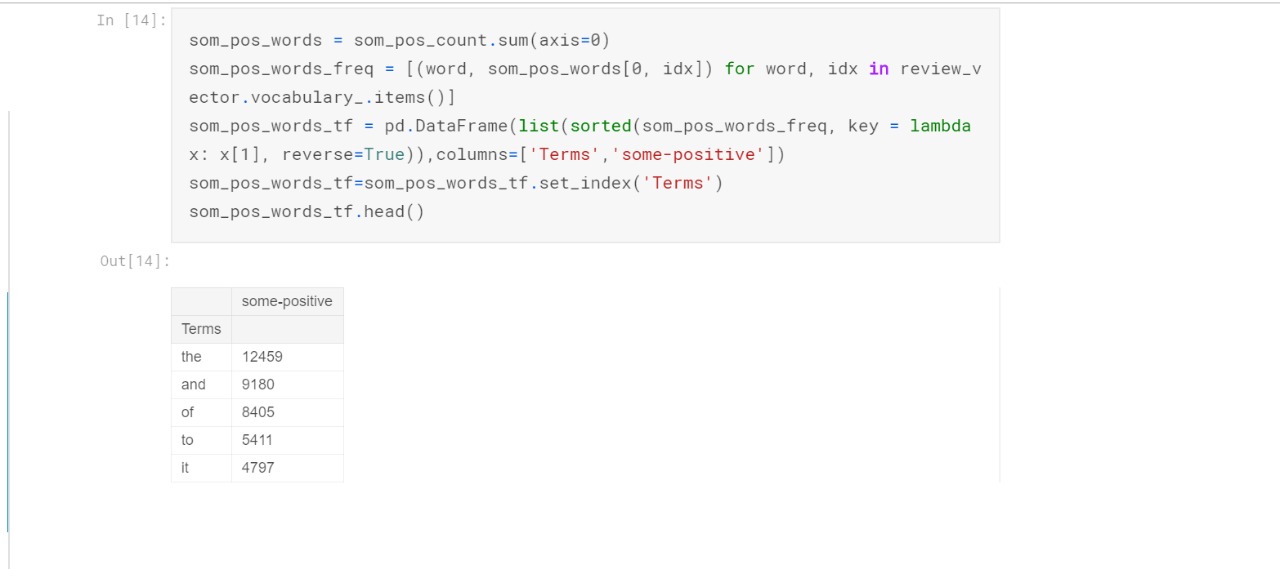




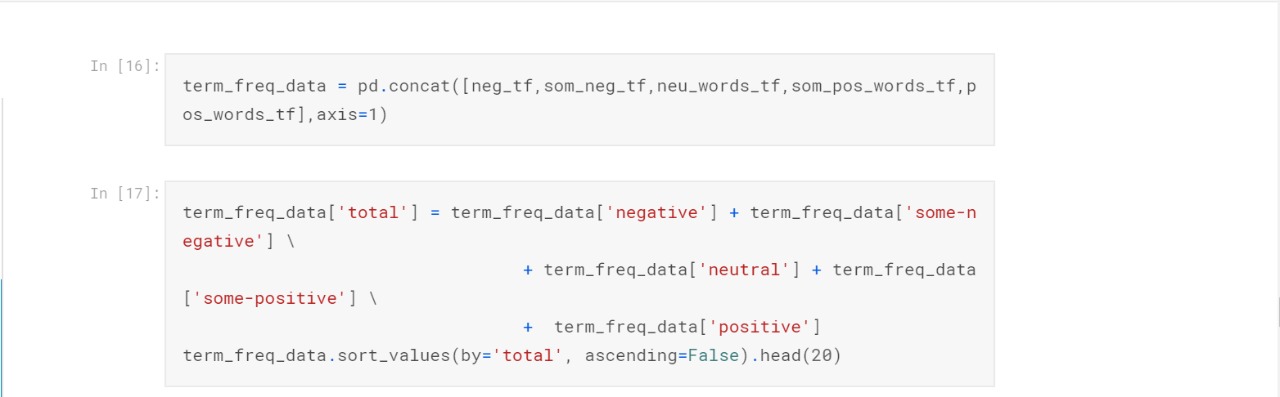


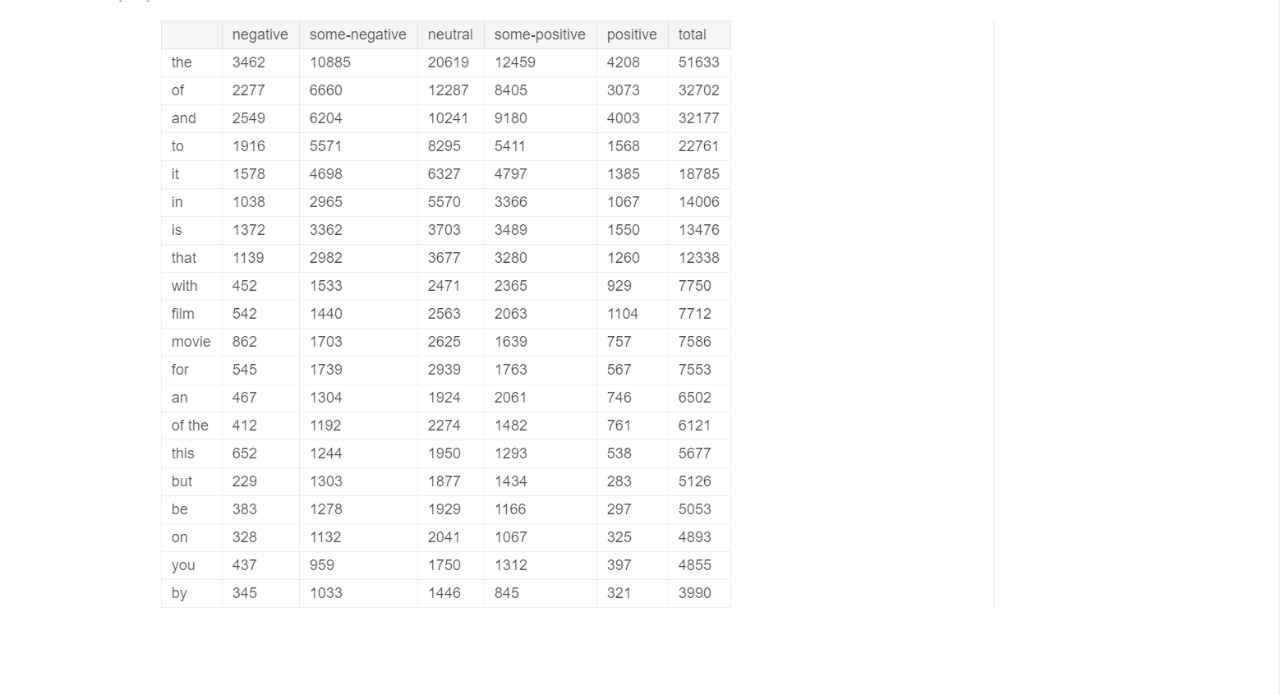




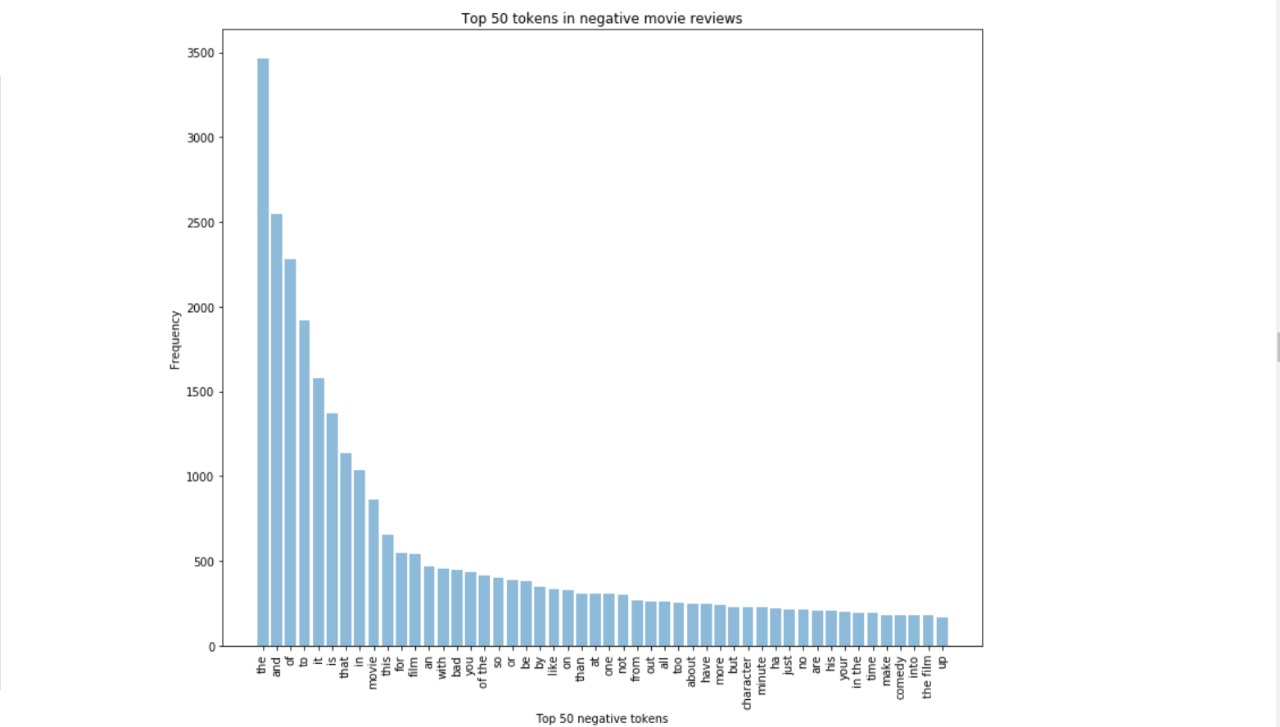


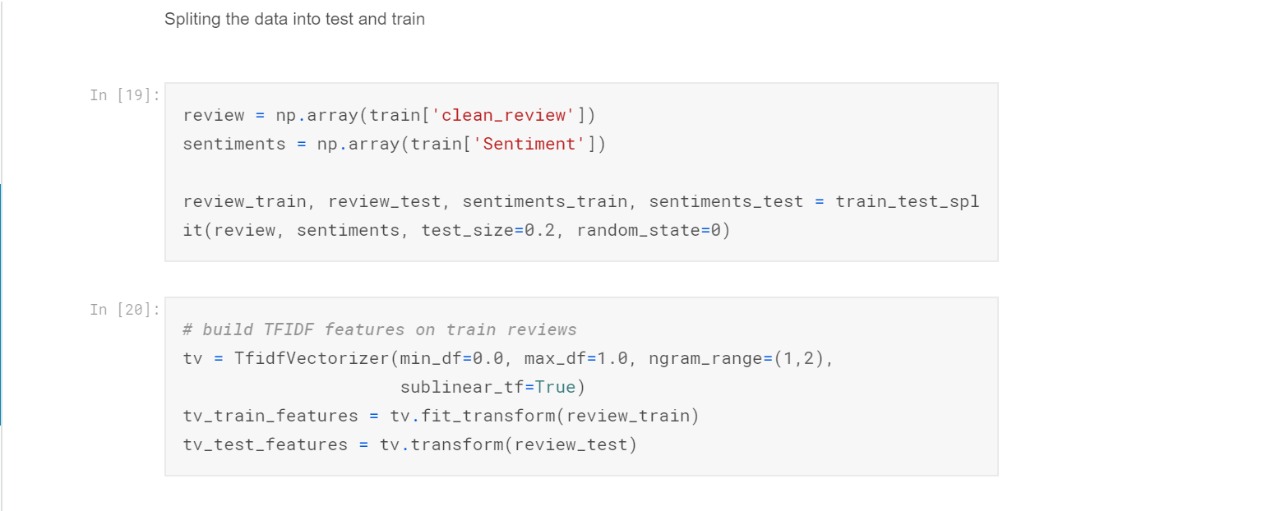


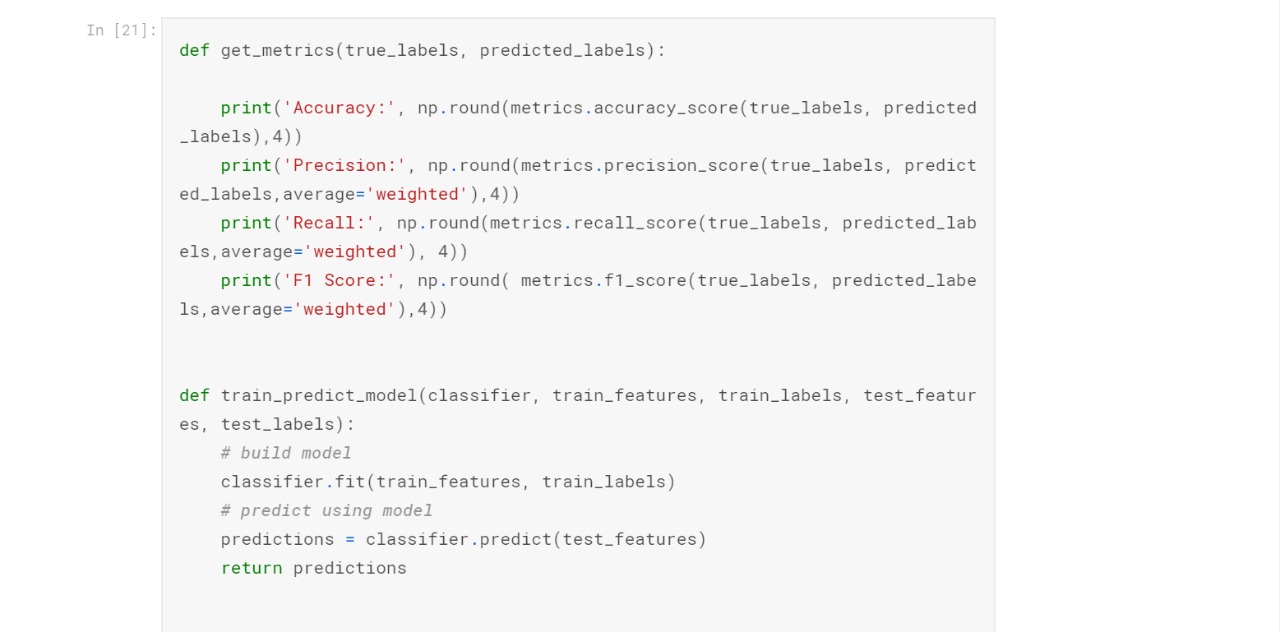






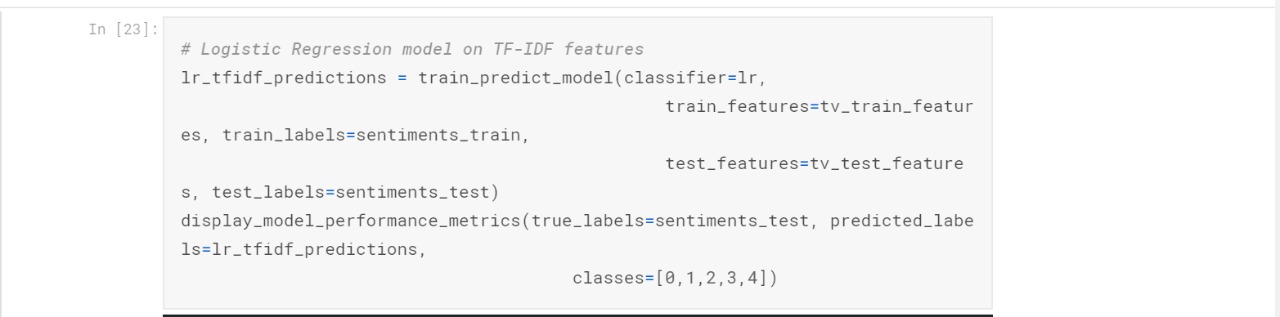


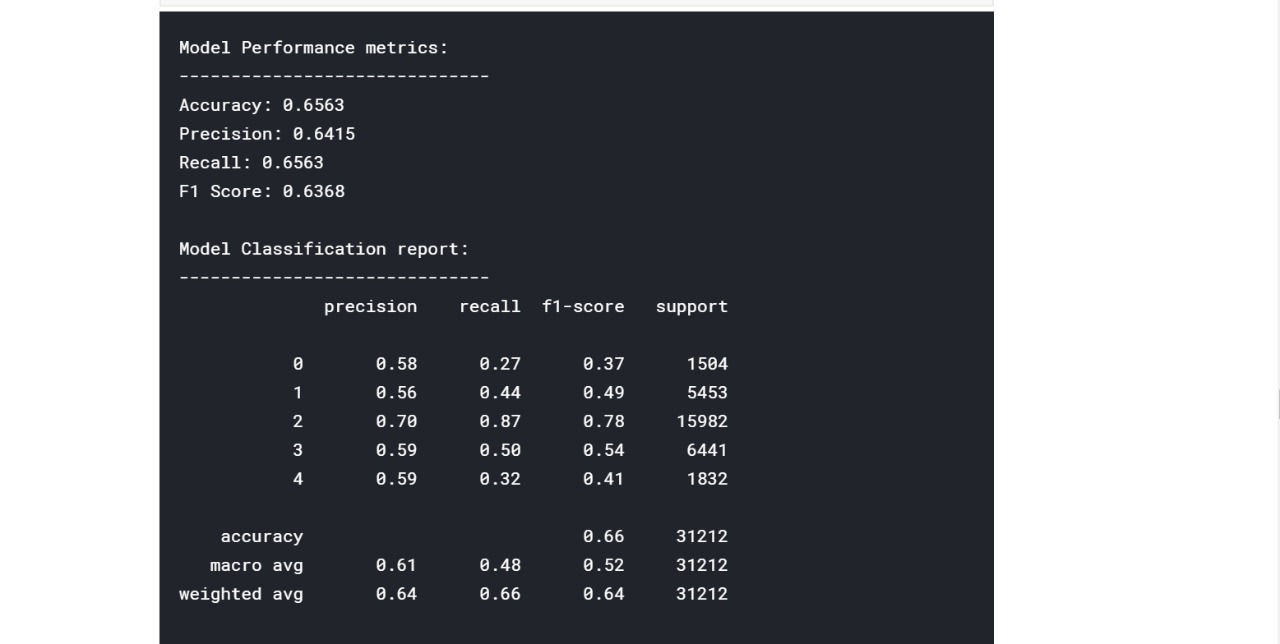


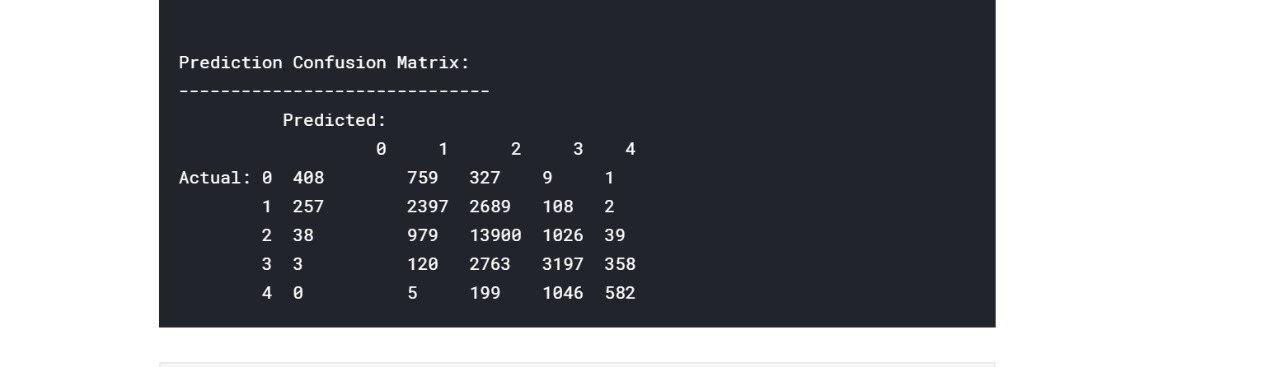


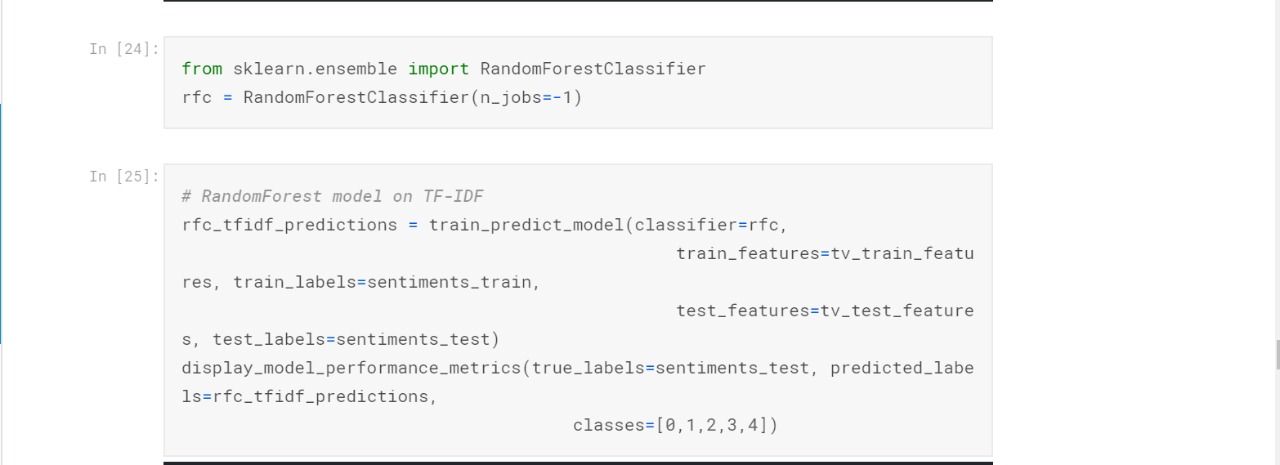


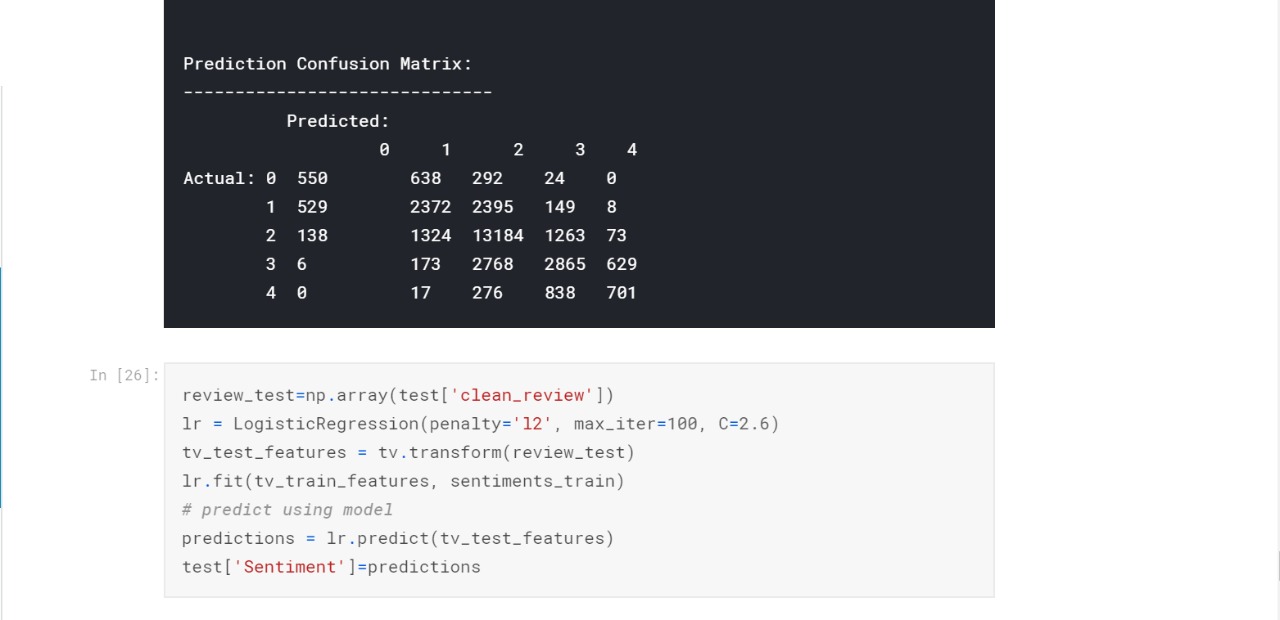














**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Summary of how you did it?

* First we load the input file Train.csv. Then we proceed with cleaning and preprocessing of reviews by converting the reviews to lower case, splitting into words and removing stop words and removing null values.
* We then find the Term Frequency data to see what kind of words are used in the movie reviews and how many times have been used. We use CountVectorizer to calculate term frequencies for each class.
* Count Vectorizer counts the number of words in the document i.e. it converts a collection of text documents to a matrix of the counts of occurrences of each word in the document.
* We perform feature engineering and select review as feature and sentiment as target.
* TF-IDF weighs down the common words occurring in all the documents and give more importance to the words that appear in a subset of documents. It works by penalising these common words by assigning them lower weights while giving importance to some rare words in a particular document. We build TFIDF features on train reviews and transform test reviews.We then run Logistic regression and predict the model using testData.csv
* Finally, we store the result in the csv file which at the end will be submitted to the competition.
* The limitations we face were the fact that we could not incorporate things like negation, sarcasm, language ambiguity etc. Also accuracy of classification can improve if set n-gram range to bigrams or trigrams