**New Jersey Institute of Technology**

**CS 644 Introduction to Big Data**

**Fall 2021**

**Date: 20 December 2021**

**Twitter Sentiment Analysis Using Spark Streaming**

**Professor:** Yao Shen

**By:**

Swapnil Aryan Sinha

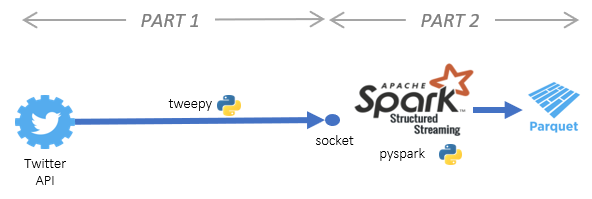
Introduction:

This is a group project based on implementing Machine Leaning techniques like Classification Algorithms and Pattern Recognition on Twitter data to perform sentiment analysis. For this project, we have used Python 3.8, Apache Spark (2.3.1), a Big Data Analytics tool. We are using Python to stream real time Twitter data into VM instance using Apache Spark. MLlib was then used to train the model with labeled training data (Apache Spark Machine Learning Library). We performed sentiment analysis on batches of streamed data using the trained model to classify them as positive, negative, or neutral sentiment values.

Data Streaming is a method of delivering data in a steady and continuous stream that may be processed. With the advent of the Internet, streaming technologies are becoming increasingly significant. Spark Streaming specifically helps us in streaming real-time data from sources such as Twitter, the stock market, and geographic systems, and perform advanced analytics to assist enterprises.

According on the sentiment of the contents of the tweets, all of them are labeled as Positive, Neutral, or Negative. The Sentiments of Tweets output is organized into folders and files based on when they were created. As needed, this output can be saved to the local file system or HDFS. The Natural Language Processing library from Stanford is used to perform sentiment analysis. It takes text input and sends it to this library, which returns the sentiment. This library creates a tree-like structure out of the plain text it receives; this structure is built after the data has been cleaned and all stop words have been removed.

**Architecture:**



**Acquiring tweets and sending them from the Twitter API:**

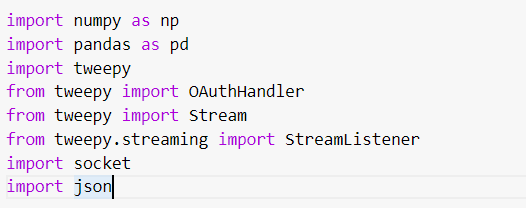
First, we have to register on Twitter Apps, then, go to https://developer.twitter.com/en/apply-for-access.html and apply for a twitter developer account. Log in to the account. Then, click “Create New App”, fill out the form, and click “Create your Twitter application”. On the next page, click on the “API keys” tab, and copy your “API key” and “API secret”. After getting the keys, scroll down and click “Create my access token”, copy your “Access token” and “Access token secret”.

We utilize our developer credentials to authenticate and connect to the Twitter API in this section. We also establish a TCP socket between the Twitter API and Spark, which waits for the Spark Structured Streaming call before sending the Twitter data. For connecting to the Twitter API and retrieving tweets, we use Python's Tweepy package.

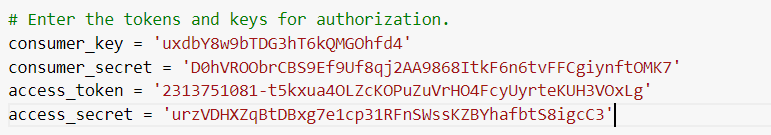
**Tweet preprocessing and sentiment analysis:**

We receive data from the TCP socket in this section and preprocess it with the pyspark package, which is Python's Spark API. We then use textblob (a Python package for processing textual data) to do sentiment analysis. We save the tweet and the sentiment analysis scores in a parquet file (data storage type) after sentiment analysis.

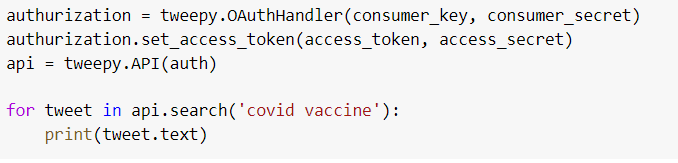
**Working:**



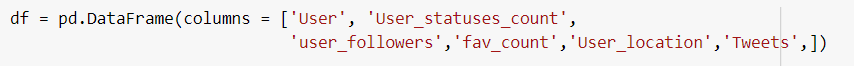
Import the necessary libraries.



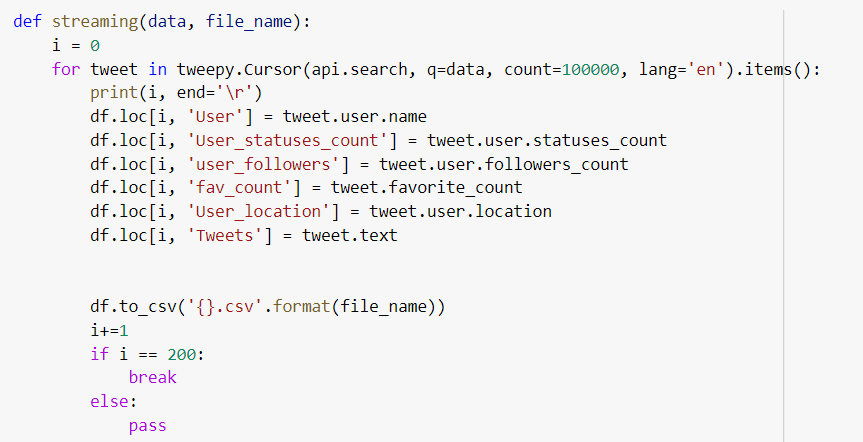
Enter the respective keys obtained from the twitter developer account.



Authorize the consumer key and consumer secret key along with access token and access secret tokens. Now, we are searching for all the tweets which have the words “Covid Vaccine” in the API. We use the classes Stream and StreamListener to create the stream, and OAuthHandler to authenticate with Twitter. To construct a communication channel between our local system and the Twitter API, we import the socket module. To handle the data of JSON objects, we'll require the JSON module.



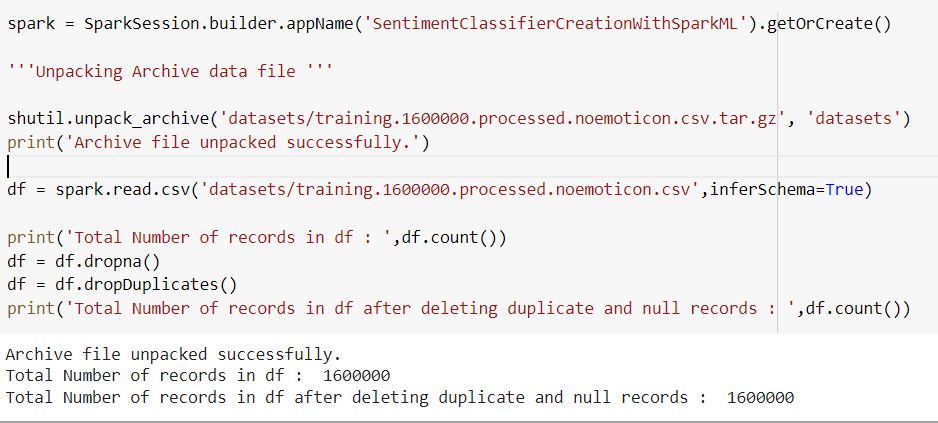
Assign the data to a pandas dataframe df specifying all the individual columns.



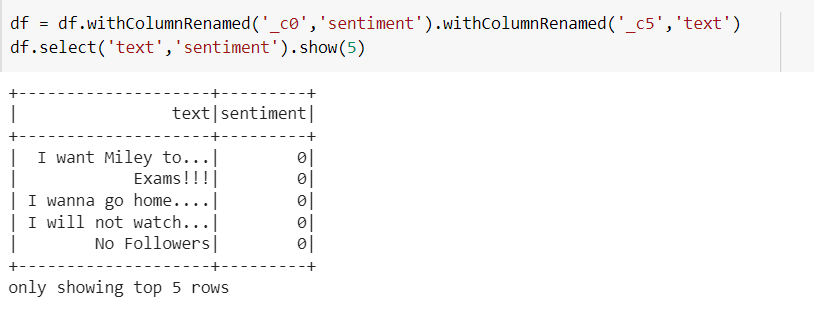
Define a function to stream the data which is in English Language and save them as new csv files.



Test the approach on tweets containing the words “Joe Biden”.

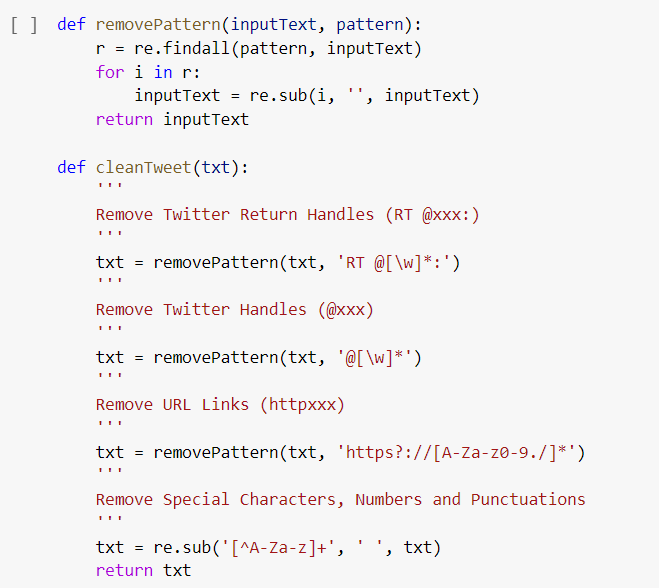


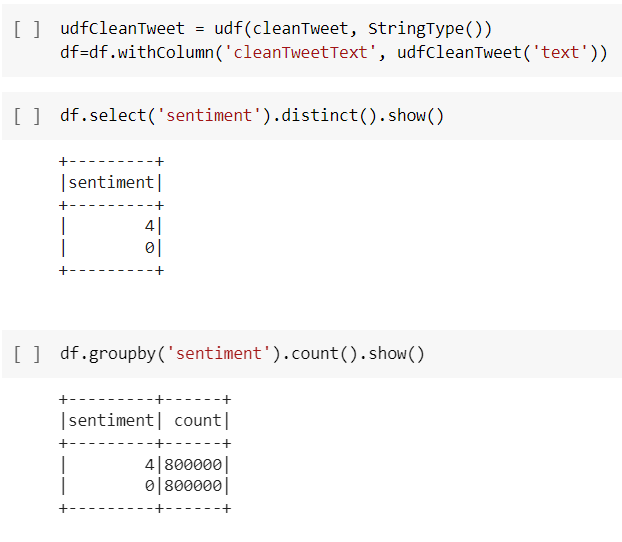
We load the data to the spark dataframe.

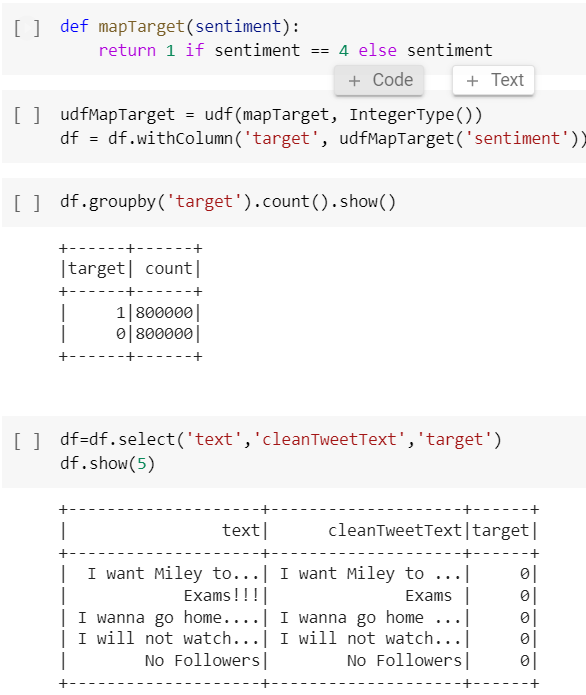


The dataframe is set to display the text and its corresponding sentiment.

Now, the data is cleansed and prepared to be used.

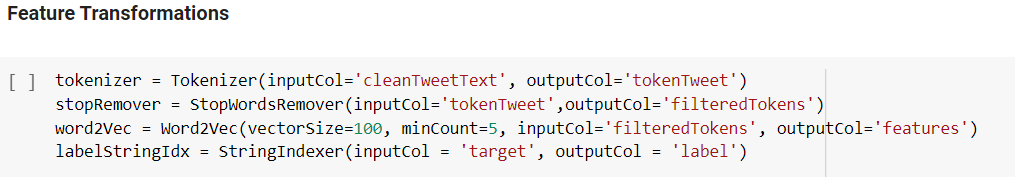






Now that the data is clean and processed, we can split it into train and test samples using randomSplit where 80% is the training data and 20% is the testing data.



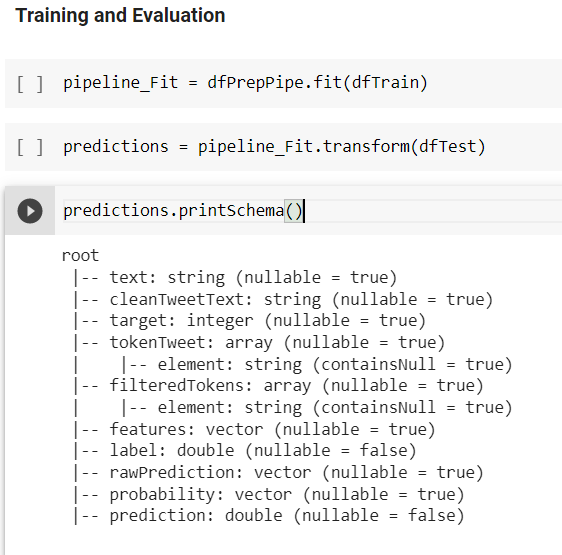


Creating the model and the pipeline

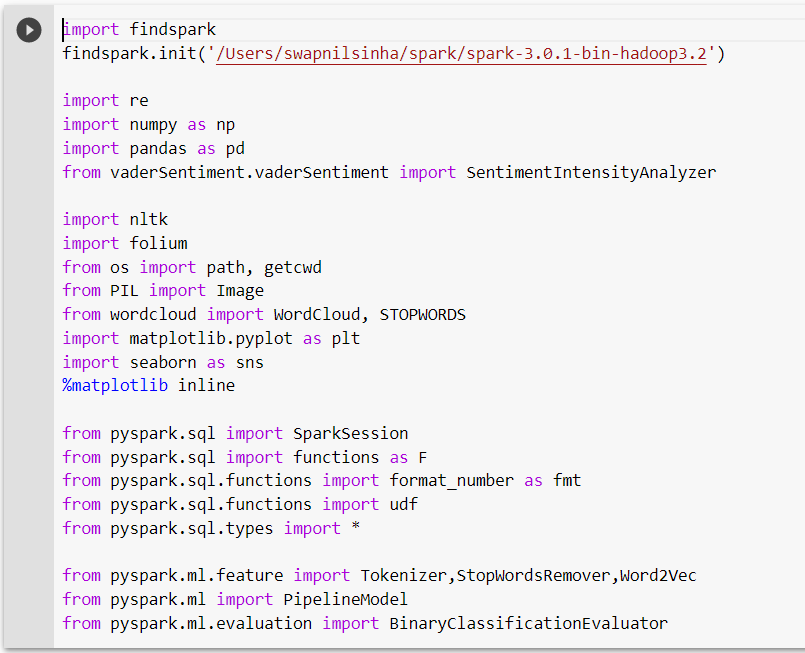


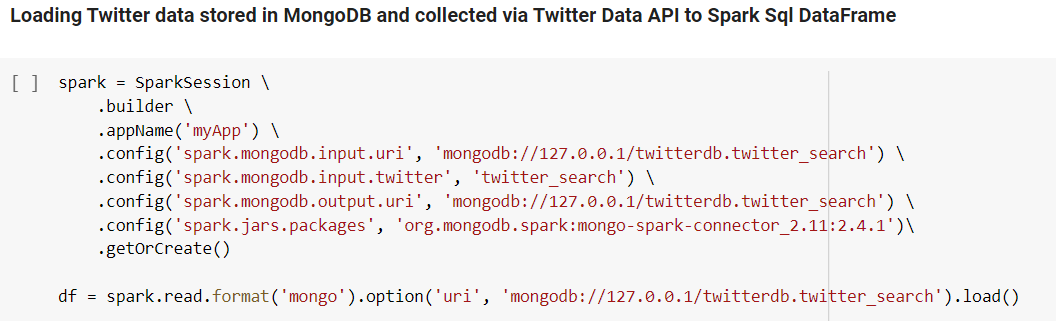
We use the logistic regression model and then create a pipeline with defined stages.

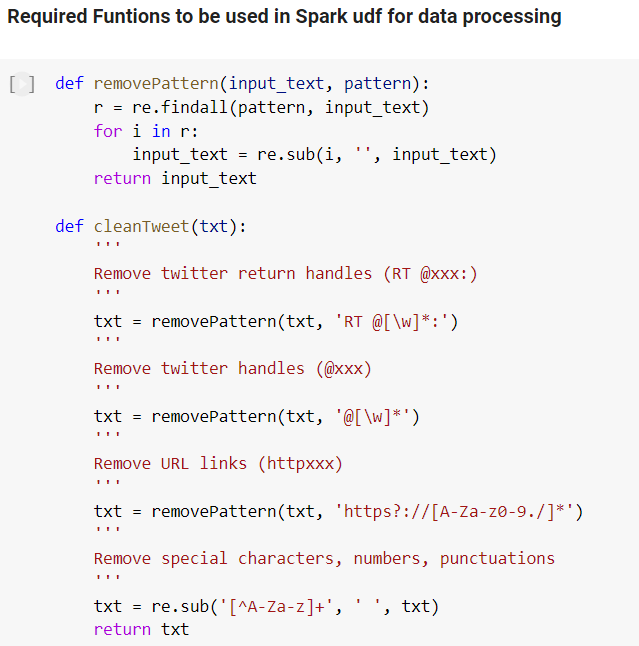
Now, we perform the training phase.

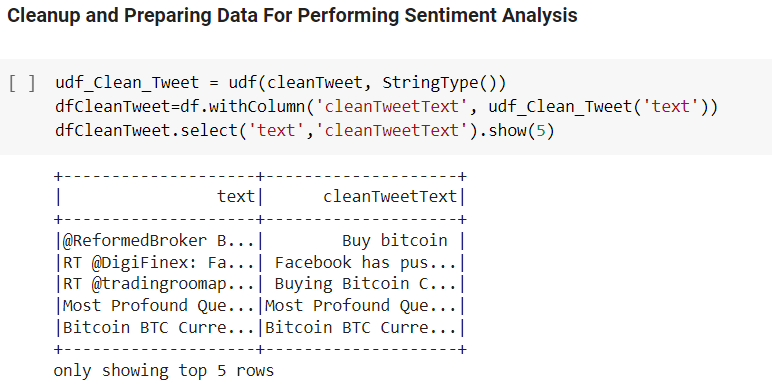






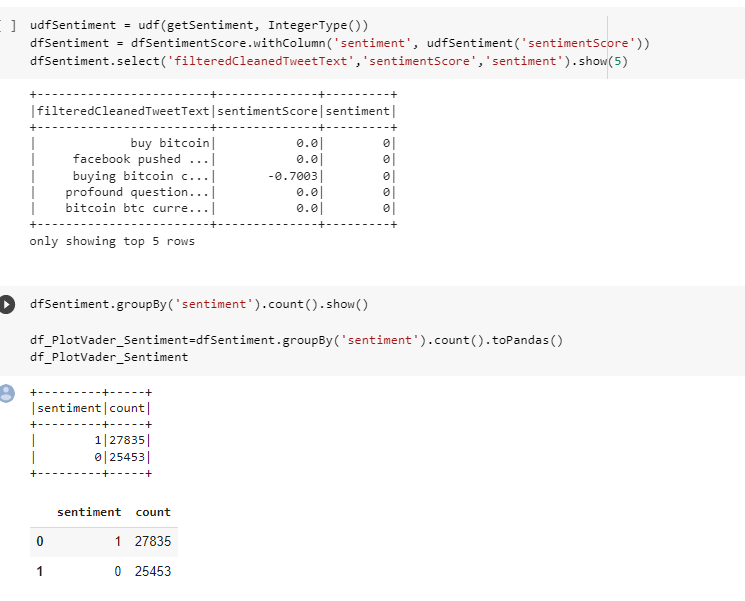


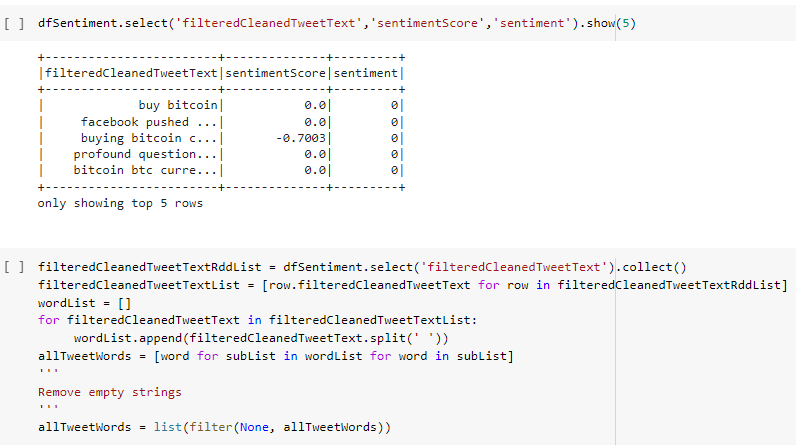




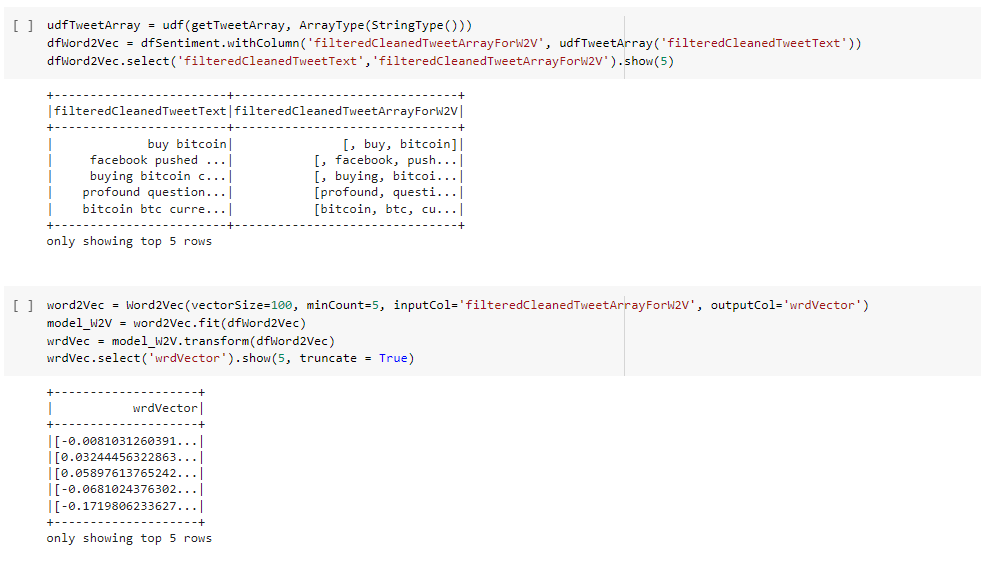


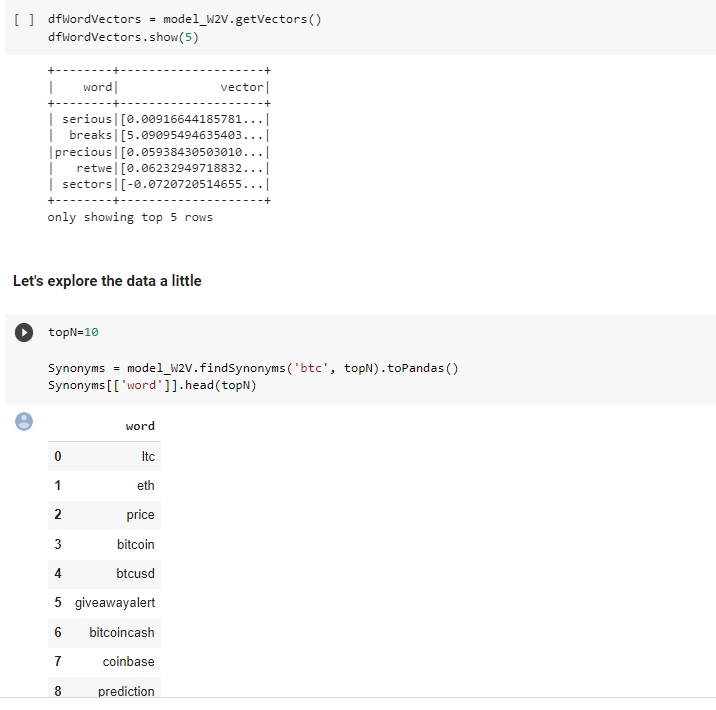


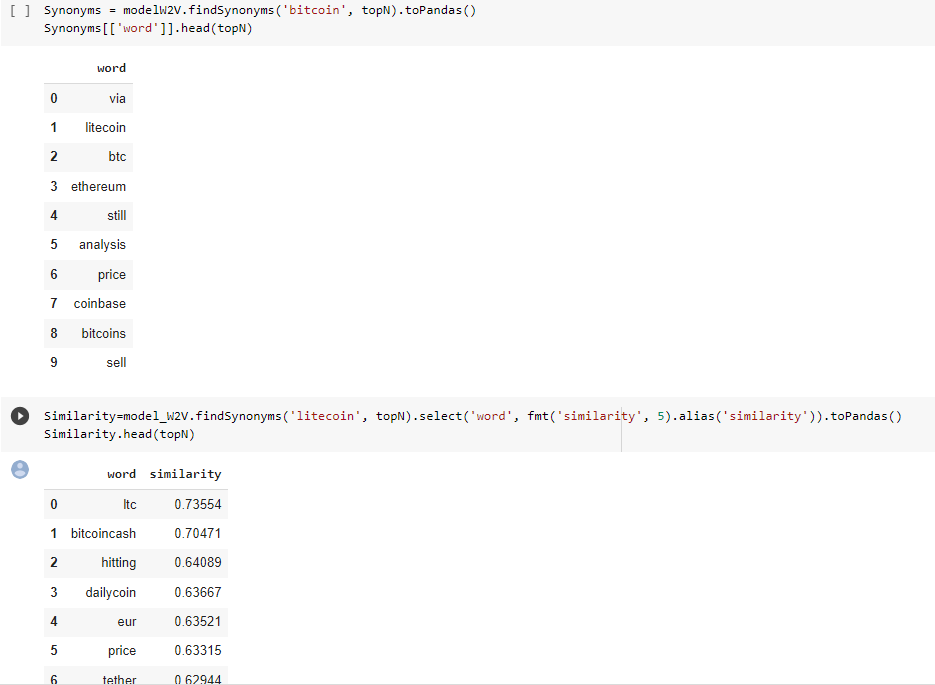


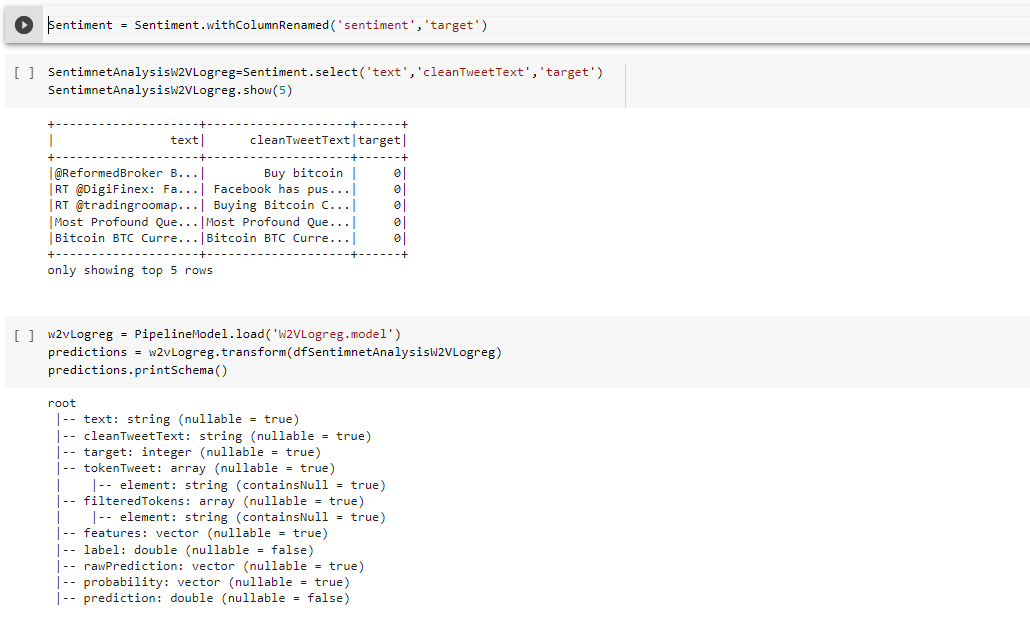


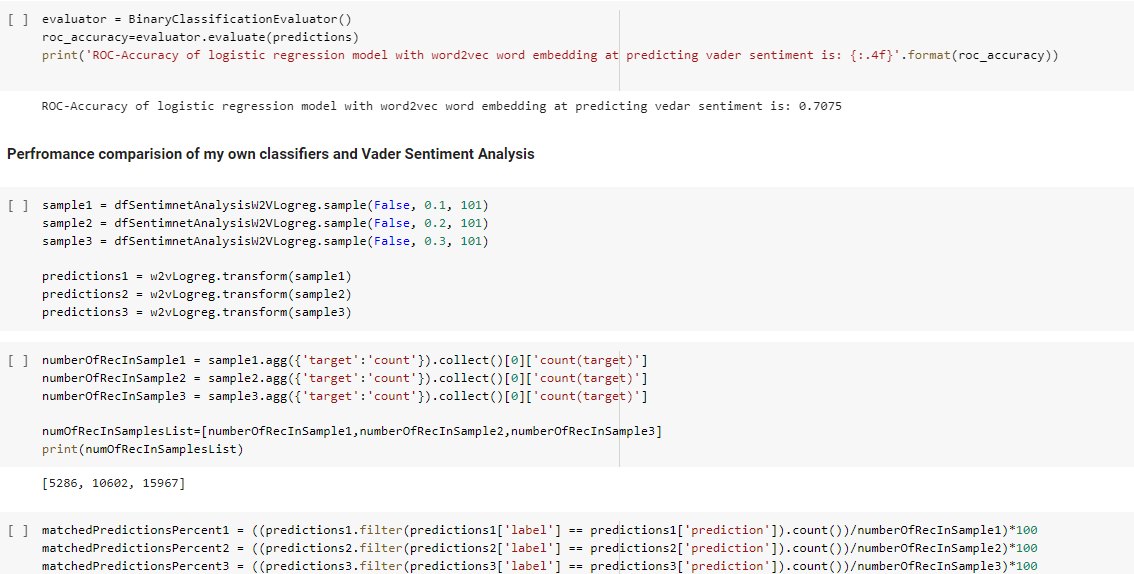


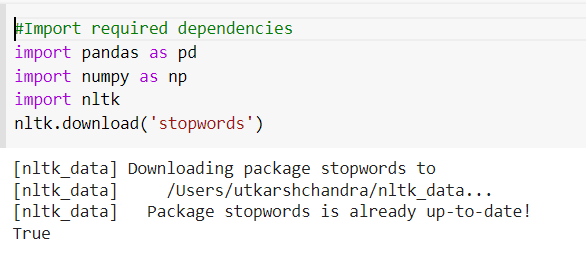


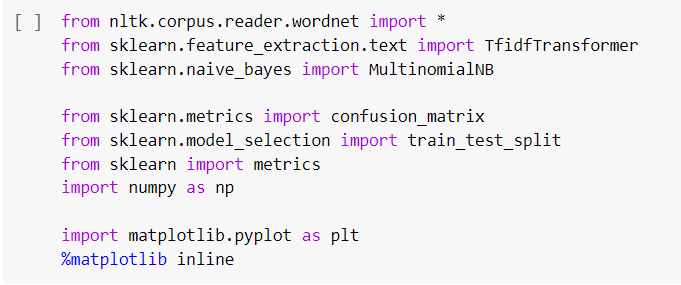


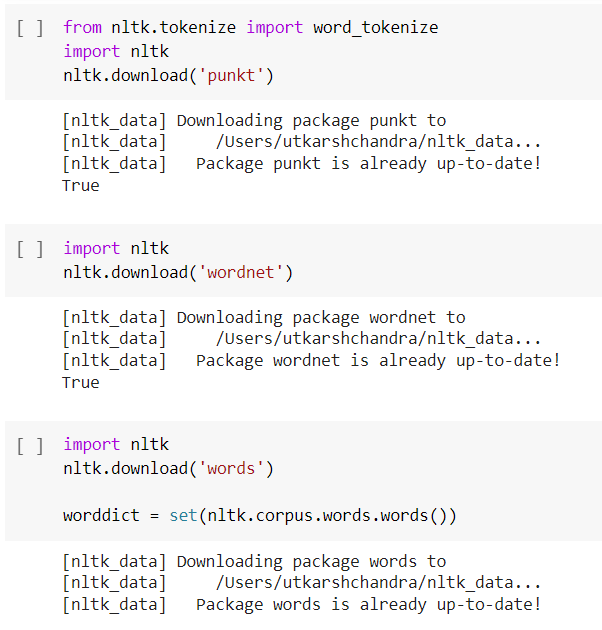


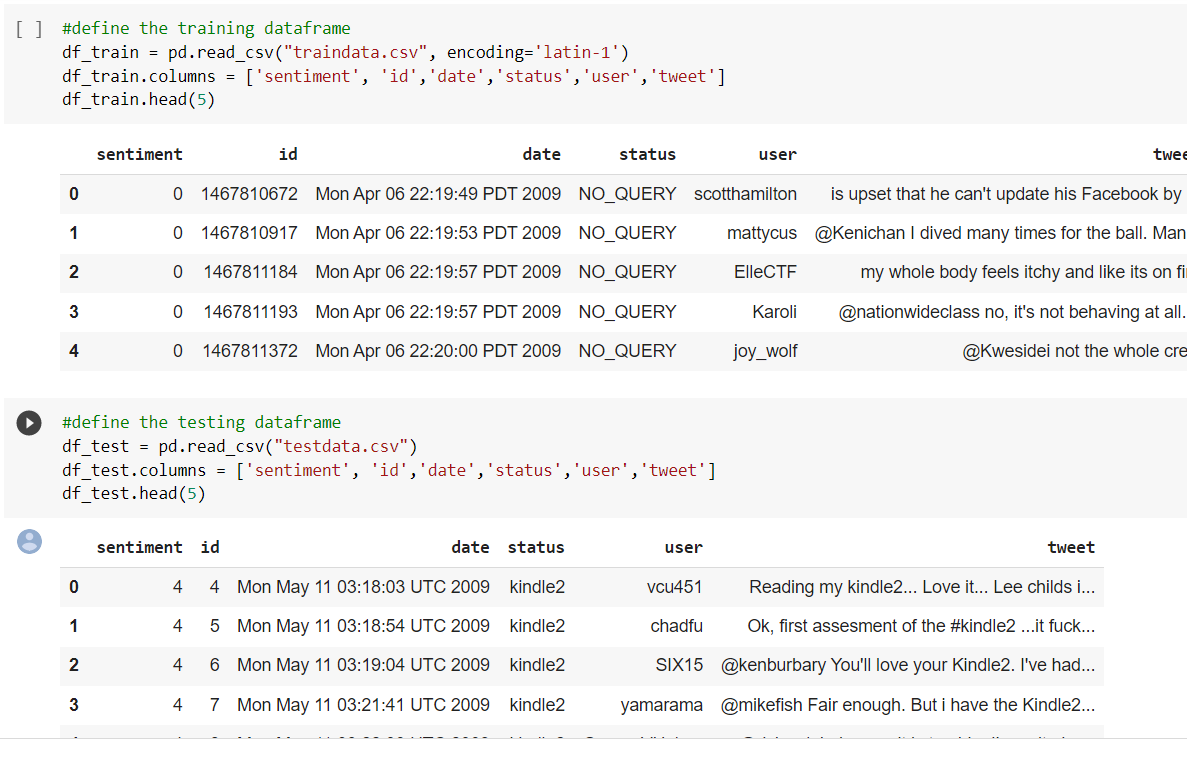


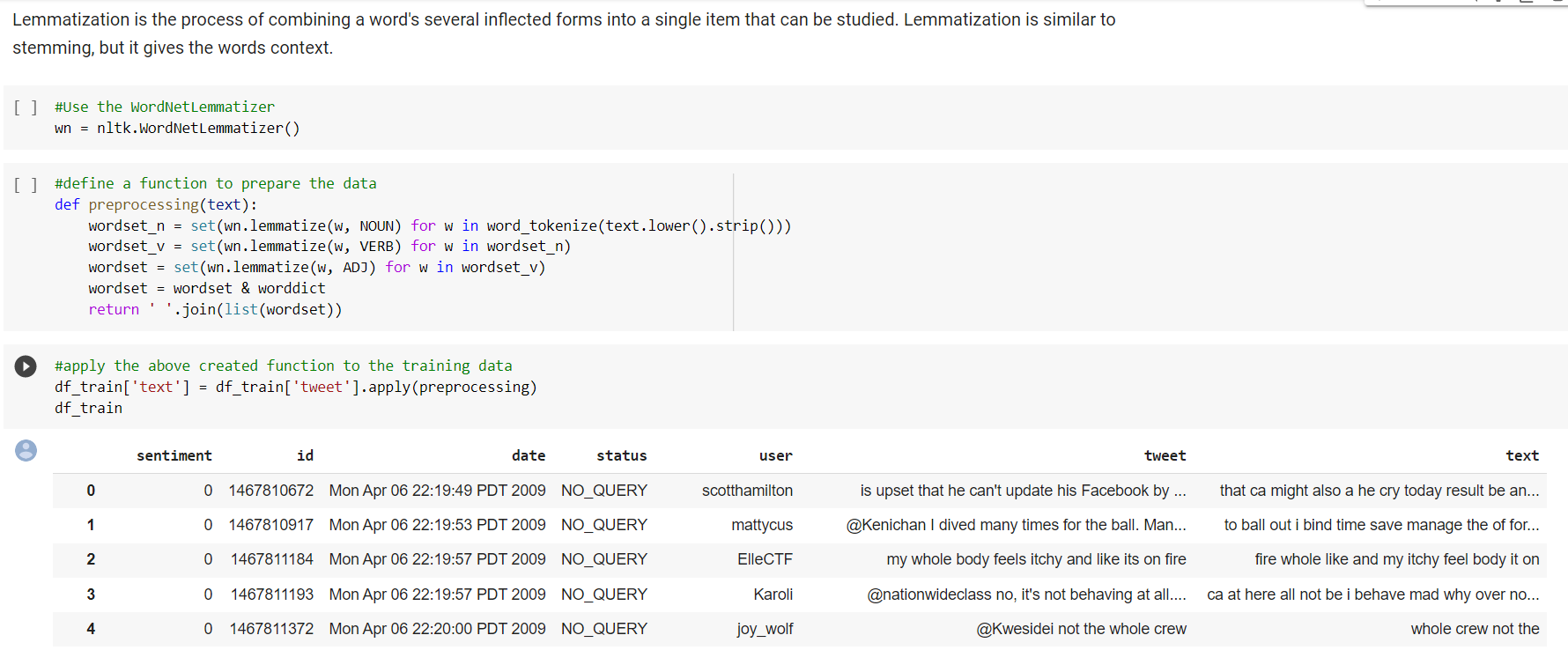


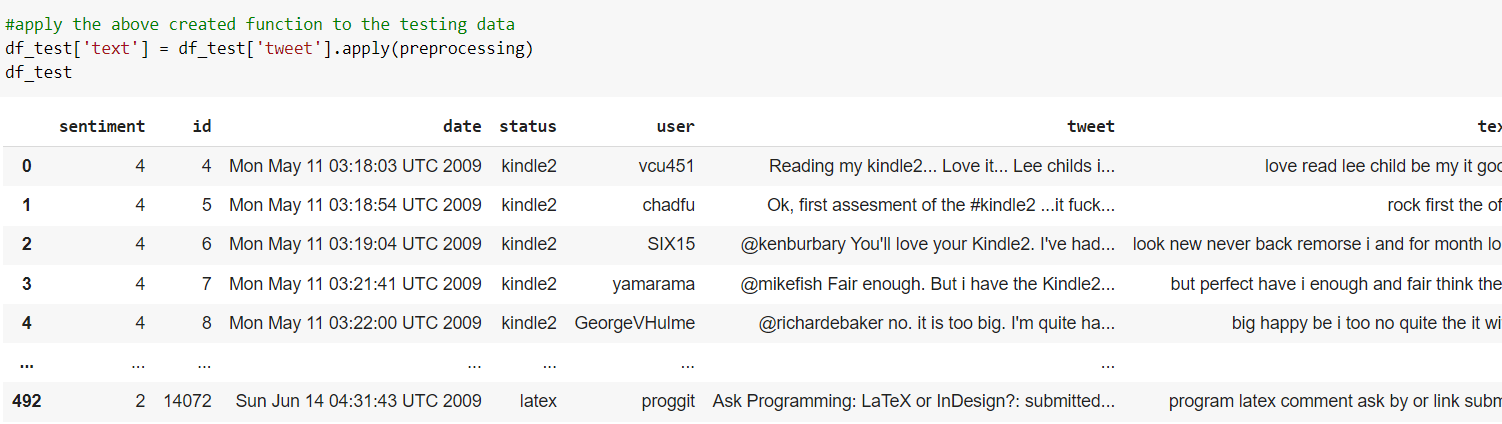


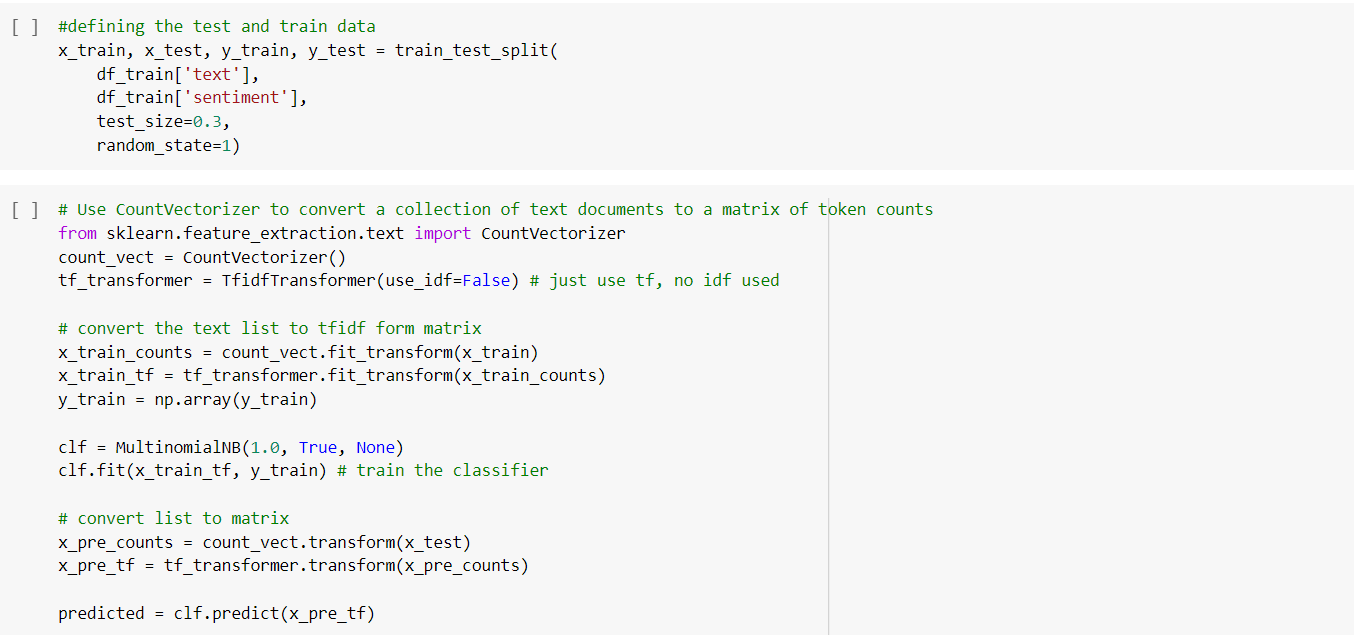


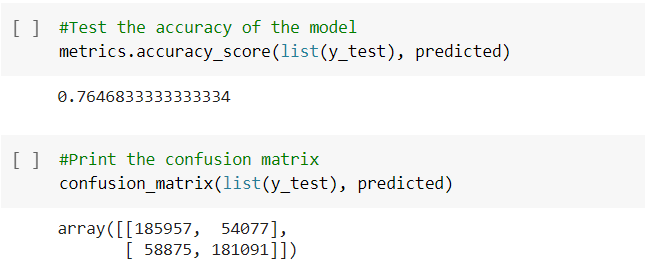












**Conclusion:**

Upon evaluating all the models, we can conclude that Naïve Bayes is the best model for the above given dataset, The naïve Bayes model is following the principle of Bayes theorem which defines that it uses the probability of an event based on its association with another event. Since our dataset has sentiment analysis, we draw an association between positive and negative. Therefore, Bayes theorem concepts achieve better performance on this dataset.