# 

**CSCI 5408**

**Summer 2018**

**Assignment 3**

**Classification through Apache Spark streaming using**

**Twitter data**

**Submitted by:-**

**Nikhil Dhirmalani: B00775542**

**Khushboo Siwal: B00781497**

**TABLE OF CONTENTS**

1. Task Description
2. Twitter Tweet Extraction
3. Sentiment Analysis Algorithm
4. Labelling training Data
5. Feature Selection
6. Output
7. Code Submission
8. References

**LIST OF FIGURES**

[Figure 1 Flow diagram for tasks performed 4](#_Toc517626739)

[Figure 2 : New app created on apps.twitter.com 5](#_Toc517626740)

[Figure 3 : Screenshot from the app on apps.twitter.com 5](#_Toc517626741)

[Figure 4 Access Token details 6](#_Toc517626742)

[Figure 5 :Consumer Key and Consumer Secret details 6](#_Toc517626743)

[Figure 6 Code for Socket 6](#_Toc517626744)

[Figure 7 Another Code Snippet for Socket 7](#_Toc517626745)

[Figure 8 Tweets Stream 7](#_Toc517626746)

[Figure 9 S-Shaped Curve for Logistic Regression 8](#_Toc517626747)

[Figure 10 Training Data Screenshot 9](#_Toc517626748)

[Figure 11 Model with Count Vectorizer 10](#_Toc517626749)

[Figure 12 Model with Word2Vec 10](#_Toc517626750)

[Figure 13 Model with TFIDF 11](#_Toc517626751)

[Figure 14 Model on Tweet Stream 11](#_Toc517626752)

1. **TASK DESCRIPTION**

The tasks for this assignment involved us to stream data from social media platforms and learn about tool Apache Spark. We have to train our model based on some machine learning technique from the tweet data provided and use that pre-trained model on streaming tweets in order to classify them as positive, negative and neutral.

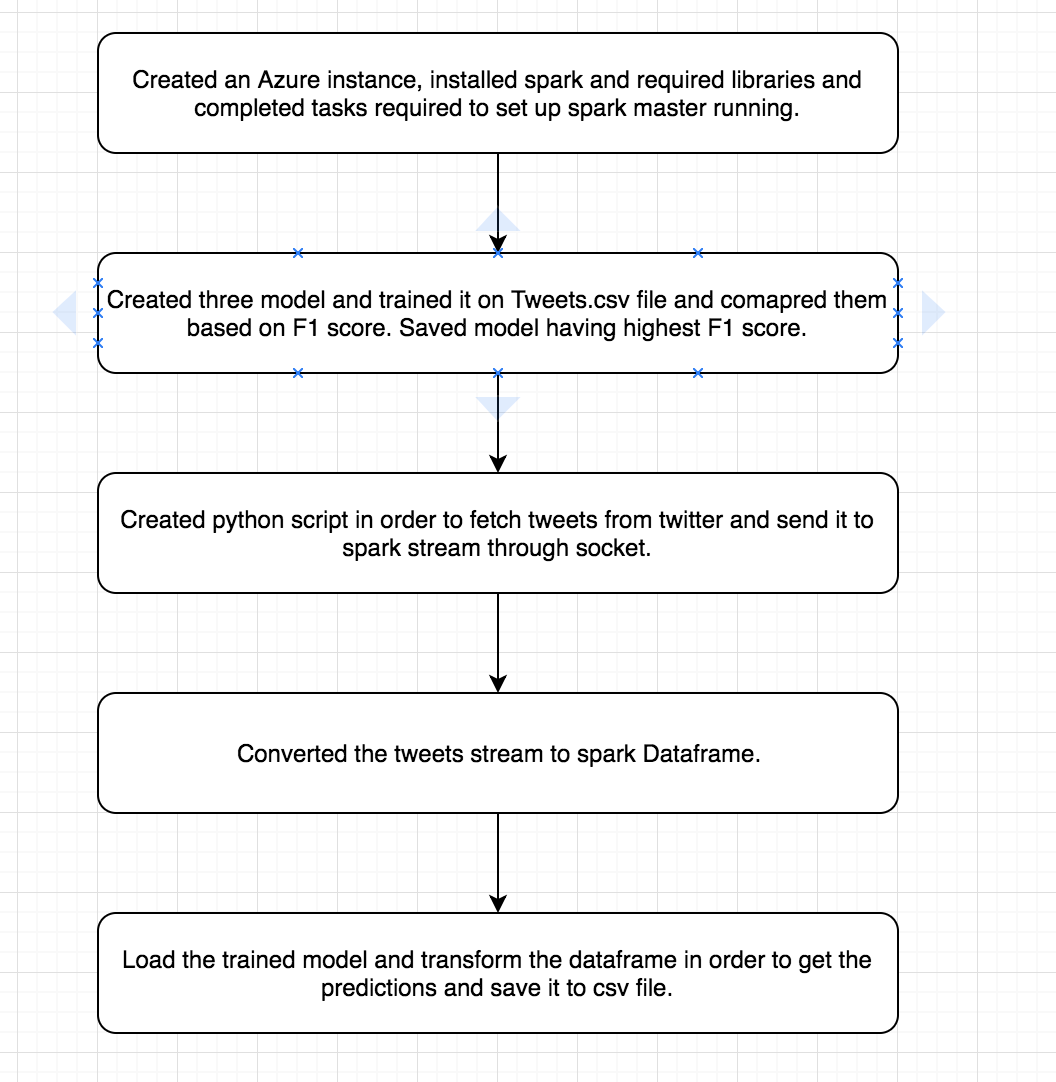


Figure 1 Flow diagram for tasks performed

1. **TWITTER TWEET EXTRACTION**

We followed the steps given in tutorial for assignment 2, first we created an account on twitter and after that we created a new app as shown in figure.

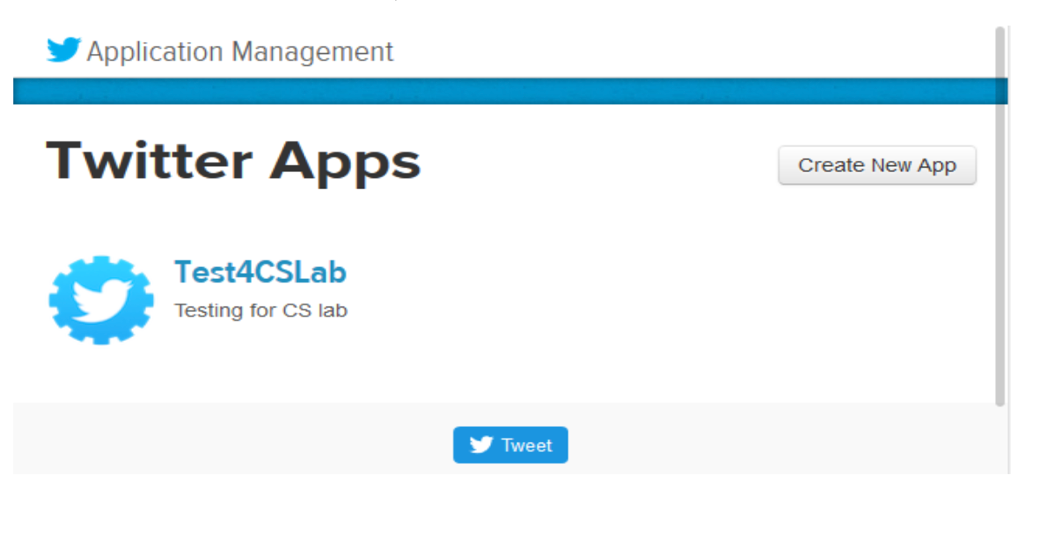


Figure 2 : New app created on apps.twitter.com

We used the same app CSCI5408\_Assignment2 created for assignment 2 and generated the key and access token required for authentication, while fetching the tweets as shown in figure.

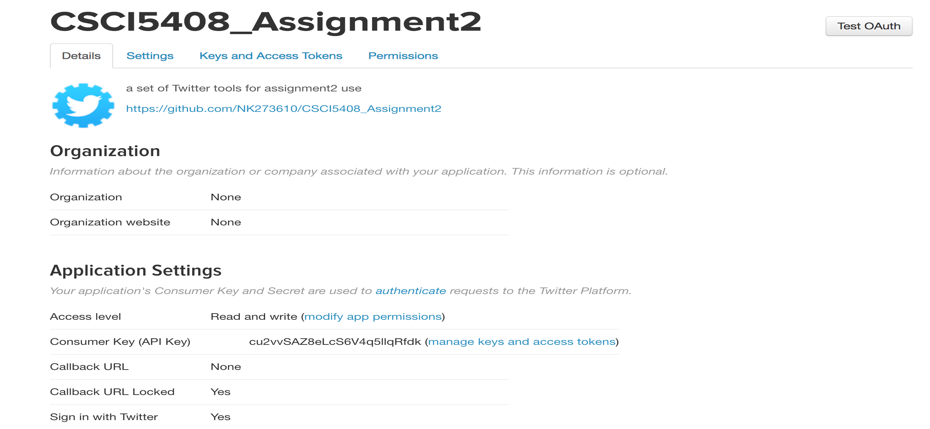


Figure 3 : Screenshot from the app on apps.twitter.com

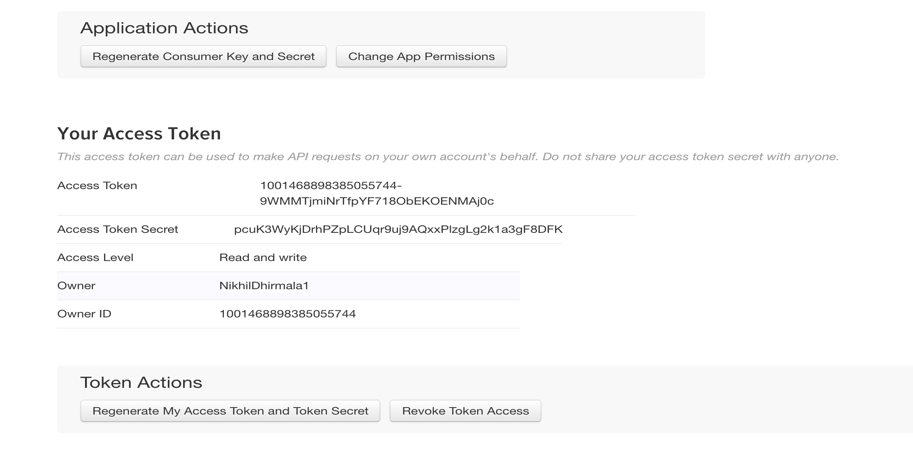


Figure Access Token details

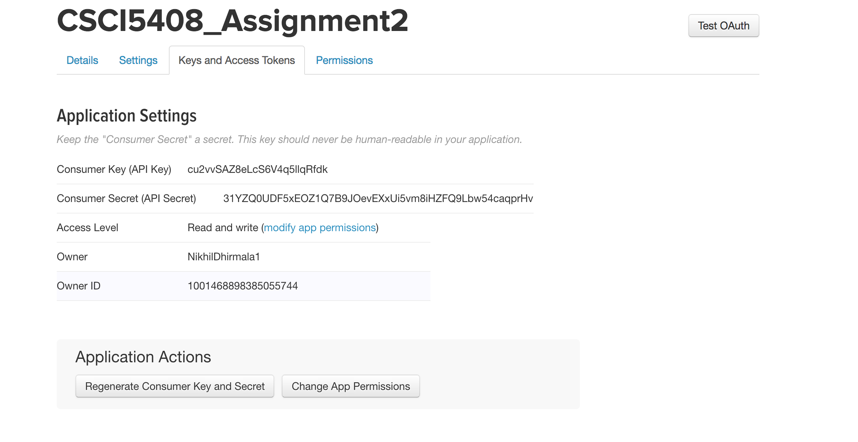


Figure :Consumer Key and Consumer Secret details

After completing this step, we imported required libraries and wrote a script in python to stream tweets, mentioned in assignment for sample query movies. Before doing this, we have also created a socket in order to send the streaming tweets to spark stream. The code for socket and sending it to spark stream is given below:

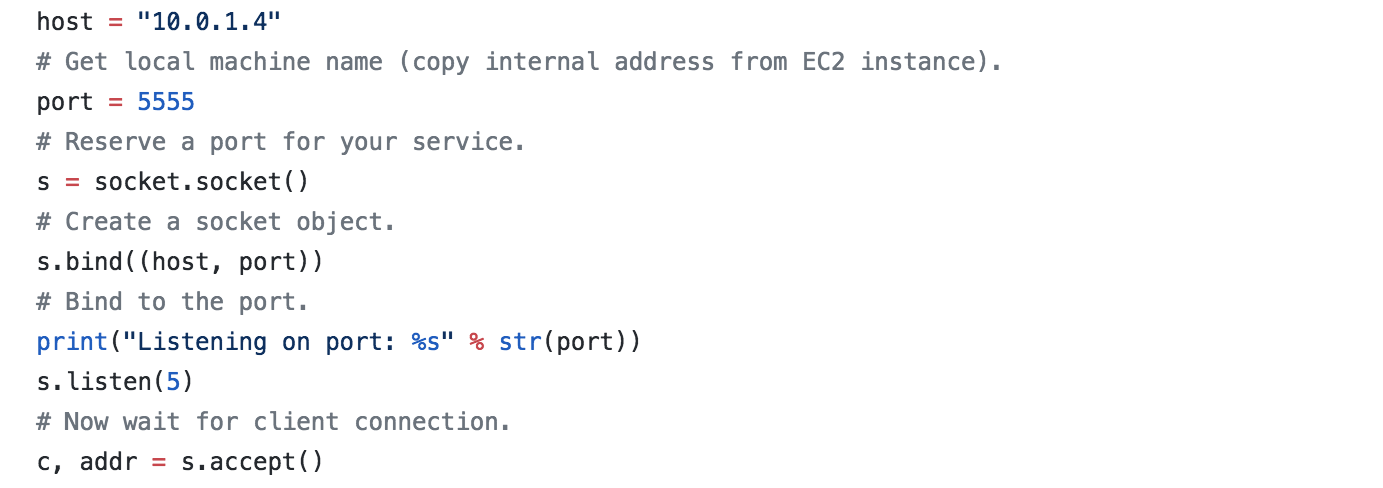


Figure 6 Code for Socket

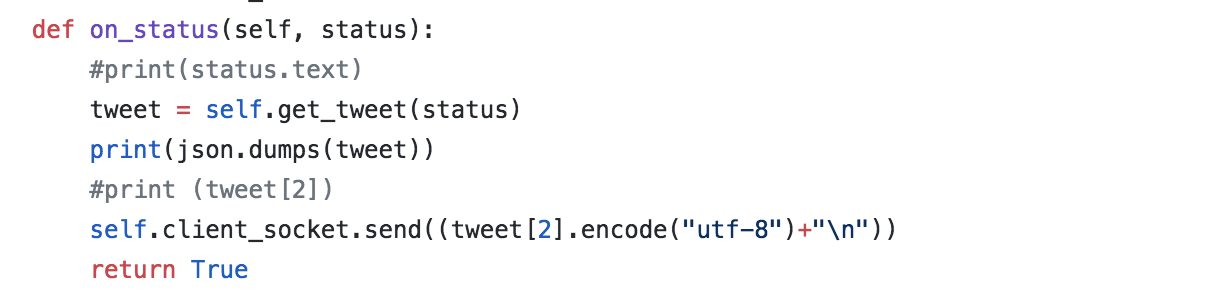


Figure 7 Another Code Snippet for Socket

Stream of tweets:

****

Figure 8 Tweets Stream

1. **SENTIMENT ANALYSIS ALGORITHM**

In this assignment we first created temp.py file which trains three models from Tweets.csv data and saves the model. Here for all the three models we have used Logistic Regression as cost function but for first model we have used CountVectorizer, for second one we used Word2Vec and for third one we have used TFIDF.

**Logistic Regression Algorithm**:-

Logistic regression is a simple algorithm that can be used for binary/multivariate classification tasks. It predicts categorical outcomes (binomial / multinomial values of y).

The predictions of Logistic Regression are in the form of probabilities of an event occurring, i.e. the probability of y=1, given certain values of input variables x. Thus, the results of Logistic Regression range between 0-1.

Logistic Regression models the data points using the standard logistic function, which is an S- shaped curve given by the equation:

https://www.kdnuggets.com/wp-content/uploads/EQN.-std-logistic-fn.png



Figure 9 S-Shaped Curve for Logistic Regression

The equation to be solved in Logistic Regression is:

https://www.kdnuggets.com/wp-content/uploads/fla_LogR-300x17.png

where:

* p = probability that y=1 given the values of the input features, x.
* x1,x2,..,xk = set of input features, x.
* B0,B1,..,Bk = parameter values to be estimated via the maximum likelihood method. B0,B1,..Bk are estimated as the ‘log-odds’ of a unit change in the input feature it is associated with.
* Bt = vector of coefficients
* X = vector of input features

Since here we want discrete output we have used logistic regression as our cost function.

In CountVectorizer we create tweet-word matrix where we map the no of times the word occurs in tweet to the respective tweet. In TFI[[1]](#footnote-1)DF we normalize it with the number of tweets in which the word occurs. In Word2Vec we randomly generate the vectors for each words and try to learn it.

We then split our data from csv into train and test data and train our model using train data. Then we test our model using test data and check the accuracy for each of the models on our test data. Based on this we have selected the model with highest accuracy i.e. Logistic Regression with CountVectorizer. [1]

1. **LABELLING TRAINING DATA**

We have used Tweets.csv as our training data that was given during tutorial. The data is given below. We have used String Indexer for column airline\_sentiment which generates three values 0, 1 and 2 for negative, positive and neutral tweets. We have also used RegexTokenizer in order to get words from the texts and CountVectorizer, TFIDF, Word2Vec etc to generate tweet-word matrix.

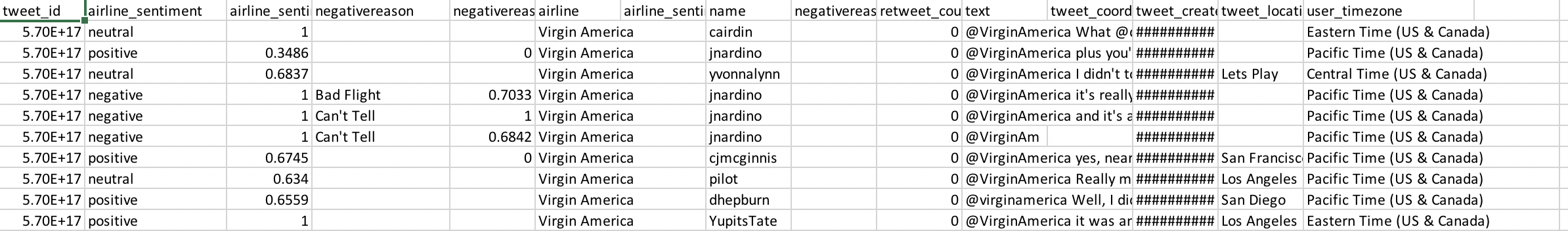


Figure 10 Training Data Screenshot

1. **FEATURE SELECTION**

From the data we selected text and airline sentiment features and used it in order to train our model. We also removed the rows having null values. Moreover, we have only taken into consideration text from the csv as feature on which model is trained. The airline sentiment is used as class label.

1. **OUTPUT**

We calculated the F1 score for each model on our test data and based on that we selected the model having highest F1 score.

Model with CountVectorizer.

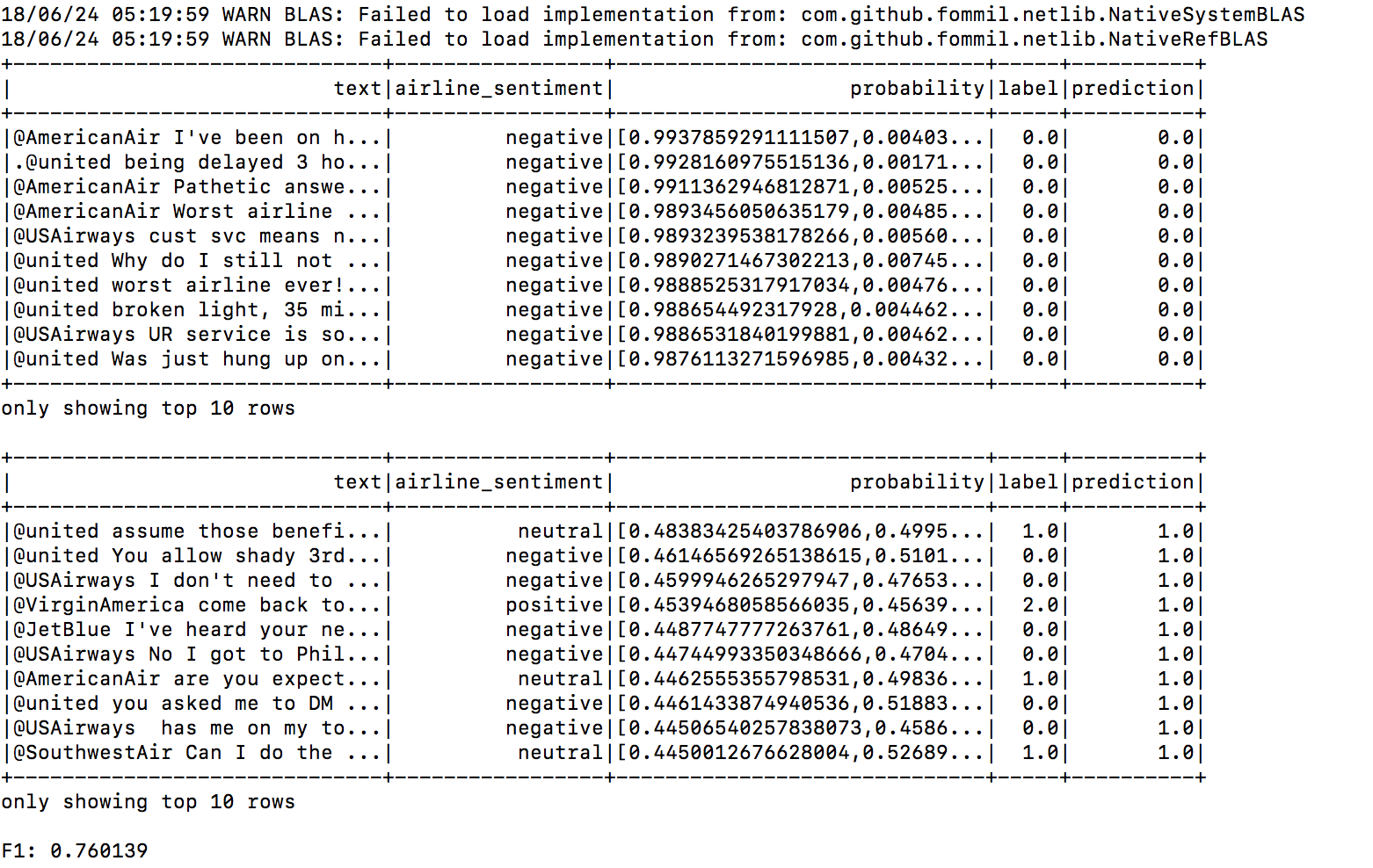


Figure 11 Model with Count Vectorizer

Model with Word2Vec

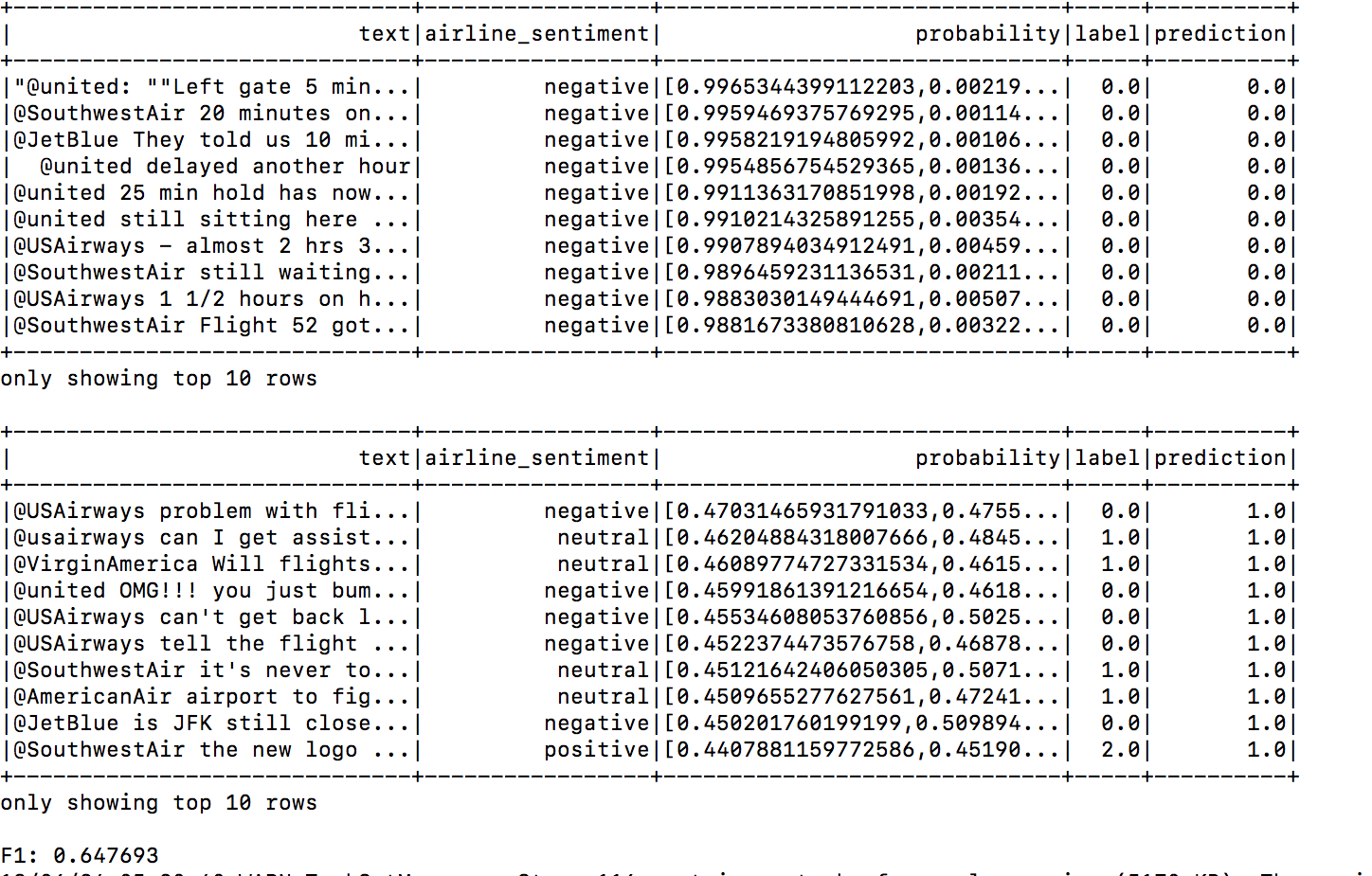


Figure 12 Model with Word2Vec

Model with TFIDF:

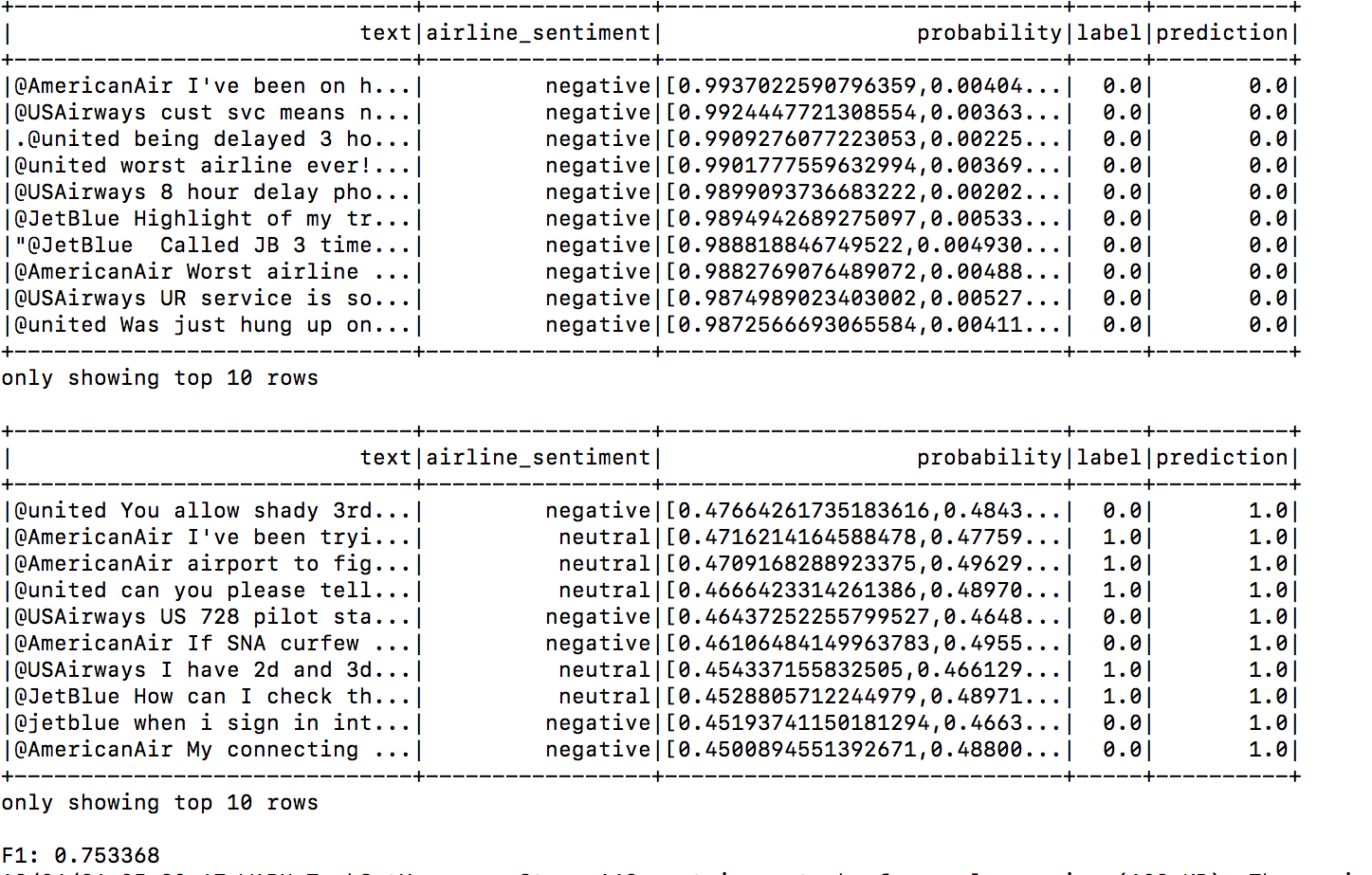


Figure 13 Model with TFIDF

As shown Model with CountVectorizer has highest F1 score so we selected this model in order to perform sentiment analysis on stream of tweets.

The output of our model on tweet stream is shown below:

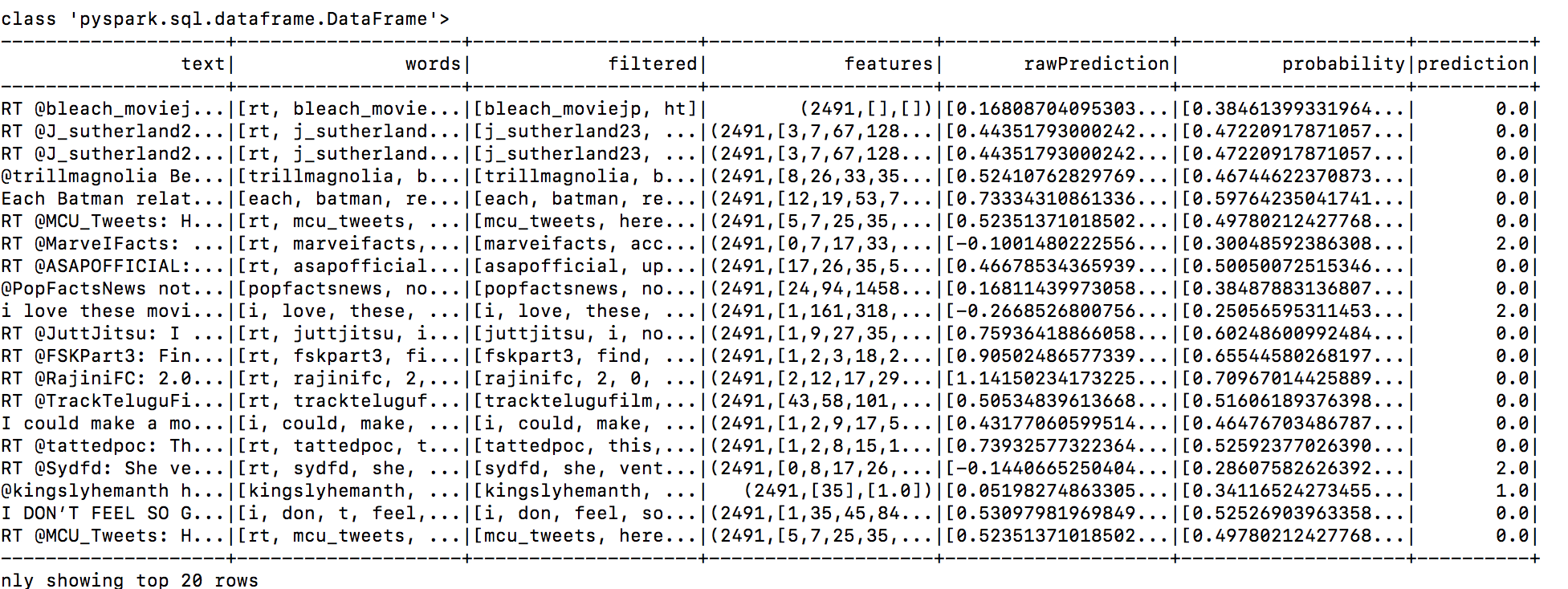


Figure 14 Model on Tweet Stream

Where 0 means if tweet has negative sentiment, 1 means if tweet have positive sentiment and 2 means if tweet has negative sentiment.

1. **CODE SUBMISSION**

Below is the link for the GitHub repository and all the outputs and code is submitted here.

<https://github.com/NK273610/DataWarehouseAssignment3>

1. **REFERENCES**

**[1]SHAW, R.**

Logistic Regression: A Concise Technical Overview

**In-text:**(Shaw, 2018)

**Your Bibliography:**Shaw, R. (2018). Logistic Regression: A Concise Technical Overview. Retrieved from https://www.kdnuggets.com/2018/02/logistic-regression-concise-technical-overview.html

1. Referred from Shaw, R. (2018). Logistic Regression: A Concise Technical Overview. Retrieved from https://www.kdnuggets.com/2018/02/logistic-regression-concise-technical-overview.html [↑](#footnote-ref-1)