Twitter Sentiment Analysis Based on Ordinal Regression

On internet opinion (sentiments on topic) mining is helping users in knowing the quality of any organization or products, if any user has good experience on any product or company then he will express good reviews/opinion and by seeing this opinion others users can know the quality of the product, in today’s online social networks like twitter all peoples expressing their opinions and social networking sites developing new techniques to detect sentiments from this opinions, all existing techniques used to discover either Positive or Negative or Neutral sentiments from topics but this paper proposes 5 levels of sentiments detection such as High Positive, Moderate Positive, Neutral, High Negative and Moderate Negative. To detect sentiments author is using 4 Ordinal Regression machine learning algorithms such as Softmax, Decision Tree, Random Forest and Support Vector Regression.

Ordinal Regression means classifier used many independent variables to predict class of given data, In this paper also we give tweets as input and classifier predict sentiment by using all independent words from this tweets. Ordinal regression is a statistical technique that is used to predict behaviour of ordinal level dependent variables with a set of independent variables. The dependent variable is the order response category variable and the independent variable may be categorical or continuous.

From above algorithms Decision Tree is giving better prediction result and to train all algorithms we are using publicly available twitter dataset from NLTK library. We are using many features from NLTK (Natural Language Processing Tool Kit) library such as cleaning tweet text by removing special symbols, removing stop words (such as the, then, where etc.), word stemming which means removing ing, tion etc from words. After cleaning tweets then we will convert all tweets to BOG (Bag of Words Dictionary) and then convert BOG to vector by calculating TF/IDF (Term Frequency/Inverse Document Frequency).

TF/IDF = number of times word occur in tweet / total number of times word occur in all tweets.

For example

Tweet1 = An apple a day keep the doctor away.

Tweet2 = Apple is good for health.

From above 2 tweets if we want to calculate TFIDF then we need to find all unique words from 2 tweets and count them. Remove all stop words and in below example I am keeping all unique words in column and their count in rows

Apple day keep doctor away good health (unique words from 2 tweets)

T1 1 1 1 1 1 0 0 (count of all words from tweet1)

T2 2 0 0 0 0 1 1

In above table we can see if word appear in tweet then I put value 1, if word not appear in tweet then I put value 0. Now from above table we can calculate TFIDF

TFIDF of word Apple in tweet1 = ½ = 0.5

Apple word appear in tweet1 is 1 time and Apple appear in all tweets (means tweet1 and tweet2) is 2 times, so ½ will give 0.5 TFIDF for Apple Word. Similarly application calculate TFID for all words and form a vector. This vector will be trained with SVR, Random Forest and Decision Tree algorithms with all positive and negative tweets dataset. Whenever we give new tweets then application convert new tweets also in TFIDF vector and then apply new tweet vector on train vector to predict sentiment type. Once we got sentiment value then we can calculate High positive or moderate positive or negative by calculating polarity. Polarity value will come from TFIDF, if new tweet polarity percentage is closer to positive (80 to 100%) then tweet is HIGH Positive, if 50 to 80 then moderate positive else neutral. Similarly negative ratio also calculated.

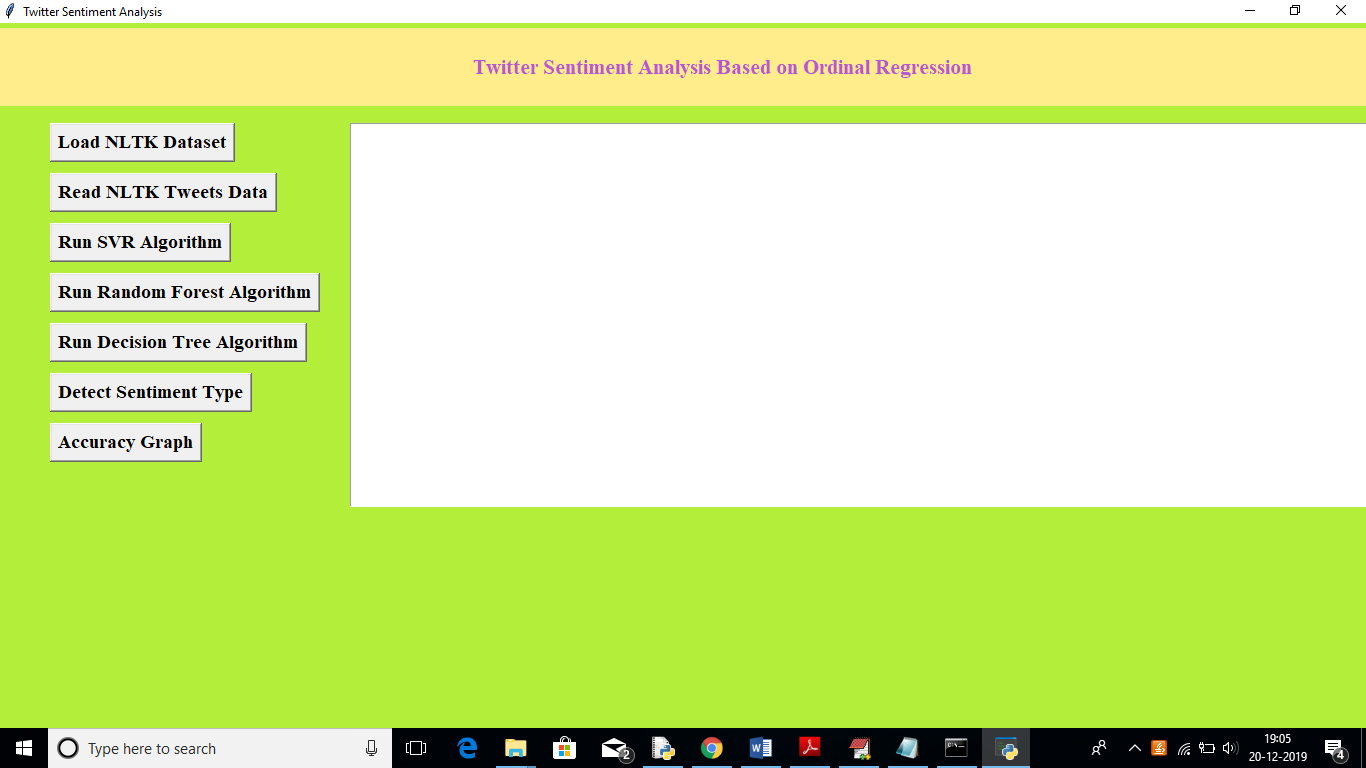
To run this project we need to download NLTK library, to download it just double click on ‘download.bat’ file then one window will appear and you just click on ‘download’ button and wait for 10 minutes to allow the library to download.

Modules Information:

1. Load NLTK Tweets: Using this module we will load twitter sentiment corpora dataset from NLTK library.
2. Read NLTK Tweets: Using this module we will read tweets from NLTK and then clean tweets by removing special symbols, stop words and then perform stemming (stemming means removing ing or tion from words for example ORGANIZATION word will become ORGANIZE after applying stem) on each words. Then we will calculate TFIDF vector.
3. Run SVR Algorithm: In this module we will give TFIDF vector as input to train SVR algorithm. This algorithm will take 80% vector for train and 20% vector as test. Then algorithm applied 80% trained model on 20% test data to calculate prediction accuracy.
4. Similarly we will build model for Random Forest and Decision tree to calculate their accuracy.
5. Detect Sentiment Type: Using this module we will upload test tweets and then application will apply train model on those test tweets to predict sentiment of that tweet.
6. Accuracy Graph: Using this module we will display accuracy graph between all algorithms.

Screen shots

To run this project double click on ‘run.bat’ file to get below screen



In above screen click on ‘Load NLTK Dataset’ to load tweets dataset from NLTK library



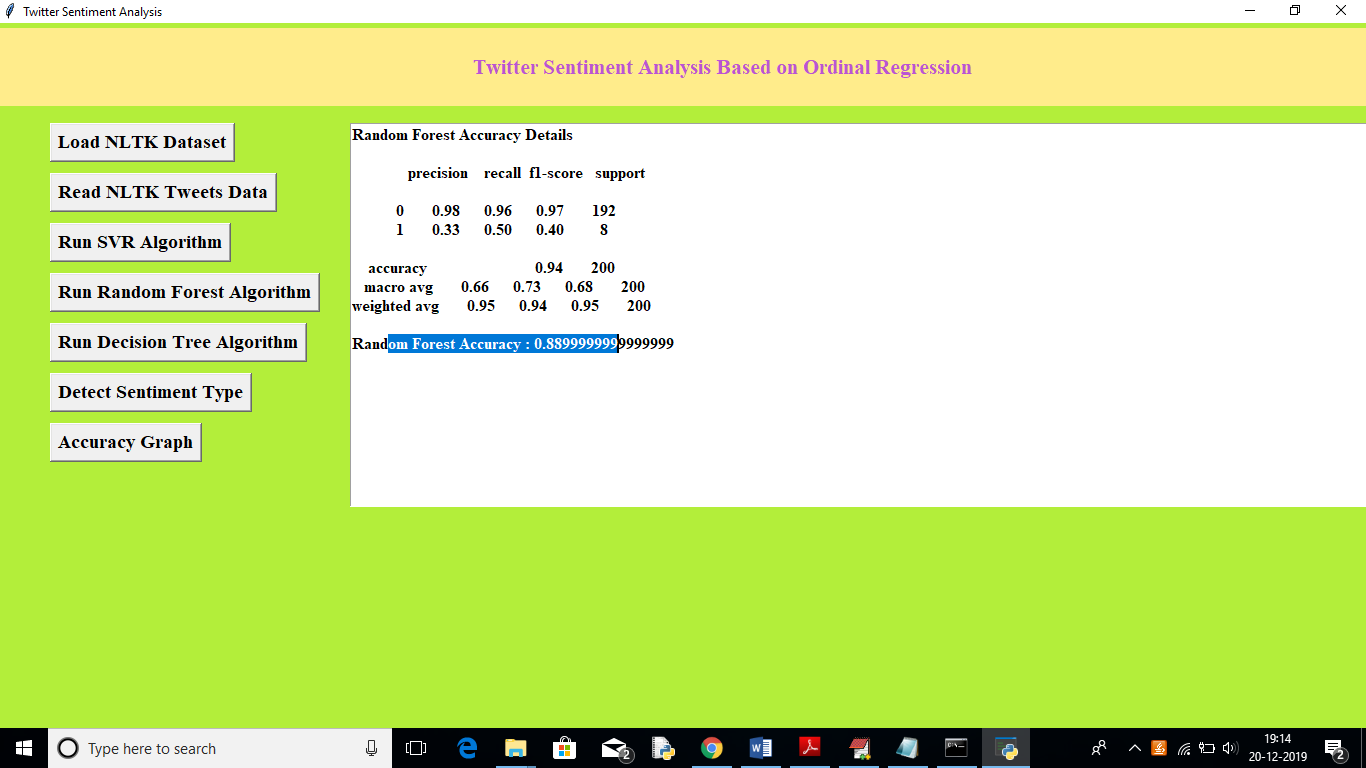
In above screen we can see total 10000 tweets are there in NLTK library, now click on ‘Read NLTK Tweets Data’ button to read all tweets and to build TFIDF vector. Upon each button click you need to wait for some seconds to get output. See below screen



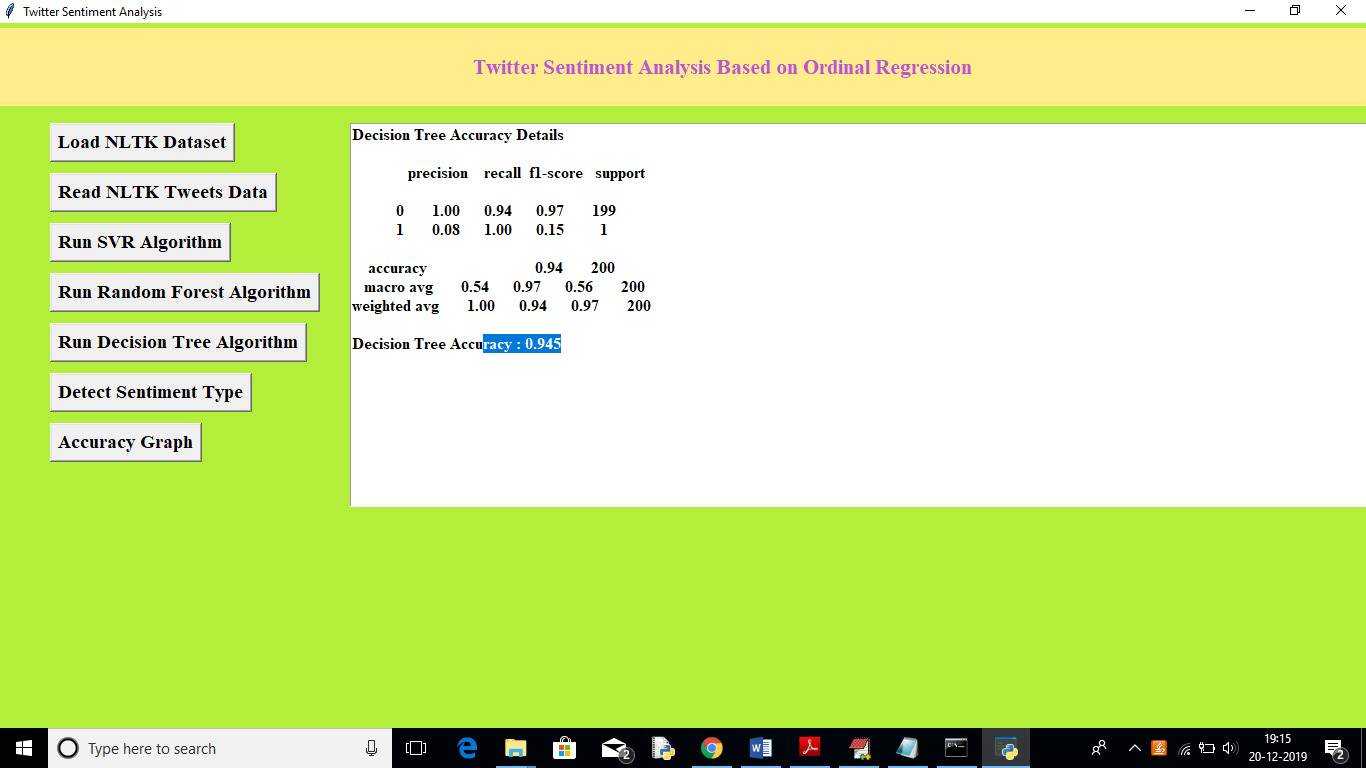
In above screen we can see total 8000 tweets vector used for training purpose and 2000 tweets used for testing purpose. Now click on ‘Run SVR Algorithm’ to build train model on that dataset and to calculate accuracy



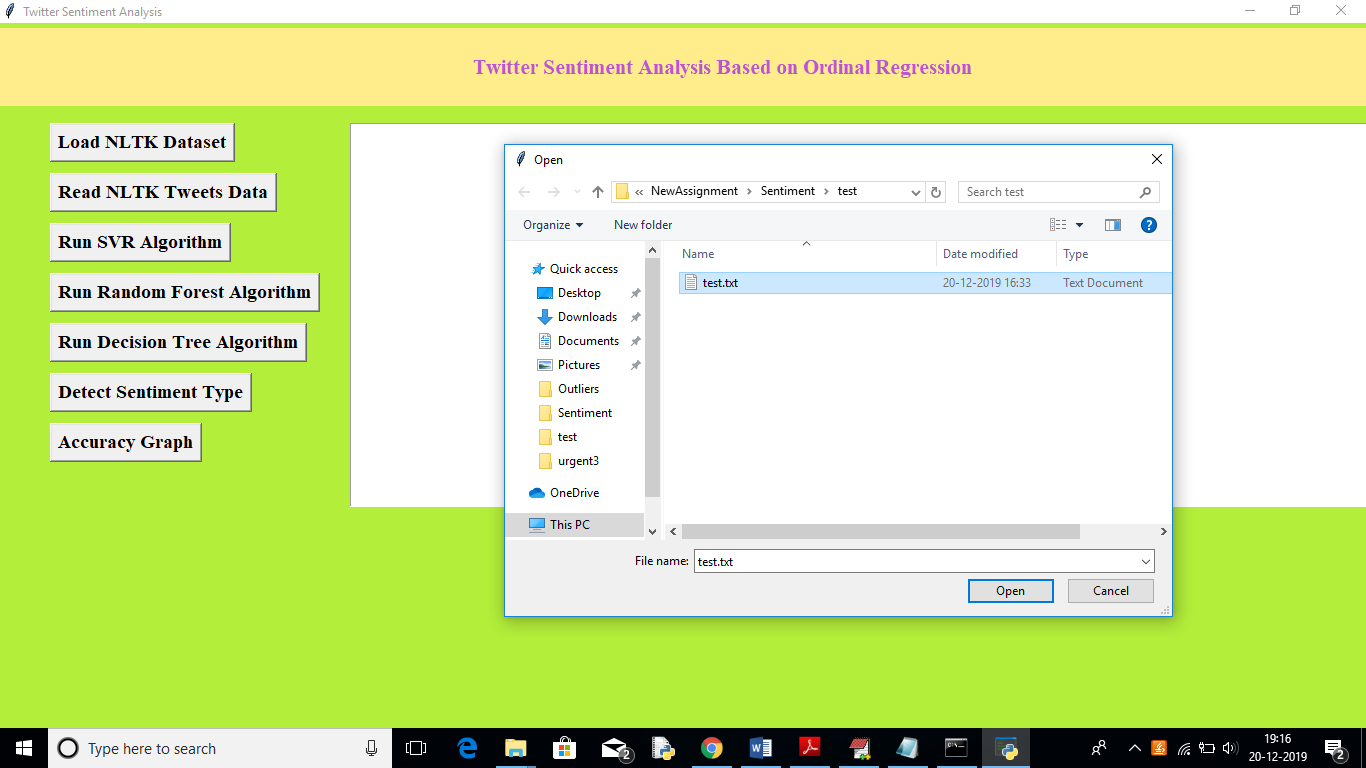
In above screen we can see SVR generate 0.71% prediction accuracy, now click on ‘Run Random Forest Algorithm’ button to calculate its accuracy



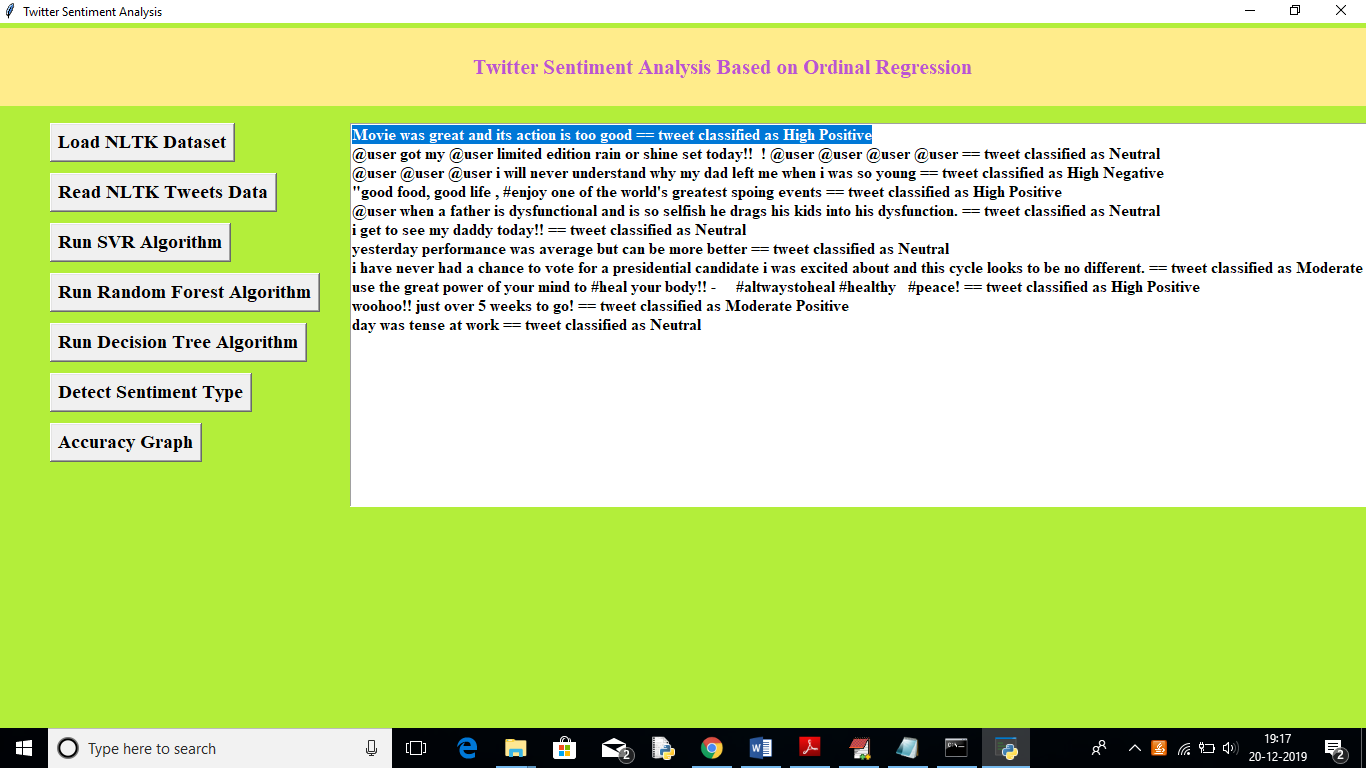
In above screen Random Forest got 0.88% accuracy, now click on ‘Run Decision Tree Algorithm’ button to calculate its accuracy



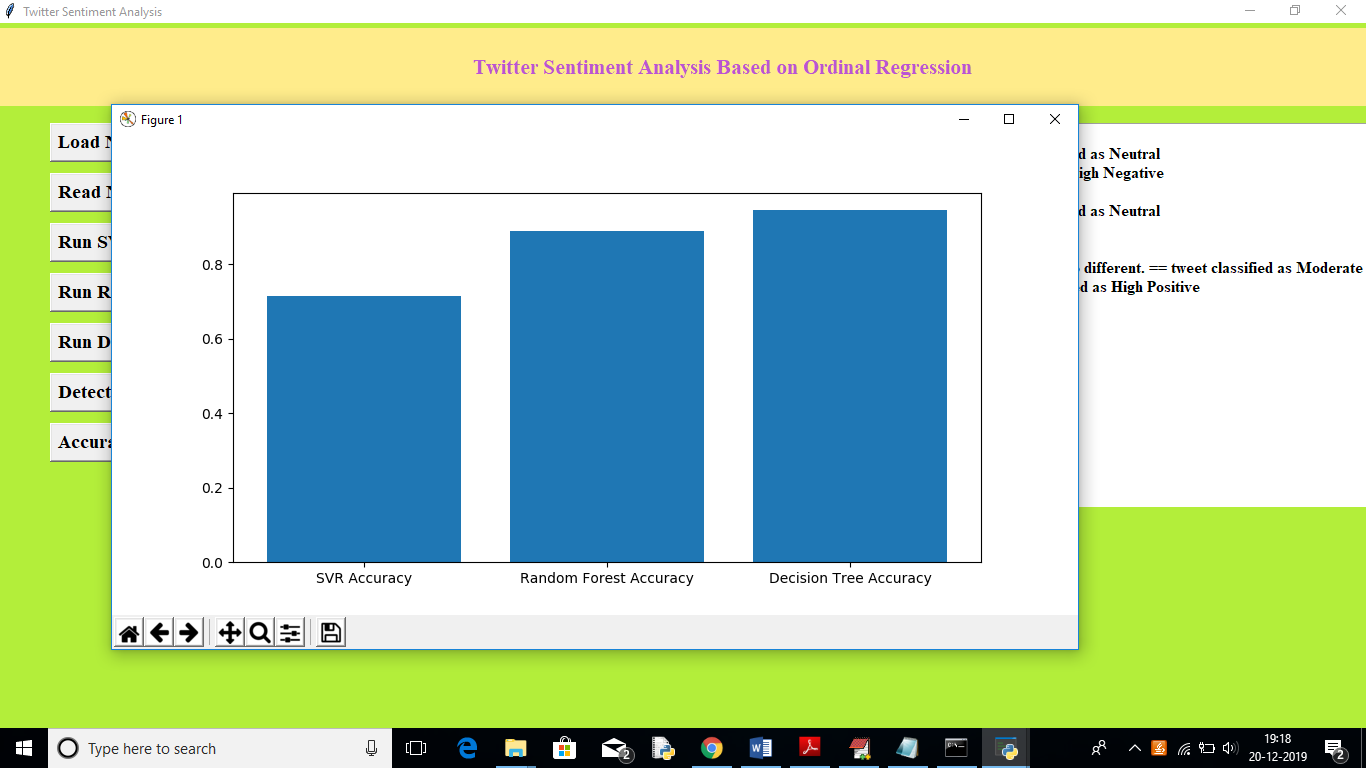
In above screen Decision Tree got 0.94% accuracy, Now click on ‘Detect Sentiment Type’ button and upload test tweets to predict it sentiment. In test folder inside test.txt you can see there is no sentiment label and application will detect it.



In above screen uploading test tweets file and below are the prediction results



In above screen for each tweet we can see the classified/predicted sentiments. Now click on ‘Accuracy Button’ to get below accuracy graph



In above graph x-axis represents algorithm name and y-axis represents accuracy, from above graph we can see decision tree got better prediction compare to other algorithm.

In this paper author using another algorithm called SOFTMAX but its not a classifier algorithm, so I am not implementing it