TASK 1- TWITTER API SENTIMENT ANALYSIS:

(code explaination)

* Firstly, import nltk and download it .Then, import random and import pickle
* Create an empty list say ‘documents’ and just to train the model I saved two types of files namely positive.txt (it contains some short positive sentences) and negative.txt (it contains some short negative sentences). So in first for loop, split the sentences and if the sentence is positive then append ‘pos’ (positive sentiment) in documents and append ‘neg’ (negative sentiment) if the sentence is negative in the other for loop. Now pickle the documents as: document = open(“documents.pickle”, “wb”), this opens up a pickle file, and prepares to write some data in bytes. Then use pickle.dump(documents,document) to dump the data in the file .The first parameter ‘documents’ is what are you dumping and the second parameter ‘document’ is where are you dumping it. Then close the file. You can open the file by following the above process, just loading (pickle.load) it except of dumping.
* Now for cleaning the data, create another empty list say ‘all\_words’ and get most common used words from the data. Create a variable say word\_features and get first 5000(any number) common words. By picking, dump word\_features in words.pickle.
* Create a function say find\_features, using documents and word\_features, this will return the reviews of sentences as positive or negative in featuresets. Dump featuresets in features.pickle file.
* For training and testing, I’ve used naïve bayes classifiers. Create a traning and testing set as per featuresets and split the data.
* Import all the classifiers:

1. Naivebayesclassifier: train the training set using this classifier and pickle it by dumping the classifier, this will give the accuracy of the data.
2. Import all the Naivebayesclassifiers, linear\_model classifiers, svm classifiers using sklearn.
3. MultinomialNB: used for classification of discrete features. Train the training set using this classifier and pickle it by dumping the classifier, this will give the accuracy of the data.
4. BernouliiNB: used for classification of binary/Boolean features. Train the training set using this classifier and pickle it by dumping the classifier, this will give the accuracy of the data.
5. Logistic Regression classifier: predicts the probability of categorial variables. Train the training set using this classifier and pickle it by dumping the classifier, this will give the accuracy of the data.
6. SGDC classifier: it regularises linear models with gradient decent.Train the training set using this classifier and pickle it by dumping the classifier, this will give the accuracy of the data.
7. LinearSVC classifier: gives best fit hyperplane. Train the training set using this classifier and pickle it by dumping the classifier, this will give the accuracy of the data.

* Create a voting system in voting\_classifier to get the most voted classifier, and that most voted classifier will be considered for accuracy of data.
* In the voting\_classifier, by appending in the votes from classifying the features we’ll get the confidence percent of the votes. The one with most confidence is the most voted classifier.
* Create a function called sentiment of text, it will return the text with the feature(‘pos’ or ‘neg’)and the confidence of voted classifier.
* For twitter, install tweepy and import Stream, OAuthHandler and streaming the StreamListener. Get all the api keys for twitter. Import json. Access the tweets using OAuthHandler.
* Create class listener for StreamListener and load the json data for twitter and print the tweet, its sentiment value and confidence(only if the confidence is more that 80%, return all these values).
* Using twitterStream filter the word that has to be queried e.g : lets take the word ‘technology’,

we’ll get the output of the tweets that include this word and its sentiment value and confidence.

SUMMARY:

1. It’s a supervised machine learning type. Sentiment analysis gives us the reviews of a text as positive, negative or neutral.
2. I stored the positive and negative text data and cleaned the data to get most common words for data pre-processing.
3. I summoned up the sentiments of the sentences from the data.
4. Training the algorithms by using different classifiers to get the acuuracy of the result.
5. Pickilng process is the next step to dump the data in bytes pickled as it can be used further.
6. Creating a function to get the most voted classifier and a function to classify the text, return the sentiment value and confidence.
7. Getting twitter data and querying the texts, then stream the tweets and get the sentiment value and confidence.
8. Filtering any word from twitter stream and getting the tweets that include that word, along with its sentiment value and confidence.
9. By this we can study about tweets and the sentiment values, and can detect any inappropriate tweets.