CS-583 Project 2
SENTIMENT CLASSIFICATION OF TWEETS

ANUSHA VOLOJU (UIN: 677775723)
YUKTHI PAPANNA SURESH (UIN: 675253362)

ABSTRACT :

Classification is the process of automatically classifying the observations into defined categories. In our project, the tweets are the observations and the positive, negative and neutral sentiments are the categories. The goal of this project was to predict the sentiment for the given tweet. We have built various classifiers using python scikit learn, out of which SVM classifier achieved higher accuracy for the given training sets. The detailed description of the techniques including data pre-processing, features used, classification methods tried, and the evaluation results are given in the following sections of the report.

INTRODUCTION :

We were provided with two sets of training data, one for Obama tweets and one for Romney tweets. The training data consists of the sentiment labels for each tweet. The labels were of three types [+1, 0, -1] corresponding to positive, neutral and negative sentiments. The given training data must be used to train a classifier and then use it to predict the sentiment of each tweet given in the test set. To build a classifier, the data was first pre-processed, followed by feature extraction and then fed to the classifier and the evaluation results were generated using 10-fold cross validation.

DATA PRE-PROCESSING :

1. <u>Removal of Tags and URLS</u>:
   The tags and URLS in the tweet were removed.

2. <u>Normalization of Text</u>:
   All the characters in the tweet text were converted to lowercase.

3. <u>Replacing emoticons with their respective text meaning</u>:
   All the emoticons in the tweet text were replaced with their respective text meaning like “face with tears of joy”.

4. <u>Removal of Non-ASCII characters</u>:
   All the Non-ASCII characters were removed from the tweet.

5. <u>Removal of Stop Words</u>:

A list of stop words were selected from the NLTK list and only those are removed from the tweet text. For example, words like ‘not’, ‘never’ are retained.

6. Stemming:
   The words in the tweet text were converted to their stemmed forms using NLTK PorterStemmer.

7. Lemmatization:
   The words in the tweet text were converted to their lemmatized forms using NLTK WordNetLemmatizer.

FEATURE EXTRACTION :

1. Word Feature:
   Each word in tweet considered as a binary feature. For counting the word feature of a tweet, it compares with a dictionary which is used to detect which words are stop words and which are not. Stop words are out of consideration. Otherwise, every word is considered for word feature. Also called the Bag-of-words model.

2. N-Gram Feature:
   N-Gram Feature, a sequence of 2-5 consecutive words in a sentence is regarded as a binary n-gram feature. The tweet which contains more rare words that have a higher weight than which contain common words and it has made a greater effect on the classification task.

3. CountVectorizer:
   Converts the collection of texts into a Count vector which is a matrix notation of the collection in which every row represents a tweet from the collection and every column represents a term from the collection and the value of every cell represents the term frequency of a term tweet.

4. TfidfVectorizer:
   Vectorization is a general process of converting a collection of texts into numerical feature vectors. Every tweet is converted into a feature vector consisting of TF-IDF scores for each word. The TF-IDF score represents relative importance of a word in the tweet and the entire collection of tweets. TF-IDF score is the product of TF(normalized term frequency) and IDF(Inverse Document Frequency). TF-IDF vectors can be generated at different levels of input tokens, out of which we are using n-grams(uni, bi and trigrams).

CLASSIFICATION METHODS TRIED:

1. Logistic Regression classifier:
   Sentiment classification polarity of text can be defined as a value that says whether the expressed opinion is positive (*polarity=1*), negative (*polarity=0*), or neutral. The logistic regression classifier is a discriminative model where it only requires one feature that neatly separates the classes and then the model is satisfied. We tried certain parameter tuning such as

penalty (l1 & l2) and C -inverse regularization (0.001,0.01,0.1,1.0), but worked best for default parameters.

2. SVM:
   SVMs are known to perform well in the sentiment analysis problems. The objective of the support vector machine algorithm is to find the hyperplane that has the maximum margin in an N-dimensional space(N-the number of features) that distinctly classifies the data points. SVM performed the best among all the experimented classifiers with Linear kernel and with introduction of n-grams.

3. Multinomial Naïve Bayes classifier:
   Naïve Bayes is a generative model. A generative model makes use of the likelihood term, which expresses how to generate the features of a document if we knew it was of a class. Introduction of bigrams and trigrams enabled higher weight representation, thus producing better results.

4. Decision tree classifier:
   Decision tree classifier takes top down approach and uses divide and conquer method to arrive at a decision. It was one of the oldest techniques and used for its ability to select the features with most importance for the classification process. It needs several key nodes, while it's hard to find "several key tokens" for sentiment classification and thus gave poor accuracy.

5. AdaBoost classifier:
   It is one of the ensemble methods. It was implemented with the decision tree (base-estimator parameter set to None meaning default set to decision tree) classifier as the base classifier. However, this method produced lower accuracy comparatively as the base classifier did not align well.

6. Random Forest classifier:
   Random forests (a *bagging decision tree*) is regarded as a rather weak classifier, especially if only a couple of features are truly significant to determine the outcome such as in sentiment classification. Thus, produced comparatively lower accuracy.

7. kNN classifier:
   The kNN classifier assumes that the classification of an instance is most like the classification of other instances that are nearby in the vector space. Compared to other sentiment categorization methods such as Bayesian classifier, kNN does not rely on prior probabilities, and it is computationally efficient. However, this method produced lower accuracy comparatively.

EVALUATION RESULTS:

10-fold cross validation results of different classification methods tried on the given training data set are as follows:

Classifier		Positive class			Negative class			Neutral class			Accuracy
		P	R	F1	P	R	F1	P	R	F1	
Logistic regression	Obama	0.60	0.62	0.60	0.58	0.64	0.59	0.65	0.22	0.32	57.84
	Romney	0.65	0.22	0.32	0.57	0.91	0.26	0.48	0.20	0.28	56.64
SVM	Obama	0.58	0.66	0.61	0.59	0.63	0.60	0.54	0.49	0.50	**58.75**
	Romney	0.60	0.39	0.47	0.62	0.82	0.69	0.45	0.32	0.35	**58.67**
Naive Bayes	Obama	0.61	0.62	0.61	0.57	0.70	0.61	0.55	0.45	0.48	58.41
	Romney	0.67	0.25	0.35	0.57	0.92	0.69	0.50	0.18	0.26	57.61
AdaBoost	Obama	0.54	0.38	0.44	0.47	0.66	0.41	0.56	0.31	0.29	41.60
	Romney	0.48	0.12	0.18	0.53	0.93	0.66	0.32	0.80	0.11	51.80
Decision Tree	Obama	0.44	0.59	0.50	0.53	0.48	0.49	0.47	0.40	0.42	49.76
	Romney	0.35	0.32	0.33	0.60	0.58	0.58	0.37	0.41	0.37	48.14
Random Forest	Obama	0.50	0.64	0.55	0.60	0.48	0.52	0.50	0.51	0.49	53.40
	Romney	0.45	0.31	0.36	0.62	0.64	0.61	0.38	0.44	0.38	51.50
kNN	Obama	0.49	0.64	0.55	0.56	0.50	0.51	0.49	0.45	0.45	52.25
	Romney	0.53	0.32	0.39	0.59	0.79	0.66	0.40	0.27	0.30	54.55

*****P – average precision, R- average recall, F1- average F1 score, Accuracy- average accuracy

CONCLUSION:

Out of all the models, SVM performed best with an accuracy of 58.75% and 58.67% on Obama and Romney datasets. Further work could involve better-weighted approach given multiple aspects in a sentence, taking variable window size like Google’s word2vec implementation, opinion lexicon incorporation and sampling. Additionally, deep learning methods could be utilized to gain higher accuracy.

REFERENCES:

1. Web data mining: exploring hyperlinks, contents, and usage data (Liu, Bing)
2. Text Classification - A Comprehensive Guide to Classifying Text with Machine Learning
3. An introduction to machine learning with scikit-learn
4. 6 Practices to enhance the performance of a Text Classification Model (Shivam Bansal)