**TWITTER SENTIMENT CLASSIFICATION**

**Abstract**

Twitter is an online news and social networking tool in which users post 280-character updates called tweets, of what is going on in their lives along with links to things they think are interesting, funny, or useful to their followers. A Twitter Sentiment Classification on these tweets relating to US Presidential election between Barak Obama and Mitt Romney is carried out in this project. The tweets are labelled as positive, negative, neutral and mixed tweets for both Obama and Romney separately. The tweets are pre-processed, vectorized and classified through various classifiers. The performance of these classifiers is evaluated based on 10-fold cross-validation. The performance parameters are accuracy, precision, recall and F1-score for each of the labeled classes.

**Introduction**

Twitter is a favored social networking [microblogging](https://searchmobilecomputing.techtarget.com/definition/microblogging) platform for news outlets because of its ability to send real-time updates. Sentiment analysis refers to the use of [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), [text analysis](https://en.wikipedia.org/wiki/Text_analytics), [computational linguistics](https://en.wikipedia.org/wiki/Computational_linguistics), and [biometrics](https://en.wikipedia.org/wiki/Biometrics) to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis within [microblogging](https://en.wikipedia.org/wiki/Microblogging) has shown that [Twitter](https://en.wikipedia.org/wiki/Twitter) can be seen as a valid online indicator of political sentiment. Tweet’s political sentiment demonstrates close correspondence to party’s and politician’s political positions, indicating that the content of Twitter messages plausibly reflects the offline political landscape. Thus, twitter is perfect tool for sentiment classification especially on political issues. In this project, the training data collected from US Presidential elections are labelled on the following classes: positive, negative, neutral and mixed. These data are stored separately for Obama and Romney. The mixed class containing both positive and negative sentiment is not considered in this project for simplicity. The objective is to correctly classify an unseen test data into positive, negative or neutral tweet. The numeric labels for positive, negative and neutral are 1, -1 and 0 respectively. The raw data needs to be pre-processed and vectorized before being given as inputs to the classifier models. These steps are discussed in the following sections.

**Techniques**

The classifier models require the data to be pre-processed and features to be provided. Various models were experimented for Obama and Romney separately. Each model implemented was pre-processed and vectorized differently. The following sub-sections will discuss the all the pre-processing, features used, and the classification methods tried.

**(i) Data Pre-processing**

**1. Removal of Punctuations, digits and RT tags:** The punctuations and digits present in the tweet are unrelated to that tweets sentiment and hence are removed. Certain tweets also contain “RT” word, which represents that it was retweeted. This extra word also in no way contributes to sentiment and therefore is removed. Regex was used to handle this.

**2. Removal of Usernames:** The usernames of a twitter account are followed by the ‘@’ symbol. The symbol and the username are removed because of lack of sentiment on them. Regex was used to handle this.

**3. Removal of HTML tags:** The HTML tags do not add any other significant information to the tweet and hence are removed. Regex was used to handle this.

**4. Tokenization:** Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. Tokens can be individual words, phrases or even whole sentences.Tokenization provides a convenient method for other pre-processing like stemming and lemmatization.

**5. Stopwords:** The stopwords which represent the most common words in a language are filtered out in certain models because they do not usually contribute to the sentiment of the tweet but they also contain some key sentiments and hence were not removed. The list of stopwords for English is obtained from NLTK.

**6.Stemming:** It is the process of reducing inflected words to their [word stem](https://en.wikipedia.org/wiki/Word_stem), base or [root](https://en.wikipedia.org/wiki/Root_(linguistics)) form. This will reduce the vocabulary size and is efficient in handling similar words. The Porter Stemmer from the NLTK package was used for stemming

**7. Lemmatization:** It is the process of grouping together the inflected forms of a word so they can be analyzed as a single item, identified by the word's [lemma](https://en.wikipedia.org/wiki/Lemma_(morphology)), or dictionary form. The lemmatization was performed using WordNet Lemmatizer from NLTK.

**(ii) Features**

**TF IDF Vectorizer**: TF-IDF is a Term Frequency-Inverse Document Frequency and is a very common algorithm to transform text into a meaningful representation of numbers. The technique is widely used to extract features across various NLP applications.

**Count Vectorizer**: Count Vectorization involves counting the number of occurrences each word appears in a document. The result of this will be very large vectors and we will get very accurate counts of the word content of our text data.

Unigrams and bigrams (words occurring between window range of one and two) text feature extraction were employed with TF-IDF and the Count Vectorizer to improve the performance of the classifiers.

**(iii) Classifiers Implemented**

**Logistic Regression**: Logistic regression is an algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

**Decision Tree:** In decision trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with record’s attribute. Based on comparison, we follow the branch corresponding to that value and jump to the next node. We continue comparing our record’s attribute values with other internal nodes of the tree until we reach a leaf node with predicted class value.

**Random Forest:** A random forest is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size.

[**BernoulliNB**](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html#sklearn.naive_bayes.BernoulliNB) **classifier:** [BernoulliNB](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html#sklearn.naive_bayes.BernoulliNB) implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued variable. This class requires samples to be represented as binary-valued feature vectors. If handed any other kind of data, a BernoulliNB instance may binarize its input.

[**MultinomialNB**](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB)**:** [MultinomialNB](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB) implements the naive Bayes algorithm for multinomially distributed data, and is one of the two classic naive Bayes variants used in text classification where the data are typically represented as word vector counts. The distribution is parametrized by vectors for each class.

**SVM (Linear, poly, rbf):** A Support Vector Machine is a discriminative classifier formally defined by a separating hyperplane. Given a labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples. In two-dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

**Neural Network MLP**: A multilayer perceptron is a [deep, artificial neural network](https://skymind.ai/wiki/neural-network). They are composed of an input layer to receive the signal, an output layer that decides or prediction about the input, and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP. Training involves adjusting the parameters, or the weights and biases, of the model to minimize error. Backpropagation is used to make those weight and bias adjustments.

[**SGD Classifier**](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn.linear_model.SGDClassifier)**:** The [SGD Classifier](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn.linear_model.SGDClassifier) implements a plain stochastic gradient descent learning routine which supports different loss functions and penalties for classification.

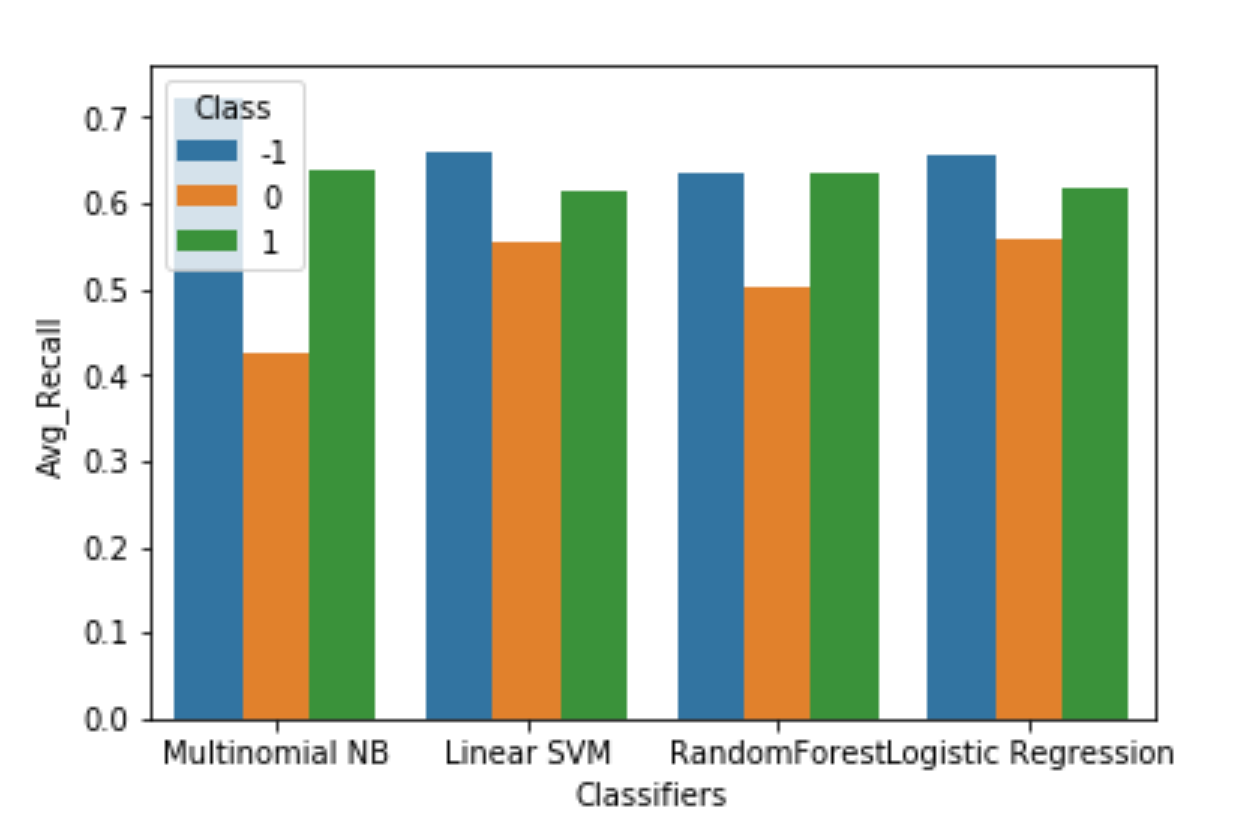
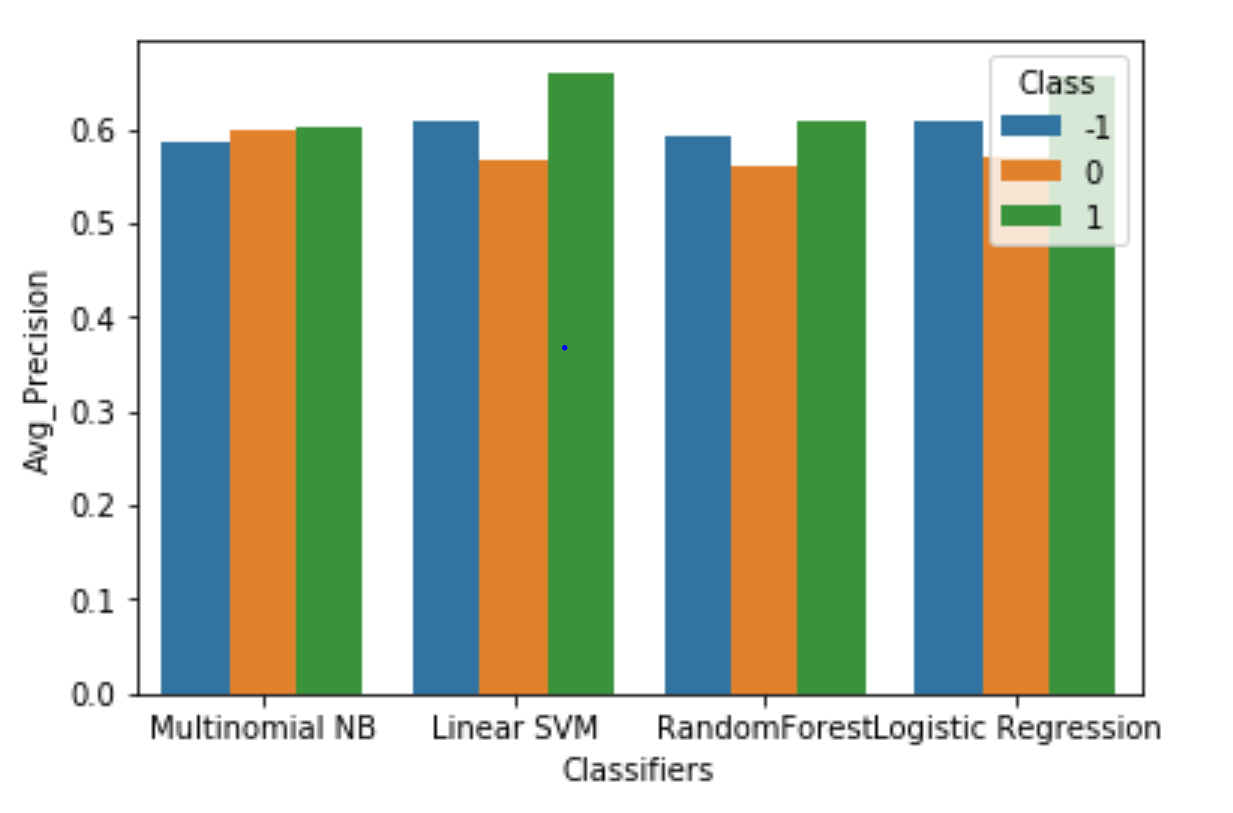
All the above machine learning algorithms were used to classify the tweets in this project and the results of the best performing classifiers are displayed below.

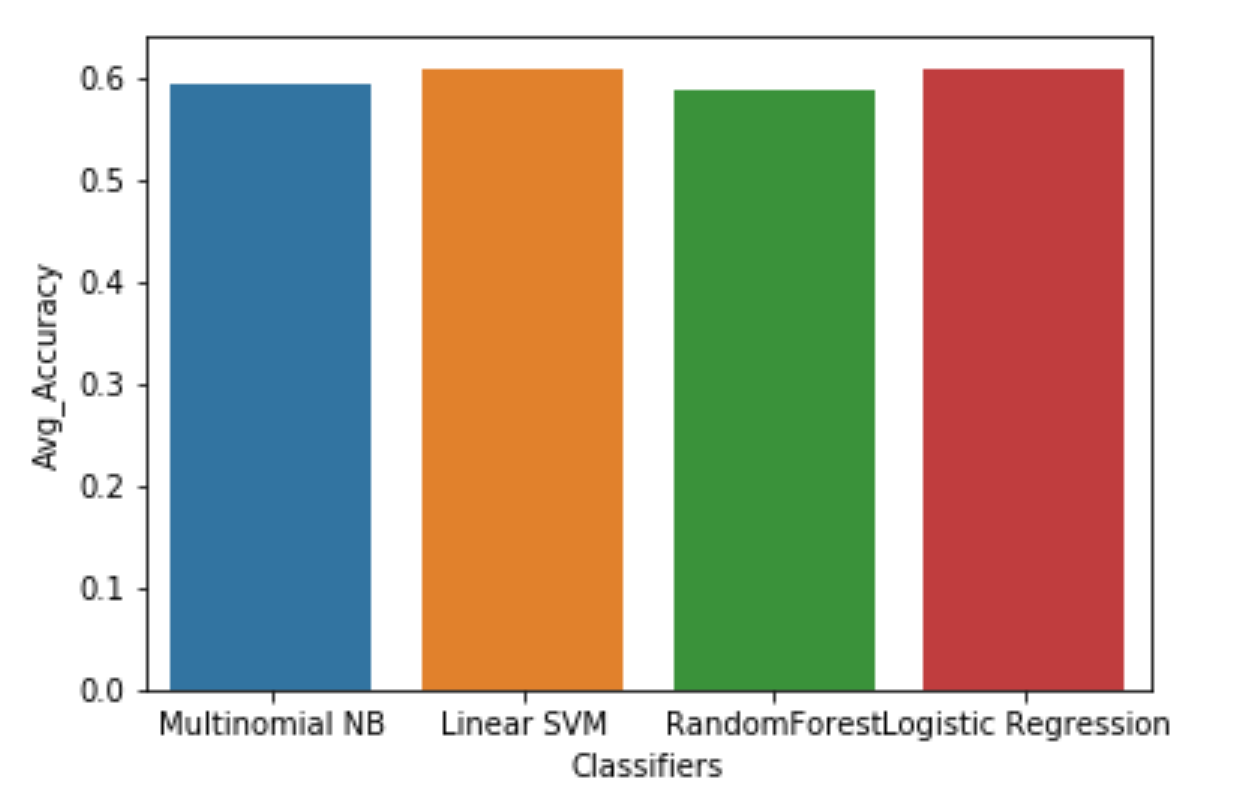
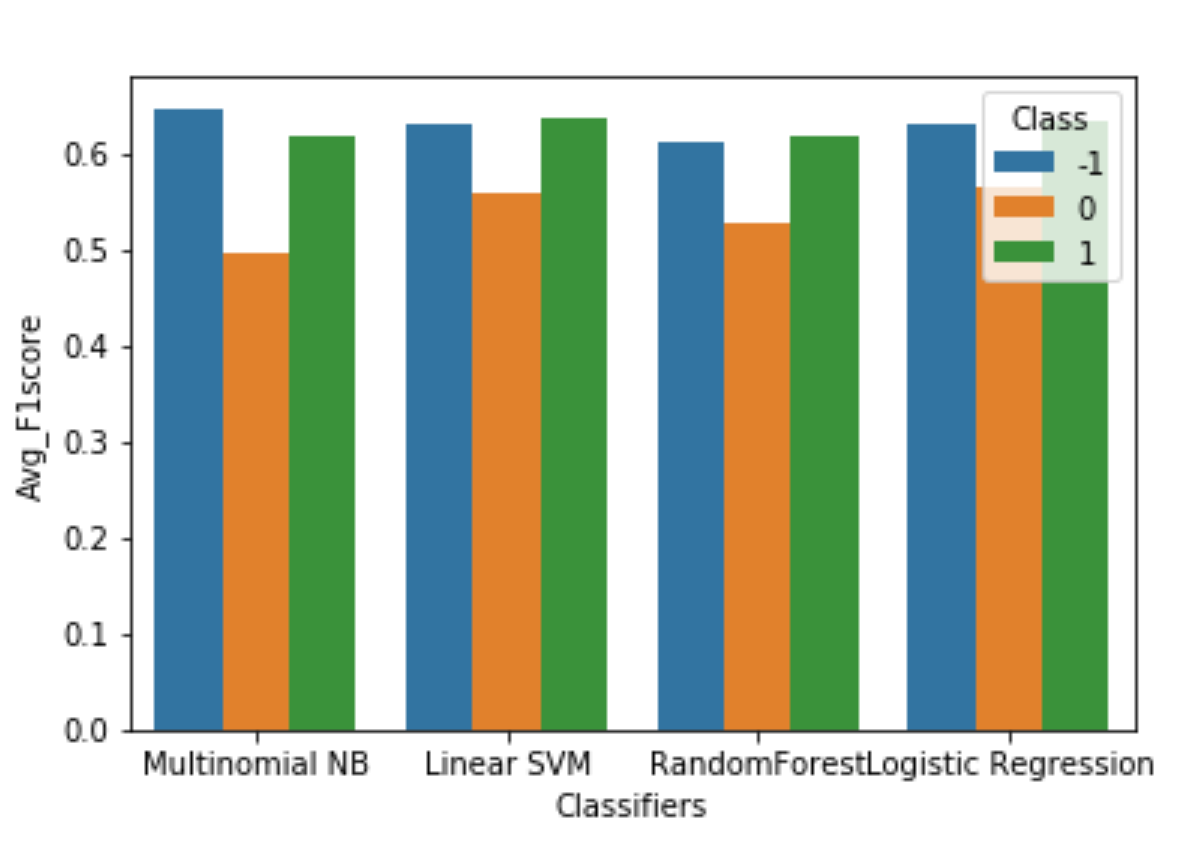
**Evaluation**

The parameters that are used in the evaluation are accuracy, precision, recall and F1-score. These parameters are obtained for the classes positive, negative and neutral for both Obama and Romney. An average across all the three classes gives a comprehensive result on the model’s performance. These results were obtained based on 10 fold cross-validation on the given dataset.

**Obama Dataset:**

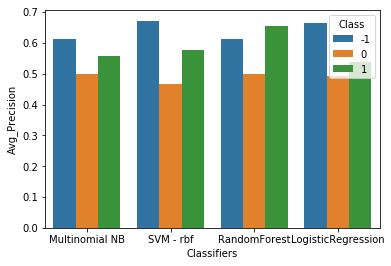
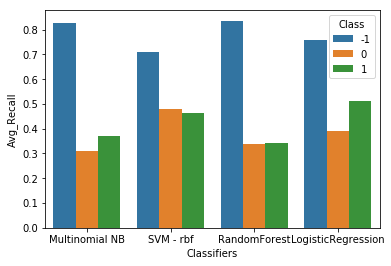
The below charts show the evaluation scores of the Obama dataset for various classifiers. Multinomial NB, Linear SVM, Random Forest and Logistic Regression are the best performing classifiers obtained for the Obama data set. Out of which Logistic Regression showed significant results as shown in the bar plots.

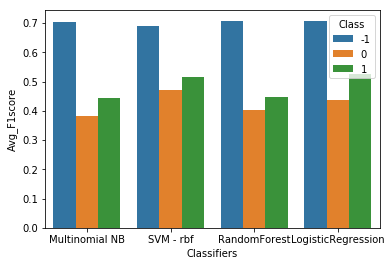




**Romney Dataset:**

The below charts show the evaluation scores of the Romney dataset for various classifiers. Multinomial NB, SVM with rbf kernal, Random Forest and Logistic Regression are the best performing classifiers obtained for the Romney data set. Out of which Logistic Regression showed significant results as shown in the bar plots.

**Conclusion**

We experimented with the various pre-processing techniques mentioned above on Obama and Romney dataset separately. Only the relevant features (tweets, class) were chosen for building the model. The processed tweets were vectorized using either TF-IDF or Count Vectorizer based on the highest accuracy results. All the above mentioned models were implemented to predict the tweet classification using k-fold cross validation with a k-value set to 10. To obtain optimal results for each model the pre-processing steps were altered, and parameter tuning was performed. A comparative study was made on the results obtained for Obama and Romney separately and the best performing model was chosen.

For Obama, Logistic Regression, Linear SVM and Random Forest performed better than the other implemented models. Among which Logistic Regression showed consistent accuracy of 61% across all the classes. Multinomial Naïve Bayes, Logistic Regression and SVM with rbf kernel produced the best results for Romney. In this case too, Logistic Regression outperformed the other classifiers with 60.2% accuracy.

**References**

[1] NLTK: <https://www.nltk.org/>

[2] Scikit-Learn: <https://scikit-learn.org/stable/>

[3] Pandas: <https://pandas.pydata.org/>

[4] Numpy: <https://www.numpy.org/>