# CS583 Data Mining and Text Mining
# Tweet Classification

Murali Krishna Valluri – 671441135; Naga Spoorthi Pendyala - 679836791

## Abstract

Twitter is one major social networking service which has served as a common platform for everyone to express their views and has been mostly exploited for brand campaigning, elections and news media. It is very interesting to know what topics are trending or what people in other parts of the world are interested in or what people expect from a product/service. The tweet dataset can thus play an important role to uncover how users feel about a particular product or a service. A company can then use this information to make informed decisions about its products. By mining for such information, a company can work more towards its customer base satisfaction and end up with higher profits and a better service. This project aims to achieve something similar, by classifying a given tweet dataset into positive, negative and neutral tweets in regards to the election campaigns by presidential candidates Barack Obama and Mitt Romney.

## Introduction

Twitter users tweet in total over 200 million tweets a day. People often express their opinions in form of tweets which are short text messages. This makes it an excellent data source for various problems like sentiment analysis and topic classification. Given below is the problem statement and objective of this project –

### Problem Statement

The input data given is a collection of tweets about Presidents Barack Obama and Mitt Romney's election campaign. The training data for the project is in the form of two excel sheets containing around 7100 tweets, where every tweet holds an opinion and can be labelled as 1, -1, 0 and 2 for positive, negative, neutral and mixed opinions respectively.

### Objective

The objective of the project is to build and train a classifier to classify the given tweets into three classes - positive, negative and neutral and then use the classifier on the test set to generate the Accuracy, Precision, Recall, F-Score for the classes. The mixed opinion tweets can be ignored.

## Techniques

In order for us to get a better result, the raw data provided must be cleaned and transformed into a format that will be more easily and effectively processed. There are many approaches which can be used for this. The following combination of approaches are used in this project –

**Data Pre-processing Steps**

- Rows with class values ‘2’, ‘IR’, ‘Irrelevant’, ‘!!!!’ are removed from the excel sheets first. Next, the Date and Time columns are removed.
- The excel files are converted into csv files – The files now consist of the class labels and tweets only.
- All the tweets are converted into lowercase.
- Any special characters encountered are removed and other characters like ‘#’, ‘@’, ‘”’, ‘’’ are replaced with blank characters.
- Hyperlinks and html tags are removed.
- The tweets are stripped to remove any extra spaces at the beginning and end of every tweet.
- Stop words are removed. This would get rid of common words and articles like ‘a’, ‘is’. ‘at’, ‘which’, ‘on’ etc.
- The encoding type of the input csv file is changed to UTF-8 to preserve all tweets.
- Stemming is done using the NLTK library. This would reduce inflected words to their stem, root or base form. Example – ‘voting’ and ‘voted’ would be reduced to ‘vote’.
- Emoticons and slang text are replaced using predefined set of values obtained from online sources.

After the data has been pre – processed, it can now be used for classification. Python’s scikit-learn library has numerous in built classifiers which have been used in this project.

- Scikit-learn’s CountVectorizer is used to convert all the text into a matrix of token counts.
- Scikit-learn’s TfidTransformer is used to transform the count matrix into a normalized tf or tf-idf representation.

Next, the following classifiers have been tested and the best model was reported.

**Classification methods tried**

- **Decision Trees**
  The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

- **Logistic Regression**
  Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

- **Multinomial Naïve Bayes**
  All naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. It is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set.

- **K – Nearest**
  The principle behind nearest neighbor methods is to find a predefined number(k) of training samples closest in distance to the new point, and predict the label from these.

- **SVM**
  Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

- **SGD**
  Stochastic gradient descent is a stochastic approximation of the gradient descent optimization method for minimizing an objective function that is written as a sum of differentiable functions.

- **Voting**
  The idea behind the voting classifier implementation is to combine conceptually different machine learning classifiers and use a majority vote or the average predicted probabilities (soft vote) to predict the class labels. Such a classifier can be useful for a set of equally well performing model to balance out their individual weaknesses.

**Methods to Improve F-score and Accuracy**

- Tweaking Default parameters for the classifiers
  - Decision Trees – ‘max_features’, ‘random_state’, ‘class_weight’
  - Logistic Regression – ‘class_weight’
  - Multinomial Naïve Bayes – ‘class_prior’
  - K – Nearest – ‘n_neighbours’, ‘weights’
  - SVM – ‘kernal’, ‘class_weight’, ‘cache_size’, ‘probabiltlity’
  - SGD – ‘class_weight’, ‘alpha’
  - Voting – ‘voting’, ‘weights’, ‘n_jobs’
- Over Sampling
  - The training data for Romney was very skewed. To handle this imbalance, over sampling was used. Under Sampling was tested too, but it harmed the accuracy and scores of the classifiers.
  - There was no improvement observed for the Obama dataset upon using sampling.
- Stratified K-fold
  - 10 cross-fold validation was used to evaluate the models, in which the original tweet data was randomly partitioned into 10 equal sized subsamples. It was observed that there was an improvement in the performance of the models on using stratified cross-fold validation, where the folds are selected so that each fold contains roughly the same proportions of class labels.
- Class Weights(Balanced/unbalanced)
- Voting – weights

# Evaluation and Performance Benchmarks

Listing down the scores and accuracies of the best classifier determined at the points where performance hike was observed on training data only using 10 – fold cross validation –

**Obama Data**

| Pre - Processing Method | Best Model Determined | Positive Class | | | Negative Class | | | Accuracy |
|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F-Score | Precision | Recall | F-Score | |
| Before Encoding | Logistic Regression | 0.57 | 0.68 | 0.61 | 0.64 | 0.58 | 0.61 | 58.47 |
| After Encoding | Logistic Regression | 0.58 | 0.67 | 0.61 | 0.64 | 0.58 | 0.6 | 58.77 |
| After Stemming | SVM | 0.59 | 0.65 | 0.61 | 0.61 | 0.61 | 0.61 | 59.18 |
| Replacing Emoticons | SVM | 0.59 | 0.65 | 0.61 | 0.61 | 0.61 | 0.61 | 59.18 |
| Replacing Slang words | Voting (LR, SVM, NB) | **0.59** | **0.67** | **0.61** | **0.63** | **0.59** | **0.61** | **58.98** |

**Romney Data**

| Pre - Processing Method | Best Model Determined | Positive Class | | | Negative Class | | | Accuracy |
|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F-Score | Precision | Recall | F-Score | |
| Before Encoding | Logistic Regression | 0.64 | 0.28 | 0.39 | 0.59 | 0.88 | 0.69 | 57.21 |
| After Encoding | Logistic Regression | 0.62 | 0.27 | 0.37 | 0.59 | 0.87 | 0.68 | 56.28 |
| After Stemming | Logistic Regression | 0.64 | 0.27 | 0.37 | 0.58 | 0.86 | 0.68 | 56.07 |
| Replacing Emoticons | Voting (LR, SVM, NB) | 0.59 | 0.36 | 0.44 | 0.6 | 0.83 | 0.68 | 56.81 |
| Replacing Slang words | Voting (LR, SVM, NB) | **0.6** | **0.36** | **0.44** | **0.6** | **0.84** | **0.68** | **56.94** |

Listing down the top 4 scores and accuracies of the best classifier determined on test data –

**Obama Data**

| Algorithms | Positive Class | | | Negative Class | | | Accuracy |
|---|---|---|---|---|---|---|---|
| | Precision | Recall | F-Score | Precision | Recall | F-Score | |
| **Logistic Regression** | **0.61** | **0.5** | **0.55** | **0.57** | **0.64** | **0.61** | **56.92** |
| **SVM** | 0.59 | 0.49 | 0.54 | 0.57 | 0.62 | 0.59 | 56.05 |
| **SGD** | 0.57 | 0.6 | 0.59 | 0.58 | 0.51 | 0.54 | 56.41 |
| **Voting (LR, SVM, SGD)** | 0.61 | 0.50 | 0.55 | 0.58 | 0.63 | 0.60 | 56.82 |

**Romney Data**

| Algorithms | Positive Class | | | Negative Class | | | Accuracy |
|---|---|---|---|---|---|---|---|
| | Precision | Recall | F-Score | Precision | Recall | F-Score | |
| **Logistic Regression** | **0.61** | **0.52** | **0.56** | **0.66** | **0.8** | **0.73** | **62.35** |
| **SVM** | 0.55 | 0.58 | 0.57 | 0.7 | 0.67 | 0.68 | 60.19 |
| **SGD** | 0.62 | 0.47 | 0.53 | 0.66 | 0.8 | 0.72 | 61.19 |
| **Voting (LR, SVM, SGD)** | 0.61 | 0.5 | 0.55 | 0.58 | 0.63 | 0.73 | 62.24 |

# Conclusion

For the Obama dataset, **Logistic Regression** gave the best accuracy of **56.92%.** The f-scores obtained were **0.55 for the positive class and 0.61 and negative class** respectively. Voting classifier combining Logistic Regression, SVM and SGD also gave the very similar results.

For the Romney dataset, **Logistic Regression** gave the highest accuracy of **62.35%** with **f-scores of 0.56 for the positive class and 0.733 and negative class**. Even here, voting classifier combing Logistic Regression, SVM and SGD gave similar results.

# References

- http://scikit-learn.org/stable/modules/classes.html
- https://github.com/sifei/Dictionary-for-Sentiment-Analysis/tree/master/slang
- https://stackoverflow.com/questions/tagged/scikit-learn