**Sentiment Analysis of Movie Reviews**

Sentiment analysis of IMDB movie reviews using TFIDF and word2vec.Our dataset contains 50,000 IMDB movie reviews which have been divided into train (labeledTrainData.tsv) and test (unlabeledTrainData.tsv).

The training data is supervised and contains 12,500 positive and 12,500 negative reviews. The test data is unsupervised and contains 25,000 reviews which are to be classified as 0(negative) or 1(positive).

**Project files** - SentimentAnalysis.py, SentimentAnalysis.ipynb

**Libraries/packages** needed to run the code -

1. NLTK

2. BeautifulSoup

3. Gensim word2vec

4. GridSearchCV, sklearn

5. scikitplot, matplotlib

Also, for word2vec the pre-trained word2vec model is needed which can be downloaded from <https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit>

**Code flow -**

* The code first displays the output for all 4 classifiers - Multinomial Naive Bayes, Random Forest, Logistic Regression and Linear SVC using TFIDF.
* The best obtained accuracy is using Linear SVC (0.87468) which is then stored in a csv file - tfidf\_svc.csv.
* The code then uses pre-trained word2vec model which again runs on all above classifiers except for Multinomial Naive Bayes. Multinomial Naive Bayes is not used with word2vec as it cannot work with negative values.
* The best obtained accuracy is again obtained with Linear SVC (0.85988) which is then stored in word2vec\_svc.csv.
* The last part of the code tests a custom movie review which is to be given by the user. This review is being tested using TFIDF and the classifier used is Multinomial Naive Bayes.

**Observations**

* For word2vec, the pre-trained model loading and vectorization has been performed (this part is commented in the code as it takes time to execute) The results of vectorization are stored in train.pkl and test.pkl files which are being read in the code.
* When running the SentimentAnalysis.py file the time taken is slightly more as compared to the time taken when running the SentimentAnalysis.ipynb.
* Our project report documents the execution times based on the SentimentAnalysis.ipynb results.
* The part of code for confusion matrix has been commented as this takes time to execute. The confusion matrices however have been shown in the project report.
* When running the .py file the precision curve plots need to be closed for the code to execute further.