CMPE-255 Program 1

Basic Info:

Name : Aaditya Deowanshi
SJSU-ID : 011815440
Rank & Accuracy : 1st & 0.8593

Goal:

To implement a k-Nearest Neighbor Classifier to predict the sentiment for 25000 movie reviews from given train set.

Methodology:

- Reading train and test data set into pandas dataframe by dividing each line on tab space to store polarity and review into two columns.
- Merging review column of train & test data set into one (to achieve same dimension for cosine similarity calculation) as data list.
- Removed all html tags for every review in data list using regex expression.
- Removed all other character other than words from every review.
- On analysing data set, we have calculated frequency of most common words occurring in all reviews and added some particular words to stop_word list like : movie,movies,films,film,hollywood,scenes,series etc. These words are common in both positive and negative review.
- Used nltk library stop words and stop_word list to remove common stop words from reviews.
- For words with length less than 3 are removed from review.
- For words with frequency greater than 5 are removed from review as these are also common words which don't contribute to polarity of review.
- Built sparse matrix using scipy library and professor's built function.
- Used IDF to decrease importance of popular words
- Normalised matrix.

- Divided matrix into train_mat for first 25000 (data[0:25000]) rows and train_mat for last 25000 rows (data[25000:])
- Calculated cosine similarity for test and train matrix using sklearn library by dividing into batch of 10 to avoid memory error (Better accuracy than jaccard and euclidean).
- In cosine similarity matrix every 25000 row represent test review and 25000 column represent train reviews.
- Using Numpy arg partition we will calculate indices of top K element (Neighbours) for each row in cosine_similarity matrix.
- Using indices of top K element we calculate polarity of review by comparing it with train dataframe polarity column values.
- If review's negative value is greater than positive value we will assign label as "-1" otherwise "+1". For review having negative = positive value, we will be assigning "+1".
- Write labels to file format.dat for every review.

Other Methods: (Reduced Accuracy- Not Included in final output)

- Used Nltk Porterstemmer to stem each word to their root words
- Used Nltk pos_tags to assign tags to each word like : Noun , Adjective etc as noun will not add to polarity of review