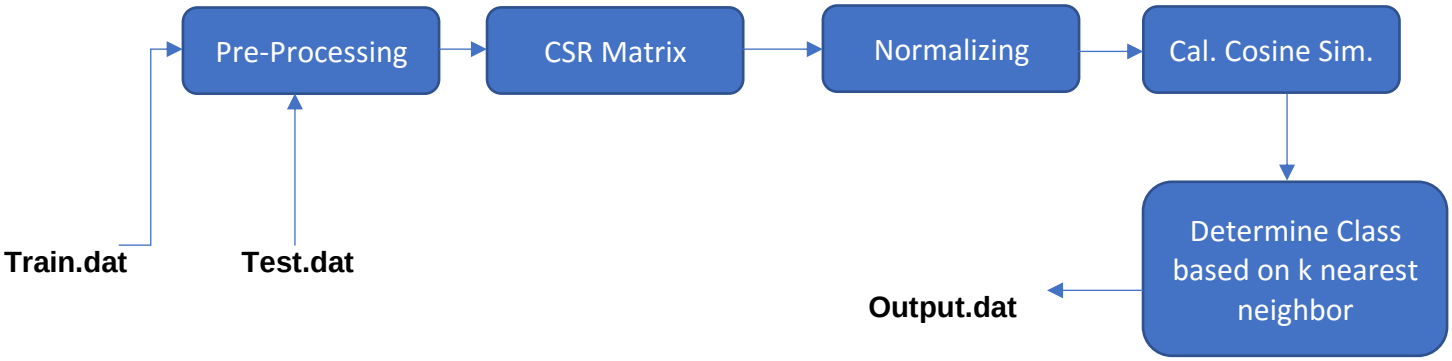
Program-1

Name :- Foram Mehta
Student Id :- 011548446
Rank : - 15
Accuracy:- 74.70%
Goal :- Movie Review Classification based on KNN classification Algorithm.

Approach & Methodology Used:-



PRE-PROCESSING:-

The training dataset was loaded using pandas library and converted into two columns class and reviews. Each review was processed in various stages using **nltk** library. It first removed all the **stopwords** from the review. eg('there','what','this' etc). It then removed all the special and html tag characters from the review and split the review into list of words. The words were then categorized into noun ,verb, adjective and adverb using **nltk.wordnet** library. It then lemmatized the word to its base word using **nltk.lemmatize**. Eg (finding -> find, sees -> see). I also tried to add the **dictionary words** for the verb and adjective words but it couldn't train the data in my laptop. The program went on for 7 hours eventually crashing my laptop. So I removed the dictionary from the preprocessing. I also tried autocorrecting the word using **autocorrect** library but even that took longer hours and couldn't train so couldn't use it in my solution output.
The same preprocessing steps were used for processing text from test file as well.

CSR Matrix :-

The matrix is used for converting the data into a low dimensional sparse data matrix for training as well as test dataset. It gives us index and value for every word in the review. This is much faster approach with such a large dataset. I also tried doing with the conventional method of calculating cosine similarity by its formulae but the code went into endless

functioning until it crashed as it ran out of memory.

Normalization :-

Normalization of the matrices will normalize the matrices in the range 0-1.

Calculation :-

Once the matrices are normalized then we just need to do the dot product of test review with every training set review as the denominators is already normalized. Once the similarity is found for a test review, it is sorted in descending order and only then only first k similarities are considered for find out the majority class candidate. ‘k’ I found out by uploading my code on clp for various k inputs. I got highest accuracy for k=31. If majority class under k are ‘+1’, then ‘+1’ is returned else ‘-1’ and if equal then adding the total similarities for both the classes under k and returning the higher ones class. This process continues unless the class for all test reviews are identified. The output is appended in the file that is loaded in CLP for accuracy. Since it was a text mining cosine similarity was used as it proves better over other algorithms for text mining.

Few challenges faced were during pre-processing as the code wouldn’t run if we tried to fine-tune data more. Finding out value of k was also a challenge. Had to do many series of trial and error runs.