Rank & Accuracy

Accuracy - 0.726

Goal

The objective of this program is to implement K-Nearest Neighbour Classification algorithm and choosing the best model on different parameter such as best K for Nearest Neighbour , best similarity function (cosine similarity , euclidean distance) and feature selection .

Dataset

Dataset consist of two file Train dataset and Test dataset with each contains 25000 rows
Train dataset contains both polarity and reviews whereas Test Dataset contains only reviews

Apporach

Step 1 : Cleansed the data using regular expression , it will remove all the html tags and special character except spaces from the review text .

Step 2 : Filter all the shorter words which have length less than 5 character as it doesn’t contribute anything to the model .

Step 3 : Perform above two steps for both train and test dataset and merge both list before creating sparse matrix .

Step 4 : Now create CSR matrix and split the matrix into two halve so that will have two sparse matrix with same dimensions .

Step 5 : Use cosine similarity built in function from sklearn to calculate cosine similarity .

Step 6 : This will calculate cosine similarity value for each test review with 25k train review and will create 25k x 25k matrix .

Step 7 : Now for each row in cosine matrix find top k similarity value and store their indices in the list using numpy arg partition .

Step 8 : For each index from top index list check the polarity in the traindata set and count the negative polarity or positive polarity .

Step 9 : On the basis of count of negative polarity and positive polarity predict the polarity test review .

Methodology

Cleared train review and test review using regular expression and removed all words having who's having length less than 5 .Combined both training and test dataset before creating CSR matrix in order to have equal dimensionality . Now once we have CSR sparse matrix , calculate similarity using similarity function . Cosine Similarity function from sklearn is used as it was faster and more accurate than euclidean distance similarity function. After cosine similarity it will have 25k x 25k matrix . Now using numpy arg partition sort the each row of matrix as it was much efficient than python sorted function.Now with top K neighbours count the polarity .