HW1 - Yelp Review Classification

Name : Divya Malyala
GUID : G01390473
Miner id : Doraemon
Accuracy : 81
Rank : 117

Introduction:

The Homework is about implementing the KNN classifier for the Yelp reviews classification dataset to predict the new reviews of the customers. The train dataset contain 18000 rows with 2 columns. The test dataset having 18000 rows with review column has text data of customer reviews on Yelp Classification.

Methodology:

Required toolkit and Libraries:

- Numpy: It's a python package that we use for mathematical operations on 2D arrays like transpose, sort, and arithmetic operations which perform efficiently. In the code, I used this package for sorting the 2 array index positions of cosine similarity of train and test data by the argsort() function.

- Pandas: It's a useful library to perform data analysis on most of the files like .csv, .json, and .excel files. It's easy to use and understandable to perform operations on data like a filter, replace, and slice the data based on the requirement of the user.

- nltk: NLTK is a toolkit that works on text data to perform natural language processing by importing required libraries such as tokenization, stemming, and stopwords. That helps to create root words for text with stemming and remove the stopwords from the text file.

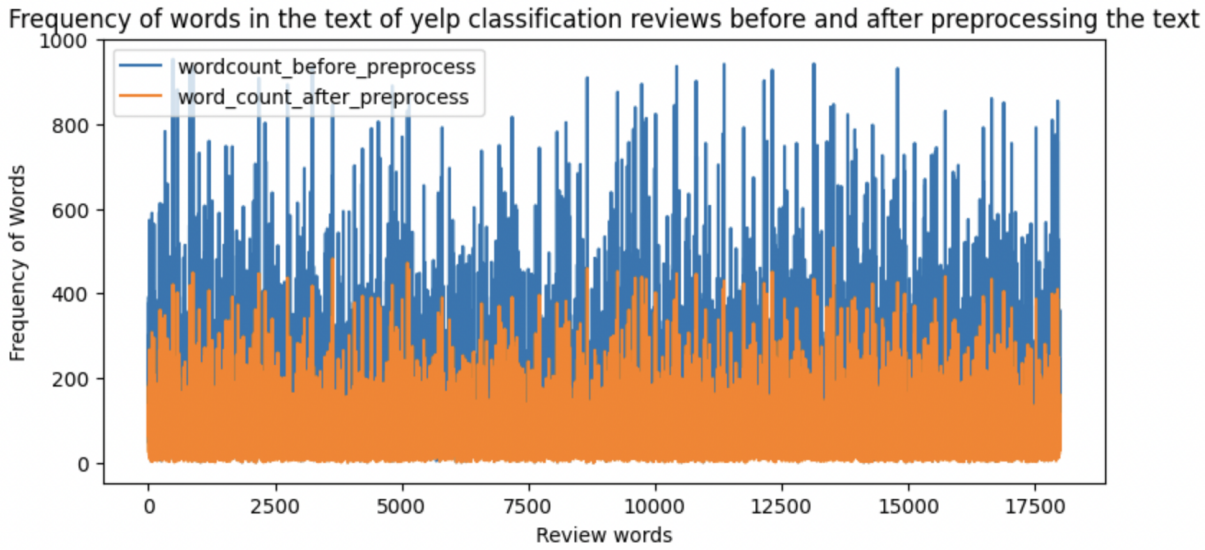
- sklearn: It is a useful library that mainly uses statistical data and machine learning for classification, clustering, and regression models. In this library, I've imported feature_extraction, model_selection, and metrics for TfidfVectorizer,

KFold, and cosine_similarity. TfidfVectorizer is used for converting the text into the sparse matrix by transforming and cosine_similarity gives the similarity of test data with the train data. KFold is used for k-fold cross validation to accuracy of all splits in the data without any shuffle that helps to choose the K value.

- Matplotlib: It's a plotting library that we use for visualization of any report or classification as bar charts, or line graphs by importing pyplot in python that helps to visualize the data of accuracy and K values to choose the best K value in our algorithm and to represent the positive and negative reviews.

Preprocess:

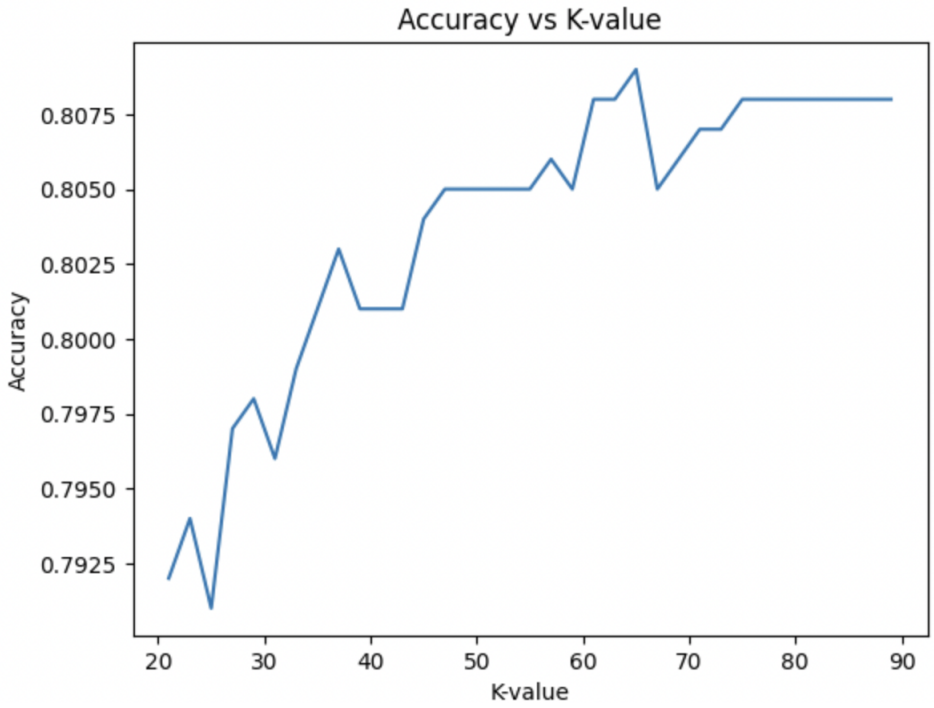
- Load the .csv files and create the pandas data frames of train and test data by read_csv() with the pandas package
- Creating the column names as review and label of train dataset and preprocess the data by removing special characters, and digits, and replacing the helping words like " 've ", " 'll ", " 're ", and " n't " with 'have', 'will', 'are', 'not'.
- Calling the PorterStemmer function and removing the stopwords and replacing the words with stem words in the text in train and test data.
- Creating TfidfVectorizer object for review column with max_features parameter and transforming the training text into sparse matrix
- Fit the Test data after preprocessing the text to the TfidfVectorizer object that transforms the data frame that returns the data in the sparse matrix that helps to improve the efficiency and competency of the code.



KNN Implementation:

1. Created the loop to choose the best K value for the k-Nearest Neighbor Classifier algorithm through k-fold cross validation with n_splits of 5
2. Plot the graph of accuracy and K value to pick the K value as there should not be any overfitting and underfit the k value so i have chosen k = 77 after k =77 it remains constant it only differs with 0.002 accuracy.
3. After choosing the K value, find the similarity of the train and test the sparse matrix with cosine similarity as we have similar text which helps to find the similarity of documents. Apart from Euclidean distance measures the distance whereas cosine similarity gives angle where Small angle gives high similarity of the text to identify the prediction label.
4. Creating a KNN function with cosine similarity and performing transform function and returns the nearest neighbors for that test row and with respect to K value, I took all K nearest values and predicted the y_pred of test data.

```
: Text(0, 0.5, 'Accuracy')
```



5. Writing prediction function by taking the nearest indexes of test data and getting the prediction values of the nearest values. Calculating the prediction label by taking a max of positive and negative reviews
6. Save all the prediction labels in the data.txt file to save a copy of the results by opening the file with write mode.