Yelp Review Classification Report

Name : Billakanti Nikhil

GNumber : G01393314

Miner ID : nbillakanti

Accuracy : 72

Rank : 303

Aim: Develop a knearestneighbor classifier from scratch to find the sentiments of 18000 reviews from test.csv files.

Explanation:

- I have carried out the process in 4 major steps:
    1. Preprocessing of data.
    2. Transforming the train and test data.
    3. Measuring cosine similarity.
    4. Predicting the test set values.
- Initially imported few important libraries such as pandas, re for regular expressions, nltk library for stopwords, snowball stemmer library for stemming of words, count vectorizer library for vectorizing the text data, cosinesimilarity from sklearn.metrics.pairwise for finding the similarity between two vectors and csv library to store the final predicted values into .csv file.
- In next step, imported train csv file and test csv file using pandas read_csv function and it returned the dataframes of train and test.
- Train set consists of 18000 rows and 2 columns where one first column describes rating as +1 for positive review or -1 for negative review and second columns has reviews.
- And the test csv file has 18000 rows of reviews and rating has to be predicted.

Data Cleaning

- Later in preprocessing, I have followed the below steps:
    1. Using regular expressions extracted the words without considering special characters.

2. The regular expression[^a-zA-Z] considers all the words starting with alphabets of any case
3. Cleared all the new line characters by replacing \n with a space character.
4. Using the split method returned the words into the list to make the later preprocessing part easier.
5. Then stemming is done to obtain meaningful words. In this project, snowball stemmer is used to extract meaningful words.
6. Then stopwords are removed from the list using the library.
7. Removed the words with word length less than 2 as there are no adjectives with word length less than 2 which might mislead the model.
8. Converted all the words to lowercase letters.
9. Finally all the processed words are joined together to form a sentence and appended into train_list to carry out transformation.
10. Train_list and test_list are the processed data with no outliers, stopwords, or punctuations.

Transformation

- After the preprocessing step, obtained text data is converted into vector form based on the frequency of words.
- Countvectorizer is used in this project to transform the data and extract important features. Countvectorizer has helped in improving the accuracy of the code.
- Max_features is the parameter considered in this code with 2000 as the feature value.
- Transformation is done for both train and test set.

Model building and training

- Using K-fold cross validation, splitting is carried on train data and the set with best accuracy is considered for prediction.
- Then made use of cosine similarity to obtain the similarity between two vectors namely train and test vectors.
- Flatten() method is used to convert 2D array to 1D array.
- Later argsort() function is used to get the index values of the sorted array which later is used to return the rating of train data. The index values of k closest elements is considered and sum is calculated.
- The sum of neighbor values is calculated and +1 is returned for positive review and -1 is returned for negative value.

- The k value for nearest neighbors is choosen by trial and error method.

- Taken different values of k and obtained accuracy for the test set and the K value with best accuracy is choosen. I have got the best accuracy for k=87.
- Finally the predicted values are taken into result.csv file and checked for accuracy on miner and obtained an accuracy of 72.
- From this assignment I got to learn about the importance of data cleaning and how outliers in text data affect the model and also learnt to build KNN algorithm from scratch.
- Most importantly got to know the functional behaviour of K-fold cross validation and its effect on model.