AdityaVetukuri

Accuracy : 0.76

Rank: 63

**CS 584 (HOMEWORK-1)**

Amazon Review Classification using K-neighbours classification

**Libraries used for this Project**

* Pandas
* Numpy
* Nltk
* String
* Re
* Tqdm
* Sklearn
* scipy

**Step by Step procedure:**

* **Pre-processing the data** : After reading our input files we send them to a preprocessing functions. I used the below two functions to clean data
* **Cleaning method** : In this method I’ve used some in built functions of “**re**” and removed all the punctuations, numbers and special characters. Also I have changed all the text to lower case.
* **Stemming method**: In this method I’ve used Porter Stemmer and stemmed each word and also at the same point I’ve removed stopwords.
* **Vectorizing the data** : For the Vectorizing of data I tried using three feature extractors Count Vectorizer, Binary Vectorizer and tf-idf Vectorizer.
* **Distance matrix** : For the Distance I’ve used pairwise distance function. It either takes a vector or a matrix and returns a distance matrix. So I’ve sent my training data set and calculated all the distances using this.
* **KNN Algorithm** : Firstly, I have calculated the distances using pairwise distance function.
* I have mapped the distances to their labels and sorted them key wise.
* Later I took a majority of the top K distances and retrieved their labels and found the majority of label for classification and appended it to the resultant list.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| K value | 5 | 10 | 25 | 50 | 100 |
| Count Vectorizer | 77.097861 | 78.067721 | 77.672321 | 75.786987 | 72.43187 |
| Binary Vectorizer | 70.032761 | 72.78999 | 69.783212 | 68.73212 | 70.32992 |
| tf-idf | 80.045321 | 81.563208 | 80.12879 | 78.67432 | 76.43521 |

**Cross- Validation:**

For the parameter choices I used **K-fold cross validation** where I divided my dataset into multiple batches and running each batch on different K values and used all the feature extraction methods and appended the accuracy scores to a list and figured out the best score.

I have observed that I was getting the best score for **tf-idf Vectorizer** using **K** value as 14 with an Accuracy of **76** on **Miner**. Where as on local training test split I was getting a best Accuracy of **85**.

**How I optimized my algorithm**: At first I was looping all over the distance matrix and storing the class labels in a separate array. And to get the class majority I was iterating the class array once again this costed me a lot of run time. It was literally 4 iterations/sec. Later I used map and pairwise-distances functions which saved me a lot of runtime literally 8 iterations/sec.

**TO RUN MY CODE:**

* **Python executable.py.**
* **I’m also submitting the preprocessed data in the submission so that it will be easy for you to run my code and save your time.**
* **Test\_data.dat , Train\_data.dat and ad\_data\_pre.pkl**

**Reference:**

[**https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/**](https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/)

I followed the above link to get a better understanding of this Assignment.