HW 1 : Amazon Review Classification

Name : Venkata Pratyush Kodavanti

Miner username: vkodavan

Accuracy: 0.77

I have used two separate files one for validation (Cross_Validation.py) and the main program(knn.py).

Approach:

The project can be broken down into 4 parts

1) Preprocessing the data
2) Vectorizing the preprocessed data
3) Calculating the K-nearest neighbor.
4) Validation (Cross-validation)

Preprocessing the data:

For preprocessing, I have used standard existing libraries. Pandas for reading the data files (Training and Testing). Once I have the data, next step is to preprocess the data. For preprocessing I have used NLTK library (word_tokenize and stopwords) to tokenize the data and remove stop words from these tokens. Further, I have used porter stemmer from nltk library to normalize the terms when passed to TFIDF vectorizer. After cleaning the data, we send it for vectorizing the reviews.

Vectorizing the preprocessed data:

After preprocessing the data, I have used TFIDF vectorizer from sklearn.feature_extration.text which calculates the term frequency- Inverse document frequency matrix.

Term-Frequency calculates how many times a word appears in the document (which might be problematic for common words (such as articles), to avoid this problem we use Inverse document frequency.

Inverse Document Frequency: It calculates the log {number of docs in the corpus /number of docs in which the term appears}

Tfidf = tf*idf which would generally be between 0 to 1.

Hyper-parameters used/ chosen: (stop-words: English),

Max_df (1.0): Ignores the terms whose document frequency is more than 1.0

Min_df (0.0015): Ignore the terms whose document frequency is less than it.

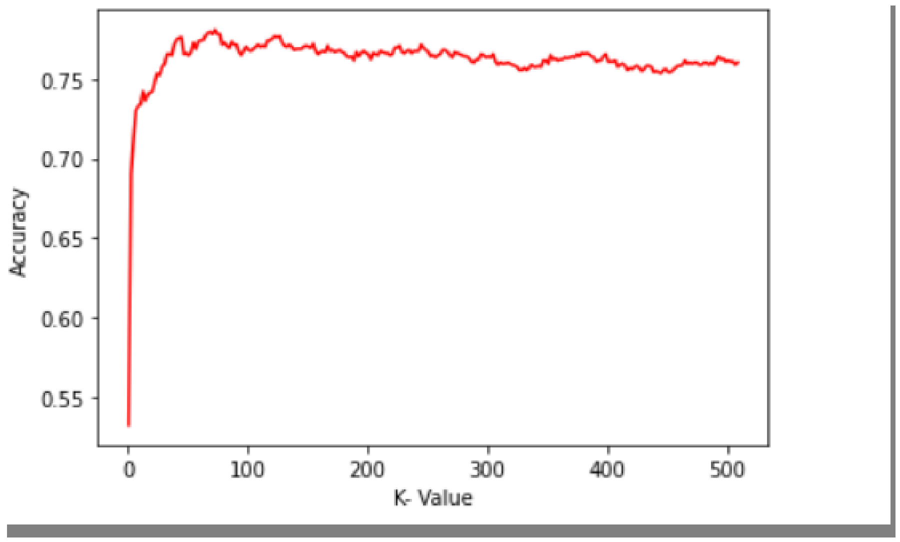
Max_features(2500) : select top 2500 frequent words.

(To normalize the data and reduce the number of computations while calculating the cosine distance).

Calculating the K-nearest neighbor: To calculate the k-nearest-neighbor, I used cosine similarity of each record in the test TFIDF matrix with the train TFIDF matrix. From the cosine similarity of each record in test TFIDF I selected the indices of top k values. After receiving the k nearest neighbor indices, based on the sum of the neighbors I have decided to classify the review.

Calculating the cosine similarity: For that I have used the sklearn.pairwise.cosine_similarity to calculate the cosine similarity if the training and test TFIDF vectors.

Choosing the K-value: I have tried and tested a bunch of k-values and for K = 69 I got an accuracy of 77.83.



Validating the performance of the algorithm:

For validation, I have used K-cross validation (10 folds), I have tried for different value of K nearest neighbors(ranging from 1 to 511). For K-cross validation I have used scikit learn's KFold library to split the training into k parts for calculating the accuracy.

For calculating the accuracy, I have used scikit learn's accuracy_score.

References:

1) I have gone over the documentation of nltk, scikit learn to understand some of the pre-existing libraries.

2) https://www.analyticsvidhya.com/blog/2018/02/the-different-methods-deal-text-data-predictive-python/

3) https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-classification