Name: Saeed Alneyadi.11

Project Report

Detailed Description

The homework 3 program uses some common methods and functions to achieve what it is asked for. *First*, it partitions the original dataset into training, validation, and testing datasets, each based on its ratio. *Second*, it applies feature-based selection on copy of the original dataset that consider the top K frequent words (I choose K = 500) after removing words 'the' and 'a'. This is done to produce the pruned dataset by summing the frequency of each word over all sentences, and then consider the K number of words with the highest sum frequencies. These words are the feature kept in the pruned dataset and the other is discarded. *Third*, the pruned dataset is separated into other training, validation, and testing datasets. *Forth*, two functions are implemented, "KNN()" and "best_KNN()". The "KNN()" apply K nearest neighbor by taking the input: (Original/Pruned) Training Dataset, (Original/Pruned) (Training/Validation/Testing) Dataset, Sentences Training Matrix, Sentences (Training/Validation/Testing) Matrix, and L value, and it outputs: the accuracy of KNN. The "best_KNN()" finds the K of nearest neighbor algorithm with the best accuracy among other K's for a the inputted datasets. Same input as KNN(), but it outputs the best K with the best accuracy. *Fifth*, sentences list is divided into training, validation, and testing matrices. These matrices hold the sentences and the actual labels associated with them. *Sixth*, the program runs "best_KNN()" for both for original and pruned dataset to calculates the time it takes to find the best KNN classifier model which then is printed to the terminal. *Seventh*, its runs "KNN()" for both for original and pruned dataset to calculates the classifier time of the new tuple which then is printed to the terminal. *Eighth*, the program runs "KNN()" to find the training, validation, and testing accuracies for the original and pruned dataset. *Ninth*, the program initialize Gaussian Naïve Bayes Model for the original dataset and the pruned dataset. *Tenth*, "fit()" are called to calculate the total training time for the model of original dataset and of the pruned dataset. *Eleventh*, "predict()" are called to calculate the total classification time for the models of original dataset and pruned dataset. Twelfth, the program reruns "predict()" to find the training, validation, and testing accuracies for the original and pruned dataset.

Program Performance

I ran the program with these varaibles:-

- Feature-base selection variable (top K number of words with highest frequencies) : top_k = 500
- Dataset Ratios (training p%, validation q%, testing r%): p = 0.70, q = 0.15, r = 0.15

Name: Saeed Alneyadi.11

```
KNN Classifier:
----------------------
The total building time for the best KNN model of the original D dataset is: 2547.395462989807 seconds
The total building time for the best KNN model of the pruned D dataset is: 1070.9015510082245 seconds
The classifier time of the new tuple for the original D dataset is: 0.026772260665893555 seconds
The classifier time of the new tuple for the pruned D dataset is: 0.005464076995849609 seconds
The training accuracy of the best KNN model for the original D dataset is: 0.6390476190476191
The training accuracy of the best KNN model for the pruned D dataset is: 0.6371428571428571
The validation accuracy of the best KNN model for the original D dataset is: 0.6755555555555556
The validation accuracy of the best KNN model for the pruned D dataset is: 0.66
The testing accuracy of the best KNN model for the original D dataset is: 0.6
The testing accuracy of the best KNN model for the pruned D dataset is: 0.6044444444444445

Naive Bayes Classifier:
----------------------
The total training time for the Gaussian Naive Bayes model of the original D dataset is: 0.0682058334350586 seconds
The total training time for the Gaussian Naive Bayes model of the pruned D dataset is: 0.008477926254272461 seconds
The total classification time for the Gaussian Naive Bayes model of the original D dataset is: 0.019572019577026367 seconds
The total classification time for the Gaussian Naive Bayes model of the pruned D dataset is: 0.00203704833984375 seconds
The training accuracy of the Gaussian Naive Bayes model for the original D dataset is: 0.9285714285714286
The training accuracy of the Gaussian Naive Bayes model for the pruned D dataset is: 0.7623809523809524
The validation accuracy of the Gaussian Naive Bayes model for the original D dataset is: 0.6644444444444444
The validation accuracy of the Gaussian Naive Bayes model for the pruned D dataset is: 0.6511111111111111
The testing accuracy of the Gaussian Naive Bayes model for the original D dataset is: 0.6733333333333333
The testing accuracy of the Gaussian Naive Bayes model for the pruned D dataset is: 0.6777777777777778
```

Figure 1 Feature-base selection variable: top_k = 500; Dataset Ratios: p = 0.70, q = 0.15, r = 0.15

KNN Classifier:

- Regarding the building time, the pruned dataset was faster since it is smaller compared to the original one.
- The classifier time for a new tuple is smaller for the pruned dataset since it deals with smaller feature vectors where fewer features compared by calculating distances.
- The training accuracy of the original dataset is higher compared to the pruned one. This means that having more features may help capture more complexity.
- The validation accuracy is higher in original dataset, which give an indication that pruned dataset lost some information that may be useful in the prediction process.
- The testing accuracy of the pruned dataset is bit higher than the original one as expected. It was expected to be more higher but it seems that feature-base selection didn't do that much.

Overall, there not much of difference between the two, indicating that the feature-base selection didn't give that much of an effect. The only difference or advantage made by this selection is the budling and classifier time.

Naïve Bayes Classifier:

- Roughly, both dataset have the same total training time due to the one of the most common characteristic of naïve bayes which is fast training times since it uses simple calculations
- Not that much of a difference in classification time between the original and pruned dataset.
- Original dataset Naïve Bayes Model have higher training accuracy to the pruned one.
- In validation accuracy, original dataset was little better off than the pruned one, indicating that the additional features of the original one helps on giving a correct predictions
- Both datasets have roughly the same testing accuracy.

Overall, there not much of difference between the two, indicating that the feature-base selection didn't give that much of an effect. This due to high calculation speed of Naïve Bayes Models.

Name: Saeed Alneyadi.11

```
SVM Classifier:
----------------------
The total training time for the SVM model of the original D dataset is: 10.530171155929565 seconds
The total training time for the SVM model of the pruned D dataset is: 0.8751211166381836 seconds
The total classification time for the SVM model of the original D dataset is: 2.8750851154327393 seconds
The total classification time for the SVM model of the pruned D dataset is: 0.27272796630859375 seconds
The training accuracy of the SVM model for the original D dataset is: 0.9371428571428572
The training accuracy of the SVM model for the pruned D dataset is: 0.8957142857142857
The validation accuracy of the SVM model for the original D dataset is: 0.7333333333333333
The validation accuracy of the SVM model for the pruned D dataset is: 0.7088888888888889
The testing accuracy of the SVM model for the original D dataset is: 0.7488888888888889
The testing accuracy of the SVM model for the pruned D dataset is: 0.7288888888888889
```

Figure 2 Feature-base selection variable: top_k = 500; Dataset Ratios: p = 0.70, q = 0.15, r = 0.15

SVM Classifier:

- Total Training Time: Original Dataset >> Pruned Dataset
  Possible Reason: Complexity from large number of features
- Total Classification Time: Original Dataset >> Pruned Dataset
  Possible Reason: Fewer dimensions of pruned dataset → Less Computations
- Training Accuracy: Original Dataset little higher compared to the Pruned Dataset
  Possible Reason: Original Dataset has more information.
- Validation Accuracy: Original Dataset little higher compared to the Pruned Dataset
  Possible Reason: Original Dataset has additional features.
- Testing Accuracy: Original Dataset little higher compared to the Pruned Dataset
- Possible Reason: Pruned Dataset has less features.

Overall, there is not that much of difference in Accuracies even though the original dataset is little higher. However, the Pruned Dataset Model is very fast compared to the Original one in building and classification time.

Difficulties Encountered

One of the most common issues encountered is maintaining the right dimensions, shapes, lengths of the matrices or list in the python program, especially the matrices related to the dataset. The dataset needed to be built in specific dimensions so that it works for both my implemented functions and the built-in functions.