Contents

1. Abstract

Activity Classifier classifies the activity of a person based on wrist worn accelerometer readings. These activities include walking, sitting, lying and climbing stairs. The classifier is implemented in **Python** using **scikit-learn. Scikit-learn** is a free software machine learning library for the Python programming language. This report includes the methodology adopted for training data set using different algorithms, a comparison of the performance of these algorithms and predictions made on unseen data by using of the algorithms whose performance was better than the other algorithms.

2. Methodology

Adopted methodology consisted of following steps:

- Data Collection
- Data Wrangling
- Training the Models
- Model Analysis
- Algorithms/Models Comparison
- Predictions on New/Unseen Data

The details of the above mentioned steps are as follows:

2.1. Data Collection

The data set was taken from UCI Machine learning repository. The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. The link for the data set is attached in references.

2.2. Data Wrangling

The data set was uneven i.e. it included more readings for some activities than the other activities. So to maintain an even representativeness of our samples, we took the readings of 25 persons for each activity and each person’s readings included 300 measurements of the accelerometer.

We read all these readings in our program firstly as individual data frames and then we merged those data frames into a single large data frame which included all of the training data. The final data frame was then shuffled randomly so that the order of the examples does not affect our model.

2.3. Training the Models

The data set was then trained on different algorithms to generate models. The models generated were then analyzed to find out which algorithm gives better performance. We trained the data set on following algorithms:

- Logistic Regression
- Linear Discriminant
- K Neighbors Classifier (KNN)
- Decision Tree Classifier
- Naïve Bayes
- Support Vector Machine (SVM)
- Multilayer Perceptron (MLP) Classifier
- Random Forest Classifier

2.4. Model Analysis

The models produced by each algorithm were then analyzed and a comparison was made between the results of all the models. This analysis gave better insight about which algorithm is generating a better model which can generalize the data i.e. which algorithm is giving a good accuracy on training data but is not giving good results on cross validation data.

The model produced by each algorithm was analyzed based on following metrics:

- Training Accuracy
- Cross Validation Score
- K-Fold Cross Validation Score
- Precision
- Recall
- F1 Score

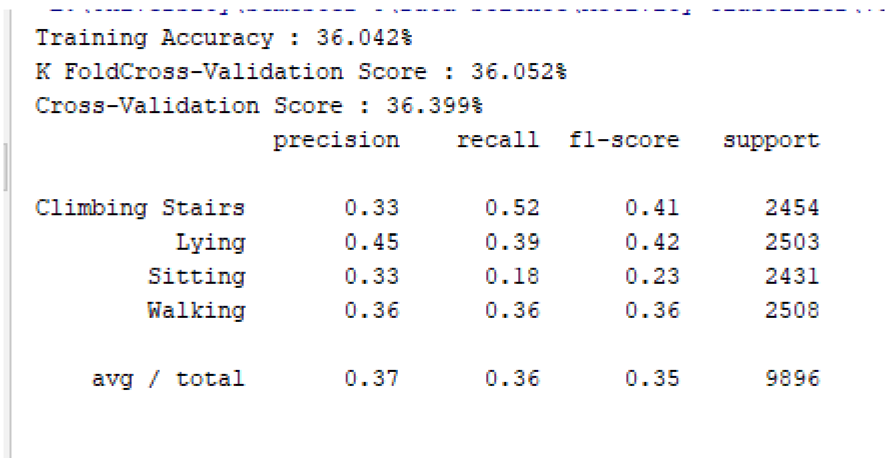
All of the above metrics were compared to gain a better insight about which algorithm is performing better than the others. Moreover learning curves for each algorithm were also plotted to analyze the performance of the models on training set and cross validation set.

The files named by the algorithms name contain the code for training data on that algorithm and analyzing the model produced by that algorithm. For example “KNN.py” file contains the analysis of model produced by K Neighbors Classifier algorithm.

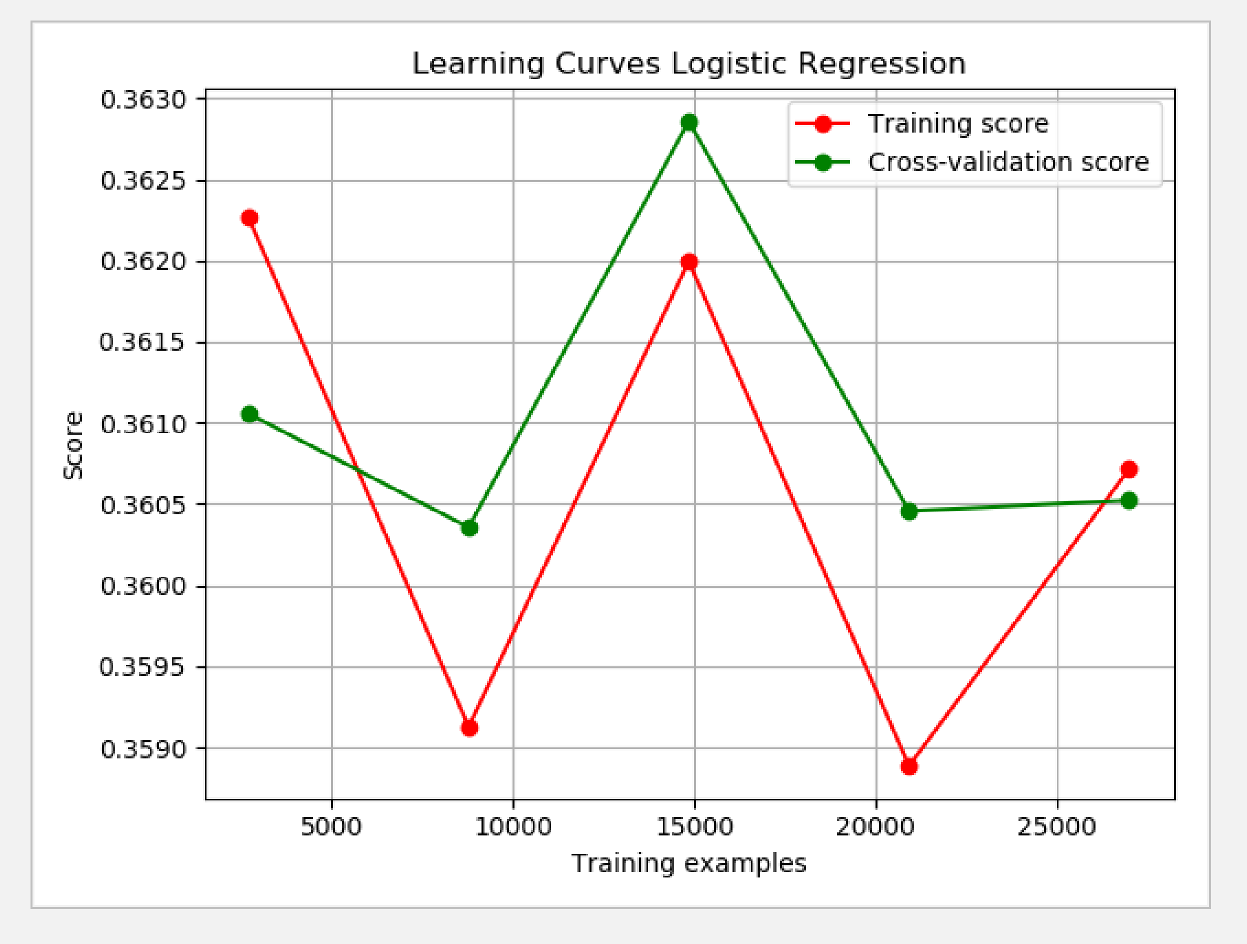
2.4.1. Results

The results of analysis performed on models generated by different algorithms are as follows:

2.4.1.1. Logistic Regression

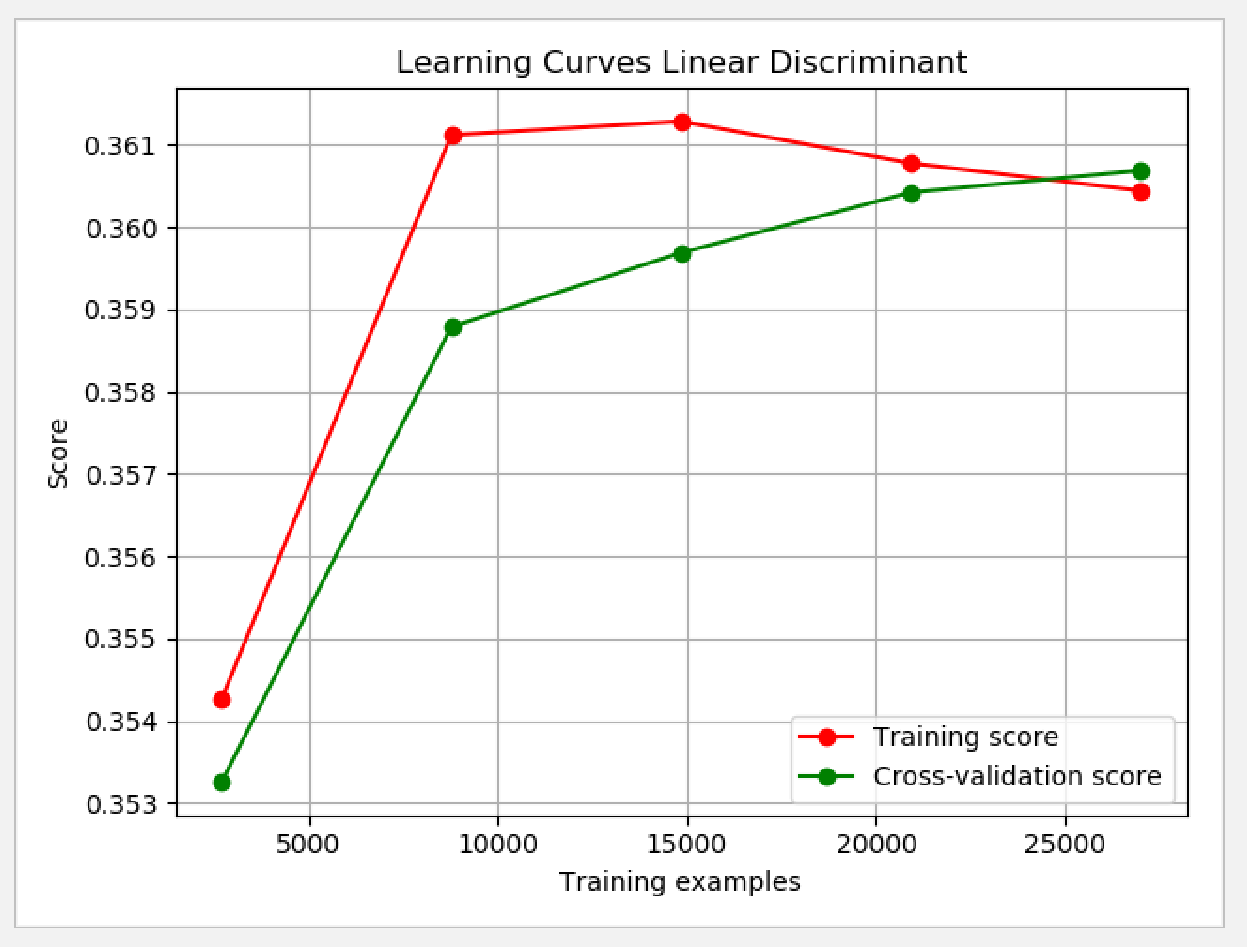


```
Training Accuracy : 36.042%
K FoldCross-Validation Score : 36.052%
Cross-Validation Score : 36.399%
                 precision    recall  f1-score   support

Climbing Stairs       0.33      0.52      0.41      2454
          Lying       0.45      0.39      0.42      2503
        Sitting       0.33      0.18      0.23      2431
        Walking       0.36      0.36      0.36      2508

    avg / total       0.37      0.36      0.35      9896
```

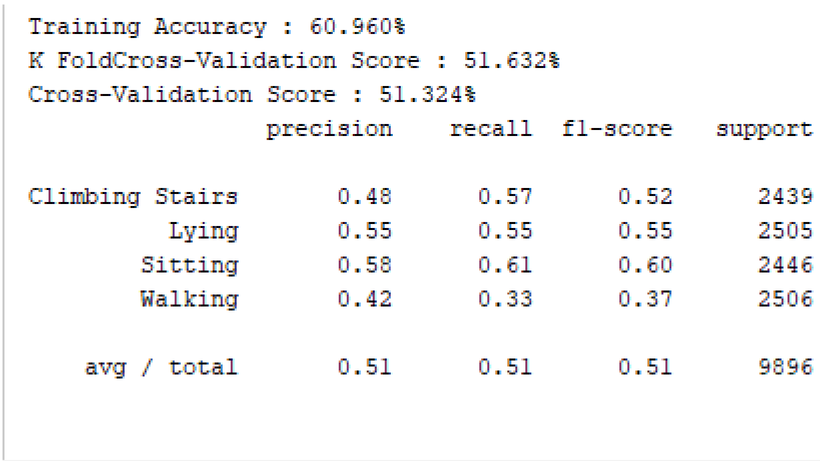


2.4.1.2. Linear Discriminant

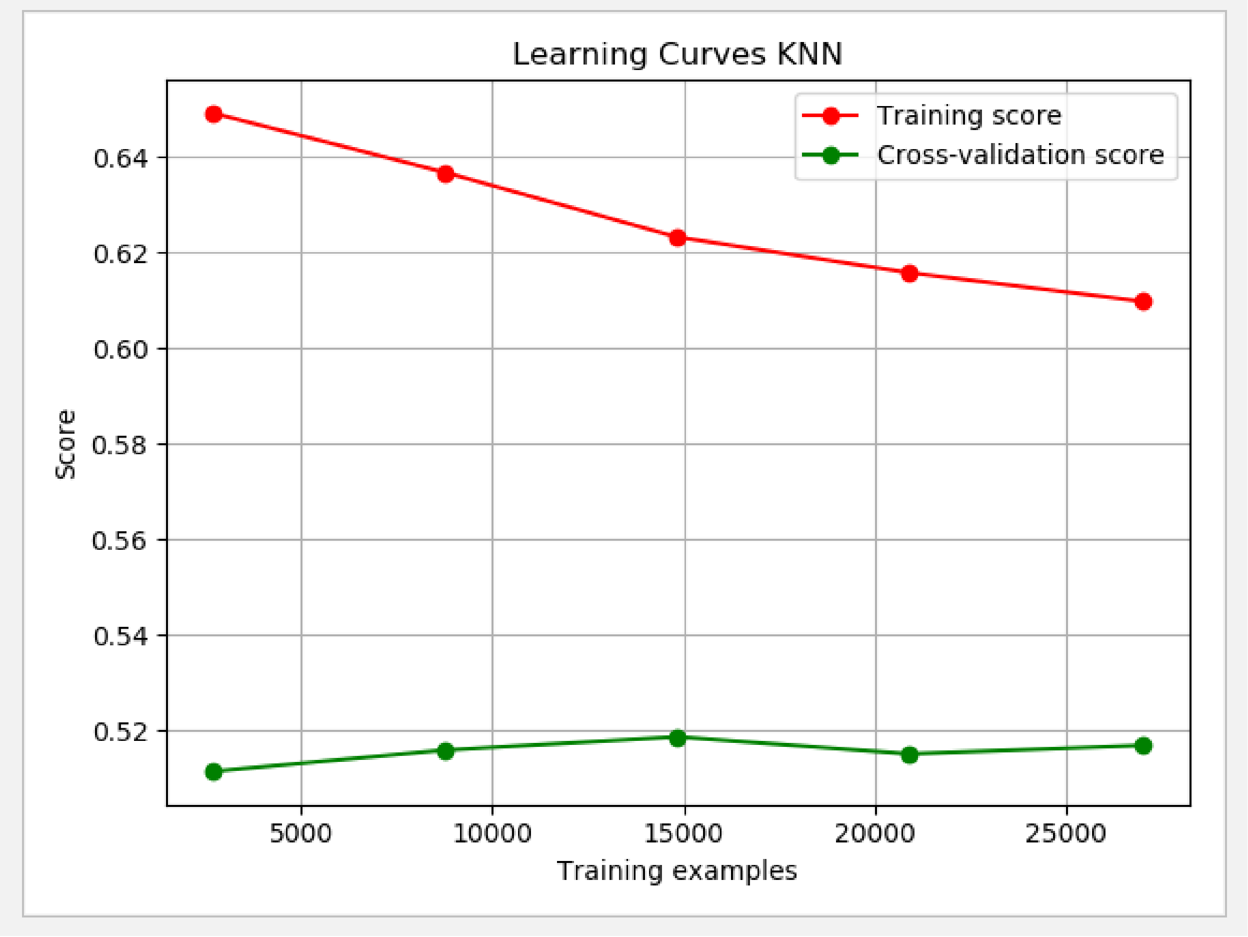
```
Training Accuracy : 36.052%
K FoldCross-Validation Score : 36.116%
Cross-Validation Score : 36.530%
                 precision    recall  f1-score   support

Climbing Stairs       0.35      0.48      0.40      2482
          Lying       0.44      0.40      0.42      2472
        Sitting       0.34      0.19      0.24      2507
        Walking       0.34      0.39      0.36      2435

    avg / total       0.37      0.37      0.36      9896
```



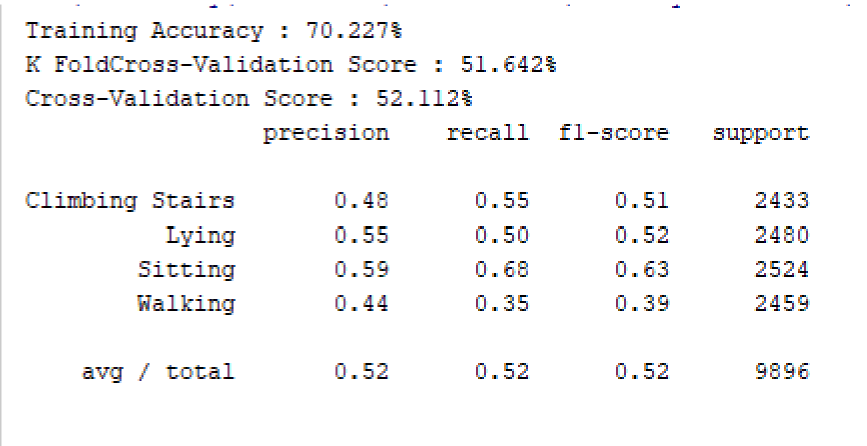
2.4.1.3. K Neighbors Classifier (KNN)



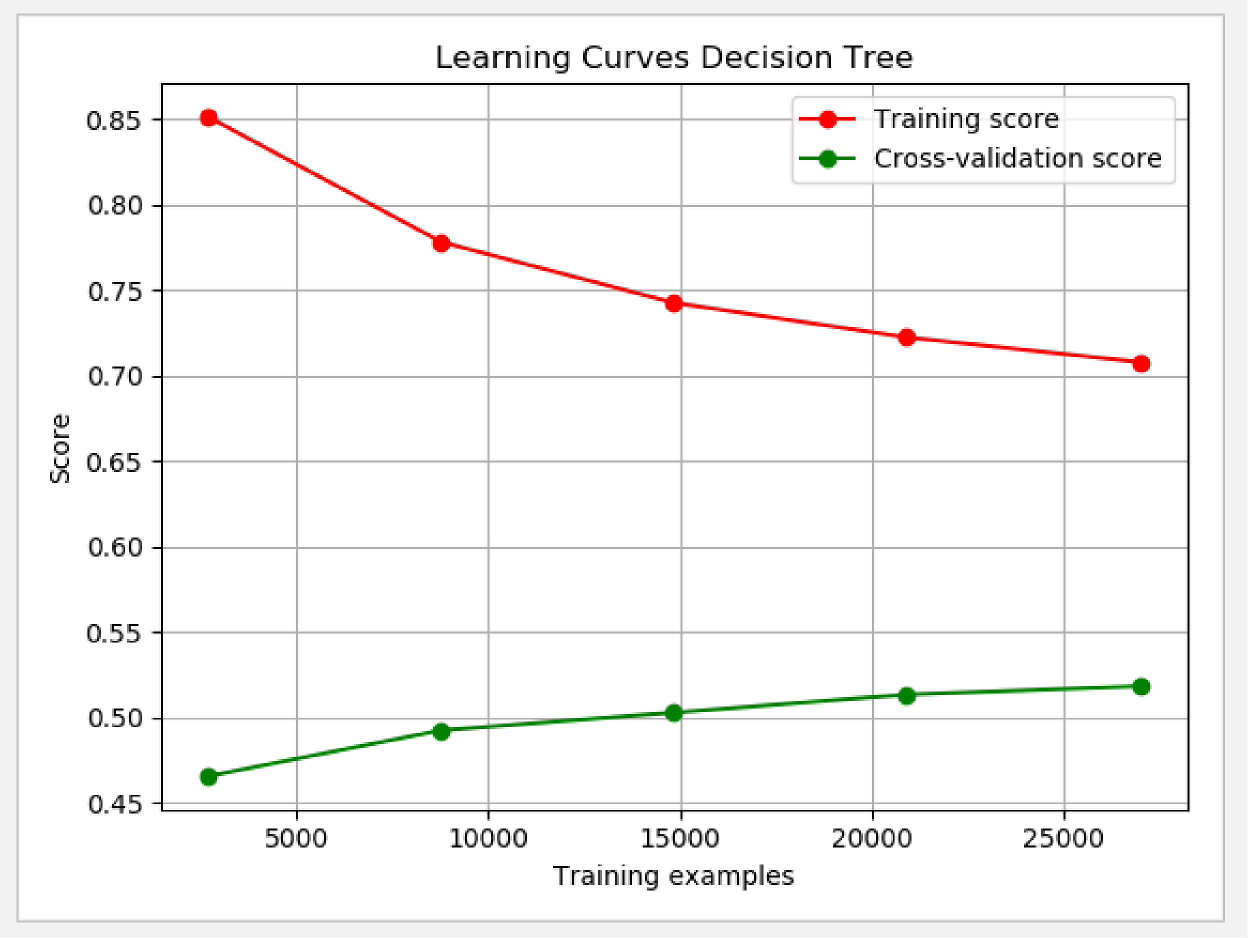
```
Training Accuracy : 60.960%
K FoldCross-Validation Score : 51.632%
Cross-Validation Score : 51.324%
                 precision    recall  f1-score   support

Climbing Stairs       0.48      0.57      0.52      2439
          Lying       0.55      0.55      0.55      2505
        Sitting       0.58      0.61      0.60      2446
        Walking       0.42      0.33      0.37      2506

    avg / total       0.51      0.51      0.51      9896
```



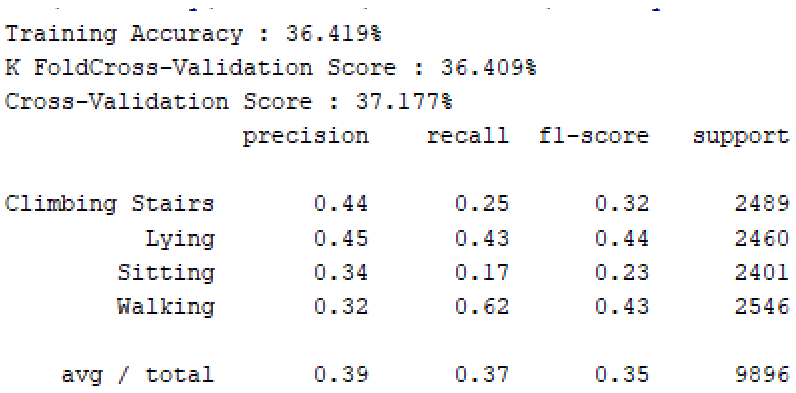
2.4.1.4. Decision Tree Classifier



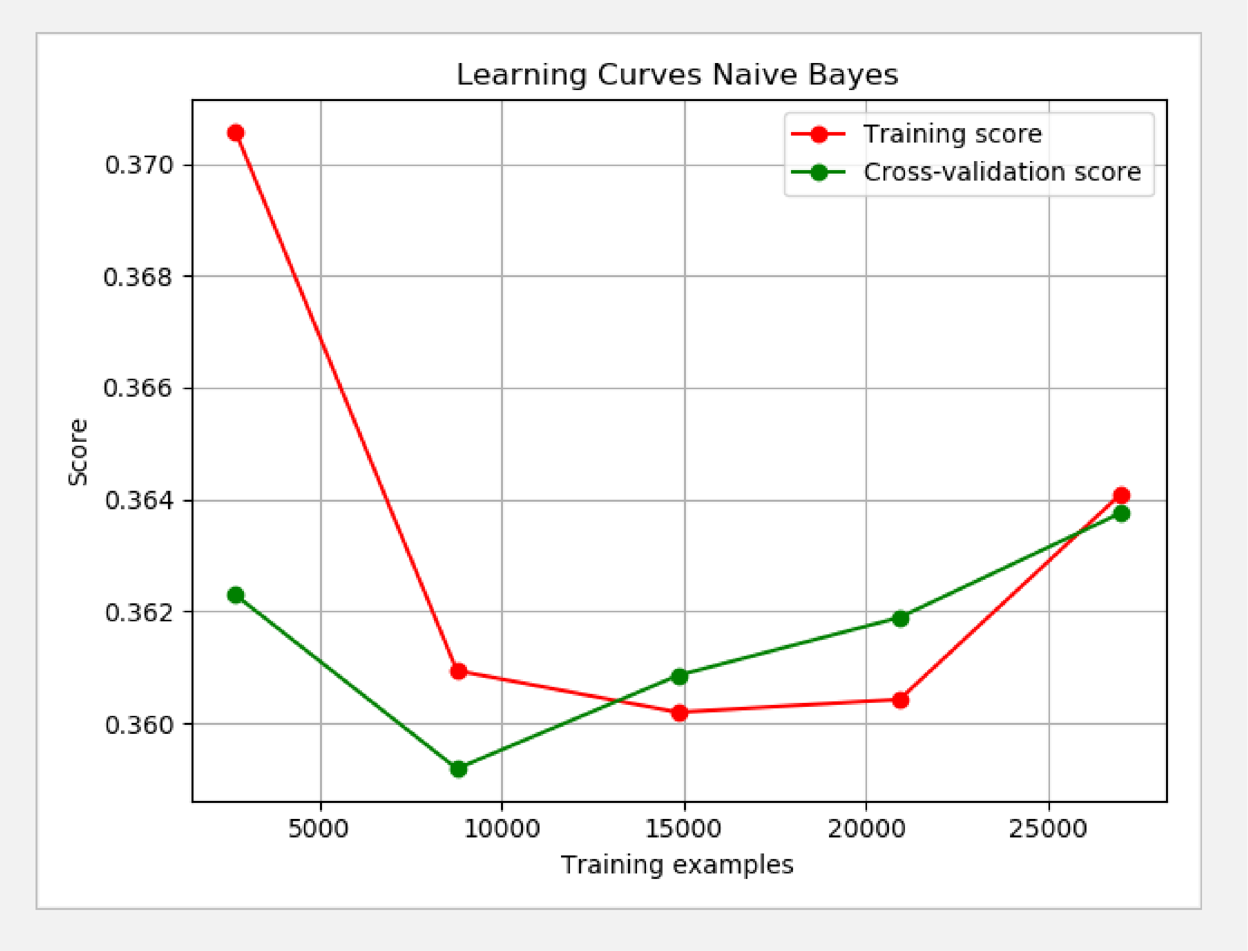
```
Training Accuracy : 70.227%
K FoldCross-Validation Score : 51.642%
Cross-Validation Score : 52.112%
                 precision    recall  f1-score   support

Climbing Stairs       0.48      0.55      0.51      2433
          Lying       0.55      0.50      0.52      2480
        Sitting       0.59      0.68      0.63      2524
        Walking       0.44      0.35      0.39      2459

    avg / total       0.52      0.52      0.52      9896
```



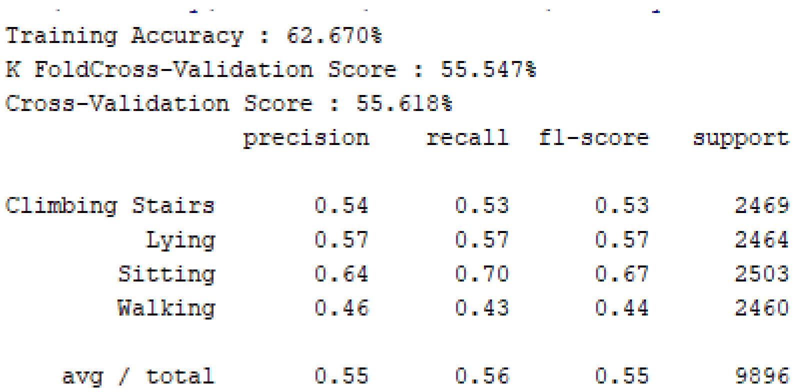
2.4.1.5. Naïve Bayes



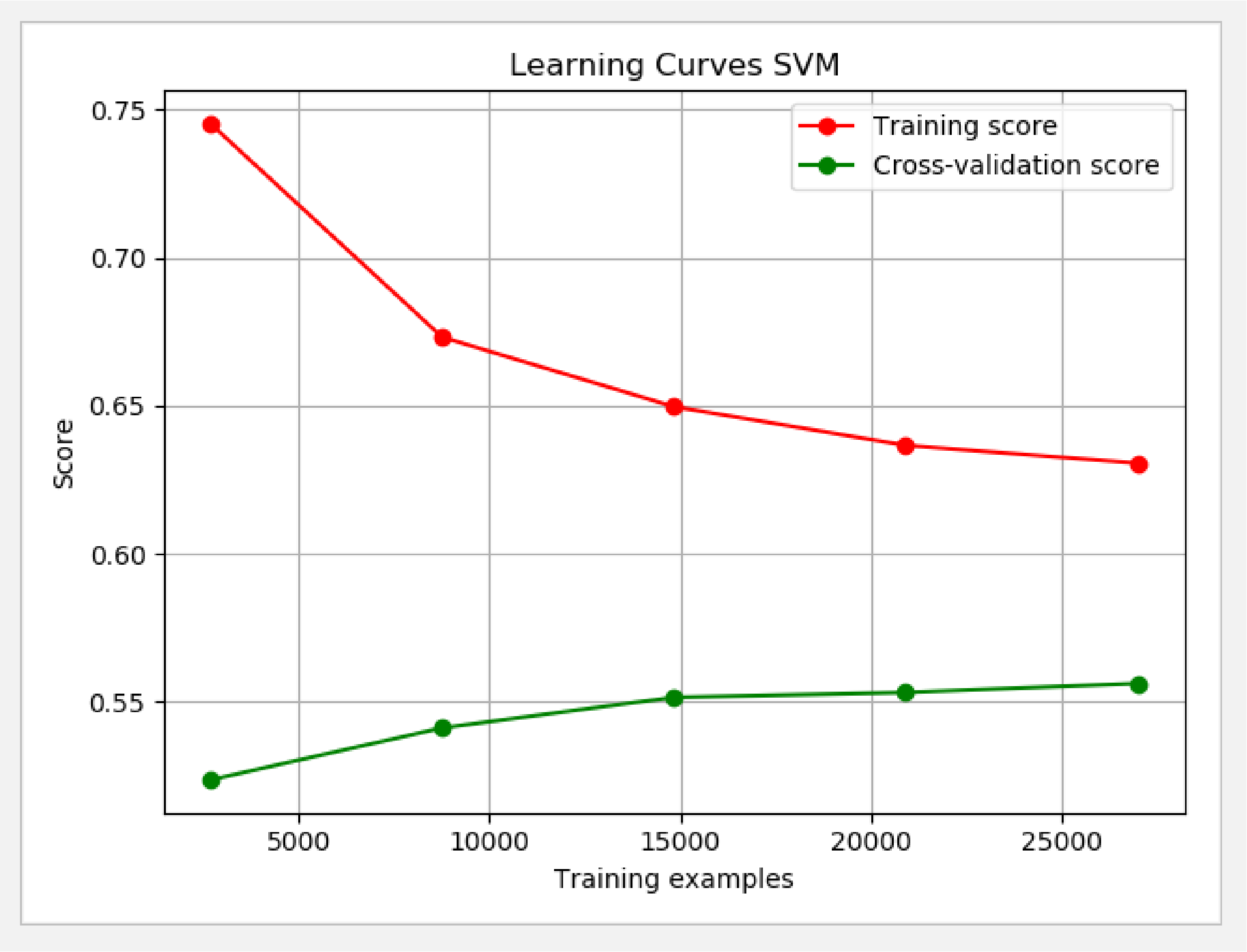
```
Training Accuracy : 36.419%
K FoldCross-Validation Score : 36.409%
Cross-Validation Score : 37.177%
                 precision    recall  f1-score   support

Climbing Stairs       0.44      0.25      0.32      2489
          Lying       0.45      0.43      0.44      2460
        Sitting       0.34      0.17      0.23      2401
        Walking       0.32      0.62      0.43      2546

    avg / total       0.39      0.37      0.35      9896
```



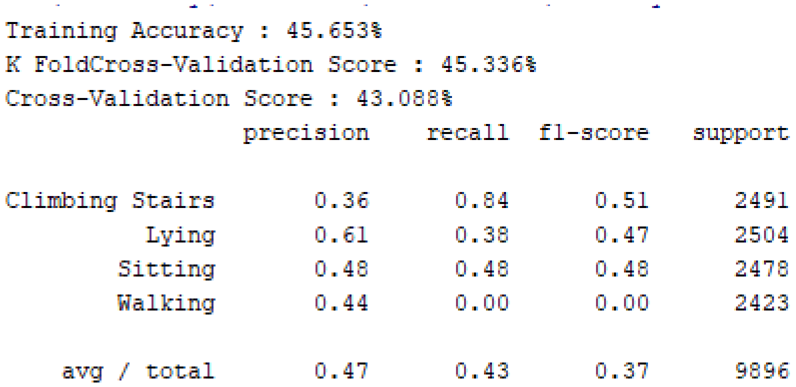
2.4.1.6. Support Vector Machine (SVM)



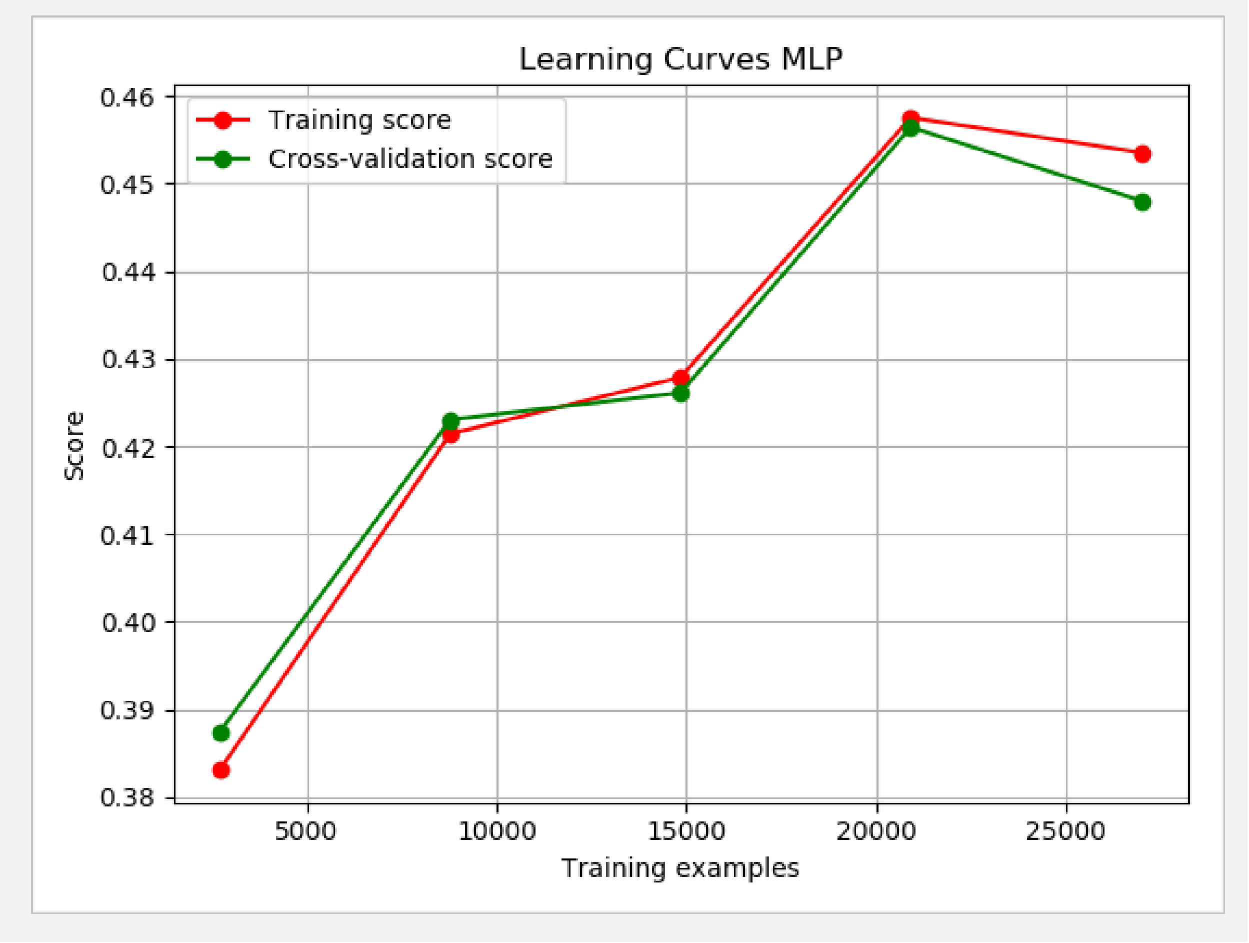
```
Training Accuracy : 62.670%
K FoldCross-Validation Score : 55.547%
Cross-Validation Score : 55.618%
                 precision    recall  f1-score   support

Climbing Stairs       0.54      0.53      0.53      2469
          Lying       0.57      0.57      0.57      2464
        Sitting       0.64      0.70      0.67      2503
        Walking       0.46      0.43      0.44      2460

    avg / total       0.55      0.56      0.55      9896
```



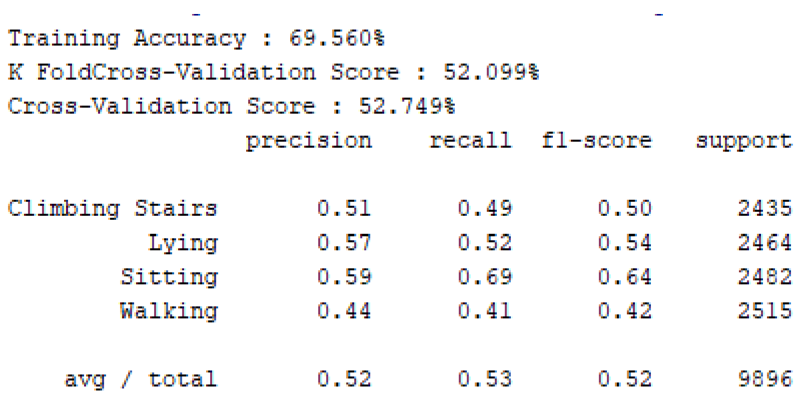
2.4.1.7. Multilayer Perceptron (MLP) Classifier



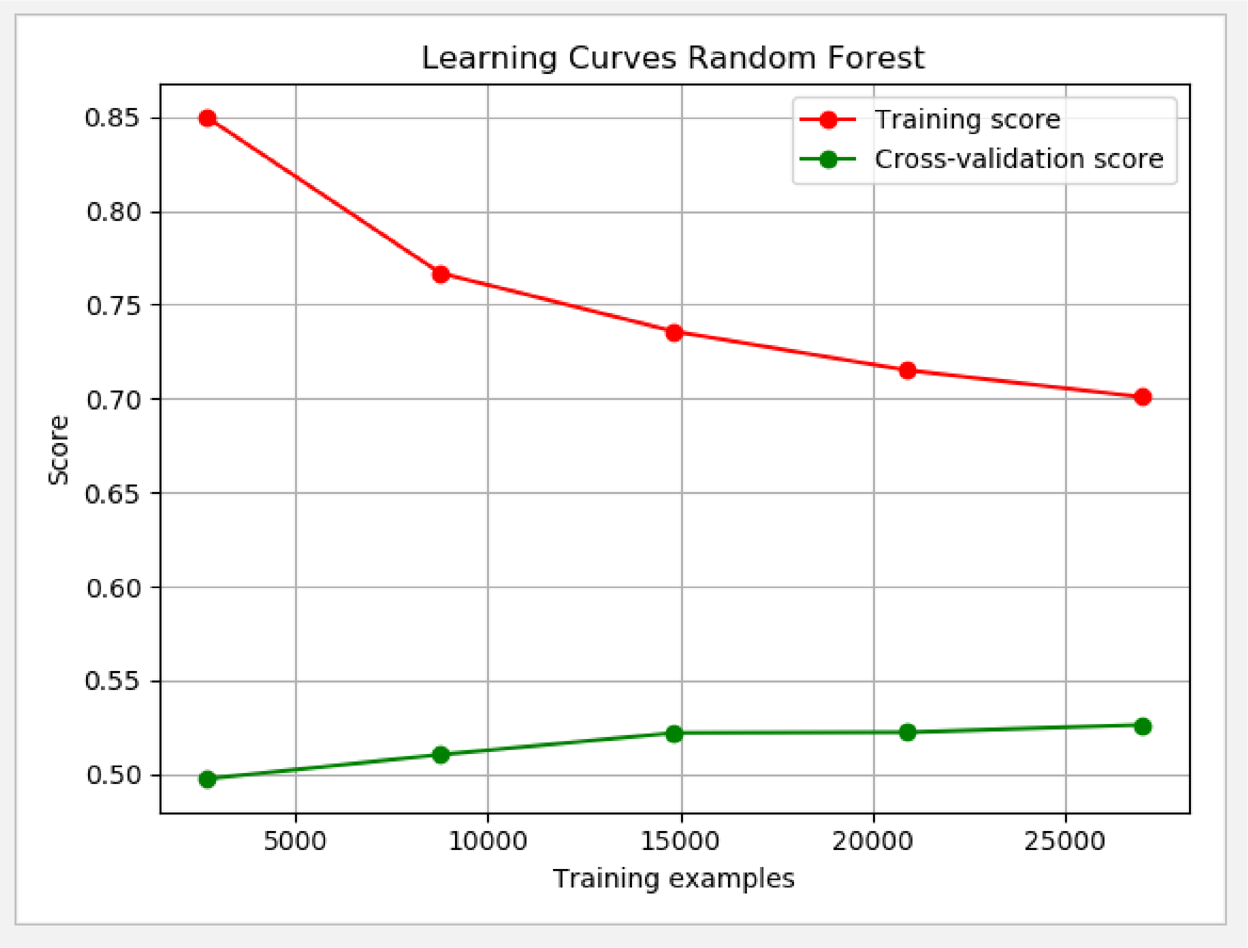
```
Training Accuracy : 45.653%
K FoldCross-Validation Score : 45.336%
Cross-Validation Score : 43.088%
                 precision    recall  f1-score   support

Climbing Stairs       0.36      0.84      0.51      2491
          Lying       0.61      0.38      0.47      2504
        Sitting       0.48      0.48      0.48      2478
        Walking       0.44      0.00      0.00      2423

    avg / total       0.47      0.43      0.37      9896
```



2.4.1.8. Random Forest Classifier



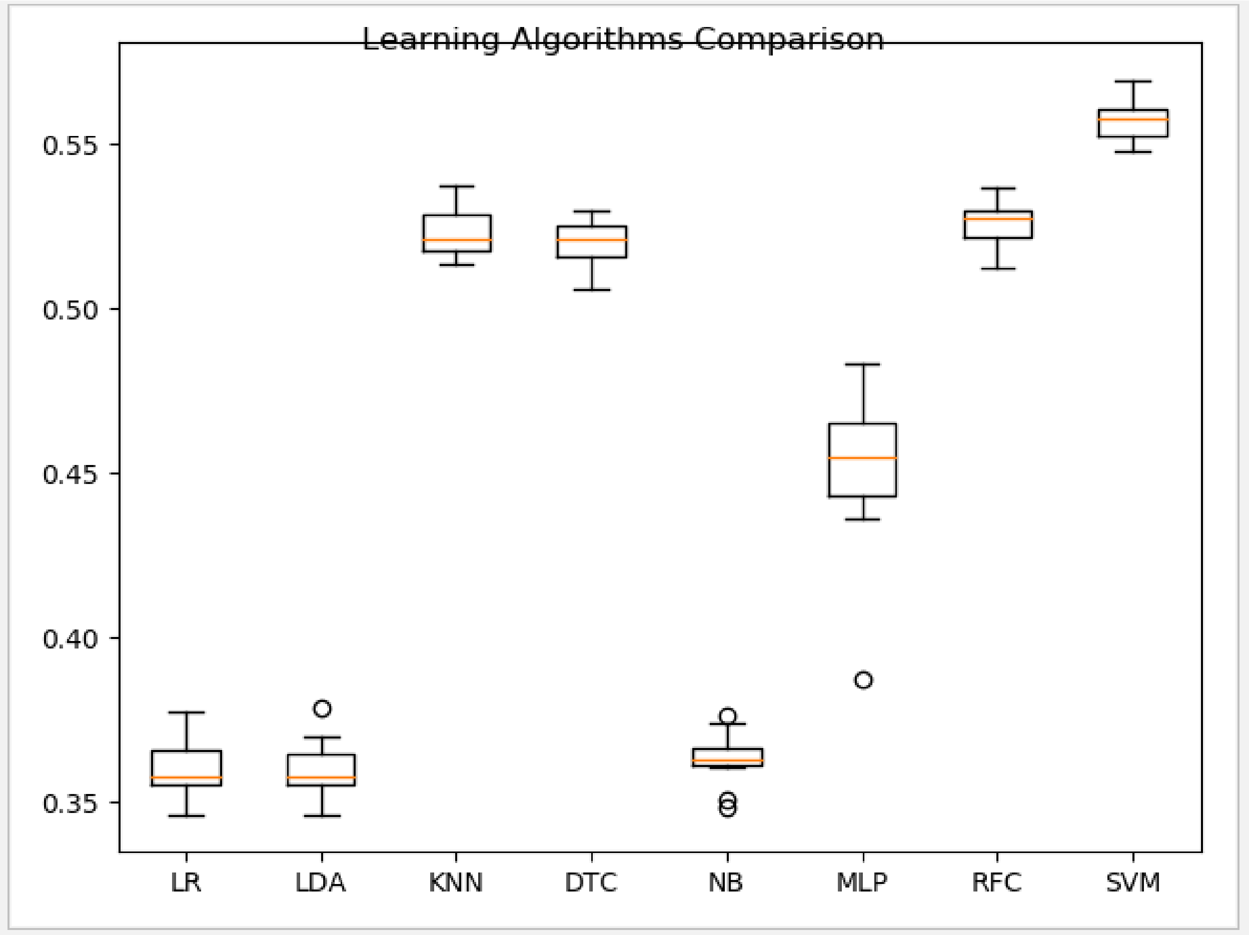
```
Training Accuracy : 69.560%
K FoldCross-Validation Score : 52.099%
Cross-Validation Score : 52.749%
                 precision    recall  f1-score   support

Climbing Stairs       0.51      0.49      0.50      2435
          Lying       0.57      0.52      0.54      2464
        Sitting       0.59      0.69      0.64      2482
        Walking       0.44      0.41      0.42      2515

    avg / total       0.52      0.53      0.52      9896
```



2.5. Algorithms/Models Comparison

After analyzing the model generated by each algorithm individually, the K-Fold cross validation scores of all the algorithms were also compared side by side in the form of a Box and Whisker Plot to have a better understanding of each algorithm's performance in comparison to other algorithms.

The file "algorithmsComparison.py" contains the code for this comparison. The resulted Box and Whisker Plot is as follows:

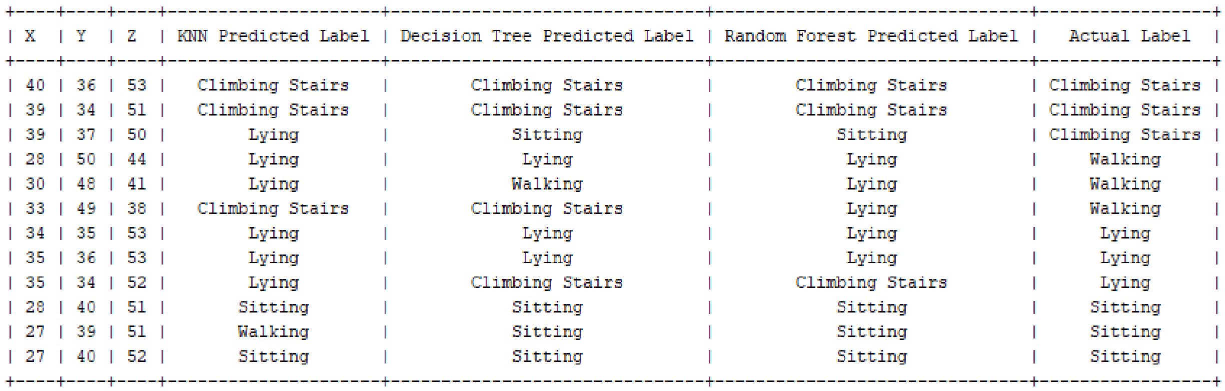


3. Predictions on Unseen Data

As the performance of K-Neighbors Classifier, Decision Tree Classifier and Random Forest Classifier were better as compared to other algorithms; therefore the models generated by these algorithms were used to make predictions on an unseen data set. SVM was not used for this purpose because it was taking a lot of time to generate model.

The file "predictor.py" contains the code for this purpose.

The result of prediction on unseen data set is as follows:



```
+----+----+----+---------------------+-------------------------------+-------------------------------+-----------------+
| X  | Y  | Z  | KNN Predicted Label | Decision Tree Predicted Label | Random Forest Predicted Label |   Actual Label  |
+----+----+----+---------------------+-------------------------------+-------------------------------+-----------------+
| 40 | 36 | 53 |   Climbing Stairs   |        Climbing Stairs        |        Climbing Stairs        | Climbing Stairs |
| 39 | 34 | 51 |   Climbing Stairs   |        Climbing Stairs        |        Climbing Stairs        | Climbing Stairs |
| 39 | 37 | 50 |        Lying        |            Sitting            |            Sitting            | Climbing Stairs |
| 28 | 50 | 44 |        Lying        |             Lying             |             Lying             |     Walking     |
| 30 | 48 | 41 |        Lying        |            Walking            |             Lying             |     Walking     |
| 33 | 49 | 38 |   Climbing Stairs   |        Climbing Stairs        |             Lying             |     Walking     |
| 34 | 35 | 53 |        Lying        |             Lying             |             Lying             |      Lying      |
| 35 | 36 | 53 |        Lying        |             Lying             |             Lying             |      Lying      |
| 35 | 34 | 52 |        Lying        |        Climbing Stairs        |        Climbing Stairs        |      Lying      |
| 28 | 40 | 51 |       Sitting       |            Sitting            |            Sitting            |     Sitting     |
| 27 | 39 | 51 |       Walking       |            Sitting            |            Sitting            |     Sitting     |
| 27 | 40 | 52 |       Sitting       |            Sitting            |            Sitting            |     Sitting     |
+----+----+----+---------------------+-------------------------------+-------------------------------+-----------------+
```

4. Conclusion

Different algorithms showed varying behavior on the data set. The accuracy of some models was better while the accuracy of some algorithms was not up to the mark. The best cross validation scores were given by SVM and Decision Tree Classifier with the cross validation score of approximately 56% and 52% respectively. This score is decent but it is not the best score which means there is a room of improvement. Better models can be generated by training the model with more examples.[i]

5. References

- Data Set Link: https://archive.ics.uci.edu/ml/datasets/Dataset+for+ADL+Recognition+with+Wrist-worn+Accelerometer
- www.re3data.org/repository/r3d100010960
- en.wikipedia.org/wiki/Scikit-learn