project_markdown

Siddhartha Sampath

6/12/2019

Executive Summary

The dataset chosen was the UCI Human Activity Recognition Dataset that attempts to classify six types of human activities based on various metrics collected by smartphones from 30 human subjects within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (a Samsung Galaxy S II) on the waist. The experiments were video recorded and labeled manually,

From the dataset description, we learn that sensor signals from smartphones were sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). Then, from each window, a vector of features was obtained by calculating variables from the time and frequency domain. More on the dataset can be found at
https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

THe data contains a training set that contains about 70% of the data and a test set that contains the remaining 30%. First Singular Value Decomposition and Principal Component Analysis was performed to analysze the variability within features and plotted with respect to the classes to see if there was enough natural separation that could be exploited or if new features needed to be engineered. Then four different algorithms were tested with five fold cross validation on the training data, and the best algorithm was chosen and applied on the never seen before test data to achieve a prediction accuracy of 93%.

Methodology

The dataset was divided into a `train_set` and a `test_set`. Each set was dividied into a `x` and a `y` dataframes for the features and the response. An extra subjectID column shows us which subject each line of data was recorded from. A quick look at the dimensions of the training set and the test set show us that there are 561 features in both sets and about 71% of the data is in the training set. The following diagrams show us the amount of data collected by activity and by test subject and how it was divided into the training and testing set.

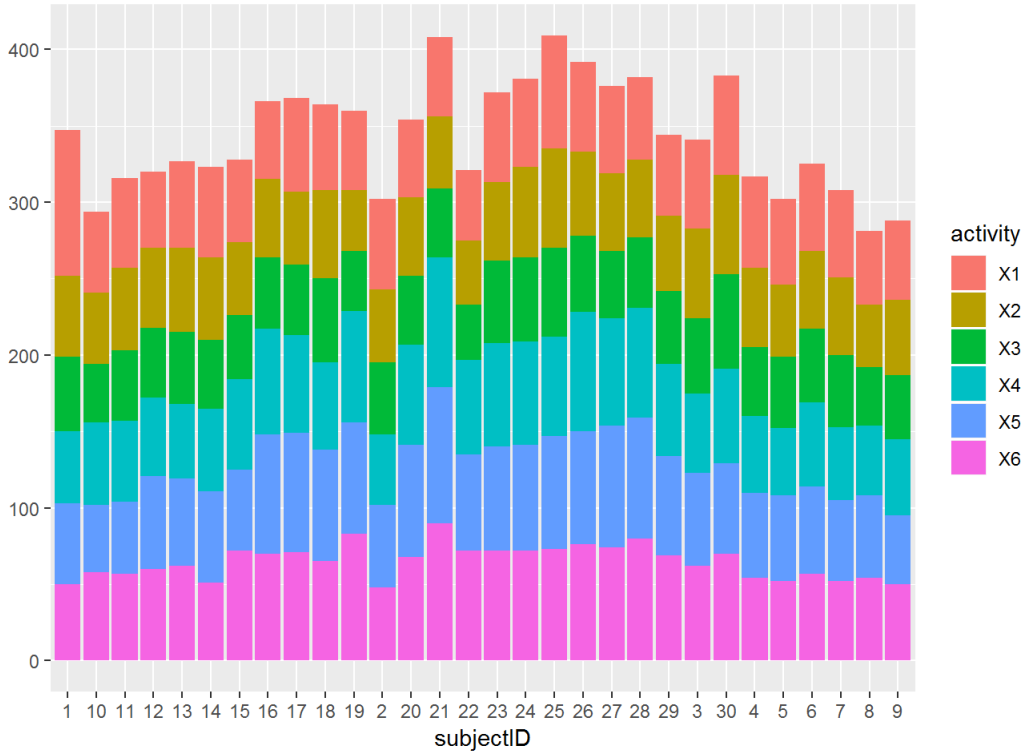
```
dim(train_x)
```

```
## [1] 7352  561
```

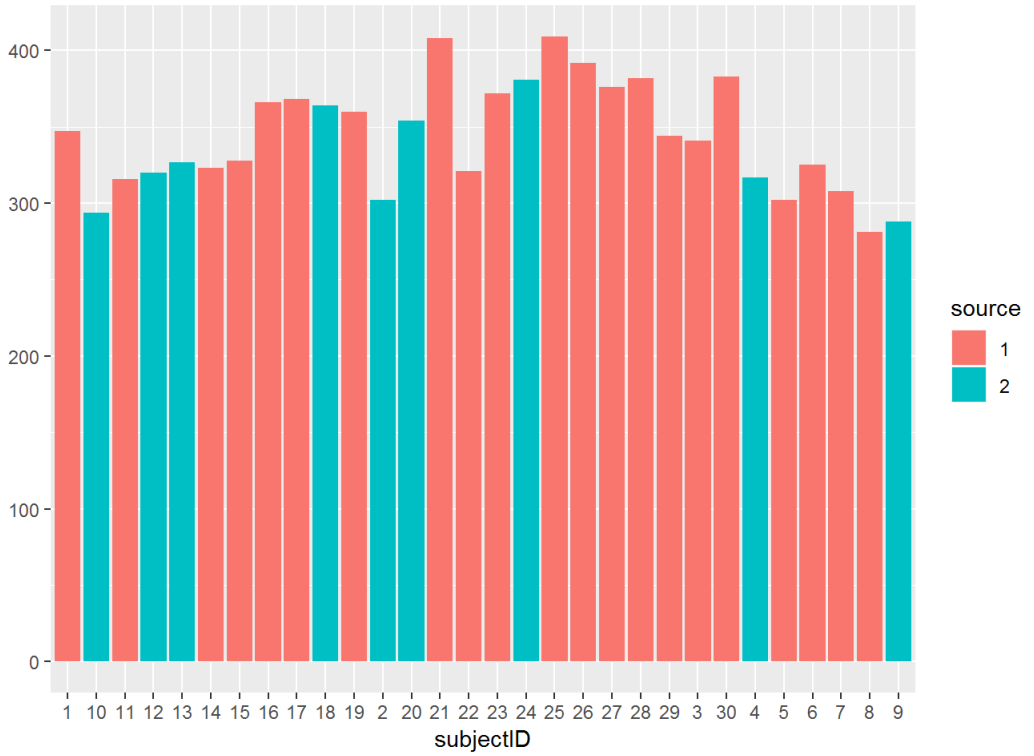
```
dim(test_x)
```

```
## [1] 2947  561
```

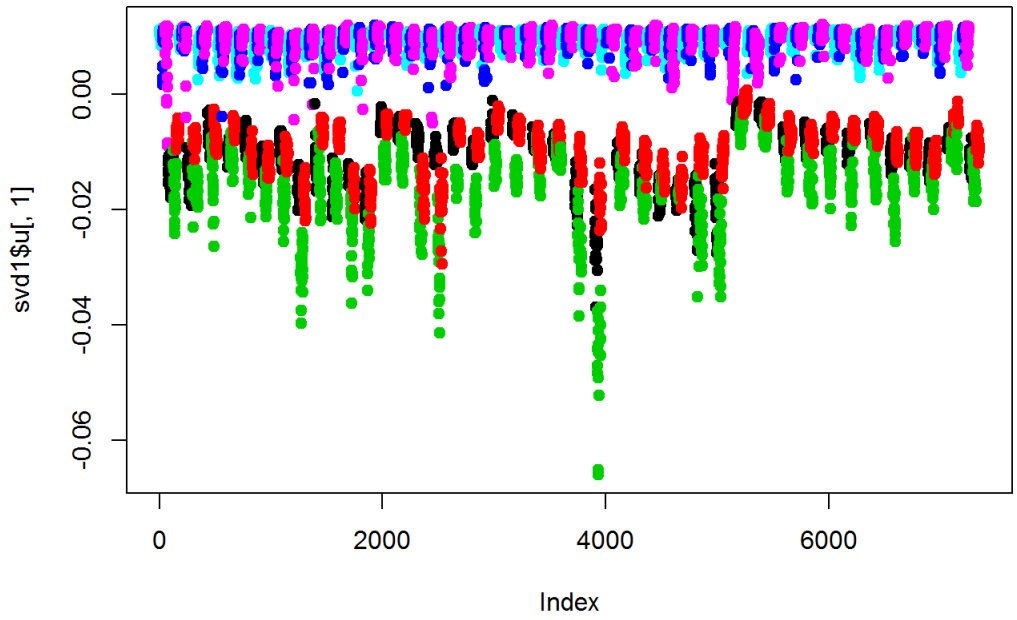
```
qplot(data = all_data, x = subjectID, fill = activity)
```

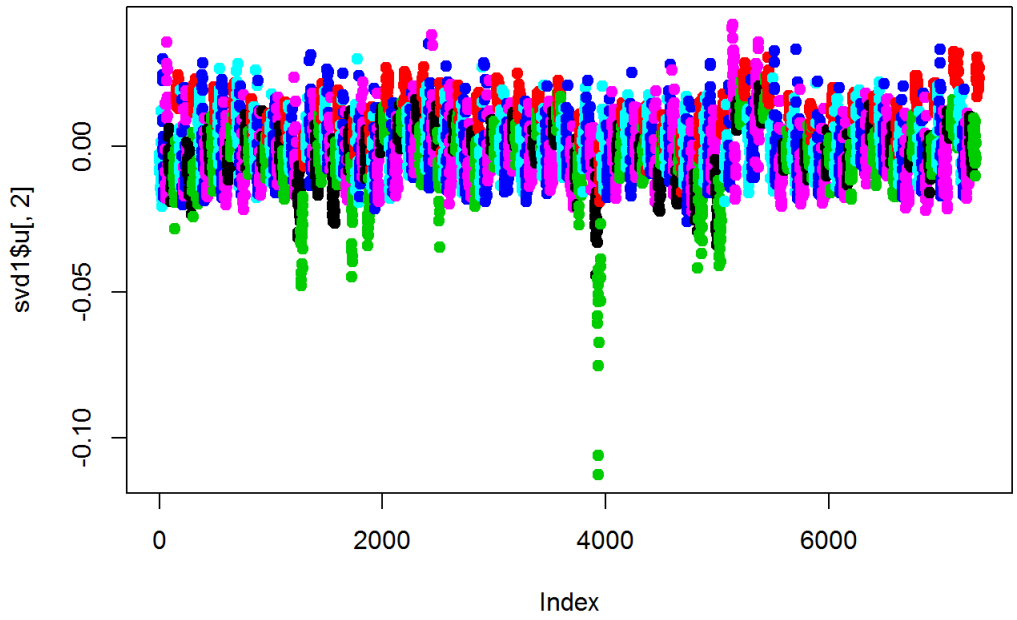



```
qplot(data = all_data, x = subjectID, fill = source)
```

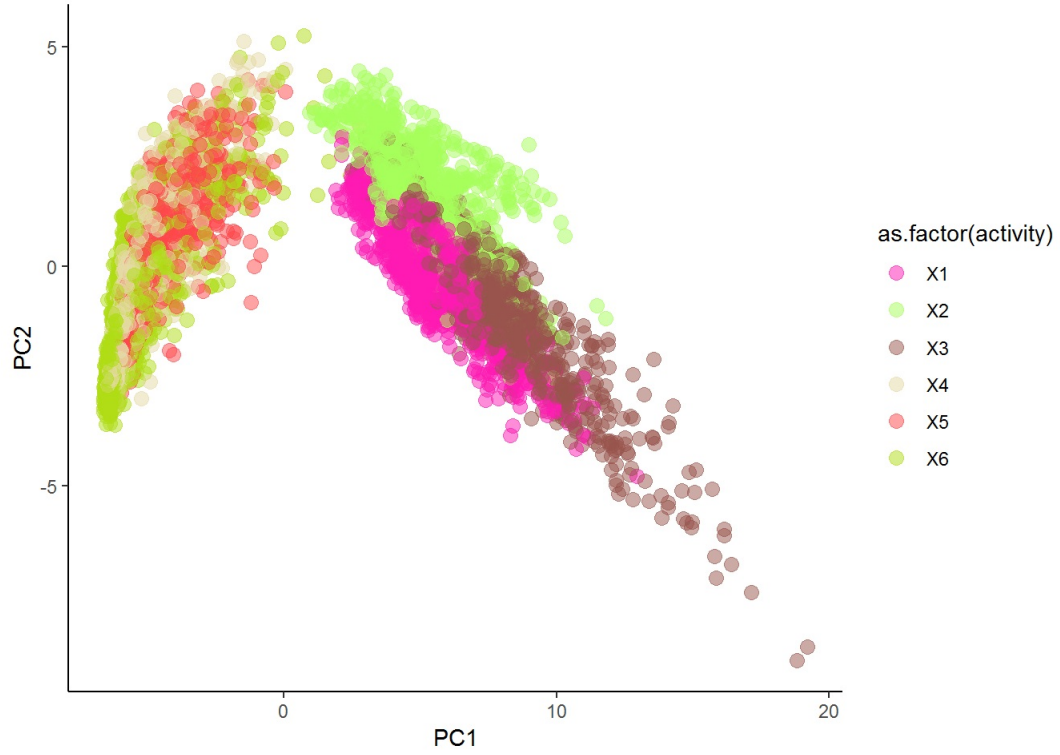


Next, the feature space was analyzed using SVD and PCA and plotted against the classes to see if there was enough separation within the features.



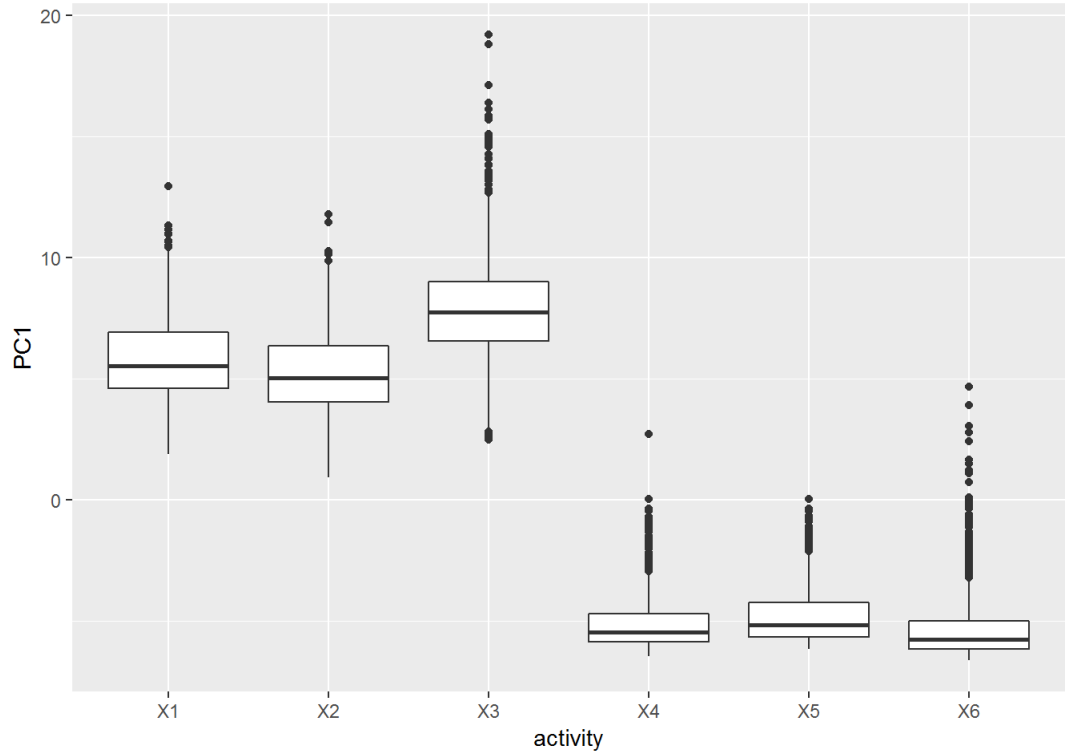


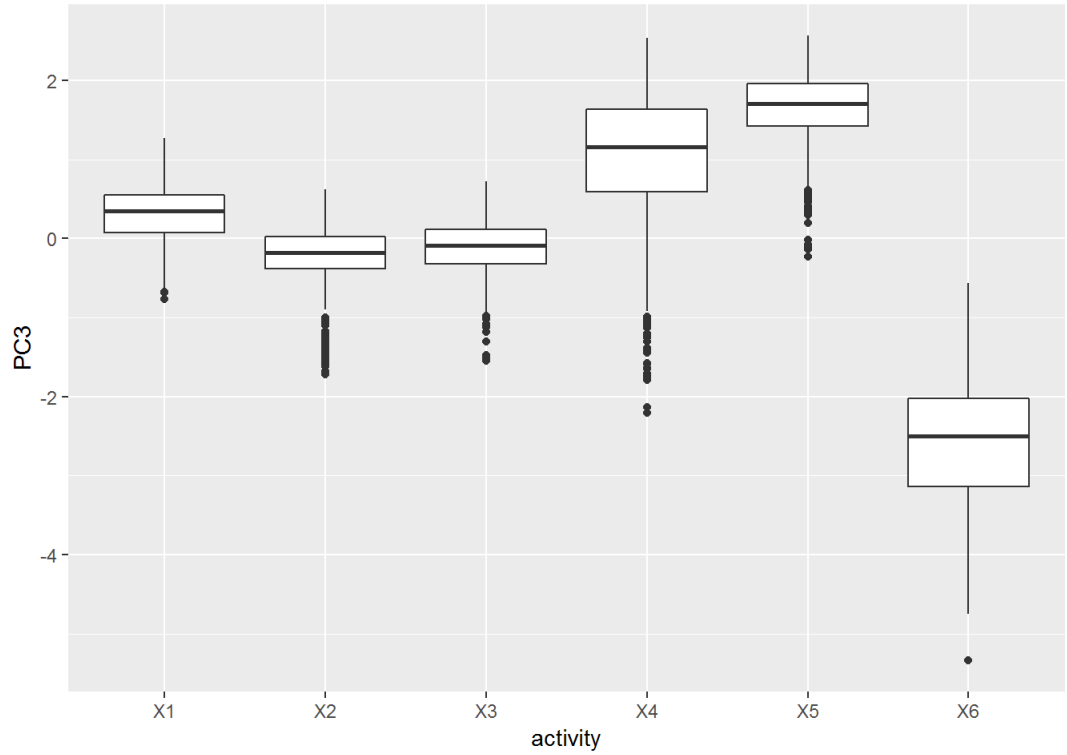
The two plots above show the first and second dimension plotted against the index for the training set obtained after an SVD analysis and color coded by activity. Clearly from the first dimension there is a clear separation between two sets of three activities. We calculate the the principal components and plot again to confirm.



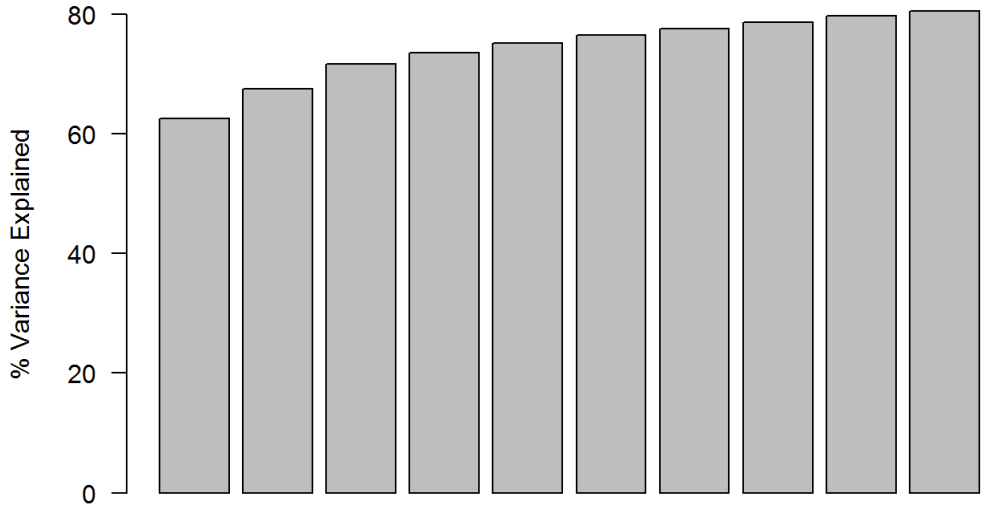
Here the activities X1, X2 and X3 relate to the more energetic activities of walking, walking upstairs, and walking downstairs and X4, X5 and X6 relate to the more sedentary activities of sitting, standing and laying. The plot above shows the observations plotted against the first two principal components and color coded by activity.

Boxplots of the first and third principal components by activity give us hope that ML algorithms should be able to achieve some high level of accuracy when classifying the observations into one of the six different activities.





The plots above show that for the first principal component, the energetic activities have a positive value and the sedentary activities have a negative value. The remaining components have to thus only help us separate the activities from within their sub groups of energetic and sedentary. In fact plotting the third principal component shows that a threshold value of PC3=0, we can separate walking from the activities of walking upstairs and downstairs, while a threshold value of PC3= -1, we can seperate the activity of laying from sitting and standing. So clearly there is enough variability within the features to exploit for classification, so we will not attempt to engineer any more features. Further we will go back to working with the original features as they are more explainable. The following plot shows us that that the first 10 principal components explain about 80% of the variance.



This leads us to the intuition that there are quite a few useful features. Since the feature size is not too large, we will not perform any feature extraction, but use all the features as is.

Four classification models were chosen and fitted to the data with their accuracy guaged via five fold cross validation with 75% of the data being used for training and 25% of the data being used for validation. The four classificatoin models chosen were Stabilized linear Discriminant Analsysis, Multi Layer Perceptron, Support Vector Machine with a linear kernel and gradient boosted trees implemented via the xgboost package. The best performing model was the XGBOOST model and this was trained on the full training set and used to predict the activity labels for the test set. The results are detailed in the next section.

Results

The tables below detail the confusion matrices and the accuracy of the four models after five fold cross validation performed via the caret package's `train` method. For the Multi-Layer-Perceptron, three to four hidden layers were explored. For the gradient boosted tree, between 50 to 70% of the columns were sampled, and tree depth was varied betweeen three and six.

```
table(prediction_slda, train_lab)
```

```
##                train_lab
## prediction_slda    0    1    2    3    4    5
##               0 1197   17   20    0    0    0
##               1   27 1046   29    1    0    0
##               2    2   10  937    0    0    0
##               3    0    0    0 1102  102    0
##               4    0    0    0  167 1272    0
##               5    0    0    0   16    0 1407
```

```
table(prediction_mlp, train_lab)
```

```
##               train_lab
## prediction_mlp    0    1    2    3    4    5
##              0 1180    5    0    0    0    0
##              1   46 1068    0    0    0    0
##              2    0    0  986    0    0    0
##              3    0    0    0 1279  297    0
##              4    0    0    0    7 1077    0
##              5    0    0    0    0    0 1407
```

```
table(prediction_svm, train_lab)
```

```
##               train_lab
## prediction_svm    0    1    2    3    4    5
##              0 1226    0    0    0    0    0
##              1    0 1073    0    0    0    0
##              2    0    0  986    0    0    0
##              3    0    0    0 1278    7    0
##              4    0    0    0    8 1367    0
##              5    0    0    0    0    0 1407
```

```
table(prediction_xg, train_lab)
```

```
##              train_lab
## prediction_xg    0    1    2    3    4    5
##            X1 1226    0    0    0    0    0
##            X2    0 1073    0    0    0    0
##            X3    0    0  986    0    0    0
##            X4    0    0    0 1286    1    0
##            X5    0    0    0    0 1373    0
##            X6    0    0    0    0    0 1407
```

```
print(acc)
```

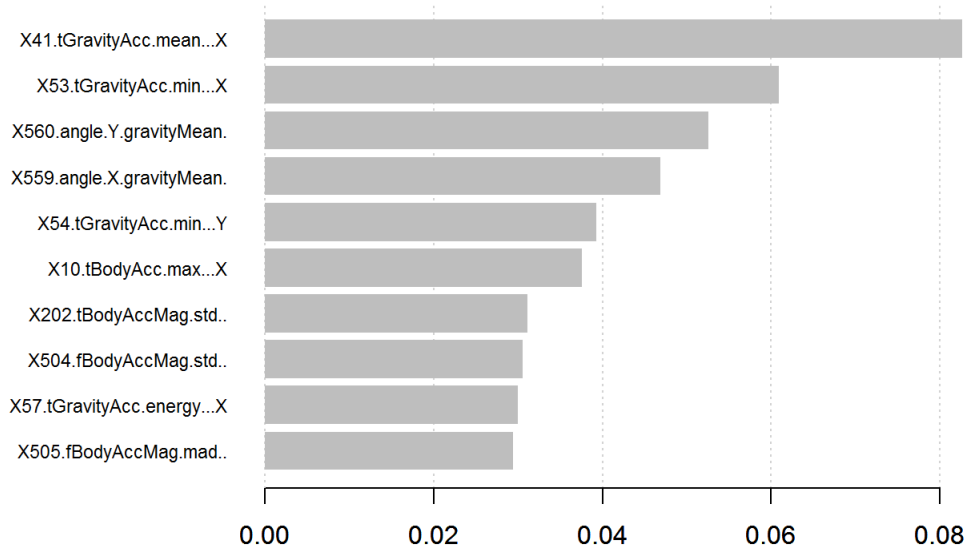
```
## # A tibble: 4 x 2
##   method                                 ACC
##   <chr>                                <dbl>
## 1 Stabilized linear Discriminant Analysis 0.947
## 2 Multi Layer Perceptron                  0.952
## 3 Support Vector Machine                  0.998
## 4 Gradient Boosted Trees                  1.000
```

All models do well, but the gradient boosted trees perform the best. The labels 0 to 5 correspond to the activities X1-X6 detailed earlier. The XGBOOST model was then trained on the whole training dataset and this trained XGBOOST model was used to predict activity labels for the test_set.

```
##       test_lab
## y_pred   0   1   2   3   4   5
##      0 488  39   5   0   0   0
##      1   4 428  34   2   0   0
##      2   4   4 381   0   0   0
##      3   0   0   0 427  36   0
##      4   0   0   0  62 496   0
##      5   0   0   0   0   0 537
```

```
##  Accuracy
## 0.9355277
```

As seen from the table, we achieve an accuracy of 93.5%. The major source of error seems to be coming from a few cases of misclassifying between sitting and standing, and also between walking upstairs and walking downstairs, though there are less instances misclassified between these two activities. The XGBOOST model also shows us the top ten most important features as shown below.



These windows would probalby need to be examined in more detail to help us further refine the model to better distinguish between the activities of sitting and standing.

Conclusion

The UCI Human Acitivy Resource dataset that tracks six different human activities of 30 human subjects over multiple time windows was divided into a training set and a testing set and analyzed using Principal Component Analysis and four different Machine Learning Models using cross validation on the training set. The best performing model was chosen and used to predict activities for the testing set with an accuracy of 93%. While this is a good result, the main source of inaccuracy seems to be stemming from the fact that most models some times misclassify standing as sitting and sitting as standing. Moreover, sometimes, they also seem to confuse the activities of walking upstairs and downstairs. The activities of simply walking or lying down are always clearly classified which seems intuitive.

The model has an accuracy of 93% on previously unseen data which is impressive, but can be improved. The XGBOOST package gives us the most important features required for classification and these would have to explored in more detail to see if further information can be gleaned from them either by improving their accuracy during measurement or capturing more realted information. Further the recommenderlab package can be used to investigate whether more feature engineering can be utilized to come up with better composite features for prediction.

END