Title: Predictive model for Human Activity Recognition

Introduction

This assignment grow out of an experiment [1] that have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed the six physical activities (*standing, walking, laying, walking, walking upstairs, walking downstairs*) wearing the smartphone Samsung Galaxy S2 on the waist. Using its embedded accelerometer and gyroscope, the authors captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz through a smartphone application based on the Google Android O.S., developed by themselves [1], for the acquisition of the sensor signals pre-processed and transformed for finding the signal frequency components. The process pipeline of the research is as shown in Fig.1 [1].

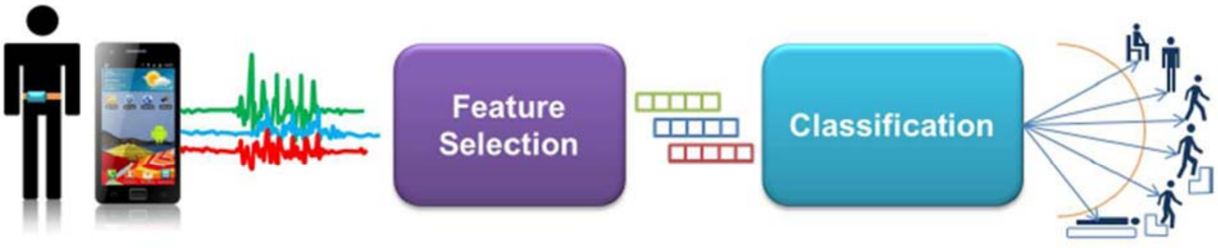


Fig. 1. Activity Recognition process pipeline.

The purpose of this assignment is to provide a predictive model able to identify the correct action (*activity*) performed by a person (*subject*) from the data associated with his movements as provided by the sensors of the smartphone.

Methods:

Data Collection

For our analysis we used the Samsung activity dataset available from the course website:
https://spark-public.s3.amazonaws.com/dataanalysis/samsungData.rda
These data are slightly processed to make them easier to load into R. Is possible also find the raw data here:
http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

All of the columns of the data set (except the last two: *subject*, *activity*) represents one measurement from the Samsung phone. The variable *subject* indicates which subject was performing the tasks when the measurements were taken. The variable activity tells what *activity* they were performing, that is : laying, setting, standing, walk, walk downstairs and walk upstairs.

The dataset consists of 561 relevant independent variables over 7,352 observations! So the first issue is to reduce the number of variables.
In this analysis I will not deal with the best strategy to choose the basket of variables with more variability.

The contest was designed in a Training and Testing Data Sets format to include the data:

- from subjects 1,3,5,6,7,8,11,14 for the Training Data with 2543 observations:

```
dataTrain <-with(samsungData, na.omit(samsungData[subject %in% c(1,3,5,6,7,8,11,14),]))
```

- from subjects 23,25,26,27,28,29,30 for the Testing Data with 2658 observations:

```
dataTest <-with(samsungData, na.omit(samsungData[subject %in% c(23,25,26,27,28,29,30),]))
```

Exploratory Analysis

The first approach was to rename some duplicate variable names through a function found on the Discussion Forums about the variable name issues [2].

Exploratory analysis was performed by examining tables and plots the observed data

dataTrain:

```
  laying      Sitting     standing       walk       Walkdown      walkup
   435          399          441          489          369          410
```

dataTest:

```
  laying      Sitting     standing       walk       Walkdown      walkup
   514          475          499          421          362          387
```

The Data sets result consistent.

Several plots was examined to individuate possible patterns. As suggested by the R-function of Mayer from Discussion Forums [3] I examined lots of each variable histograms and plots. Here are some examples for three variables:

Fig.2. tBodyAcc.mean.X

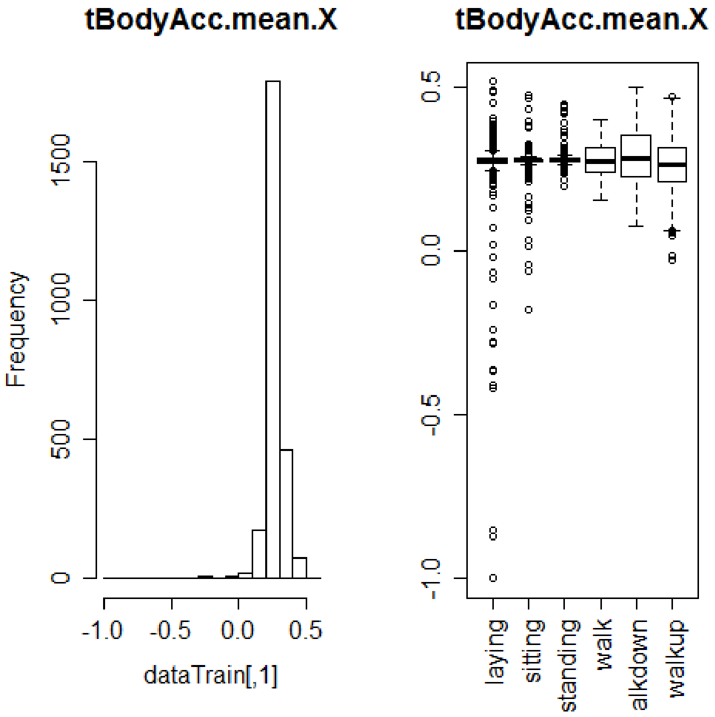
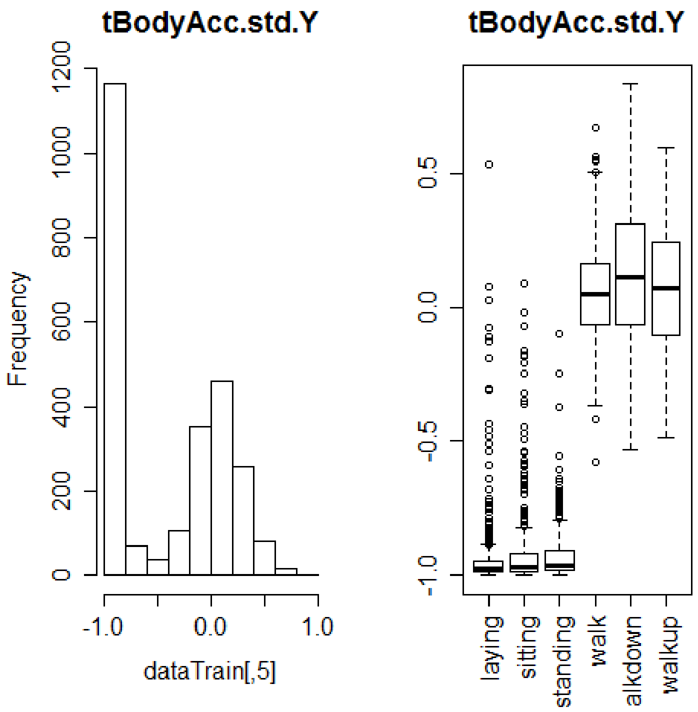


Fig.3. tBodyAcc.std.X



The second approach was to calculate standard deviation of the means of each variables group by activity:

```
> round(trainMeanDev[1:12], 5)
      fBodyAccJerk.entropy.X       fBodyAccJerk.entropy.Y           fBodyAcc.entropy.X 
                     0.80967                      0.78952                      0.76363 
     tBodyAccJerkMag.entropy  fBodyBodyAccJerkMag.entropy           fBodyAccMag.entropy 
                     0.75475                      0.71830                      0.70903 
        tGravityAcc.energy.X           fBodyAcc.entropy.Y      tBodyGyroJerkMag.entropy 
                     0.70218                      0.69270                      0.69018 
      tBodyAccJerk.entropy.X       fBodyAccJerk.entropy.Z fBodyBodyGyroJerkMag.entropy 
                     0.67452                      0.66187                      0.65928 
```

The higher the standard deviation the greater the variable could split the activities.

Statistical Modeling

Two different approach:

1. Predicting with Regression models (Linear Least Squared) by selecting randomly 100 statistic variables from the 561 available.

2. Predicting with decision trees by selecting randomly 100 statistic variables from the 561 available.

Results:

1. <u>Regression - Predicting with Regression models (Linear Least Squared):</u>

After transforming the *activity* factor as numeric, I run the Linear Least Squared model with *lm* function and then improved the model with a step wise regression (*step* function) .

Step-wise model using the Training set:

```
lm(formula = as.numeric(activity) ~ fBodyAccJerk.bandsEnergy.Y.17.32 +
    fBodyGyro.bandsEnergy.Z.17.24 + fBodyGyro.mean.Z + tBodyGyroJerkMag.mad +
    tBodyAccJerk.min.X + tBodyGyroJerk.iqr.Y + fBodyAcc.energy.Y +
    tBodyAcc.iqr.X + tBodyAccJerk.std.X + tBodyAcc.mean.X + tGravityAccMag.mean +
    tBodyAccMag.arCoeff.3 + fBodyBodyAccJerkMag.mean + tBodyGyroMag.energy +
    fBodyAccJerk.mean.Z + tBodyAccMag.entropy + fBodyAccJerk.bandsEnergy.Z.1.24 +
    tBodyAcc.std.X + tBodyAccJerk.iqr.X + fBodyGyro.skewness.Y +
    fBodyAcc.bandsEnergy.Y.49.56 + tGravityAcc.arCoeff.Y.1 +
    fBodyGyro.bandsEnergy.Y.57.64 + tBodyGyro.min.Z + tBodyGyroJerk.std.Z +
    tBodyAcc.arCoeff.Y.3 + tBodyAcc.entropy.X + fBodyGyro.bandsEnergy.Z.33.40 +
    fBodyBodyGyroMag.mad + tBodyGyroJerk.mean.X + tGravityAcc.mean.X +
    tGravityAccMag.energy +tBodyAccJerk.correlation.X.Z +BodyAccJerk.bandsEnergy.Y.9.16 +
    tBodyGyro.max.X + fBodyAccMag.entropy + fBodyAccJerk.kurtosis.X +
    fBodyGyro.bandsEnergy.Y.1.24 + fBodyAccMag.sma + fBodyAcc.meanFreq.Z +
    fBodyAcc.bandsEnergy.Z.57.64 + tBodyGyroMag.entropy + tBodyGyroJerk.entropy.X +
    tBodyGyroJerk.std.X + fBodyAcc.skewness.Z + fBodyAcc.bandsEnergy.Y.9.16 +
    tBodyGyroMag.iqr + fBodyAccJerk.bandsEnergy.Z.17.32 + fBodyGyro.bandsEnergy.Y.1.16 +
    tBodyAcc.std.Y + tBodyAcc.max.X + tGravityAcc.arCoeff.X.4 +
    tBodyGyro.arCoeff.X.1 + tBodyAccJerkMag.arCoeff.4 + fBodyGyro.min.Z +
    fBodyAcc.energy.X + fBodyAcc.bandsEnergy.Z.25.32 + tBodyGyroMag.mean +
    fBodyAcc.bandsEnergy.Z.33.40 + tGravityAcc.std.Y, data = dataTrain)
```

The results seem pretty good for the Training dataset. The model ends up using 60 variables and the statistical significance is very high.

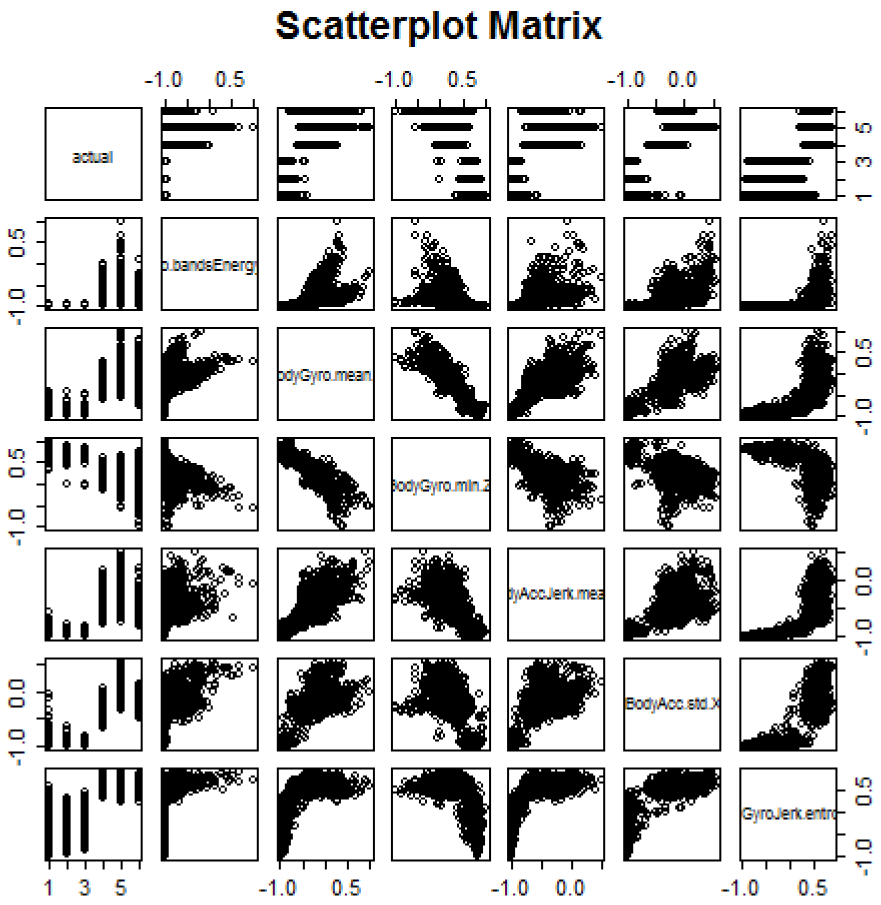
```
Residual standard error: 0.4543 on 2482 degrees of freedom
Multiple R-squared:  0.9292,     Adjusted R-squared:  0.9275
F-statistic: 543.1 on 60 and 2482 DF,  p-value: < 2.2e-16
```

```
> table(dataTrain$activity, predict1)
          predict1
             1   2   3   4   5   6
  laying   395  40   0   0   0   0
  sitting    0 257 140   2   0   0
  standing   0  97 344   0   0   0
  walk       0   0   6 364 119   0
  walkdown   0   0   0  38 297  34
  walkup     0   0   0   2 157 251
```

The root mean squared error (*rmse* function, package Metrics) is 0.5044045

The following Scatterplot Matrix (Fig. 4) shows the relationships between the actual values of *activity* variable and a set of 6 predictors (highly significant)

Fig.4. Scatterplot Matrix (dataTrain)



Finally applying the regression model to the Test data set the result results good as well.

```
> table(dataTest$activity, predict2)
          predict2
             1   2   3   4   5   6
  laying   492  22   0   0   0   0
  sitting    2 355 118   0   0   0
  standing   0  77 422   0   0   0
  walk       0   0  29 258 110  24
  walkdown   0   0   0  21 275  66
  walkup     0   0   0   4 188 195
```

And the root mean squared error (*rmse* function , package Metrics) results 0.5294204

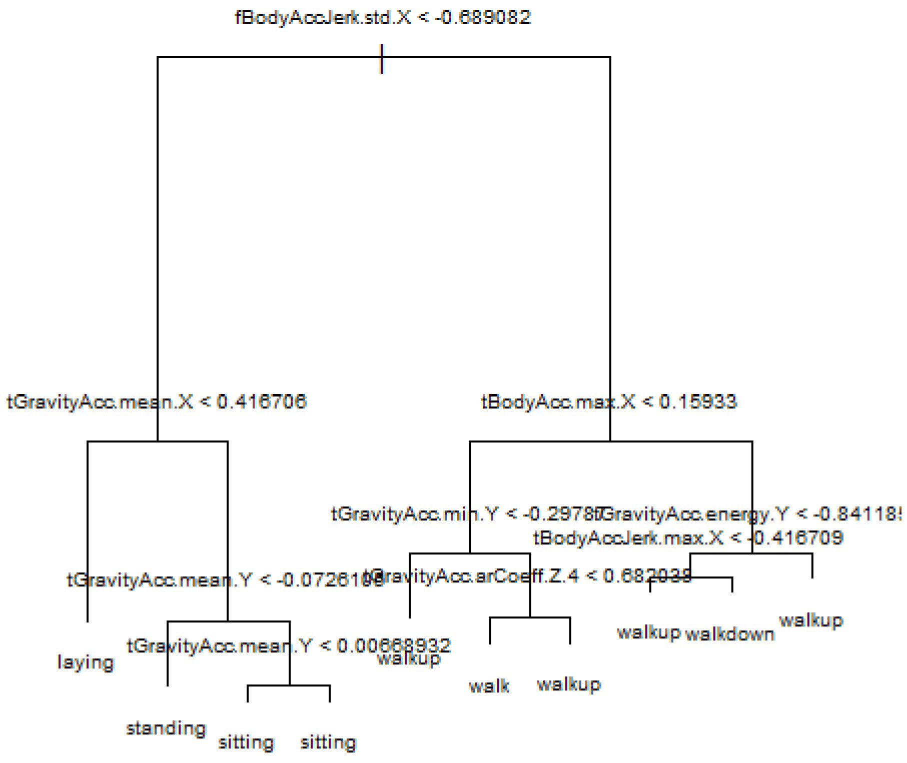
2. <u>Decision Tree - Predicting with Trees:</u>

The variable selection was based on the same regression strategy. Selecting randomly 100 variables the tree algorithm used only 8 of them:

```
Variables actually used in tree construction:
[1] "fBodyAccJerk.std.X" "tGravityAcc.mean.X" “tGravityAcc.mean.Y" "tBodyAcc.max.X"
[5] "tGravityAcc.min.Y"  "tGravityAcc.arCoeff.Z.4" "tGravityAcc.energy.Y"
[8] "tBodyAccJerk.max.X"
Number of terminal nodes:  10
Residual mean deviance:  0.3903 = 988.7 / 2533
Misclassification error rate: 0.06606 = 168 / 2543
```

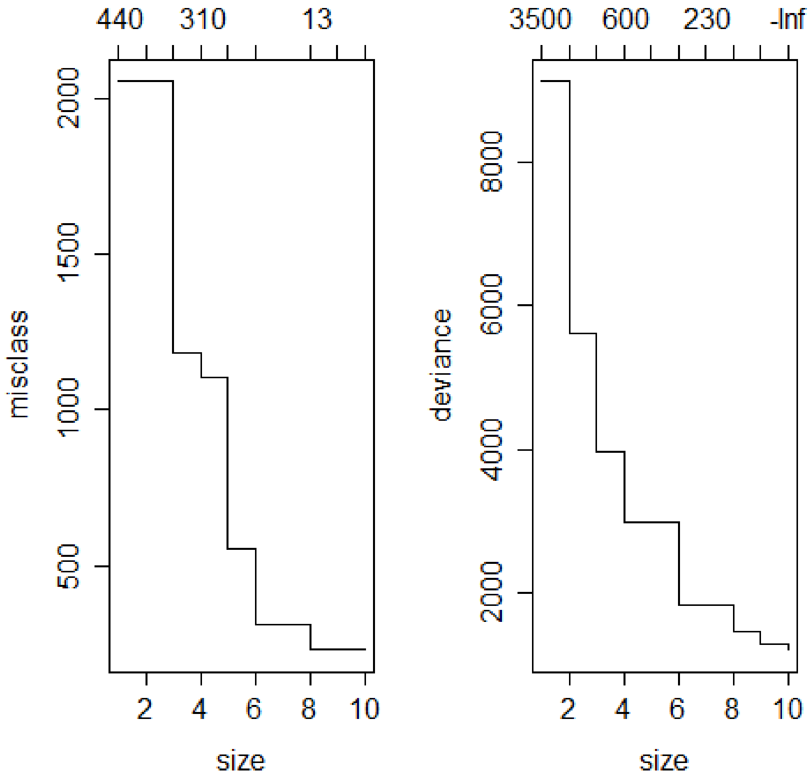
The initial decision tree (Fig. 5):

Fig.5. Plot Tree (dataTrain)


fBodyAccJerk.std.X < -0.689082
tGravityAcc.mean.X < 0.416706
tBodyAcc.max.X < 0.15933
tGravityAcc.min.Y < -0.29787
tGravityAcc.energy.Y < -0.84118!
tBodyAccJerk.max.X < -0.416709
tGravityAcc.mean.Y < -0.0726108
tGravityAcc.arCoeff.Z.4 < 0.682038
laying
tGravityAcc.mean.Y < 0.00668932
walkup
walk
walkup
walkup
walkdown
walkup
standing
sitting
sitting

This tree was improved analyzing the mis-classification errors and the deviance related to the number of leaves (Fig. 6)

Fig.6. Plot Errors (dataTrain)

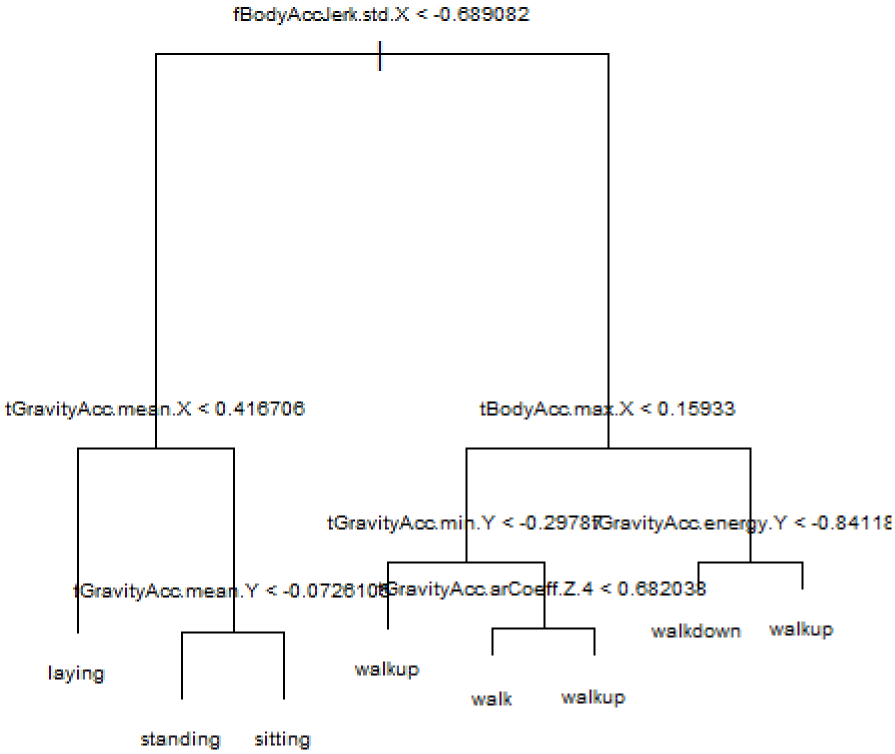


The initial tree was pruned down to 8 leaves providing this result:

```
Variables actually used in tree construction:
[1] "fBodyAccJerk.std.X" "tGravityAcc.mean.X" "tGravityAcc.mean.Y"
[4] “tBodyAcc.max.X" "tGravityAcc.min.Y"  "tGravityAcc.arCoeff.Z.4"
[7] "tGravityAcc.energy.Y"
Number of terminal nodes:  8
Residual mean deviance:  0.4966 = 1259 / 2535
Misclassification error rate: 0.07118 = 181 / 2543
```

The final tree (Fig. 7) is the following

Fig.7. Prune the Tree (dataTrain)


fBodyAccJerk.std.X < -0.689082
tGravityAcc.mean.X < 0.416706
tBodyAcc.max.X < 0.15933
tGravityAcc.min.Y < -0.29781
tGravityAcc.energy.Y < -0.8411
tGravityAcc.mean.Y < -0.0726108
tGravityAcc.arCoeff.Z.4 < 0.682038
laying
standing
sitting
walkup
walk
walkup
walkdown
walkup

```
           predictTrain
actualTrain laying sitting standing walk walkdown walkup
   laying      435       0        0    0        0      0
   sitting       0     369       30    0        0      0
   standing      0      60      381    0        0      0
   walk          0       0        0  444       20     25
   walkdown      0       0        0    8      359      2
   walkup        0       0        0   17       19    374
```

The root mean squared error (*rmse* function , package Metrics) results 0.347453

Finally applying thi model to the Test Dataset the result results good as well.

```
          predictTest
actualTest laying sitting standing walk walkdown walkup
  laying      514       0        0    0        0      0
  sitting      17     419       39    0        0      0
  standing      0      74      425    0        0      0
  walk          0       0        8  371       22     20
  walkdown      0       0        0   49      302     11
  walkup        0       0       36   39       79    233
```

And the root mean squared error (*rmse* function , package Metrics) results 0.5684851

Conclusions:

Analysing the different approaches using a very large set of variables the best option seems to be Decision Tree modelling.
The analysis could be improved further using variable reductions/transformations and a cross validation.

References:

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. *Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine*. International Workshop of Ambient Assisted Living (IWAAL 2012). Vitoria-Gasteiz, Spain. Dec 2012.

[2] by Uwe F Mayer forum post: https://class.coursera.org/dataanalysis-002/forum/thread?thread_id=1237

[3] by Uwe F Mayer forum post: https://class.coursera.org/dataanalysis-002/forum/thread?thread_id=1198