Introduction

The purpose of this assignment is to provide a predictive model able to identify the correct action performed by a person from the data associated with his movements as provided by the sensors of a smart phone[1].

The data have been collected and presented in a paper that has been written on the subject[2]. In the same paper the logic and the conventions linked to the variable names are extensively explained.

The vast majority of the method and the work flow used has heavily borrowed from the three Coursera[3] courses the author attended this year: *Introduction to data science* (prof. Bill Howe – University of Washington), *Computing for data analysis* (prof. Roger Peng – Johns Hopkins Bloomberg school of public health) and *Data analyisis* (prof. Jeff Leek – Johns Hopkins Bloomberg school of public health).

All mistakes are obviously mine.

Methods

Data collection and variables management

Data has been provided as an .rda file as part of the assignment rubric.

After loading the data from the .rda file I realized that few variable names are duplicated. The most likely cause is the missing axis for some of the measures. Using a function posted in the forum[4] this issue was sorted. The other change being performed was to transform the activity variable in a factor variable.

On top of this I standardized all the 561 potential independent variables using the *scale()* function: all variables become unit-less this way. Any coefficient from statistical methods will represent *betas*: sensitivity of the dependent variable to percentage changes in the independent ones.

I didn't change any variable names; they are fairly long but the result is that it's clear what each measure is about.

The issue is fairly clear: how best to deal with a set of 561 (I exclude the subject one) relevant independent variables over 7352 observations?

Training set/Test set

The prompt for the assignment also specify to large extent how to split the dataset between a training set and a test set.

> Your task is to build a function that predicts what activity a subject is performing based on the quantitative measurements from the Samsung phone. For this analysis your training set must include the data from subjects 1, 3, 5, and 6. But you may use more subjects data to train if you wish. Your test set is the data from subjects 27, 28, 29, and 30, but you may use more data to test. Be careful that your training/test sets do not overlap.

I preferred to have only a training set and a test set. I decided not to use a validation set.

In order to create the two sets I followed the instructions and did the following:

```
fixTrain <- c(1, 3, 5, 6)
fixTest <- c(27, 28, 29, 30)

otherSubj <- unique(samsungData[! samsungData$subject %in% c(fixTrain, fixTest), "subject"])
set.seed(1234)
index <- 1:length(otherSubj)
trainindex <- sample(index, ceiling(length(index)/2))

subjTrain <- c(fixTrain, otherSubj[trainindex])
subjTest <- c(fixTest, otherSubj[-trainindex])
samTrain <- na.omit(samsungData[samsungData$subject %in% subjTrain, ])
samTest <- na.omit(samsungData[samsungData$subject %in% subjTest, ])
```

samTrain (3779 observations) and samTest (3573) frames are the two sets of data I worked with.

Exploratory analysis

A quick run on the sample is done in order to assess if the there is anything wrong with the activity variable (the object of the modelling):

```
  laying  sitting standing     walk walkdown   walkup     <NA>
     696      631      705      668      522      557        0
```

The sample appears balanced.

The first option is to chart all the variables and have a quick look as suggested by someone[5]. In my opinion going through 561 sets of charts is hardly something exciting: how can you visually compare all the charts (an example is Figure 1) and determine which one to pick?

What can be derived from those charts is that you can basically split the variables set between those that are good to identify *laying, sitting and standing* and those that are good for *walk, walkdown, walkup.*

In my opinion a better guidance is provided by statistics performed on each of them. The relevant strategy employed is presented in the *Statistical Modelling* section.

A table for the mean for each variable, for each activity can be computed and delivers interesting information:

a) This is an example for the first few of them

```
                      laying     sitting    standing        walk    walkdown       walkup
tBodyAcc.mean.X   0.26285168  0.27368204  0.27988742  0.27642252  0.28687697  0.258900900
tBodyAcc.mean.Y  -0.01963588 -0.01140817 -0.01549443 -0.01821543 -0.01657714 -0.027137098
tBodyAcc.mean.Z  -0.10765206 -0.10627132 -0.10694535 -0.11115261 -0.10722971 -0.121808766
tBodyAcc.std.X   -0.95260705 -0.98233925 -0.98494183 -0.31959341  0.12359577 -0.242019425
tBodyAcc.std.Y   -0.91656801 -0.92958808 -0.93913215  0.01151121  0.06322413  0.004372214
tBodyAcc.std.Z   -0.92457633 -0.92886694 -0.93868548 -0.22980581 -0.15316738 -0.174602885
```

b) 21 instances of the variables appear to be duplicated. This gives you the opportunity to get rid of 21 variables (using them would lead to singularity errors).

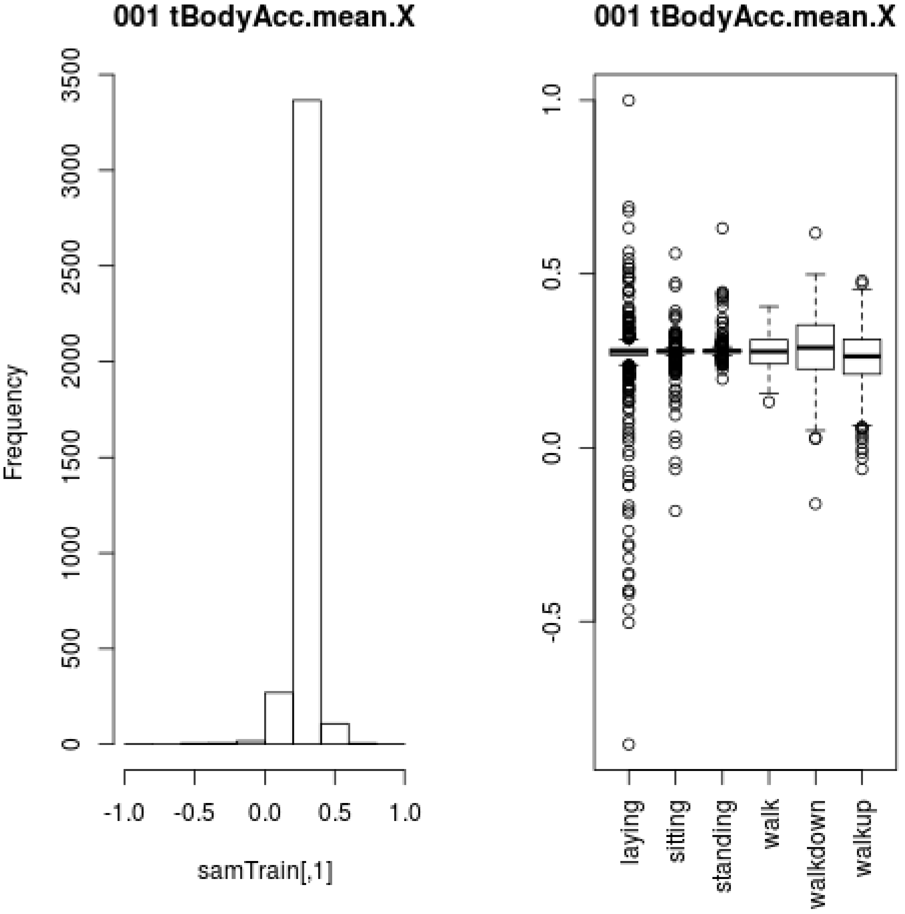

Figure 1: charting variable characteristics

c) you can calculate the standard deviation of the value of each row of this variable statistics. The higher that standard deviation the more likely the variable might split the activities.

```
  fBodyAccJerk.entropy.X        fBodyAccJerk.entropy.Y
               0.8121700                     0.7879797
 tBodyAccJerkMag.entropy            fBodyAcc.entropy.X
               0.7722359                     0.7706269
fBodyBodyAccJerkMag.entropy          fBodyAccMag.entropy
               0.7208513                     0.7152758
```

d) you can calculate the same statistic for group of activities: this can be useful in order to provide some hint on what is relevant when we do some refinement on features selection.

Strategy

In order to have a benchmark model I randomly picked 100 variables from the 561 available, transformed the activity factor into numerical, run OLS (the *lm* function) and then finalize the model with a step-wise regression (the *step* function). Rounding the predicted values (and casting activity < 0 into 1 and activity > 6 into 6) I treated them back as activity factors.

The results are pretty good: using the training set, the model ends up using 71 variables and the statistical significance is very high.

```
Residual standard error: 0.4199 on 3707 degrees of freedom
Multiple R-squared:  0.9389,	Adjusted R-squared:  0.9377 
F-statistic:   802 on 71 and 3707 DF,  p-value: < 2.2e-16
```

In terms classifying activities, this is done correctly 79% of the time.

```
      predicted
actual   1   2   3   4   5   6
     1 616  78   2   0   0   0
     2  22 455 153   1   0   0
     3   0 115 577  13   0   0
     4   0   0   7 549 112   0
     5   0   0   1  39 433  49
     6   0   0   0   0 199 358
```

The factor numbers correspond to the activities factors as per table below

```
   1      2       3       4      5       6 
laying sitting standing walk walkdown walkup
```

Not a poor outcome if you consider there isn’t any thinking in terms of feature selection!

Statistical modelling

Two options have been pursued:

a) **refining the OLS model** by selecting variables in some smart way: this was done by picking variables that provides the biggest diversification among their mean across the six activities. I ranked the variables (in descending order) based on the standard deviation of their mean value for each activity.
More variable have been added based on those that provides the biggest diversification among static activities (*laying, sitting, standing*) and those providing the biggest diversification among dynamic activities (*walk, walkup, walkdown*).

b) **using a decision Tree model** applying the same criteria for feature selection specified above. An initial tree is built with a basic set of variables and then analysed and pruned to deliver the final model.

Results

Regression

The OLS model ended up providing good results but not better than the benchmark one: the feature selection started with 45 variables only and ended up with 27. There are clear advantages from a lower number of variables being used (e.g. lower chance of overfitting, better potential usefulness of the model itself) but while the statistical fit is not bad at all, the predictive power of the model is not good enough: misclassification is high.

```
Residual standard error: 0.5293 on 3749 degrees of freedom
Multiple R-squared:  0.9018,   Adjusted R-squared:  0.901
F-statistic:  1187 on 29 and 3749 DF,  p-value: < 2.2e-16
```

Only 68% of the activities are correctly identified by this regression model on the training set:

```
      predicted
actual   1   2   3   4   5   6
     1 620  75   1   0   0   0
     2  13 493 123   2   0   0
     3   0 125 543  34   3   0
     4   0   0   4 401 262   1
     5   0   0   2  77 325 118
     6   0   0   0  12 363 182
```

Tree

The tree model provided the best results. The variable selection has been based on the same strategy but I could start with a lower initial set (25 variables); the initial tree uses only eight of them:

```
Variables actually used in tree construction:
[1] "fBodyAccJerk.bandsEnergy.X.1.8" "angle.X.gravityMean"
[3] "tGravityAcc.mean.Y"             "tGravityAcc.energy.Y"
[5] "fBodyAccMag.energy"             "tGravityAcc.min.X"
[7] "fBodyAcc.bandsEnergy.X.1.8"     "tBodyAccJerk.sma"

Number of terminal nodes:  11
Residual mean deviance:  0.5746 = 2165 / 3768
Misclassification error rate: 0.1148 = 434 / 3779
```

This tree can be improved analysing the misclassification and the deviance associated with the number of leaves (Figure 2):

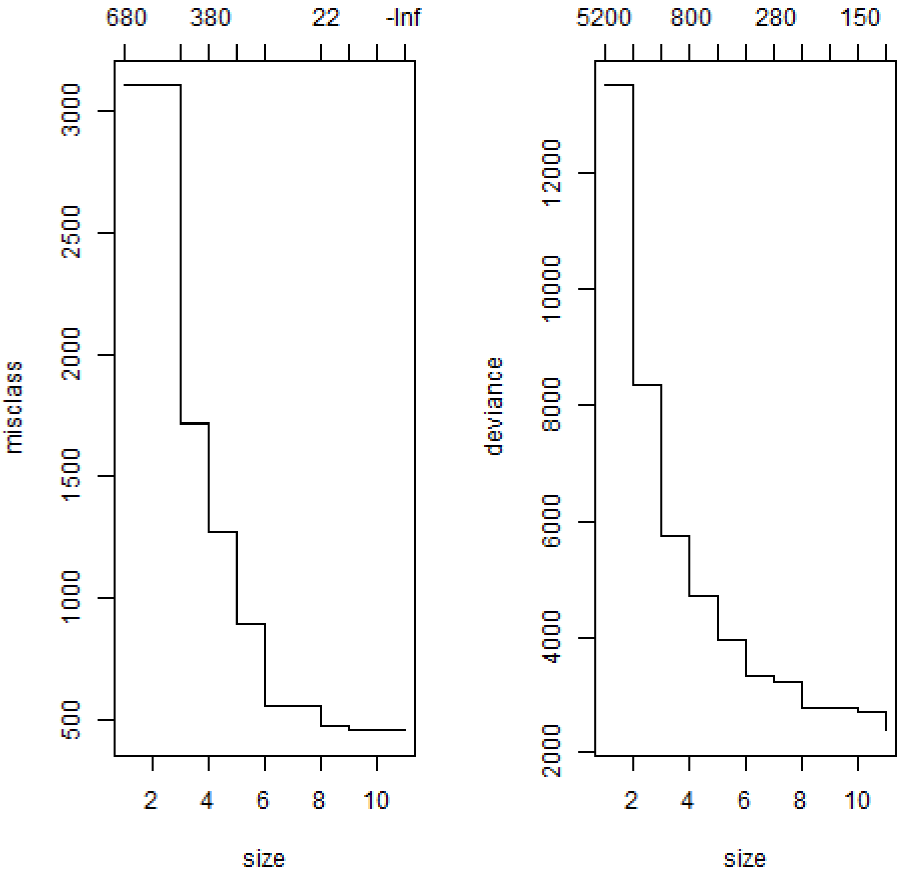

Figure 2: pruning the tree

Pruning the tree down to 8 leaves (from the initial 11) provides the following results.

```
Variables actually used in tree construction:
[1] "fBodyAccJerk.bandsEnergy.X.1.8" "angle.X.gravityMean"
[3] "tGravityAcc.mean.Y"             "fBodyAccMag.energy"
[5] "tGravityAcc.min.X"              "fBodyAcc.bandsEnergy.X.1.8"
[7] "tBodyAccJerk.sma"
Number of terminal nodes:  8
Residual mean deviance:  0.698 = 2632 / 3771
Misclassification error rate: 0.1209 = 457 / 3779
```

Effectively about 88% of the activities are correctly identified by this model.

The final tree is as follows (Figure 3):

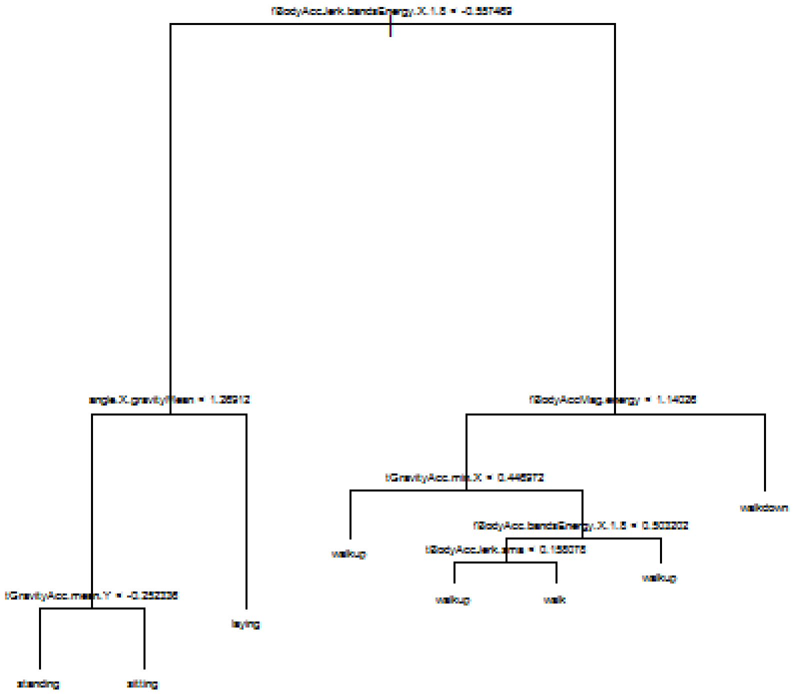
fBodyAccJerk.bandsEnergy.X.1.8 < -0.557489
angle.X.gravityMean < 1.26912
fBodyAccMag.energy < 1.14026
tGravityAcc.min.X < 0.446972
walkdown
fBodyAcc.bandsEnergy.X.1.8 < 0.503202
tBodyAccJerk.sma < 0.158075
walkup
walkup
walkup
walk
tGravityAcc.mean.Y < -0.252236
laying
standing
sitting

Figure 3: final tree

```
          predicted
actual     laying sitting standing walk walkdown walkup
  laying      695       0        0    0        1      0
  sitting       0     573       57    0        0      1
  standing      0     127      578    0        0      0
  walk          0       0        0  564       18     86
  walkdown      0       0        0   16      402    104
  walkup        0       0        0   17       30    510
```

Finally (as pudding proof), applying this to the test set deliver pleasing results as well: 87% observations identified correctly: this is both pleasing in terms of absolute results and also for how close this is to the results obtained using the training set.

```
          predicted
actual     laying sitting standing walk walkdown walkup
  laying      709       1        0    0        1      0
  sitting       0     599       56    0        0      0
  standing      0      58      611    0        0      0
  walk          0       0        0  340        1    217
  walkdown      0       0        0    4      415     45
  walkup        0       0        0   39       59    418
```

Conclusions

For this task where a factor variable needs to be predicted using a large set of explanatory variables the best option seems to be modelling with trees.

Regression deliver decent outcome but the price is to use an unreasonable amount of variables. This might lead to overfitting and, crucially, low usefulness of the model itself: with so many variables you can't easily provide the intuition and the sense check behind the model.

The tree model might be improved further by exploring the main weakness of the model which has hard time differentiating between *walk* and *walkup*. It is disturbing that this happens for the test set and not for the training one. Analysis based on S*ubjects* could have been performed and this might have suggested other choices in terms of feature selection.

The analysis could have benefitted from cross-validation and a lot more work on transforming variables, but time availability was definitely not on my side!

[1] A Samsung Galaxy SII has been used: this is actually a fairly old phone (released on April 2011) by now. Most recent models (we are close to the S5 some time early next year) should provide an even better quality for those measures.

[2] 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. Or here: http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

[3] http://www.coursera.com

[4] Uwe F Mayer post on thread https://class.coursera.org/dataanalysis-002/forum/thread?thread_id=1237 "How to deal with duplicate column names in samsungData"

[5] Again Uwe F Mayer post on thread https://class.coursera.org/dataanalysis-002/forum/thread?thread_id=1198 "Assignment 2: some pointers on getting started" is providing with an excellent procedure to get all the charts saved for us.