Exercise 4 - Decision Trees and Random Forests

For this exercise you can decide on what learning problem you want to investigate (how many samples, with or without preprocessing, smoothing, 'all persons in' and/or 'disjunct'). Remember you can explore small datasets and then validate on the complete dataset.

- Literature

- Chapter 5 (Divide and Conquer - Classification using Decision Trees and Rules) and chapter 11 (Improving Model Performance) of "Machine Learning with R" (First Edition) by Brett Lantz, Packt Publishing Ltd., second edition, 2015.
- Chapter 8 (Tree basedmethods) of "An Introduction to Statistical Learning with Applications in R" from G. James, D. Witten, D., T. Hastie, R. Tibshirani. Springer 2013.

4.1 - Decision Trees:

4.1.1 - Compute the optimal decision point for the first 5 PCAs of a dataset (e.g. a single person) and compute the information gain associated to it (plot 5 graphs, one for each component, and show the highest information gain). See slides for how to compute information gain.

4.1.2 - Compute a decision tree for the digit classification and visualize it.
You can use "rpart" for creating a tree and "rpart.plot" for visualizing the C5.0 tree.

```
datanew <- cbind(id_pca_first_5, id[,1])      # 'id_pca_first_5' is a dataframe
datanew$States <-factor(datanew[,6])
tree <- rpart(States ~ PC1 + PC2 + PC3 + PC4 + PC5, data = datanew, method = "class")
rpart.plot(tree)
```

Alternatively, you can use the "C50" lib

```
library(C50)
treeModel <- C5.0(x = id[, -1], y = id[,1])
plot(treeModel)
```

4.1.3 – Using the full data set (i.e. dataset from multiple people), evaluate a trained decision tree using cross validation. Try to train a tree with PCA, and without PCA (raw data). Discuss the important parameters.

```
predictions <- predict(tree, id_new[,-1], type = "class")
```

4.2 - Random forests:

4.2.1 - Create a Random Forest classifier and evaluate it using cross validation.
Discuss the critical parameters of “randomForest” (e.g., number and depth of trees)

```
library(randomForest)
model <- randomForest(States ~ PC1 + PC2 + PC3 + PC4 + PC5, data = id, ntree=100)
p <- predict(model, id_new)
```