DEPARTMENT OF STATISTICS

STATS 784 Data Mining

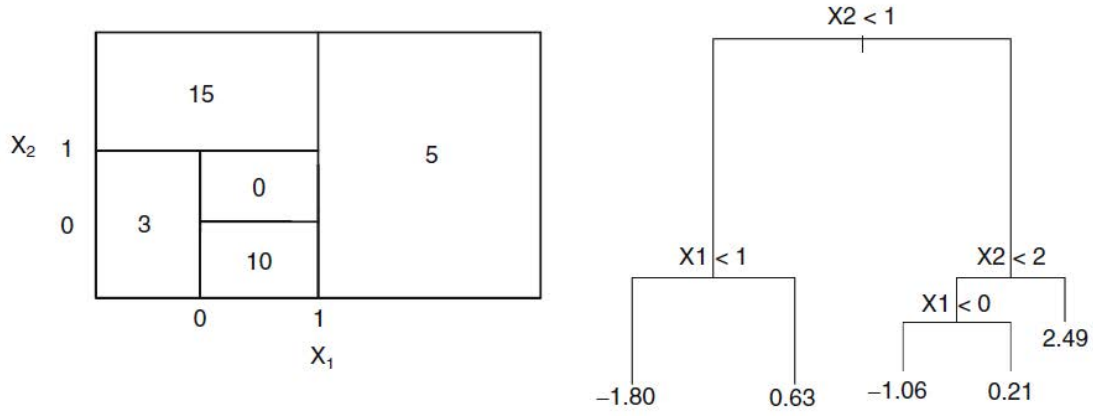
Assignment 2

Due Monday August 15. Hand in or mail to Alan by 5pm.

Q1. In the figure below

(a) Draw the tree corresponding to partition of the feature space in the left hand diagram. Show the value of the predictor at each node.

(b) Draw the partition of the feature space corresponding to the tree in the right hand diagram. Show the value of the predictor in each region.



[20 marks]

Q2. The data for this assignment are adapted from the Kaggle competition "Facial Keypoints Detection" described further at

https://www.kaggle.com/c/facial-keypoints-detection

There are both training and test sets available on the web page.

The data refer to a collection of about 1900 images of faces, represented by 96 x 96 pixel arrays. For each pixel, a greyscale value between 0 and 255 is also recorded.

In addition, the locations of the left eye in the image are also given.

The data sets supplied each have 9218 variables: These are

left_eye_center_x: the x–coordinate of the center of the left eye (from the subject's point of view)
left_eye_center_y: the y–coordinate of the center of the left eye (from the subject's point of view)
V1-V9216: The greyscale values of the 9216 pixels in the image.

Your general task is to create an x-predictor and a y-predictor of the two coordinates, using the pixel information as features, and evaluate their prediction errors. Specific steps are listed below.

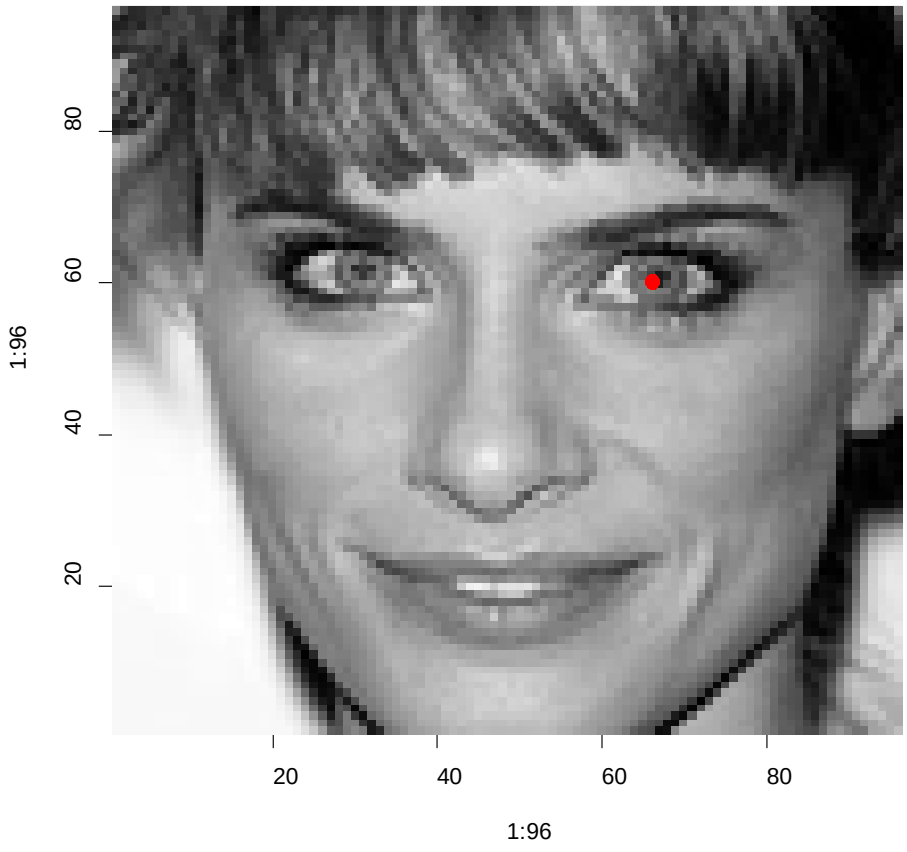
It is instructive to be able to create the image represented by the vector of coordinates. The following R function will do this: (note that the vector has to be formatted into the pixel array to get the correct image)

```
plotImage = function(imagevec){
imagemat = matrix(imagevec,96,96)
for(i in 1:96)imagemat[i,] = rev(imagemat[i,])
image(1:96, 1:96, imagemat, col = gray((0:255)/255))
}
```

To draw a face, where the values of V1, …,V9216 are in a vector **imagevec**, and plot the position of the left eye from the coordinates in the range 1-96, use

```
plotImage (imagevec)
points (xCoord,  96-yCoord, pch=19, cex=1.5, col="red"))
```

This results in the image below.



Specific tasks.

1. Read the training and test data into R.
2. Fit a tree to the x and y coordinates separately and calculate the predicted values for the test set. Print out the first 10 values of each predictor.
3. Estimate the prediction error using the test set estimate, and CV and the bootstrap on the training data. Compare.
4. Print out the first 25 images in the test data (on one page) and mark the position of the predicted eye centre.
5. Fit random forests and see if there is any improvement in the PE over fitting a single tree. You may want to use **`caret`** to tune the random forests.

Marks: 10 each for tasks 2-5.

Hint for 5. The random forest will run very slowly if you try to fit all 9216 features. You could try just using the features that appeared in the trees you fitted.