STAT201 Assignment 8

Robert Ivill 46012819

Question 1 [70% of the marks] The census dataset contains four demography characteristics of 1000 records from the US Census database in 1994. The Income variable represents two categories of annual income: “more than 50000” and “less or equal to 50000”. The aim of the classification analysis is to use demographic characteristics to predict whether the annual income of a person is greater than 50K.

a. Load the data. Check the structure. Convert the outcome variable into factor. Replace outcome_var with the name of the outcome variable in the code below and any other code where outcome_var occur. Which of the two outcomes is the “positive” outcome?

```
census <- read.csv("/Users/robbi/Downloads/census.csv")
str(census)

## 'data.frame':    1000 obs. of  4 variables:
##  $ Age           : int  38 51 38 30 38 21 34 33 66 38 ...
##  $ Education     : int  9 9 9 12 13 10 15 9 9 13 ...
##  $ Hours_per_week: int  36 40 40 40 40 20 50 40 35 20 ...
##  $ Income        : chr  ">50K" "<=50K" "<=50K" "<=50K" ...

census$Income <- as.factor(census$Income)
```

In the above code, the data is loaded and the structure is checked with the str function. The income variable is the outcome variable, therefore we replace outcome_var with Income.The positive outcome in the income variable is “>50k”.

b. Fit a logistic regression model to the census dataset (replace outcome_var with the required variable). Print the confusion matrix and calculate sensitivity and specificity.

Again, we replace outcome_var with Income in the code below:

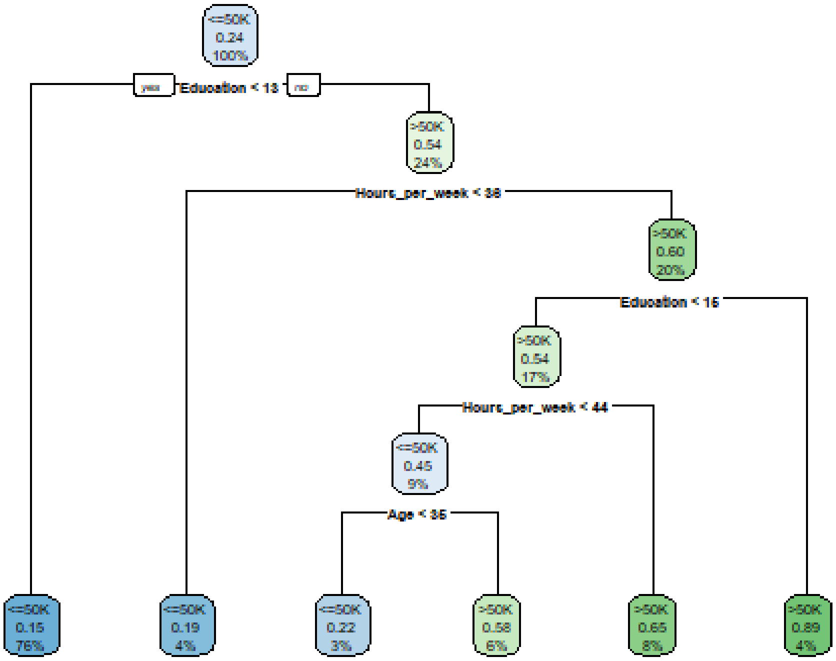
```
#install.packages("regclass")
census_glm <- glm(Income ~ ., family=binomial(link='logit'), data=census)
library(regclass)

## Loading required package: bestglm

## Loading required package: leaps

## Loading required package: VGAM

## Loading required package: stats4

## Loading required package: splines
## Loading required package: rpart

## Loading required package: randomForest

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

## Important regclass change from 1.3:
## All functions that had a . in the name now have an _
## all.correlations -> all_correlations, cor.demo -> cor_demo, etc.

confusion_matrix(census_glm)

##               Predicted <=50K Predicted >50K Total
## Actual <=50K             707             52   759
## Actual >50K              153             88   241
## Total                    860            140  1000
```

To calculate the sensitivity and specificity, we use the values from the confusion matrix. Sensitivity is calculated by TP/(TP+FN). The true positive value in the matrix is equal to 88 and the false negative value is equal to 153. So sensitivity is equal to 88/(88+153) = 0.365. Specificity is calculated as TN/(TN+FP). The true negative value is 707 and the false positive value is 52. So specificity is equal to 707/(707+52) = 0.931.

c. Fit a decision tree model to the census dataset and plot the obtained decision tree. What is the predicted income for a person who is 25 years old, has 13 years of education, and works 40 hours a week?

In the below code, we replace outcome_var with Income.

```
library(rpart)
census_tm <- rpart(Income ~ ., data=census)
library(rpart.plot)
rpart.plot(census_tm)
```



If we follow the specified nodes of the decision tree according to the definition of the person in question, we find that their predicted income is <=50K.

d. Print the confusion matrix for the decision tree model using the code below (replace outcome_var with the required variable). Calculate the accuracy of the model on the training data.

In the below code, outcome_var is again replaced with Income.

```
pred=predict(census_tm, type="class")
table(census$Income, pred)

##         pred
##           <=50K  >50K
##    <=50K    704    55
##    >50K     127   114
```

Accuracy is calculated as (TP+TN)/(TP+TN+FP+FN). The TP value is 114, the TN value is 704, the FP value is 55 and the FN value is 127. Therefore, the accuracy of the model is (114+704)/(114+704+55+127) = 0.853 or 85.3%.

e. Use 5-fold cross-validation to estimate misclassification error for the decision tree model (replace outcome_var and number with the required variable name and number in the code below). What accuracy do you expect from the decision tree model when used for future predictions (prediction accuracy)?

In the code below, outcome_var was replaced with Income and number was replaced with 5.

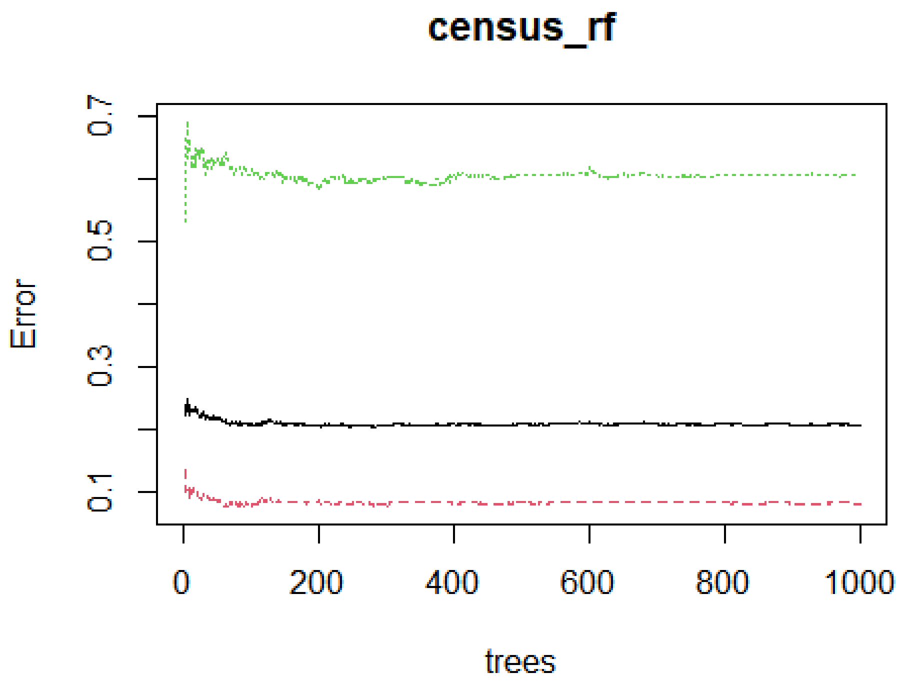
```
#install.packages("ipred")
library(ipred)
myrpredict <- function(object, newdata) predict(object, newdata,
type="class")
errorest(Income ~ ., data=census, model=rpart, predict=myrpredict,
estimator="cv",est.para=control.errorest(k=5))

##
## Call:
## errorest.data.frame(formula = Income ~ ., data = census, model = rpart,
##     predict = myrpredict, estimator = "cv", est.para = control.errorest(k
= 5))
##
##   5-fold cross-validation estimator of misclassification error
##
## Misclassification error:  0.201
```

From the 5-fold cross-validation estimator, we get an average misclassification error of 0.192. Therefore the prediction accuracy would be expected to be 1 - 0.192 = 0.808.

f. Fit a random forest model using 1000 trees (replace outcome_var and number with the required variable name and number in the code below). Plot the graph of errors for this model. Are 1000 trees enough for this analysis?

In the below code, outcome_var is replaced with Income and number is replaced with 1000.

```
library(randomForest)
set.seed(0)
census_rf <- randomForest(Income ~ ., data=census, ntree=1000)
plot(census_rf)
```



We can see that after a certain number of trees, the errors do not change much. This suggests that in this specific case the error rates begin to converge before 1000 trees, therefore 1000 trees is enough for this analysis.

g. In the questions above line set.seed(0) was used. Do you need to set the seed for your classification analyses outside of this assignment?

Setting the seed for classification analyses is necessary as it is good to ensure that the results of the models are reproducable. It makes sure that whenever the code is reran, the same outputs will be produced. When producing things like random forests that randomly chooses subsamples, it is important make sure that your analysis is reproducable.

h. The following code plots the ROC curves for the three fitted models (replace outcome_var with the required variable). Which of the models performs the best on the training data? Explain in terms of the position of the curve.

In the below code, outcome_var is replaced with Income.

```
#install.packages("pROC")
library(pROC)

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
## 
##     cov, smooth, var
```

```r
roc_glm = roc(census$Income, predict.glm(census_glm, type="response"))
```

```
## Setting levels: control = <=50K, case = >50K
```

```
## Setting direction: controls < cases
```

```r
roc_tree = roc(census$Income, predict(census_tm)[,2])
```

```
## Setting levels: control = <=50K, case = >50K
## Setting direction: controls < cases
```

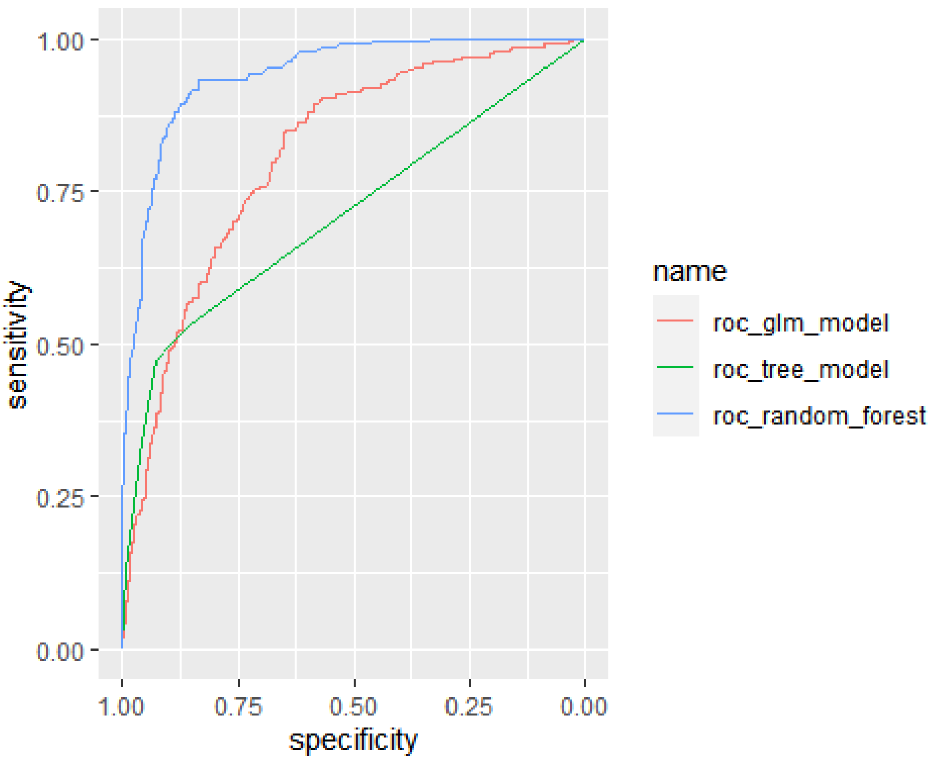
```r
roc_rf = roc(census$Income, predict(census_rf, census, type = 'prob')[,2])
```

```
## Setting levels: control = <=50K, case = >50K
## Setting direction: controls < cases
```

```r
model_list <- list()
model_list[["roc_glm_model"]] <- roc_glm
model_list[["roc_tree_model"]] <- roc_tree
model_list[["roc_random_forest"]] <- roc_rf
ggroc(model_list)
```

```
## You may need to call library(ggplot2) if you want to add layers, etc.
```

```
## Loading required namespace: ggplot2
```



From the ROC curve plot, we can see that the random forest model performs the best on the training data. This is because it is closest to the top left of the graph, meaning it has the highest true positive rate (sensitivity) with a lower false positive rate.

Question 2 [30% of the marks] File resp_disease.csv contains data of 60 patients with respiratory diseases. The variables are Age, four saliva permittivity measurements (ImaginaryPartMin,ImaginaryPartMax,RealPartMin, RealPartMax), patient ID and Diagnosis. The diagnoses are: Asthma, Infected, and Healthy Controls (HC). This dataset is going to be used to fit a classification model for diagnosing future patients.

a. Read in the data and check the structure. Convert the outcome variable to factor, replacing outcome_var with the name of the outcome variable in the code below. Which of the variables you would not use as a predictor variable?

In the code below, outcome_var is replaced with Diagnosis:

```
rd <- read.csv("/Users/robbi/Downloads/resp_disease.csv")
str(rd)

## 'data.frame':    60 obs. of  7 variables:
##  $ Diagnosis       : chr  "HC" "HC" "HC" "HC" ...
##  $ ID              : chr  "14-5" "15-3" "16-5" "18-4" ...
##  $ ImaginaryPartMin: num  -304 -330 -321 -321 -321 ...
##  $ ImaginaryPartMax: num  -288 -298 -306 -286 -303 ...
##  $ RealPartMin     : num  -514 -519 -469 -627 -469 ...
##  $ RealPartMax     : num  -453 -474 -465 -470 -465 ...
##  $ Age             : int  50 41 41 27 21 58 44 38 23 39 ...
```

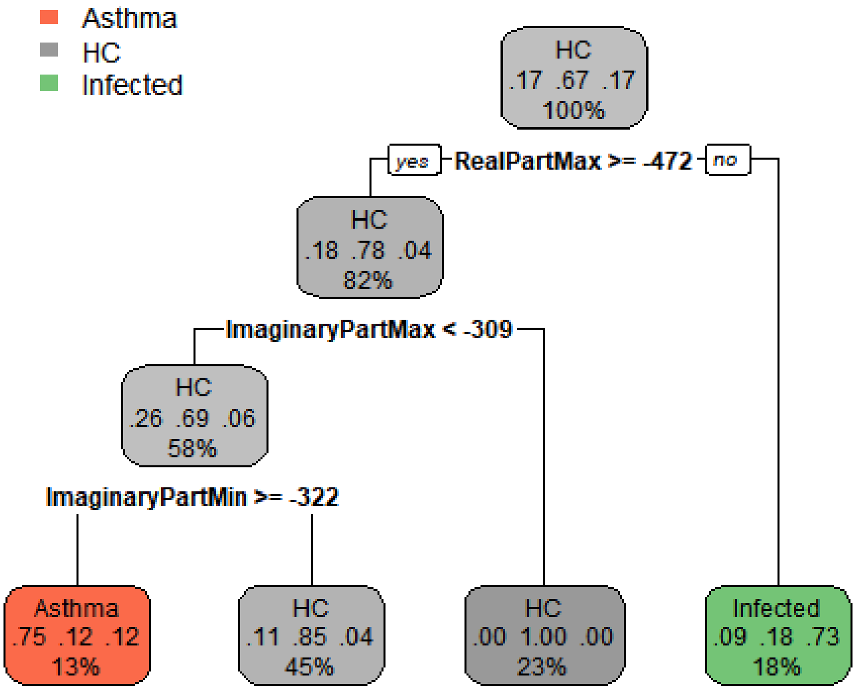
```
rd$Diagnosis <- as.factor(rd$Diagnosis)
```

The ID varaible would not be used as a predictor variable as it is more of an identifier rather than a predictor.

b. Fit a decision tree model and plot the obtained decision tree (replace outcome_var with the name of the outcome variable and exclude_var with the name of the variable you chose in a.) to answer the following questions: • Which variables do not influence the diagnosis in this decision tree? • A new patient has RealPartMax=-450, ImaginaryPartMax= -315, ImaginaryPartMin = -321. Which diagnosis will they receive? • What is the probability that this new patient has Asthma? • How many patients from the training data will be classified in the bottom left node.

In the code below, outcome_var was replaced with Diagnosis and exlude_var is replaced with ID:

```
library(rpart.plot)
rd_tm <- rpart(Diagnosis ~. - ID, data=rd)
rpart.plot(rd_tm)
```



The variables that don't influence the diagnosis in the decision tree are Age and RealPartMin as they are not involved in any of the splits in the tree. To figure out the diagnosis the patient will receive, we follow the decision path based on the values of their variables and choose the leaf node that it ends at. Thus, the patient's diagnosis should be Asthma. The probability that this patient has Asthma is 0.75. Of the patients in the training data, 13% of them will be classified into the bottom left node of the decision tree.