(https://lms.monash.edu/mod/assign/view.php?id=7560449)

FIT5197 Assignment 3 Semester 2, 2020 (https://lms.monash.edu/mod/assign/view.php?id=7560449)

Authors: Dan Nguyen, Yun Zhao

Admins (Competition): Dr. Levin Kuhlmann, Yun Zhao, Anil Gurbuz

Proofreaders: Dr. Levin Kuhlmann, Yun Zhao, and other tutors

Date: Oct 2020

(https://lms.monash.edu/mod/assign/view.php?id=7560449)

Assignment Instruction (https://lms.monash.edu/mod/assign/view.php?id=7560449)

Please read through the instructions carefully, by submitting the assignment, you are considered to have read all the instructions carefully and be aware of the penalties that entail.

Part 1: Regression (50 Marks)

This part is about regression. Specifically, you will be `predicting the fuel efficiency` of a car (in kilometers per litre) based on its characteristics. This is a practical problem as Australia is one of the largest automobile markets in the world; thus, correctly predicting the fuel efficiency is necessary to control emission rates to the environment.

The dataset has many observations and predictors obtained from many retailers for car models available for sale from 2017 to 2020. The target variable is the fuel efficiency of the car measured in kilometers per litre. The higher this value, the better the fuel efficiency of the car.

PleaseProvide working/R code/justifications for each of these questions as required.

Note: If not explicitly mentioned, libraries are not allowed

In [ ]:

```
# Read the data from students' side
remove(list = ls())
train <- read.csv("RegressionTrain.csv")
test <- read.csv("RegressionTest.csv")
```

In [ ]:

```
# PLEASE DO NOT ALTER THIS CODE BLOCK
# Please skip (don't run) this if you are a student
# Read in the data from marking tutors' side (ensure no cheating!)
remove(list = ls())
train <- read.csv("../data/RegressionTrain.csv")
test <- read.csv("../data/RegressionTest.csv")
label <- read.csv("../data/RegressionTestLabel.csv")
```

Question 1 (5 Marks)

Fit a **multiple linear model** to the fuel efficiency data using the `train` dataset. By checking the summary information, which predictors/variables do you think are possibly associated with fuel efficiency (use `0.05` significant level), and why? Which `three predictors/variables` appear to be the strongest predictors of fuel efficiency, and why?

Note: You don't have to worry about categorical variables here since R can deal with this automatically, focus your efforts on interpretation. Additionally, when explaining why features are strongly associated with the target, please refrain giving one or two sentences answers, these answers are not descriptive enough and will result in deduction of marks. Finally, please name the model here `lm.fit` for future marking purposes.

In [ ]:

```
# mutlivariate linear regression model
lm.fit <- lm(Comb.FE ~ .,data=train)
summary(lm.fit)
```

In [ ]:

```
TopPredictor.lm <- function(model){
    # Finding the best predictors/variables for fuel efficiency
    Predictor <- coef(summary(model))
    # Filtering only predictors/variables with significance of 0.05
    Best.Pred <- Predictor[Predictor[,"Pr(>|t|)"]<0.05,]
    Predictor <- sort(Best.Pred[,4],decreasing=FALSE)
    print(Predictor[1])
    print(Predictor[2])
    print(Predictor[3])
    }

TopPredictor(lm.fit)
```

SOLUTION 1:

Predictor/variables (significance level of 0.05 or less) that have possibly associated with fuel efficiency are:-

1. Model.Year `Significance: 0.05`
2. Eng.Displacement `Significance: 0.0`
3. AspirationSC `Significance: 0.0`
4. AspirationTC `Significance: 0.0`
5. AspirationTS `Significance: 0.0`
6. No.Gears `Significance: 0.0`
7. Lockup.Torque.ConverterY `Significance: 0.0`
8. Drive.SysF `Significance: 0.0`
9. Max.Ethanol `Significance: 0.05`
10. Fuel.TypeGP `Significance: 0.0`

Above variables have $\mathbf{Pr}(>|\mathbf{t}|)$ less than or equal to 0.05 when applied Linear model function.

Predictor/variables with strongest associated with fuel efficiency are:-

1. Eng.Displacement : $\mathbf{Pr}(>|\mathbf{t}|) - - - - >$ less than 2e-16
2. AspirationSC : $\mathbf{Pr}(>|\mathbf{t}|) - - - - - - >$ less than 2e-16
3. Drive.SysF : $\mathbf{Pr}(>|\mathbf{t}|) - - - - - -- >$ less than 2e-16

These are the most significant and have the least $\mathbf{Pr}(>|\mathbf{t}|)$ value when we apply linear modelling.

Question 2 (5 Marks)

Describe/discuss the effect that the year of manufacture `(Model.Year)` variable appears to have on the mean `fuel efficiency` . Additionally, describe/discuss the effect that the number of gears `(No.Gears)` variable has on the mean `fuel efficiency` of the car.

$\mathbf{Note}$: This asks for your descriptions, please refrain from using one or two lines to describe/discuss the effect. Keep answers to be 4 decimal places

SOLUTION 2:

Model.Year versus mean fuel efficiency

From below graph we can conclude that model year don't have any significant effect on fuel efficiency

Numbers of Gears versus mean fuel efficiency

From below graph we can conclude that Numbers of Gears does have significant effect on fuel efficiency. As No. of gears increases Fuel efficiency of mean car decreases significantly. Although not perfect an linear equation can be generated that might show graphs behaviour

In [ ]:

```
# grouping and finding the mean of data
df1 <- aggregate(x = train$Comb.FE,by=list(Year=train$Model.Year),FUN=mean)
df2 <- aggregate(x = train$Comb.FE,by=list(No.Gears=train$No.Gears),FUN=mean)

# Plotting graph
plot(df1,col  = "red",main = "Model.Year (Year) versus mean fuel efficiency (x)")
lines((df1))
plot(df2,col  = "red",main = "Numbers of Gears (No.Gears) versus mean fuel efficiency
 (x)")
lines((df2))
```

Question 3 (5 Marks)

Apply the stepwise selection procedure with the **BIC** penalty to prune out potentially less significant variables. Write down the final regression equation obtained after pruning, please keep the values of the parameter coefficients to 2 decimal places. Finally, also describe the pruned model.

Note: please don't change the default direction `both` in the step function, this is so that we can check your work easily. Additionally, please name this model `sw.fit`

YOUR ANSWER HERE

In [ ]:

```
# mutlivariate linear regression model with pruning done using BIC penalty
sw.fit <- step(lm.fit,k = log(nrow(train)), direction="both")
summary(sw.fit)
```

SOLUTION 3:

BIC penalty can help Prune variables which doesn't impact fuel efficency significantly. Model add or remove variables to give the equation with minimum AIC.

Step 1: `AIC=1424.68`

Comb.FE ~ Model.Year + Eng.Displacement + No.Cylinders + Aspiration + No.Gears + Lockup.Torque.Converter + Drive.Sys + Max.Ethanol + Fuel.Type

Step 2: `AIC=1413.96`

Comb.FE ~ Model.Year + Eng.Displacement + No.Cylinders + Aspiration + No.Gears + Lockup.Torque.Converter + Drive.Sys + Max.Ethanol

Step 3: `AIC=1407.38`

Comb.FE ~ Model.Year + Eng.Displacement + Aspiration + No.Gears + Lockup.Torque.Converter + Drive.Sys + Max.Ethanol

Step 4: `AIC=1405.54`

Comb.FE ~ Eng.Displacement + Aspiration + No.Gears + Lockup.Torque.Converter + Drive.Sys + Max.Ethanol

FORMULA: **Comb.FE = 16.2 - 1.28*Eng.displacement* - *0.1*AspirationOT - 0.7*AspirationSC* *-1.14*AspiratioTC - 1.12*AspirationTS* - *0.11*No.Gears -0.83*Lockup.Torque.Converter* + *0.04*Drive.SysA + 1.48*Drive.SysF* *-0.32*Drive.SysP + 0.09 *Drive.SysR* - *0.01*Max.Ethanol**

Question 4 (5 Marks)

Say we are going to buy a new car and we want to improve the fuel efficiency of our new car, what does this `BIC model` suggest we should do? Provide a detailed answers of at least `150 words` .

SOLUTION 4:

From the data formula mentioned above we get some sort of picture about an mathermatical model than can be used to figure out dependence of fuel efficiency on independent variable such as Engine displacement, Aspiration, Number of Gears ,Lockup Torque Converter , Drive System and Maximum Ethanol. These independent varaibles are selected by BIC model because there combination leads to lowest Akaike information criterion, i.e. AIC . Decreasing the values with negative coefficent such as Eng.displacement (coeff = -1.28) and increasing values with positive coefficent such as Drive.SysF (coeff = 1.48) is neccesary to improve the Fuel efficiency of our new car

Question 5 (5 Marks)

Imagine that you are looking for a new car to buy to replace your existing car. Use the test dataset to inspect the first car fuel efficiency and see whether it is a good fit for you or not.

(a) Use your BIC model to predict the mean fuel efficiency for this new car. Provide a 95% confidence interval for this prediction. [2 mark]
(b) Following the previous estimation, given that the current car that you own has a mean fuel efficiency of 9.5 km/l (measured over the life time of your ownership), does your model (BIC) suggest that the new car will have better fuel efficiency than your current car? Why? [3 marks]

In [ ]:

```
# fuel efficiency prediction with 95% confidence interval
test$Comb.FE = predict(sw.fit,test,level=0.95, interval='confidence')
print(test$Comb.FE[1,])
```

SOLUTION 5:

1. **Using BIC (95% confidence interval) car's mean F.E. value is 9.287 , lower F.E. value is 9.053 and upper F.E. value 9.521**
2. **As the value is very close to mean fuel efficiency of 9.5 km/l therefore it's not wise to buy current car**

Question 6 (Libraries are allowed) (25 Marks)

As a Data Scientist, one of the key tasks is to build models $\mathrm{most\ appropriate/closest}$ to the truth; thus, modelling will not be limitted to these steps in the assignment. To simulate for a realistic modelling process, this question will be in the form of a competition among students to find out who has the best model.

Thus, You will be graded by the performance of your model compared to your classmates', the better your model, the higher your score. Additionally, you need to write a short paragraph describing/documenting your thought process in this model building process `(300 words)` . Note that this is to explain to us why you build your current model so that we can verify that you understand the model you build and not just copy from other people.

Note **Please make sure that we can install the libraries that you use in this part, the code structure can be:**

```
install.packages("some package", repos='http://cran.us.r-project.org')
```

```
library("some package")
```

Remember that if we cannot run your code, we will have to give you 0 marks, our suggestion is for you to use the standard `R version 3.6.1`

You also need to name your final model `fin.mod` so we can run a check to find out your performance. A good test for your understanding would be to set the previous $\mathrm{BIC\ model}$ to be the final model to check if your code works Appropriately.

20 **Marks for the model performance in the competition**

5 **Marks for logically writing down the thought process in building the final model**

This is the [link (https://www.kaggle.com/t/0a3c0fc91b074816a6315bb4e9b42602)](https://www.kaggle.com/t/0a3c0fc91b074816a6315bb4e9b42602) to the competition

SOLUTION 6:

After pruning the data using BIC criterion, the R square value has observed to be 0.65. Hence, to improve the accuracy (RMSE score) first we need to

1. **transform the independent variables**
2. **Find Optimum Regression model.**

ggpairs was used to observe the trend of fuel efficiency w.r.t other independent variables. To make an optimum final regression equation an combination of logs and polynomials of independent variables were used. After carefully observing the trend of fuel efficiency w.r.t other independent variables,

Further randomForest modelling were used as final model as in comparision with other models tried, RandomForest resulted in better score when checked for output on kaggle. After that hyper parameter optimization were neccesary to extract best RMSE score.

After multiple experiments combination of parameters were found which result in best score so far.

In [ ]:

```
# Library used
# install.packages("randomForestSRC")
# install.packages("GGally")
options(warn=0)
library(randomForestSRC)
library(GGally)

# An correlation graph to make sense of data variables as well as any relation that can
effect train-label
data(train, package = "reshape")
ggpairs(
  train[, c(2, 3, 4, 5, 8, 10)],
  upper = list(continuous = "density", combo = "box_no_facet"),
  lower = list(continuous = "points", combo = "dot_no_facet")
)

# Creating extra variables that might represent fuel efficiency
train$log.Eng.Displacement = log(train$Eng.Displacement)
train$log.Gear = log(train$No.Gears)
train$log.Max.Ethanol = log(train$Max.Ethanol)
train$log.No.Cylinders = log(train$No.Cylinders)

train$SQRT.Eng = sqrt(train$Eng.Displacement)
train$SQRT.Gear = sqrt(train$No.Gears)
train$SQRT.Max.Ethanol = sqrt(train$Max.Ethanol)
train$SQRT.No.Cylinders = sqrt(train$No.Cylinders)

train$Sq.Eng = (train$Eng.Displacement)^2
train$Sq.Gear = (train$No.Gears)^2
train$Sq.Max.Ethanol = (train$Max.Ethanol)^2
train$Sq.No.Cylinders = (train$No.Cylinders)^2

train$log2.Ethanol <- log(log(log(train$Max.Ethanol)))
train$Combination <- train$log.Eng.Displacement * train$Sq.Max.Ethanol *train$Sq.Gear *
train$Sq.No.Cylinders

# For predicting the test label, new variables were introduced in test data
test$log.Eng.Displacement = log(test$Eng.Displacement)
test$log.Gear = log(test$No.Gears)
test$log.Max.Ethanol = log(test$Max.Ethanol)
test$log.No.Cylinders = log(test$No.Cylinders)

test$SQRT.Eng = sqrt(test$Eng.Displacement)
test$SQRT.Gear = sqrt(test$No.Gears)
test$SQRT.Max.Ethanol = sqrt(test$Max.Ethanol)
test$SQRT.No.Cylinders = sqrt(test$No.Cylinders)

test$Sq.Eng = (test$Eng.Displacement)^2
test$Sq.Gear = (test$No.Gears)^2
test$Sq.Max.Ethanol = (test$Max.Ethanol)^2
test$Sq.No.Cylinders = (test$No.Cylinders)^2

test$log2.Ethanol <- log(log(log(test$Max.Ethanol)))
test$Combination <- test$log.Eng.Displacement * test$Sq.Max.Ethanol *test$Sq.Gear *test
$Sq.No.Cylinders
```

In [ ]:

```
# Use this function to check the performance of your model
rmse <- function(pred.label, truth.label){
    # Lower is better
    return(sqrt(mean((pred.label - truth.label)^2)))
}
```

In [ ]:

```
# Build your final model here, use additional coding block if you want to
fin.mod <- NULL
# Creating an model with tuned variables using experiments
fin.mod <- rfsrc(Comb.FE ~ Eng.Displacement + Eng.Displacement * log.Eng.Displacement +
Combination + log2.Ethanol + Aspiration + No.Gears *
                  Lockup.Torque.Converter + Drive.Sys + Max.Ethanol + Fuel.Type + No.C
ylinders +
                  log.Eng.Displacement * log.Gear * log.Max.Ethanol * log.No.Cylinders
+ SQRT.Eng * SQRT.Gear *
                  SQRT.Max.Ethanol * SQRT.No.Cylinders + Sq.Eng * Sq.Gear * Sq.Max.Eth
anol * Sq.No.Cylinders,
    data = train,importance=TRUE,proximity=TRUE,mtry=8,nodesize=2,ntree=950,replace=TRU
E)
```

In [ ]:

```
# If you are using any packages that perform the prediction differently, please change
 the value of this variable
pred.label <- predict(fin.mod, test)
predict.label <- pred.label$predicted
```

In [ ]:

```
# PLEASE DO NOT ALTER THIS CODE BLOCK
# put this label in a csv file to commit to the Leaderboard
write.csv(data.frame("RowIndex" = seq(1, length(predict.label)), "Prediction" = predic
t.label),
          "RegressionPredictLabel.csv", row.names = F)
```

In [ ]:

```
## PLEASE DO NOT ALTER THIS CODE BLOCK
## Please skip (don't run) this if you are a student
## For teaching team use only
RMSE.fin <- rmse(pred.label, label$Label)
cat(paste("RMSE is", RMSE.fin))
```

Part 2: Classification (50 Marks)

In this part, you are going to work with "Census Income Dataset" which was originally donated by Ronny Kohavi and Barry Becker to UCI (University of California, Irvine) in 1996. This is a trimmed dataset used for machine learning students to study classification.

This dataset has collected over 40,000 records (we excluded some data in our version) regarding personal yearly income with 12 attributes (predictors). The attributes comprise many aspects of a person that may contribute to the yearly income. You can use summary() function to obtain the attributes information. Your prediction task is to determine whether a person makes over 50K a year.

We have splitted the dataset into a trainning and a testing set. There are 27245 records in the training set while 13631 records in the testing set. Besides the 12 predictors, there is one more column named Salary indicating whether a person's yearly income is over 50K. The label information is a seperated file for the testing set and will be used by us to asess your performance later. Note the label TRUE means an individual's yearly salary exceeds 50K while FALSE means an individual's yearly salary is under 50K.

Note: If not explicitly mentioned, libraries are not allowed

In [ ]:

```
# Read the data from students' side
remove(list = ls())
train <- read.csv("ClassTrain.csv")
test  <- read.csv("ClassTest.csv")
```

In [ ]:

```
## PLEASE DO NOT ALTER THIS CODE BLOCK
# Please skip (don't run) this if you are a student
# Read in the data from marking tutors' side (ensure no cheating!)
remove(list = ls())
train <- read.csv("../data/ClassTrain.csv")
test  <- read.csv("../data/ClassTest.csv")
label <- read.csv("../data/ClassTestLabel.csv")
```

Question 1 (10 Marks)

Fit a $\mathrm{Generalized\ Linear\ Model\ (Logistic\ Regression)}$ to predict level of income (salary) $(\geq 50\ \mathrm{K}, \mathrm{or}\ < 50\ \mathrm{K})$ using the `train` dataset. Using the results of fitting this model, which predictors do you think are possibly associated with the level of Salary (use `0.05` significant level), and why? Which `three variables` appear to be the strongest predictors of salary, and why?

Furthermore, you can see that you have much more predictors in this part than in the `linear model` from Part 1 $\Rightarrow$ manually checking information is counterproductive. Thus, please write a function to automate these processes (1) selecting important feature against 0.05 threshold and (2) Selecting three most important features.

Note: You don't have to worry about categorical variables here since R can deal with this automatically, focus your efforts on interpretation. Additionally, when explaining why features are strongly associated with the target, please refrain from giving one or two sentences answers, these answers are not descriptive and will result in a deduction of marks. Finally, please name the model here `glm.fit` and have the parameter in the model set to `family = binomial` .

SOLUTION 1:

Predictor/variables that have possibly associated with Salary are:-

Age, WorkClass, FinalWeight, Education, MaritalStatus, Occupation, Relationship, Gender, CapitalGain, CapitalLoss, HoursWork

Predictor/variables with strongest associated with fuel efficiency are:-

1. **CapitalGain**
2. **HoursWork**
3. **CapitalLoss**

These are the most significant and have the least $\mathbf{Pr}(>|\mathbf{z}|)$ value when we apply Generalized linear modelling.

In [ ]:

```
# Build your model, keep family = binomial, ignore the warnings, they are benign\
# mutlivariate generalized linear regression model
glm.fit <- glm(Salary ~.,data = train, family = binomial)
options(warn=0)
summary(glm.fit)
```

In [ ]:

```
TopPredictor <- function(model){
    # Finding the best predictors/variables for fuel efficiency
    Predictor <- coef(summary(model))
    # Filtering only predictors/variables with significance of 0.05
    Best.Pred <- Predictor[Predictor[,"Pr(>|z|)"]<0.05,]
    Predictor <- sort(Best.Pred[,4],decreasing=FALSE)
    print(Predictor[1])
    print(Predictor[3])
    print(Predictor[4])
    }

TopPredictor(glm.fit)
```

Question 2 (10 Marks)

Firstly, please use the model created in the previous question to predict for the labels of the train data. Consequently, our objective is to compare this `predict.label` with the `truth.label` from the test data. However, as we don't know the test label, we have to estimate model performance using train data at this moment.

Secondly, since our objective is to estimate the performance of this model in making correct predictions; thus, this question also asks you to explore different [performance metrics (https://en.wikipedia.org/wiki/Precision_and_recall)](https://en.wikipedia.org/wiki/Precision_and_recall) for classification models. The metrics we will use are Accuracy, Precision, Recall, and F1 Score, please create a function to calculate these value and print them out properly using the given structure.

Additionally, please also discuss the results of these values in the context of your model.

Note: This asks for your descriptions, please refrain from using one or two lines to describe/discuss the effect. Keep answers to be 4 decimal places

SOLUTION 2:

Accuracy measured using the true postive and true negative values. After looking at the confusion matrix value, we find that the true negative values are indeed very high as compared to the true positive value. But still my accuracy is high.

The value of precision, recall and f1-score the value is low in this confusion matrix.

In [ ]:

```
# Apply your previous model to perform prediction, keep type = "response"
# Don't worry if you receive some warnings, they are benign
predict.label <- predict(glm.fit,train,type = "response")

# Truth label from train data
truth.label <- train$Salary
```

In [ ]:

```
# Model statistics function
mod.stat <- function(predict.label, truth.label){
    # instantiate the variables
    accuracy <- NULL
    precision <- NULL
    recall <- NULL
    F1 <- NULL

    ##############################
    #Your calculatation here

    # Creating an confusion table. Predicted probabilities which are > 0.5 as true and
 vice-versa
    predict.label <- table(truth.label, predict.label > 0.5)

    accuracy <- round(sum(diag(predict.label)) / sum(predict.label),4)
    precision <- round(predict.label[2,2] / (predict.label[2,2] + predict.label[1,2]),4
)
    recall <- round(predict.label[2,2] / (predict.label[2,2] + predict.label[2,1]),4)
    F1 <- 2 * round(((precision * recall) / (precision + recall)),4)


    ##############################

    # Return a list of value
    return(list("accuracy" = accuracy, "precision" = precision, "recall" = recall, "fsc
ore" = F1))
}
```

In [ ]:

```
mod.stat(predict.label, truth.label)
```

Question 3 (5 Marks)

Use the stepwise selection procedure with the BIC penalty to prune out potentially unimportant variables. Checking the performance of your model using the created `mod.stat()` function, please give your discussion as how this model is compared with the `glm.fit` (you can run the `mod.stat()` function for this as well if you want to).

Note: please don't change the default direction `both` in the step function, this is so that we can check your work easily. Additionally, please name this model `sw.fit` . Don't worry about the warnings, they are benign

SOLUTION 3:

From the data we can see that sw.fit has slightly higher precision in comparision to glm.fit. however sw.fit has lower F1 and recall score in comparision to glm.fit. sw.fit and glm.fit don't have any significat improvement over each other

In [ ]:

```
# Setting to suppress warnings
options(warn=-1)
# Fit a stepwise model
sw.fit <- step(glm.fit,trace = 0,k = log(nrow(train)), direction="both")
# Setting to suppress warnings
options(warn=0)
# Getting the summary to understand the result
summary(sw.fit)
```

In [ ]:

```
# Making prediction using train data and view the statistics
predict.label <- NULL
predict.label <- predict(sw.fit,train,type = "response")
```

In [ ]:

```
# Only run the below if you have labels, in your submission, this must be UNCOMMENTED
mod.stat(predict.label, truth.label)
```

PROVIDE DISCUSSION HERE

Question 4 (Libraries are allowed) (25 Marks)

Similar to the first part, to simulate for a realistic modelling process, this question will be in the form of a competition among students to find out who has the best model.

Thus, You will be graded by the performance of your model compared to your classmates', the better your model, the higher your score. Additionally, you need to write a short paragraph describing/documenting your thought process in this model building process `(300 words)` . Note that this is to explain to us why you build your current model so that we can verify that you understand the model you build and not just copy from other people.

Note Please make sure that we can install the libraries that you use in this part, the code structure can be:

```
install.packages("some package", repos='http://cran.us.r-project.org')

library("some package")
```

Remember that if we cannot run your code, we will have to give you a deduction, our suggestion is for you to use the standard `R version 3.6.1`

You also need to name your final model `fin.mod` so we can run a check to find out your performance. A good test for your understanding would be to set the previous BIC model to be the final model to check if your code works perfectly.

20 Marks for the model performance in the competition

5 Marks for logically writing down the thought process in building the final model

This is the link (https://www.kaggle.com/t/1bdebc96607742dbaf47ab36cd3ae421) to the competition

SOLUTION 4:

After pruning the data using BIC criterion, the Accuarcy has observed to be 0.84. Hence, to improve the accuracy (RMSE score) first we need an better model

Few Models were tried such as SVM, h2o and train function but best non tuned accuracy was observed using ranger. Which uses randomForest techniques. Further randomForest modelling were used as final model as in comparision with other models tried, RandomForest resulted in better score when checked for output on kaggle. After that hyper parameter optimization were neccesary to extract best Accuarcy.

After multiple experiments combination of parameters were found which result in best score so far.

In [ ]:

```
# Library used

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

In [ ]:

```
# Build your final model here, use additional coding block if you want to
fin.mod <- NULL
# An example would be use the previous model as your final one
fin.mod <- ranger(Salary ~ .,data = train, mtry = 3,min.node.size = 4,num.tree = 900)
```

In [ ]:

```
# Getting the predict label for the TEST data
pred.label <- predict(fin.mod, test)
pred.label <- (pred.label$predictions > 0)
```

In [ ]:

```
# PLEASE DO NOT ALTER THIS CODE BLOCK
# Use this csv file to commit to the leaderboard
write.csv(data.frame("RowIndex" = seq(1, length(pred.label)), "Prediction" = pred.label
),
          "ClassPredictLabel.csv", row.names = F)
```

In [ ]:

```
## PLEASE DO NOT ALTER THIS CODE BLOCK
## Please skip (don't run) this if you are a student
## For teaching team use only
source("../data/modassess.r")
model.perf <- mod.stat.test(pred.label,label$Label)
print(model.perf)
```

References