R Notebook

Code

Hide

```
#Had an issue with 00Lock problems, so I had to include the this statement to prevent any furthe
r issues. I asked
# ChatGPT how to do this.
unlink("C:/Users/yoshi/AppData/Roaming/R-3.6.3/library/00LOCK-rsample", recursive = TRUE)


#Here, I am importing all of the libraries I tried save for GGPlot2 and DoParallel as I got rid
of my usage of them.
library(rpart)
library(rpart.plot)
library(dplyr)
library(rsample)
library(randomForest)
library(caret)
#library(doParallel)
library(ranger)
library(ggplot2)

#Reading my CSV files !
EarningsTrain <- read.csv("C:/Users/yoshi/OneDrive/Documents/Data101/Aligina_Finalp1/EarningsTra
in.csv")
Test <- read.csv("C:/Users/yoshi/OneDrive/Documents/Data101/Aligina_Finalp1/EarningsTest.csv")
```

Hide

```
#Here, I am splitting this into a 70-30 split of my training set with my favourite number.
set.seed(777) # My fav number <3

esplit <- initial_split(EarningsTrain, prop = 0.75)
train <- training(esplit)
val <- testing(esplit)
```

Hide

```
earn <- rpart(Earnings ~ ., data = train[, setdiff(names(train), "pred.Earnings")], method = "an
ova",
              control = rpart.control(cp = 0.00003, minsplit = 10, minbucket = 8, maxdepth = 2
0))


#Model training using the two different parts of the training dataset and creating a pred.Earnin
gs column for my
# two sections.
train$pred.Earnings <- predict(earn, train)
MSE_train <- mean((train$Earnings - train$pred.Earnings)^2)
val$pred.Earnings <- predict(earn, val)
MSE_val <- mean((val$Earnings - val$pred.Earnings)^2)

#MSES made for the two  of them!
print(paste("MSE Training: ", MSE_train))
```

```
[1] "MSE Training:  17106.8256747643"
```

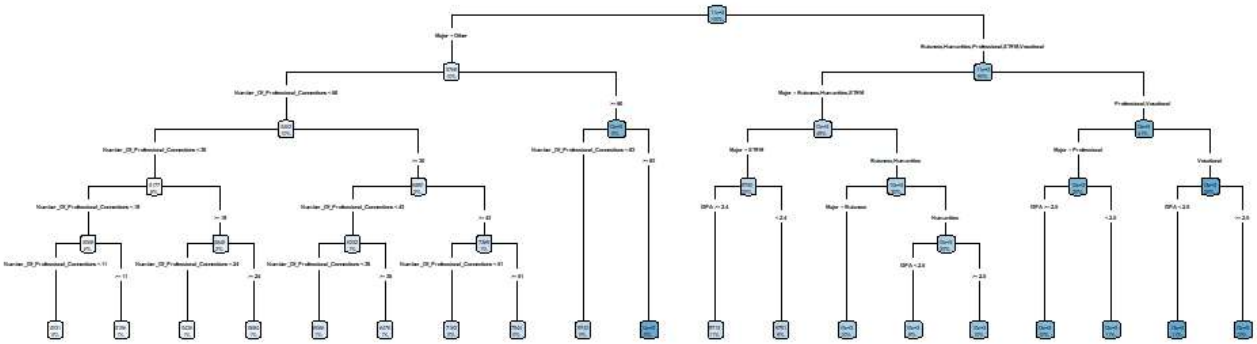
Hide

```
print(paste("MSE Validation: ", MSE_val))
```

```
[1] "MSE Validation:  13800.445544661"
```

Hide

```
#Visualising the current tree, it looks very confusing straight off the bat, this allowed me to
visualise the CP, and nodes and change as I wished.
rpart.plot(earn, type=4, fallen.leaves = TRUE)
```



Hide

```
# I think that this means I need to start working on making tweaks and trying new ideas to lower my MSE as much as possible.  Initially, I got an MSE of 17-13, which is pretty good comparing to the prior of 100 k.Then was suggested to try some tuning methods by ChatGPT which involved increasing the number of trees, and otherwise using cross validation methods. It suggested I use the ranger method in caret as it is more efficient than that of the RF method.

#From that, I got down to 7000 MSE, and now I just have to mess with the settings / tree amount to lower this to my goal of 5000. I came to realise that this  may be too high, ChatGPT then advised me to turn down my tree number, and I did. Getting up to about 6700.

#This was my initial RF model before I realised I wanted to do more tweaking.
#forest_model <- randomForest(Earnings ~ ., data = train, ntree = 10432, mtry = 4, nodesize = 5)


#Here, I created a feature with Height and Professional Connections. I feel like height is considered to be a characteristic that many people find to be important and it is often the first thing people see. I feel it could be height dependent.
train$HP <- train$Height * train$Number_Of_Professional_Connections
val$HP <- val$Height * val$Number_Of_Professional_Connections
Test$HP <- Test$Height * Test$Number_Of_Professional_Connections

#And for this, I created a feature of Grad year and Credits. This is because more credits assumes an intensive degree, and during the older years, a more impressive degree may carry more weight.
train$GC<- train$Graduation_Year * train$Number_Of_Credits
val$GC <- val$Graduation_Year * val$Number_Of_Credits
Test$GC <- Test$Graduation_Year * Test$Number_Of_Credits


train_control <- trainControl(method = "cv", number = 20, allowParallel = TRUE, returnResamp = "all")

# =Suggested by ChatGPT
tune_grid <- expand.grid(
  .mtry = c(2,3, 4,5,6),
  .splitrule = c("variance", "extratrees"),
  .min.node.size = c(2,3,4)
)

# New Ranger model, an implementation of randomForest with 700 trees
model <- train(Earnings ~ ., data = train[, setdiff(names(train), "pred.Earnings")], method = "ranger",
               trControl = train_control, tuneGrid = tune_grid, num.trees = 700)



predEarn_RF <- predict(model, val)
MSE_RF <- mean((val$Earnings - predEarn_RF)^2)
print(paste("FINAL MSE Validation with Random Forest: ", MSE_RF))
```

```
[1] "FINAL MSE Validation with Random Forest:  6788.11912554744"
```

Hide

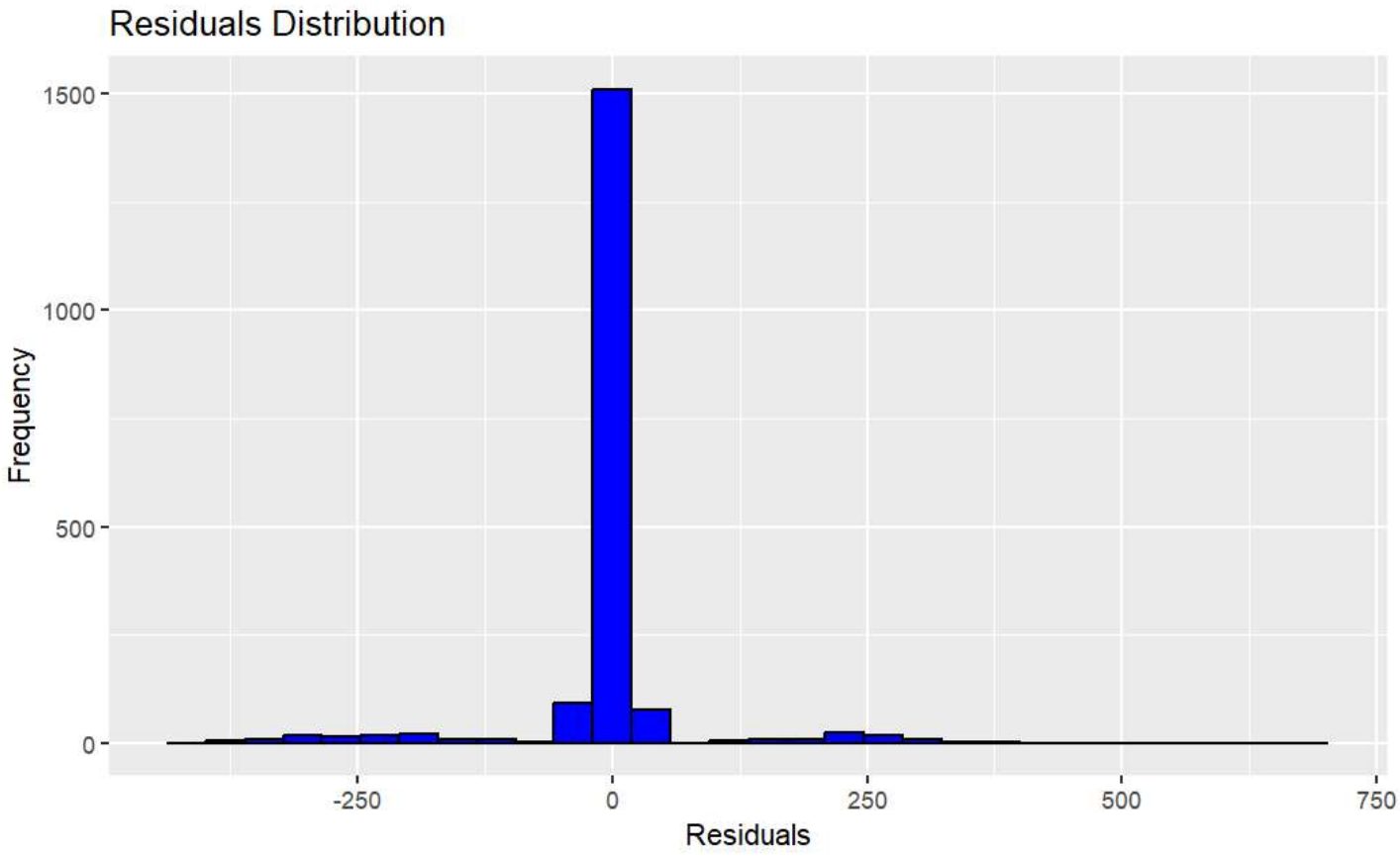
```
#CGPT Then suggested the usage of a histogram to check my accuracy, being centered around zero e
nsures it is MOSTLY accurate,
# but not overfitting.

residuals <- val$Earnings - predictions

predictions <- predict(model, val)
if (is.factor(predictions)) {
  predictions <- as.numeric(as.character(predictions))
}

# Calculate residuals
residuals <- val$Earnings - predictions
if (is.factor(residuals)) {
  residuals <- as.numeric(levels(residuals))[residuals]
}

# Plot residuals using ggplot2
ggplot(data.frame(Residuals = residuals), aes(x = Residuals)) +
  geom_histogram(bins = 30, fill = "blue", color = "black") +
  ggtitle("Residuals Distribution") +
  xlab("Residuals") +
  ylab("Frequency")
```



Hide

```
#However, when I was working on this, and attempting to lower my MSE, I was measuring the measur
e of accuracy, utilising a ggplot2 function given to me by CGPT, and the accuracy reached 100% p
rior to 5% margin of error, so I feel like it is best to reduce a few things and leave it at thi
s as too low of an MSE may cause overfitting.
```

Hide

```
#Finally, what we want to do is to try it on our training model and write our results into the n
ew file.
Test$predEarnings <- predict(model, newdata = Test)
write.csv(Test, "EarningsTest_with_PredEarnings.csv", row.names = FALSE)
```