Predicting Earnings with Machine Learning

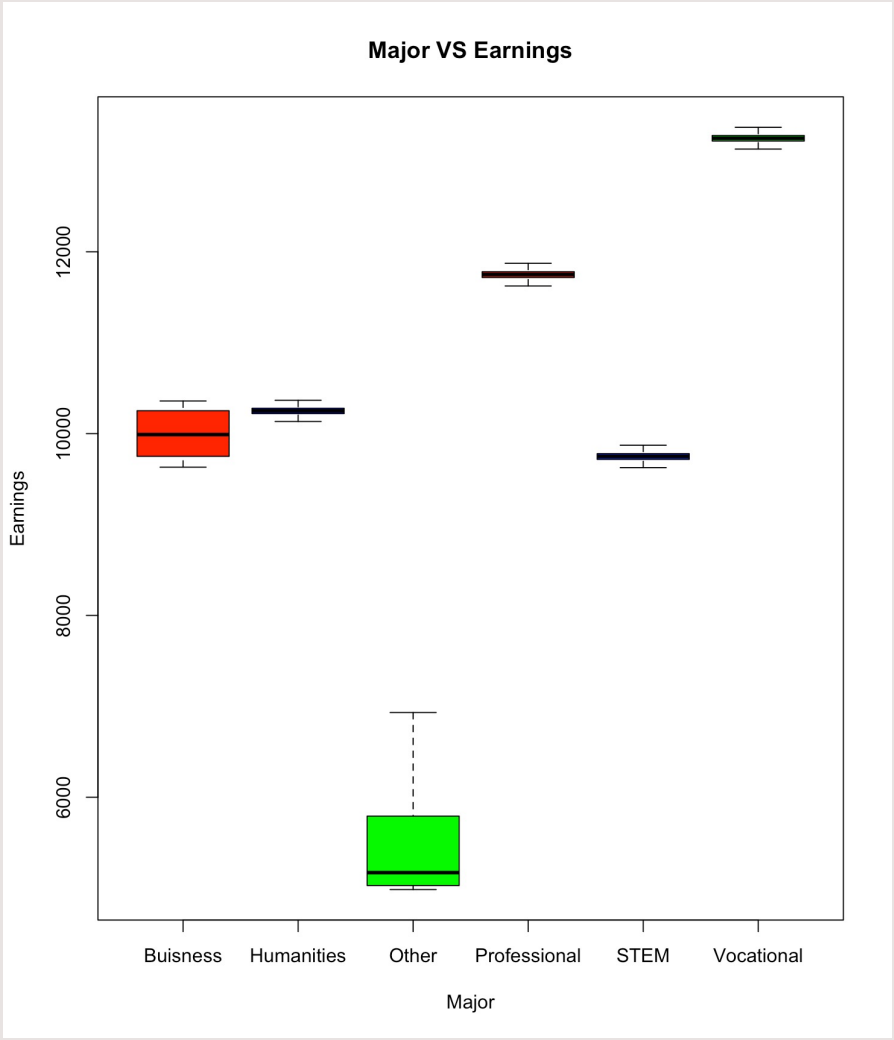
By Fauzan Amjad

The Dataset

We have a rich dataset with 9995 rows that represent people and 8 columns that represent 8 columns of data that characterize the person.

The columns include GPA, number of professional connections, earnings, major, graduation year, height, number of credits, and number of parking tickets.

The goal is to make an accurate prediction of each person’s respective earnings using the values that we’re given.



Major VS Earnings

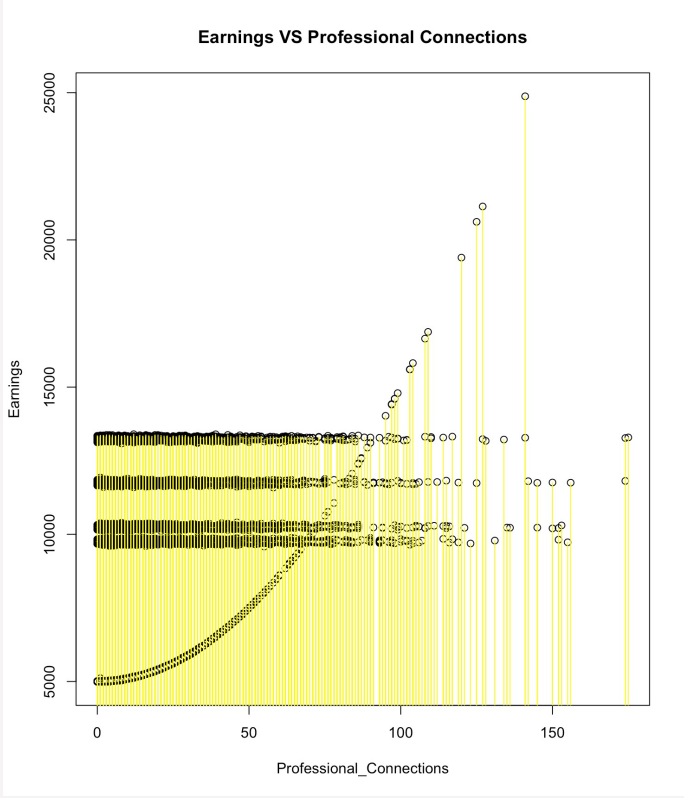
When plotting major vs earnings using a boxplot, we notice that for humanities, professional STEM, and vocation the earnings are very congested.

Business and other majors facilitate a broader spectrum of earnings.

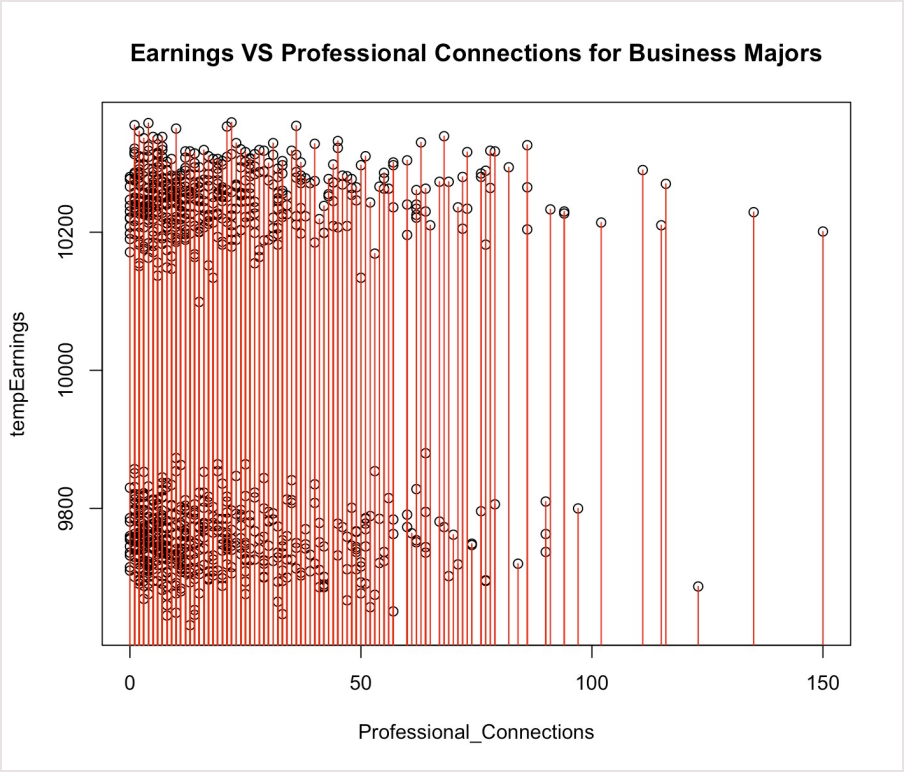
Conclusion: Major plays a role, but it’s not the sole determinant.

Earnings VS Professional Connections

Using a scatterplot to visualize earnings vs professional connections yields an interesting plot. We see 4 distinct straight lines; however, there this one line that almost exponentially increases as professional connections increases.



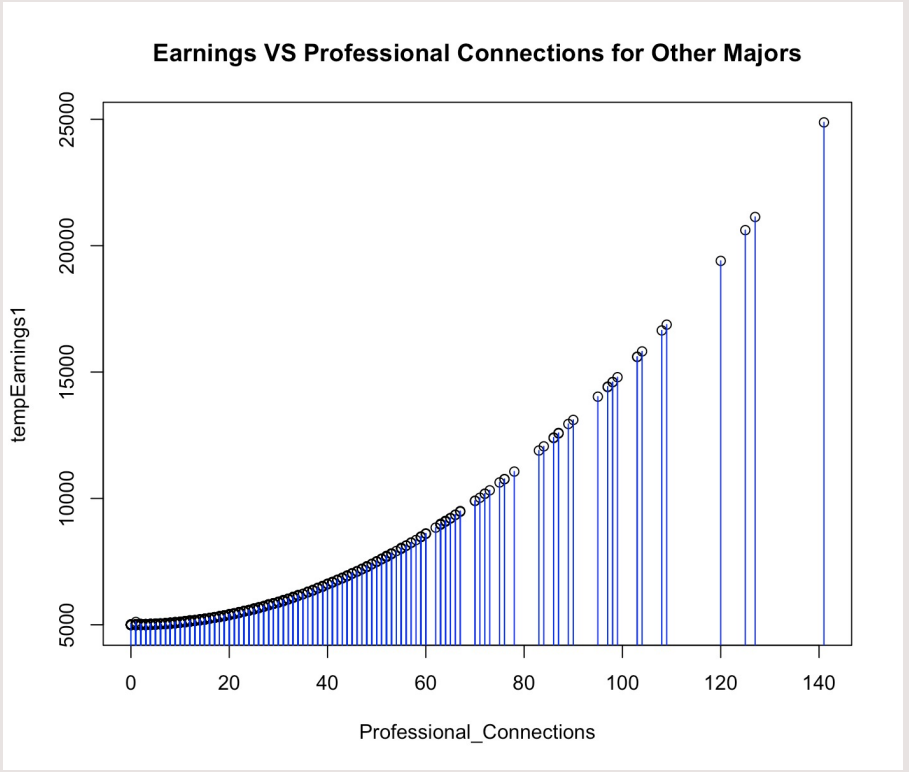
Subset the Data for Business and Other Majors

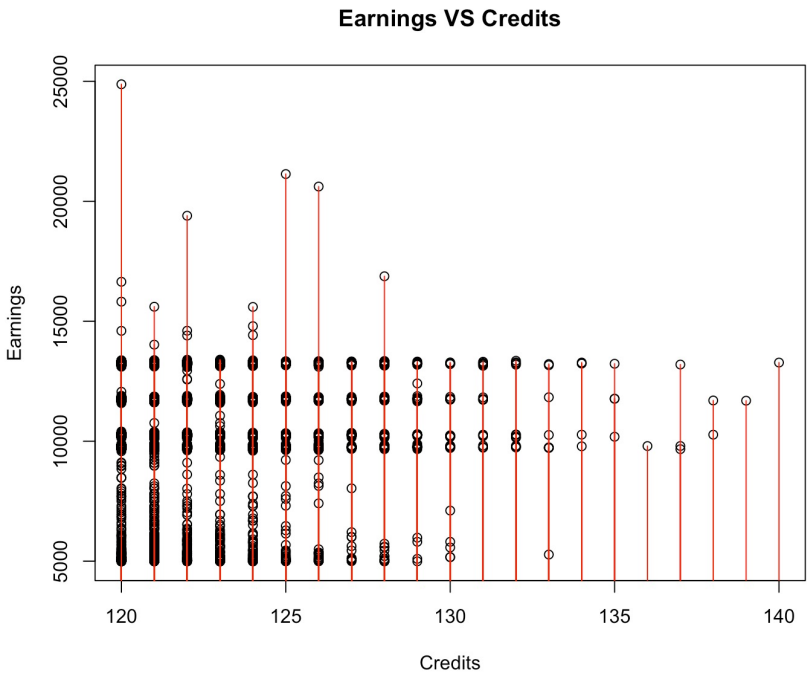


These two majors had the broadest earnings range showed on the boxplot.

Other majors exhibit a very distinct trend that relies heavily on professional connections.

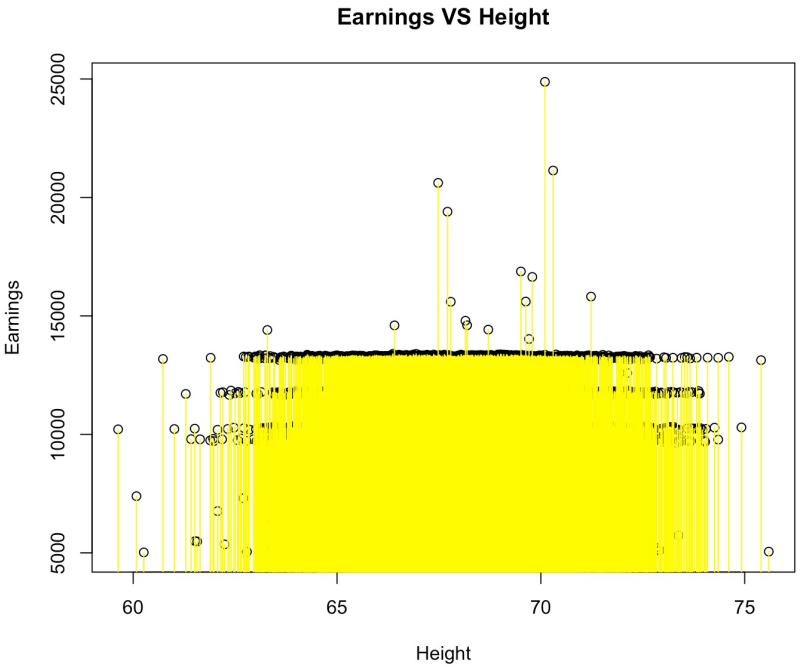
Business majors are split into two groups.

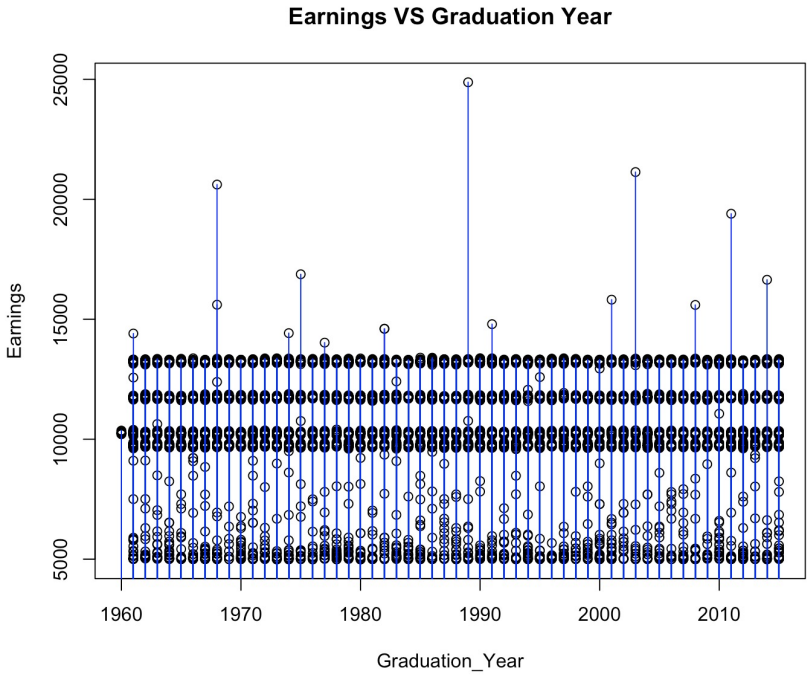


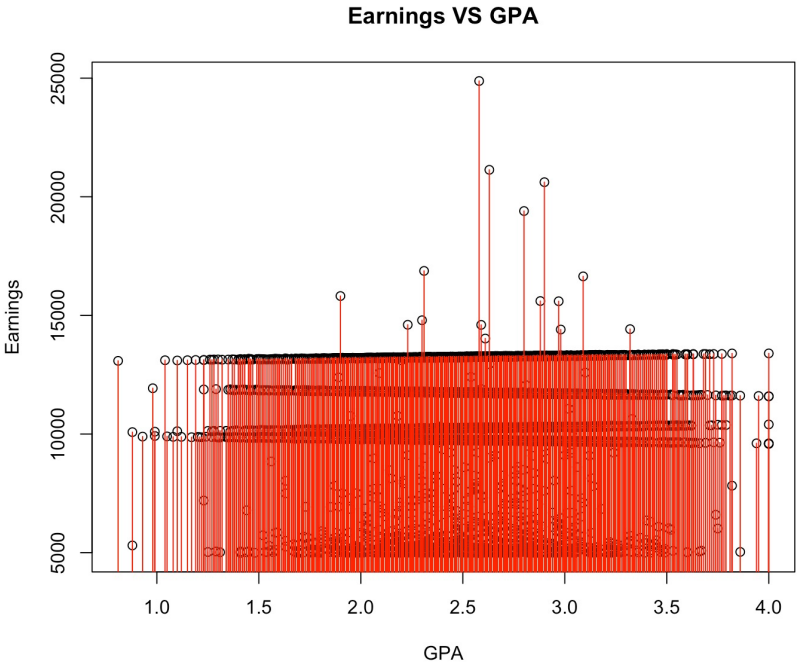


Other Plots

All of these plots illustrate roughly the same idea. Different majors will have different ranges of earnings and it's distinct since it is portrayed as straight lines within the scatterplots for whatever variable we plot against earnings.







Using Machine Learning

We will now use machine learning to try and predict the earnings given our respective training dataset.

We will pick the best model to use on our test data.

We have to be mindful about overfitting the data as well as using the appropriate model when crunching our final dataset.

Using Linear Regression w/ All Variables

```
linear_regression_model <-
lm(Earnings~Number_Of_Professional_Connections+Major+Graduation_Year+Height+Number_Of_Credits+Number_Of_Parking_Tickets, data =
Earnings_Train2021, x = TRUE, y = TRUE)

lm_predictions <-
predict(linear_regression_model, newdata =
Earnings_Train2021)

mse(lm_predictions,
Earnings_Train2021$Earnings)

cv.lm(linear_regression_model, m = 3)
```

Mean absolute error : 221.2745

Sample standard deviation : 11.60396

Mean squared error : 320053.3

Sample standard deviation : 156411.2

Root mean squared error : 549.8042

Sample standard deviation : 140.5093

Using Linear Regression w/ 2 Indicator Values

```
linear_regression_model <-
lm(Earnings~Number_Of_Professional_Connections+Major, data = Earnings_Train2021, x = TRUE, y = TRUE)

lm_predictions <-
predict(linear_regression_model, newdata = Earnings_Train2021)

mse(lm_predictions, Earnings_Train2021$Earnings)

cv.lm(linear_regression_model, m = 3)
```

```
Mean absolute error       :  221.0533
Sample standard deviation :  13.59883

Mean squared error        :  319288.8
Sample standard deviation :  140946.1

Root mean squared error   :  553.3968
Sample standard deviation :  120.3732
```

Using Random Forest Excluding GPA

```
random_forest <-
randomForest::randomForest(Earnings~Number_Of_Professional_Connections+Major+Graduation_Year+Height+Number_Of_Credits+Number_Of_Parking_Tickets, data = Earnings_Train2021)

training_predictions <- predict(random_forest, newdata = Earnings_Train2021)

mse(training_predictions, Earnings_Train2021$Earnings)
```

```
> mse(training_predictions, Earnings_Train2021$Earnings)
[1] 18478.7
```

Using Random Forest With Only 2 Indictor Values

```
random_forest <-
randomForest::randomForest(Earnings~Number_Of_Professional
_Connections+Major, data = Earnings_Train2021)

training_predictions <- predict(random_forest, newdata =
Earnings_Train2021)

mse(training_predictions, Earnings_Train2021$Earnings)
```

```
> mse(training_predictions, Earnings_Train2021$Earnings)
[1] 37634.2
```

Using SVM on All the Variables Except GPA

```
svm_fit <-
svm(Earnings~Number_Of_Professional_Connections+Major+Number_Of_Credits+Number_Of_Parking_Tickets, data =
Earnings_Train2021)

svm_training_predictions <- predict(svm_fit, newdata =
Earnings_Train2021)

mse(svm_training_predictions, Earnings_Train2021$Earnings)
```

```
> mse(svm_training_predictions, Earnings_Train2021$Earnings)
[1] 39731.76
```

Using SVM w/ 2 Indictor Values

```
svm_fit <-
svm(Earnings~Number_Of_Professional_Connections+Major, data
= Earnings_Train2021)

svm_training_predictions <- predict(svm_fit, newdata =
Earnings_Train2021)

mse(svm_training_predictions, Earnings_Train2021$Earnings)
```

```
> mse(svm_training_predictions, Earnings_Train2021$Earnings)
[1] 27107.18
```

The Model I Ended Up Using

- To my dislike, I ended up using the Random Forest model because it yielded the smallest mean squared model by a rather significant amount.
- The MSE came out to 18478.7 which is good in comparison to the others.
- However, more research and using a combination of other variables will likely yield better results with the other machine learning algorithms.
- All in all, more research will be needed to make this model perfect; however, for a quick prediction, this model is suffice.