DOKUZ EYLUL UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING

CME 4403
INTRODUCTION TO MACHINE LEARNING
SALARY PREDICTION

2019510012 Elif Dilara Akkuş
2019510094 Elif Özker
2019510108 Oğuz Kaan Şanlı

Assoc. Prof. Zerrin Işık

IZMIR
30.12.2023

1. Introduction

Project Description:

The main purpose of our project is to make salary predictions by using descriptive features in our dataset and different machine learning models and to make error rate calculations and observe how accurate these predictions are. Data was collected from people from different regions around the world in order to observe how values such as people's education status and age caused changes in their salaries.

Dataset Information:

This dataset consists of 6684 rows, 9 columns: Age, Gender, Education Level, Job Title, Years of Experience, Salary, Country, Race and Senior. This dataset is compiled from employment platforms and some surveys. It is useful to obtain information about labor market trends in different professions. It is a dataset that can be used and analyzed to obtain information about salaries and make informed decisions for business strategies. Salaries are represented in US Dollars. For Senior, it is represented as 0 if the employee is in a senior position, not 1.

For Education Level

0 : High School

1: Bachelor Degree

2: Master Degree

3 : Phd

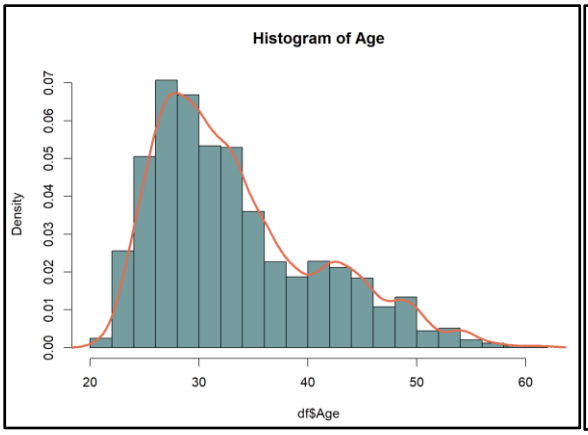
It is represented as.

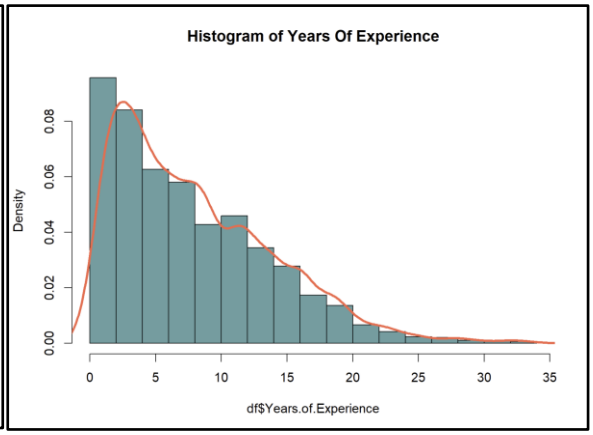
2. Data Exploration

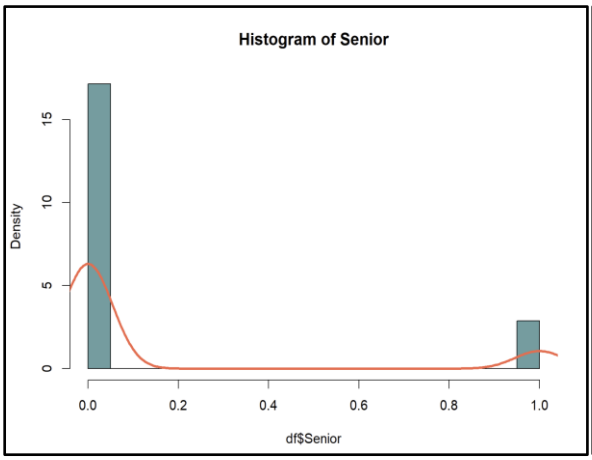
First, we summarized our raw data set and examined values such as minimum, maximum, mode, median and quarter for its features.

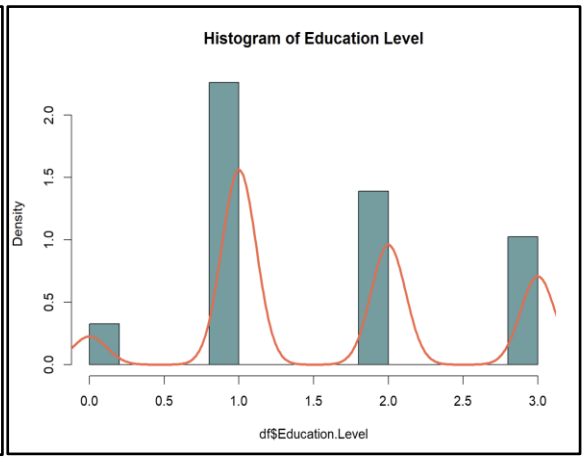
```
      Age            Gender          Education.Level  Job.Title          Years.of.Experience
 Min.   :21.00   Length:6684        Min.   :0.000   Length:6684        Min.   : 0.000
 1st Qu.:28.00   Class :character   1st Qu.:1.000   Class :character   1st Qu.: 3.000
 Median :32.00   Mode  :character   Median :1.000   Mode  :character   Median : 7.000
 Mean   :33.61                      Mean   :1.622                      Mean   : 8.078
 3rd Qu.:38.00                      3rd Qu.:2.000                      3rd Qu.:12.000
 Max.   :62.00                      Max.   :3.000                      Max.   :34.000
     Salary         Country              Race              Senior
 Min.   :   350   Length:6684        Length:6684        Min.   :0.0000
 1st Qu.: 70000   Class :character   Class :character   1st Qu.:0.0000
 Median :115000   Mode  :character   Mode  :character   Median :0.0000
 Mean   :115307                                         Mean   :0.1435
 3rd Qu.:160000                                         3rd Qu.:0.0000
 Max.   :250000                                         Max.   :1.0000
```

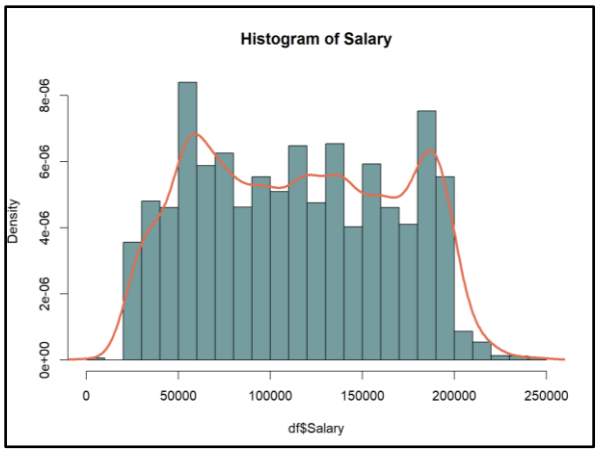
We have 5 continuous and 4 categorical feature. In order to interpret the data we have more easily, we drew a histogram for the continuous features.











The data shows that most people in our group are younger, and there are not many older individuals. Many folks seem to be in the early part of their careers. When we look at years of experience, it seems more even. But there are more people with less experience compared to those who are experts. So, most of the folks in our group are still learning and gaining experience in their jobs.

3. Data Preprocessing

Handling missing values:

Handling missing values is crucial in data analysis and machine learning, as many algorithms cannot work with missing data.The data set was examined to identify and evaluate the presence of missing values.As a result, we see that there are no missing values in any column.

Detect outliers:

Outliers are values that differ significantly from the overall data set or diverge significantly from other data points.Outliers can occur for a variety of reasons, such as measurement errors, data entry errors, or natural changes in the data. These outliers can affect the performance of machine learning models and make it difficult to obtain accurate results.

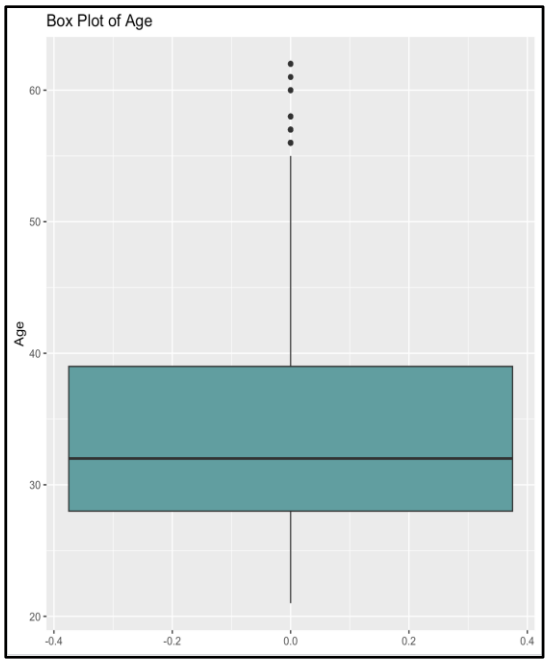


Figure 2.1

In the box plot graph created for the Age variable, we see the median, interquartile range and potential outliers. 6 outliers were detected in the graph. Outliers generally indicate that the ages of younger or older individuals differ significantly from the overall distribution.

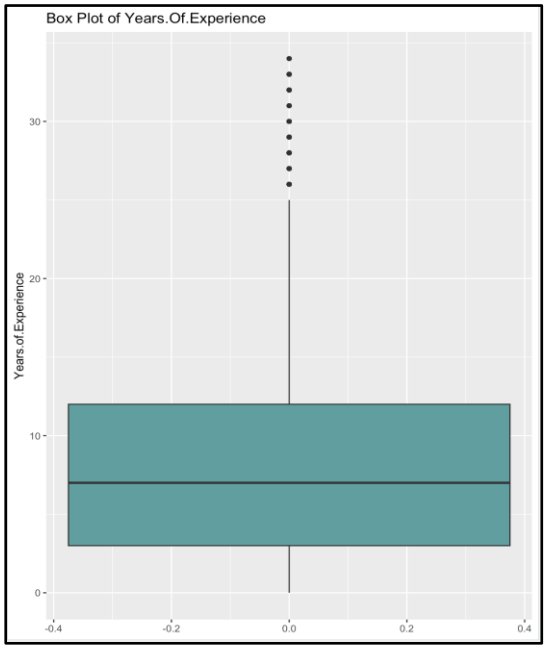


Figure 2.2

In the box plot graph created for the Years.of.Experience variable, we observe the distribution of years of experience and potential outliers.9 outliers were detected in the graph.

These values often represent individuals with years of experience that are markedly different from other employees.

Encoding categorical variables:

It can be used to convert ordinal or nominal categorical variables in a data set into numerical values. Getting categorical data into numerical format is crucial because machine learning algorithms perform better on numerical data.One-hot encoding process, which is one of some commonly used methods, was used in this coding process. One-hot encoding was applied to the categorical columns Gender, Education.Level, Job.Title, Country and Race.We can see it as an example in Figure 2.1. After encoding, our column number increased to 154.

Gender_Female	Gender_Male	Education.Level_0	Education.Level_1
0	1	0	1

Figure 2.3

Scaling numerical features:

In many data science applications, including statistical analysis, machine learning, and data mining, scaling or normalizing numerical features is a crucial step. These operations, which normalize or standardize value ranges between numerical features, can enhance model performance and analysis accuracy.In the dataset, Age and Years of Experience columns are standardized.The standardization process transforms the values in the specified columns to have a mean of 0 and a standard deviation of 1. This is known as Z-score standardization.

Numeric properties such as "Age" and "Years.of.Experience" can often have different ranges of values compared to other properties, especially when used in regression analysis or similar models. This may cause the model to give more weight or influence to certain features than others.

	Age	Years.of.Experience
1	32	5.0
2	28	3.0
3	45	15.0
4	36	7.0
5	52	20.0

Before scaling

Figure 2.4

	Age	Years.of.Experience
1	-0.2577865185	-0.53527068
2	-0.7730592801	-0.85611524
3	1.4168499566	1.06895210
4	0.2574862430	-0.21442613
5	2.3185772893	1.87106350

After scaling

Figure 2.5

4. Feature Selection

Feature selection is a crucial step in the process of building machine learning models, and it involves choosing a subset of relevant features from the original set of features in the dataset. We performed two different feature selections.

Boruta:

Boruta is a feature selection algorithm designed for identifying all relevant variables in a dataset. The main purpose of Boruta is to address the challenge of selecting the most important features from a potentially large set of candidate features.

Random Forests: Boruta utilizes random forests to assess the importance of each feature in the dataset. A random forest is an ensemble learning method that builds multiple decision trees and combines their predictions.

Shadow Features: Boruta creates shadow features by permuting the values of the original features. These shadow features represent a null hypothesis that there is no relationship between the features and the target variable.

Comparison with Shadow Features: Boruta then compares the importance of each original feature with the importance of its corresponding shadow feature. If a feature's importance is significantly higher than that of its shadow, it is considered relevant.

Iterative Process: The algorithm operates in an iterative manner, repeatedly assessing the importance of features and their shadow counterparts until all features are either deemed relevant or discarded.

Final Feature Set: At the end of the process, Boruta provides a set of relevant features that can be used for model training.

Boruta performed 99 iterations in 9.845265 mins.

58 attributes confirmed important: Age, Education.Level_0, Education.Level_1,

Education.Level_2, Education.Level_3 and 53 more;

91 attributes confirmed unimportant: Country_Australia, Country_Canada,

Country_China, Country_UK, Country_USA and 86 more;

4 tentative attributes left: Job.Title_Business Development Associate,

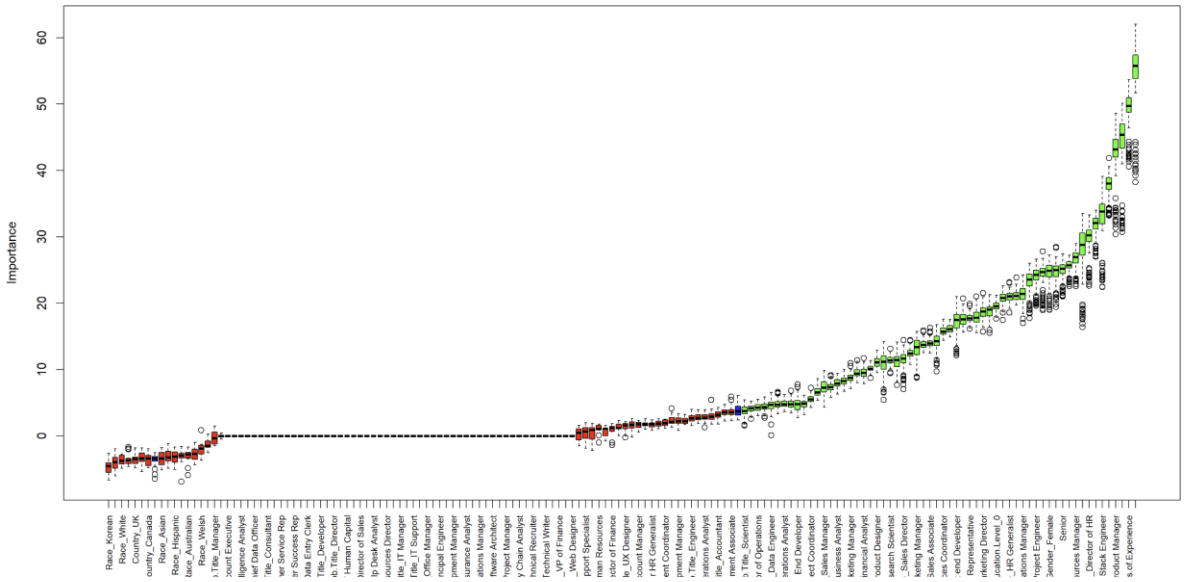
Job.Title_Financial Advisor, Job.Title_Recruiter, Job.Title_Scientist;

	meanImp	medianImp	minImp	maxImp
Age	**37.7366721**	**38.0706222**	**33.19552403**	**41.8777236**
Years.of.Experience	**49.0453391**	**49.7036681**	**40.58065362**	**53.6663873**
Senior	**24.9024639**	**25.1845990**	**20.99878187**	**27.4431387**
Gender_Female	**24.3881772**	**24.8952493**	**19.00501156**	**27.3169929**
Gender_Male	**24.2454638**	**24.7285618**	**18.96896766**	**27.8220887**
Education.Level_0	**19.5291857**	**19.5511192**	**17.69833834**	**21.1972310**
Education.Level_1	**25.6278411**	**25.7373675**	**22.60664709**	**27.2679559**
Education.Level_2	**17.5963327**	**17.5710398**	**15.64984481**	**20.7329168**
Education.Level_3	**17.9137262**	**17.8438038**	**15.56198862**	**21.0152896**

Boruta gives importance scores for each feature. Years of Experience appears to be the most important feature.

	normHits	decision
Age	**1.00000000**	**Confirmed**
Years.of.Experience	**1.00000000**	**Confirmed**
Senior	**1.00000000**	**Confirmed**
Gender_Female	**1.00000000**	**Confirmed**
Gender_Male	**1.00000000**	**Confirmed**
Education.Level_0	**1.00000000**	**Confirmed**
Education.Level_1	**1.00000000**	**Confirmed**
Education.Level_2	**1.00000000**	**Confirmed**
Education.Level_3	**1.00000000**	**Confirmed**
Job.Title_Account Executive	**0.00000000**	**Rejected**
Job.Title_Account Manager	**0.00000000**	**Rejected**

Features with a NormHits value of 1 are considered to have a significant impact on the target variable and are verified as significant.



Blue boxplots correspond to minimal, average and maximum Z score of a shadow attribute. Red, yellow and green boxplots represent Z scores of rejected, tentative and confirmed attributes respectively. After we performed boruta there are only 60 columns left in our dataset

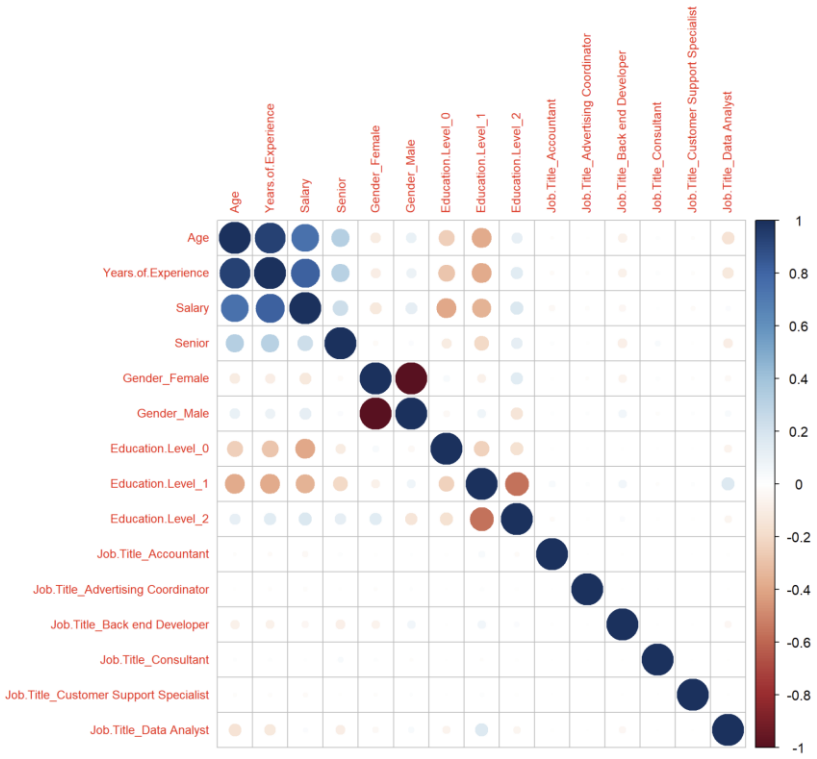
Correlation Matrix:

A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table represents the correlation between two variables.

1 indicates a perfect positive correlation: As one variable increases, the other also increases proportionally.

-1 indicates a perfect negative correlation: As one variable increases, the other decreases proportionally.

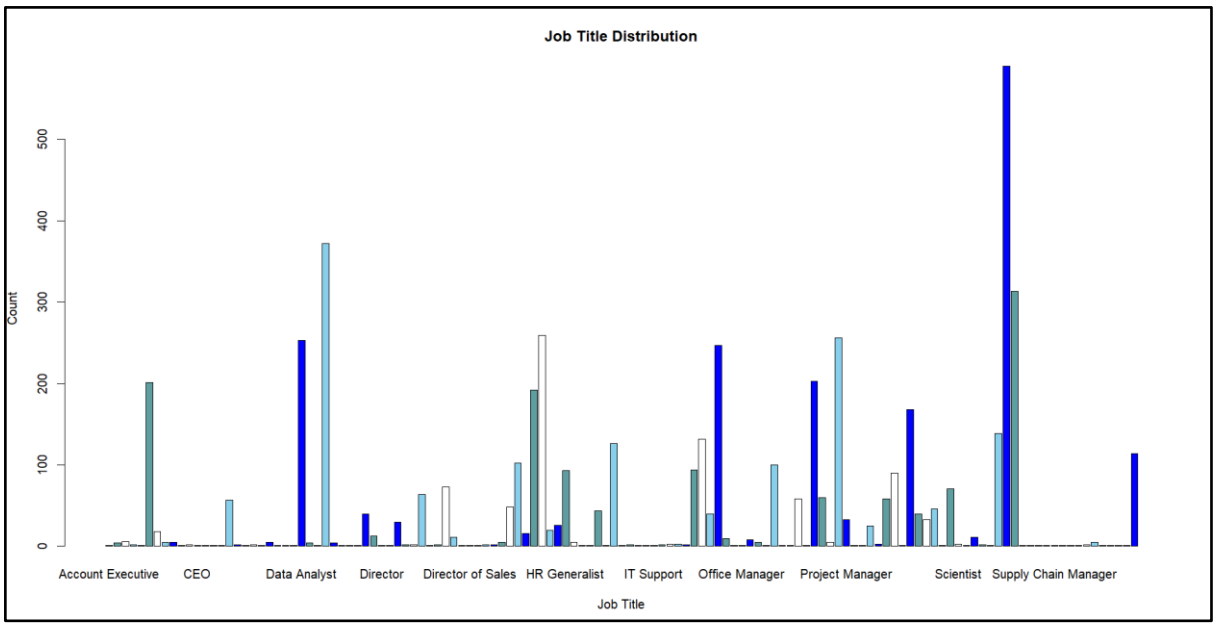
0 indicates no correlation: Changes in one variable do not predict changes in the other.



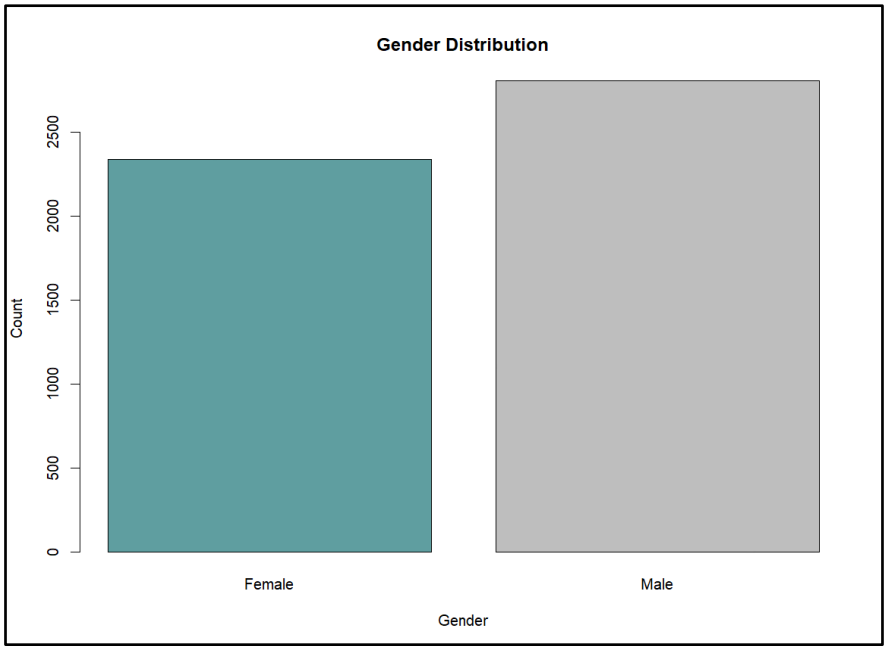
We observed that the Age and Years of Experience columns correlation is 0.94 which is pretty high value therefore one these two features would be enough for training models. We removed the Age column from the dataset.

5.Visualizations

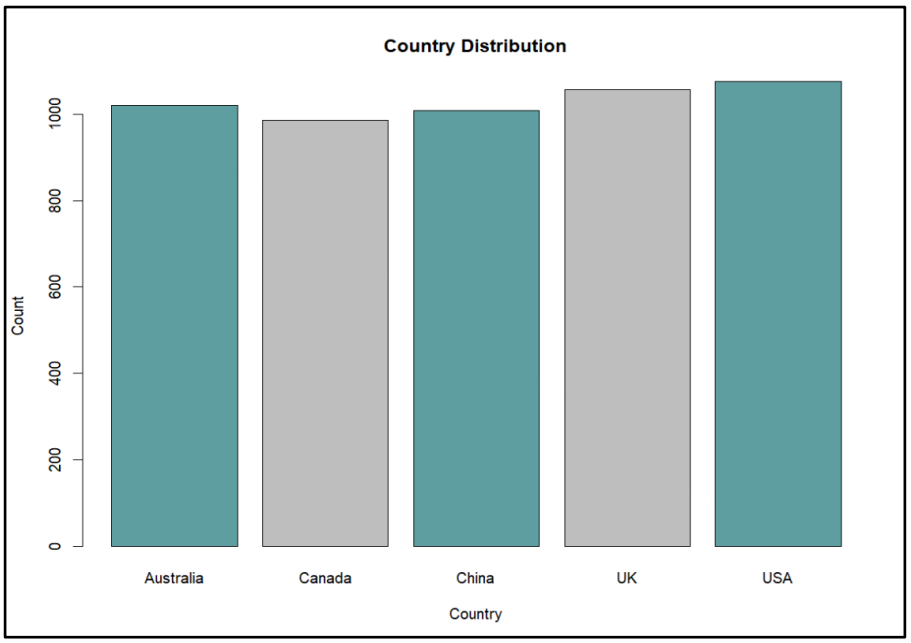
Job Title Distribution



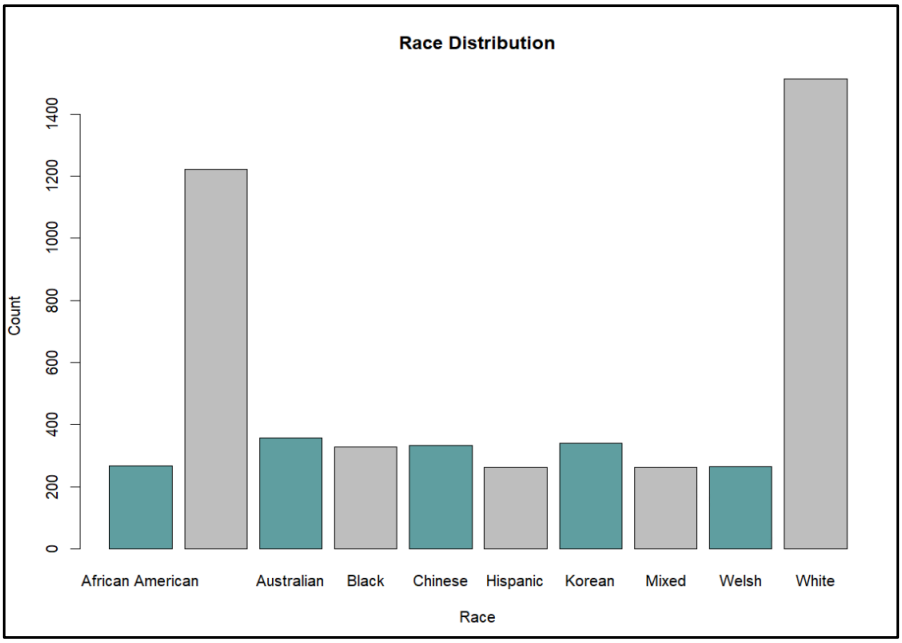
Gender Distribution



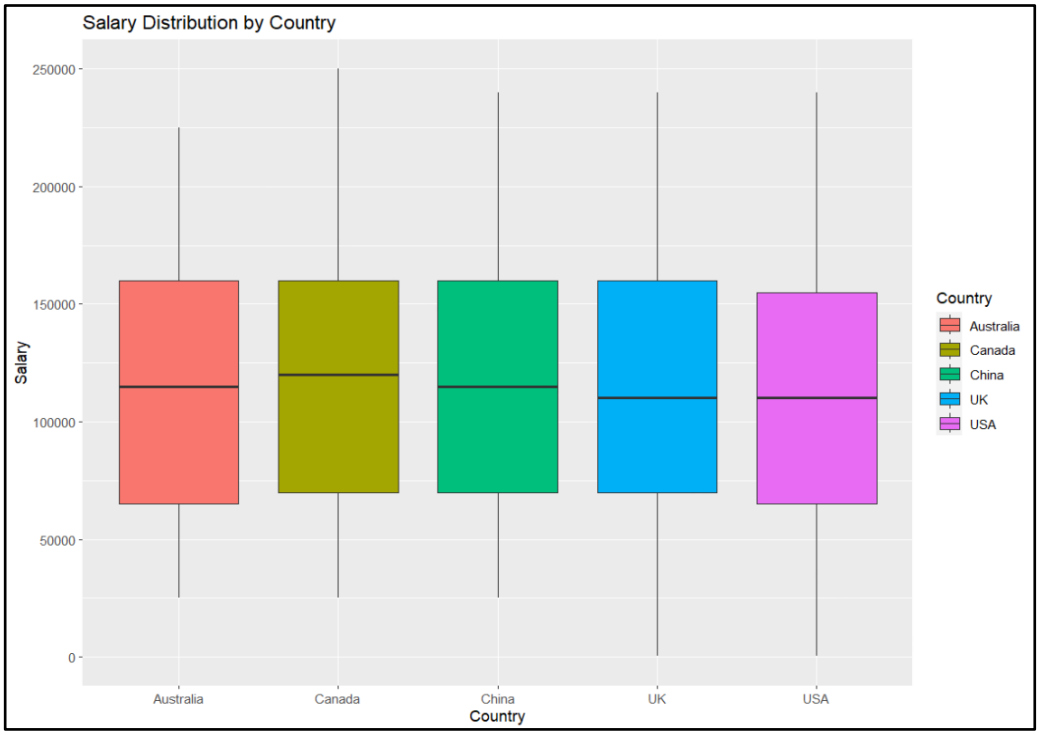
Country Distribution



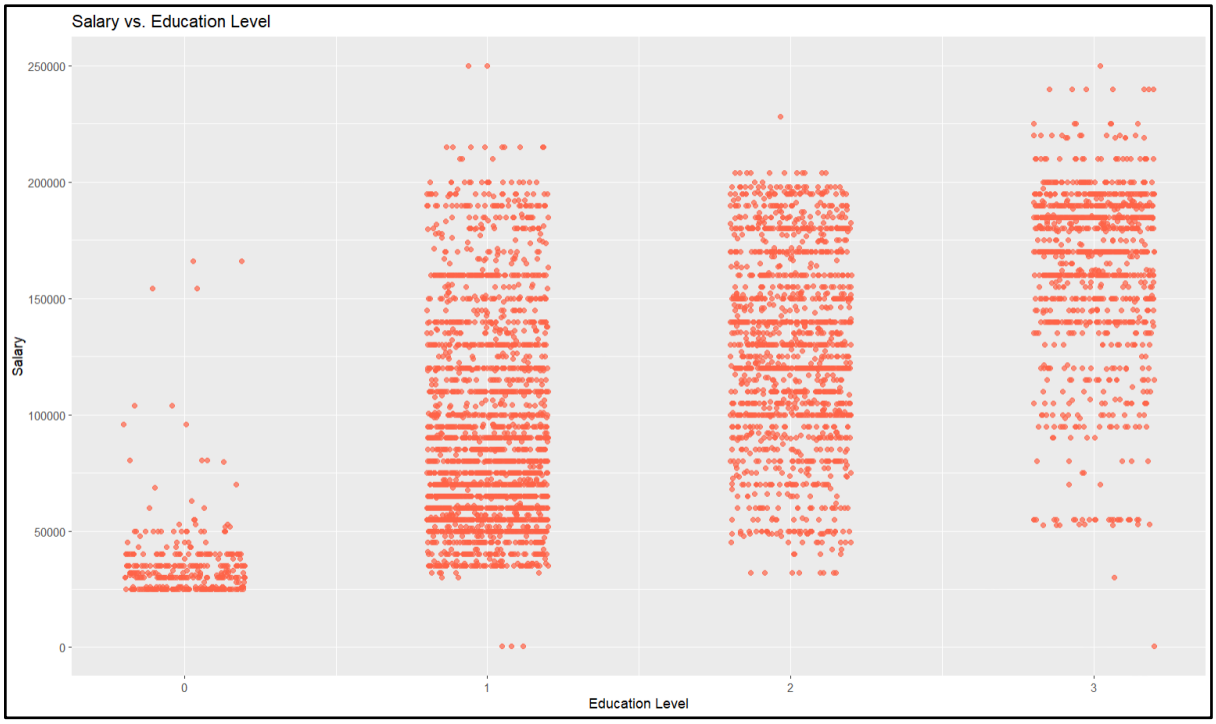
Race Distribution



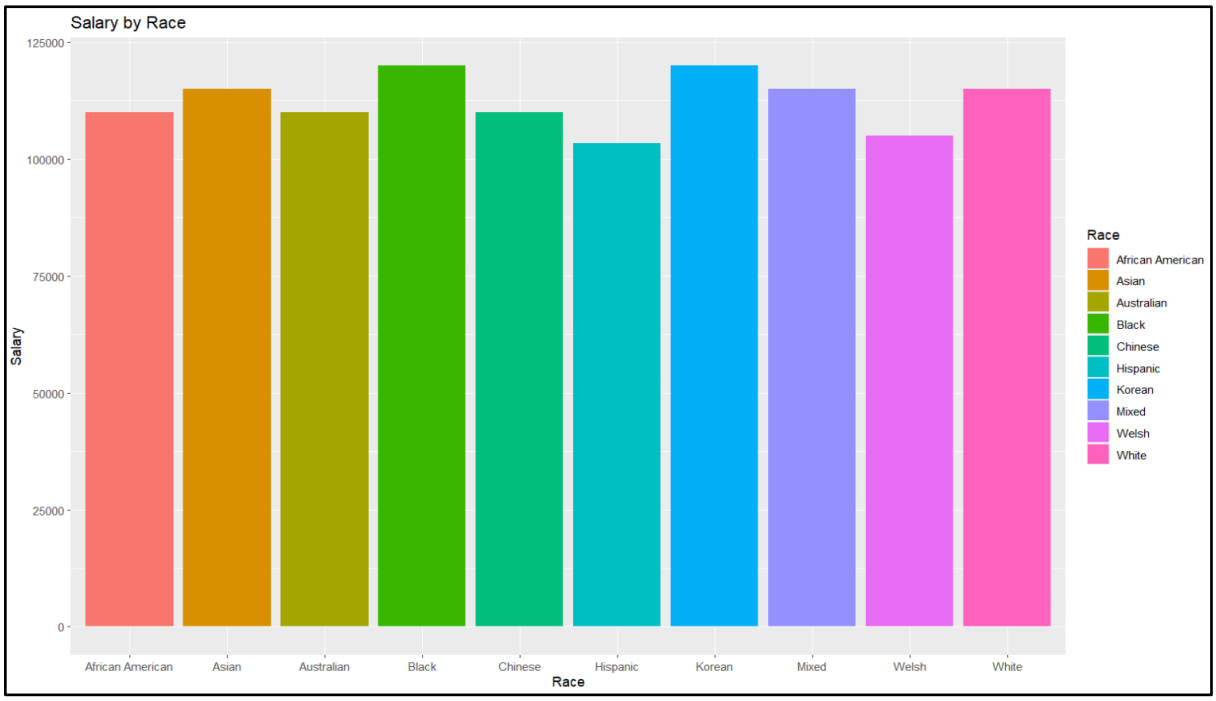
Salary Distribution by Country



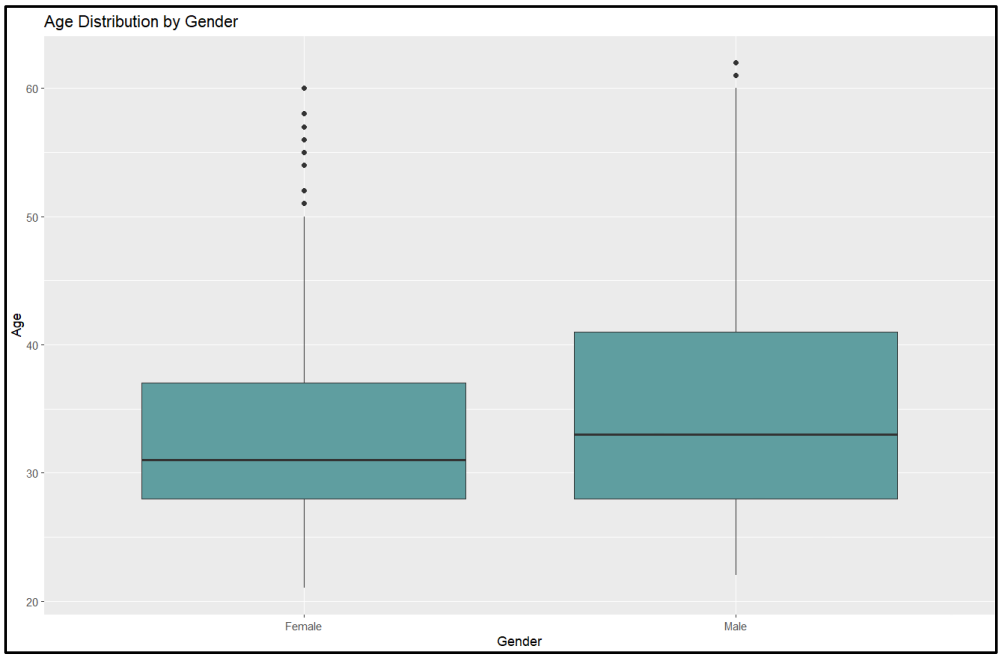
Salary Distribution by Education Level



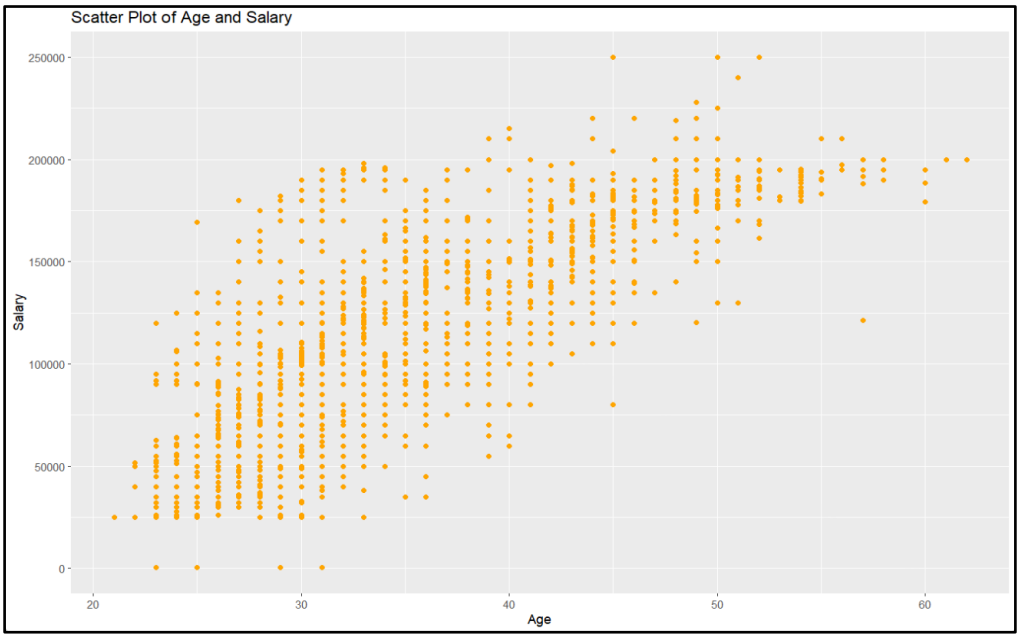
Salary Distribution By Race



Age Distribution by Gender



Salary Distribution by Age



6. Model Training and Evaluation

Splitting of the Data Set:

To build a model and assess its performance, the data set is typically divided into training, validation, and test sets.The data set that the model uses to learn is called the training set. During the model's development phase, the validation set is used to evaluate the model's generalizability and adjust hyperparameters.The test set is an independent data set allocated to evaluate the real-world performance of the model.

80% of the data set is divided into the training set, 10% into the validation set, and 10% into the testing set.

Models: Single Linear Regression, Multi-Linear Regression, Decision Tree, Random Forest, Gradient Boosting, Neural Network.

Evaluation Metrics:

R-square (Coefficient of Determination) is a measure that expresses the proportion of the variance of the dependent variable explained by the independent variables of a regression model. That is, the R-squared value shows how well the model can be explained by the independent variables. In other words, r-squared shows how well the data fit the regression model (the goodness of fit).

SSE refers to the sum of squares of the differences between the values predicted by the regression model and the actual observations. Regression analysis aims to minimize SSE; The smaller the error, the better the predictive power of the regression.

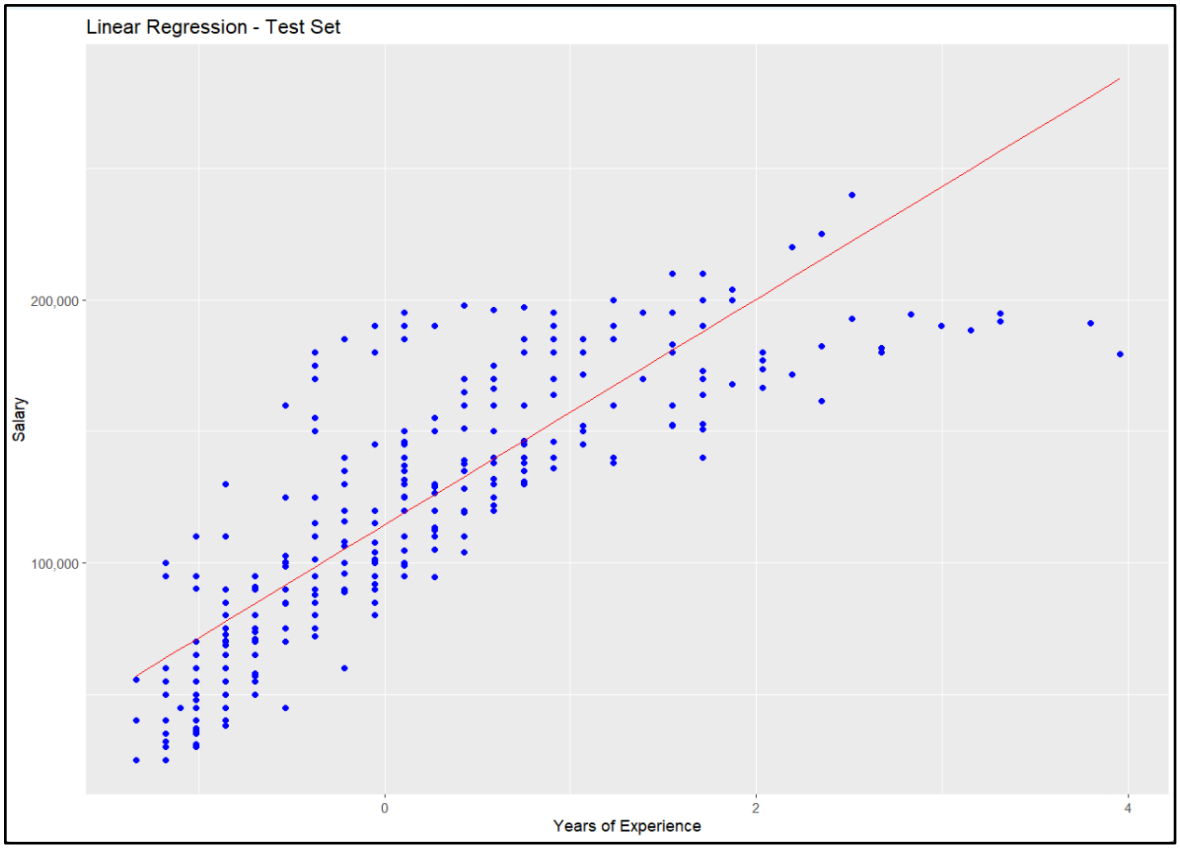
MSE is a measure of the mean square error obtained by dividing the SSE by the number of observations. MSE refers to the mean square error of the regression model. A smaller MSE is preferred because it indicates that your data points are closely distributed around the central moment (mean).

RMSE is the square root of MSE and is generally a more understandable metric. It measures the average difference between values predicted by a model and actual values. It provides an estimate of how well the model can predict the target value (accuracy).

MAE represents the average of the absolute differences between the values predicted by the regression model and actual observations.

Single Linear Regression:

Simple linear regression is a statistical method employed to estimate and quantify the association between two numerical variables. The degree of correlation between two variables indicates the strength of their relationship. This linear regression model was created to analyze the relationship between the "Years of Experience" variable and the "Salary" variable.
"Runtime: 0.00629210472106934 seconds"



While the values shown with a blue dot in the Salary-Years of Experience table on the test set above are the real values in our dataset, we see the prediction of our model with the red line. We can say that as the Years of Experience value increases, deviations increase and the accuracy of the model's outputs decreases. However, it can be commented that our linear regression model is generally compatible with our dataset.

Coefficient of Determination (R^2) = 0.6653701 so the R-square value of the model shows that the variable years of experience has a success rate of 66.54%.

Test Error: 0.6936965
Train Error: 0.6653701

Sum of Squared Residuals (SSE): 3.764944e+12
Mean Squared Error (MSE): 920299186
Root Mean Squared Error (RMSE): 30336.43
Mean Absolute Error (MAE): 24443.35

In conclusion, while the model's efficiency is good, the relatively high errors (MSE, RMSE, MAE) suggest that there is room for improvement in predictive accuracy. In order to enhance the model performance we implemented cross validation.

Single Linear Regression Cross Validation Version:

Cross-validation is a crucial technique for assessing the model's performance and comprehending its capacity for generalization. This approach improves comprehension of how the model can be applied to actual data, particularly when dealing with small data sets.

Linear Regression 5148 samples 1 predictor

Resampling: Cross-Validated (5 fold, repeated 5 times)
Summary of sample sizes: 4119, 4118, 4118, 4119, 4118, 4118, ...
Resampling results:

RMSE	Rsquared	MAE
30327.76	0.6664002	24321.8

Test Error: 0.6937824
Train Error: 0.6653579

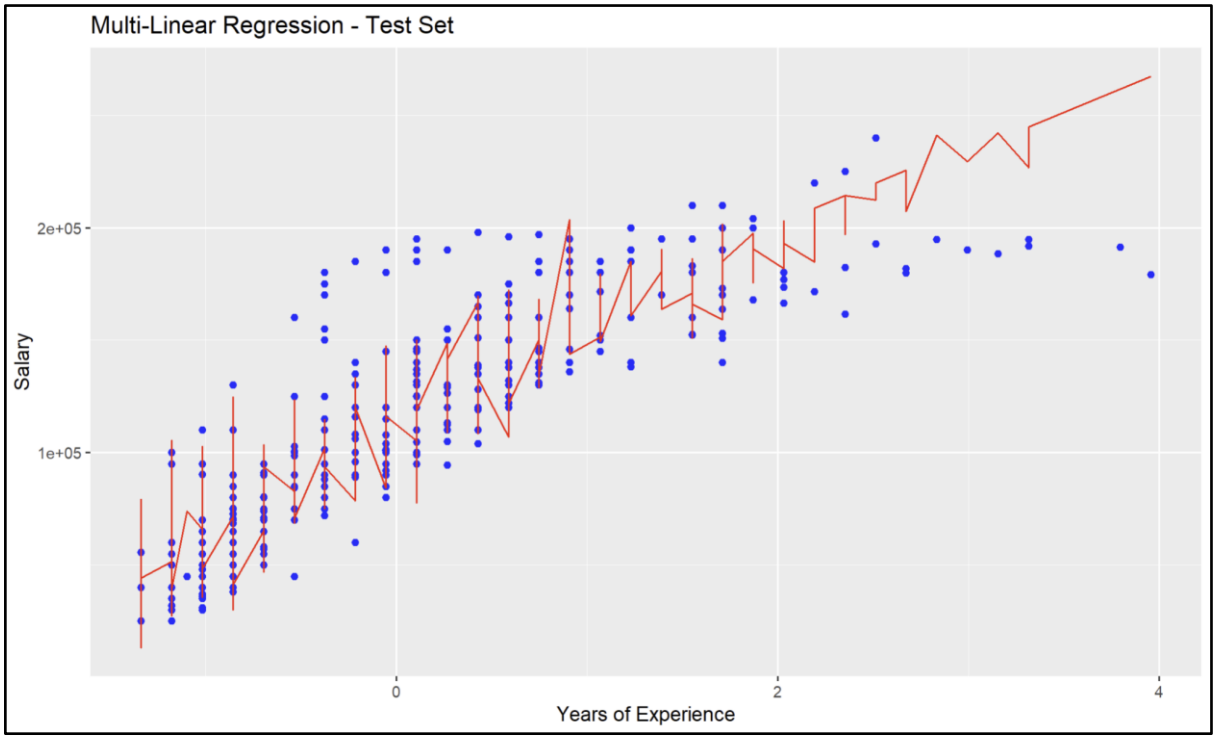
Sum of Squared Residuals (SSE): 4.73215e+12
Mean Squared Error (MSE): 919220982
Root Mean Squared Error (RMSE): 30318.66
Coefficient of Determination (R^2): 0.6659054
Mean Absolute Error (MAE): 24314.04

The RMSE and MAE have slightly decreased,while R-Squared has increased which suggests a small improvement in the model's accuracy. However the R-squared value provides additional insight into the proportion of variance explained by the model, and a value of 0.6664 is generally considered decent. Therefore we decided to implement Multi-Linear Regression and observe the difference between one predictor and multiple predictors.

Multi-Linear Regression:

Multi-linear regression is a statistical method used to model the relationship between multiple independent variables and a dependent variable. It is an extension of simple linear regression.In multi-linear regression, the goal is to fit a linear equation to the observed data that can predict the dependent variable based on the values of multiple independent variables.The effect of all independent variables in the model is evaluated on the dependent variable, Salary.

"Runtime: 0.0386838912963867 seconds"



It is possible to observe that the predictions on the Test are much more accurate with the Multi-Linear Regression model.

Test Error: 0.804075
Train Error: 0.7913035

Sum of Squared Residuals (SSE): 2.348059e+12
Mean Squared Error (MSE): 573957144
Root Mean Squared Error (RMSE): 23957.4
Coefficient of Determination (R^2): 0.7913035
Mean Absolute Error (MAE): 17777.7

The multilinear regression model appears to perform better than the linear regression model based on these metrics. It has lower errors (MSE, RMSE, MAE) and a comparable R^2 value. This suggests that the inclusion of multiple predictors has improved the model's ability to explain and predict the target variable.

Multi-Linear Regression Cross Validation Version:

Linear Regression 5148 samples 58 predictor
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 4119, 4118, 4118, 4119, 4118

Test Error: 0.806391

Sum of Squared Residuals (SSE): 2.94591e+12
Mean Squared Error (MSE): 572243552
Root Mean Squared Error (RMSE): 23921.61
Coefficient of Determination (R^2): 0.7920157
Mean Absolute Error (MAE): 17701.88

We found that the multiple linear regression model consistently outperformed the linear regression model on a variety of metrics, including cross-validated metrics and test error. This shows that adding multiple predictors and using cross-validation improves the model's ability to generalize and make accurate predictions.

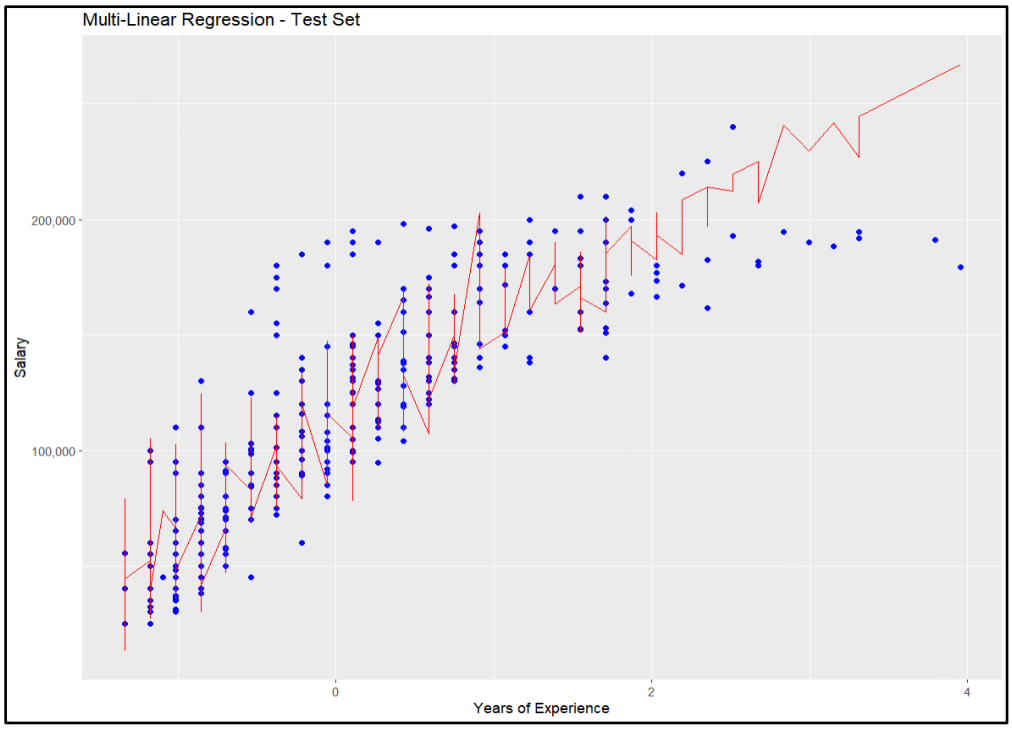
In order to observe if the Multi-Linear Regression model can be enhanced by optimizing hyperparameters, we performed parameter tuning to our model.

Multi-Linear Regression Parameter Tuning:

Hyperparameter tuning of the model takes place on alpha and lambda values. These values control the complexity of the model and the degree of regularization.

While performing hyperparameter optimization for the regularized linear regression model, we varied the alpha and lambda values. We performed five-fold cross-validation using various alpha and lambda combinations and observed the performance of each combination. For example, the five-fold cross-validation results for the combination alpha=0.8 and lambda=0.1 are as follows:

```
+ Fold1: alpha=0.0, lambda=0.1
- Fold1: alpha=0.0, lambda=0.1
+ Fold1: alpha=0.1, lambda=0.1
- Fold1: alpha=0.1, lambda=0.1
+ Fold1: alpha=0.2, lambda=0.1
.                .                .
.                .                .
.                .                .
+ Fold5: alpha=0.8, lambda=0.1
- Fold5: alpha=0.8, lambda=0.1
+ Fold5: alpha=0.9, lambda=0.1
- Fold5: alpha=0.9, lambda=0.1
+ Fold5: alpha=1.0, lambda=0.1
- Fold5: alpha=1.0, lambda=0.1

Aggregating results
Selecting tuning parameters
Fitting alpha = 0.8, lambda = 0.1 on full training set
```



RMSE: 234433044055459
R-squared: 0.804229473945893
Sum of Squared Residuals (SSE): 2.348457e+12
Mean Squared Error (MSE): 574054393
Mean Absolute Error (MAE): 17804.89

We see that parameter tuning improves the performance of the model, especially in terms of RMSE, R-squared and MAE. These improvements show us that the selected hyperparameters contribute to better prediction accuracy and a more effective model.

PREVENT OVERFITTING

Ridge Regression:
Ridge Regression, also known as Tikhonov regularization or L2 regularization, is a linear regression technique that introduces a regularization term to the traditional linear regression objective function. The purpose of ridge regression is to address the overfitting issue that can arise when the number of features (predictors) is large compared to the number of observations in the dataset.

We used glmnet package for our Ridge Regression.

"Runtime: 2.13344502449036 seconds"

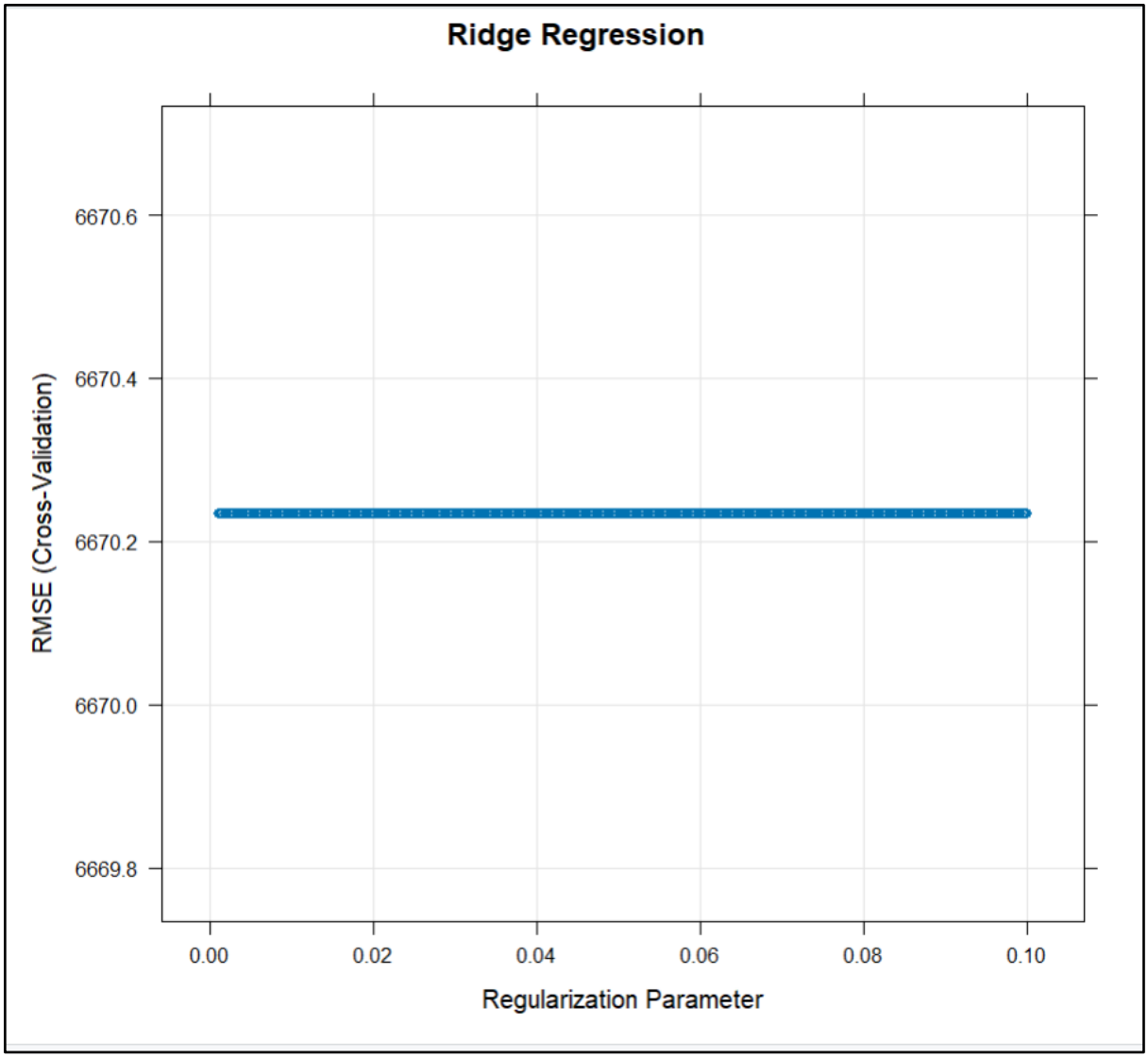
Ridge_model glmnet 4091 samples 59 predictor

RMSE = 6670.234

Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 3272, 3274, 3272, 3273, 3273
Resampling results across tuning parameters:

lambda	RMSE	Rsquared	MAE
0.0010	6670.234	0.9876128	5116.734
0.0012	6670.234	0.9876128	5116.734
0.0014	6670.234	0.9876128	5116.734
0.0016	6670.234	0.9876128	5116.734
0.0018	6670.234	0.9876128	5116.734
0.0020	6670.234	0.9876128	5116.734
0.0022	6670.234	0.9876128	5116.734
0.0024	6670.234	0.9876128	5116.734
0.0026	6670.234	0.9876128	5116.734
0.0028	6670.234	0.9876128	5116.734
0.0030	6670.234	0.9876128	5116.734
0.0032	6670.234	0.9876128	5116.734
.	.	.	.
.	.	.	.
.	.	.	.
0.0506	6670.234	0.9876128	5116.734
0.0508	6670.234	0.9876128	5116.734

Cross-validation results show that the Ridge regression model is insensitive to the lambda value. This may actually show us that after a certain level, editing no longer improves model performance.
Sum of Squared Residuals (SSE): 178811129876
Mean Squared Error (MSE): 43708416



The RMSE value of the model seems to be fixed at 6670.234. This is a metric that measures how well the model's predictions compare to actual values, with a lower RMSE indicating better model performance.

In Conclusion, there is no significant change in performance with the change of lambda value and may indicate that the model does not need regularization after a certain level.

Decision-Tree:

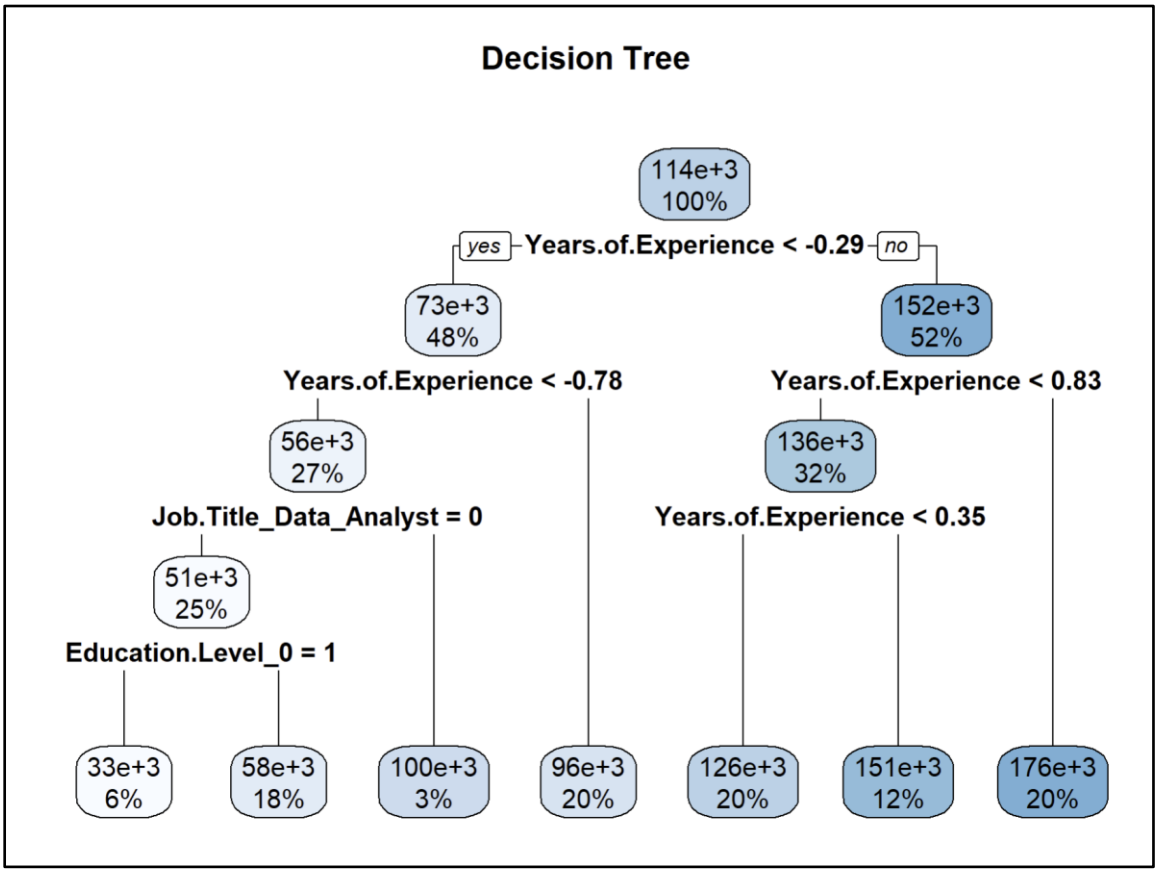
A decision tree is a popular machine learning model that is widely used for both classification and regression tasks. We performed this model and our training runtime was:

"Runtime: 0.186091899871826 seconds"

Results:

Test R2: 0.7586603 Train R2: 0.7472367 Validation R2: 0.7465533
Sum of Squared Residuals (SSE): 2843857e+12
Mean Squared Error (MSE): 695149518
Root Mean Squared Error (RMSE): 26365.69
Mean Absolute Error (MAE): 20120.9

Looking at the metrics, it is possible to say that we achieved a reasonable result. However, the model needs improvements to increase its performance.



In the Decision Tree table, branching started from the Years of Experience value as the descriptive feature with the highest importance. When we look at the table, we can see that the weight is on people whose Years of Experience value is greater than -0.29. As this value increases, the Salary of these people also increases. Later values continue to branch out based on the person's professional position and education level. While there are 3 different branches with the highest prediction probability with a rate of 20%, the lowest probability with 3% is the branch with Job Title Data Analyst with Years of Experience less than -0.78.

Cross Validation and Parameter Tuning for Decision Tree:

We applied 5-fold cross validation and the model was trained with different values for the relevant parameter, the complexity parameter, and its performance was evaluated. Starting with 0.001 cp, the cp values were changed in 0.01 increments. The model with the smallest RMSE was selected, and the cp value of the selected optimal model was determined as 0.001. With cross validation, we see a significant performance increase and optimization in our model.

CART 4091 samples58 predictor

No pre-processing

Resampling: Cross-Validated (5 fold)

Summary of sample sizes: 3272, 3274, 3272, 3273, 3273

Resampling results across tuning parameters:

cp	RMSE	Rsquared	MAE
0.001	20833.13	0.8423839	15314.06
0.011	26657.84	0.7419158	20577.79
0.021	28466.43	0.7057467	22330.80
0.031	28997.28	0.6947212	22873.72

0.041 28997.28 0.6947212 22873.72

0.051 28997.28 0.6947212 22873.72

0.061 28997.28 0.6947212 22873.72

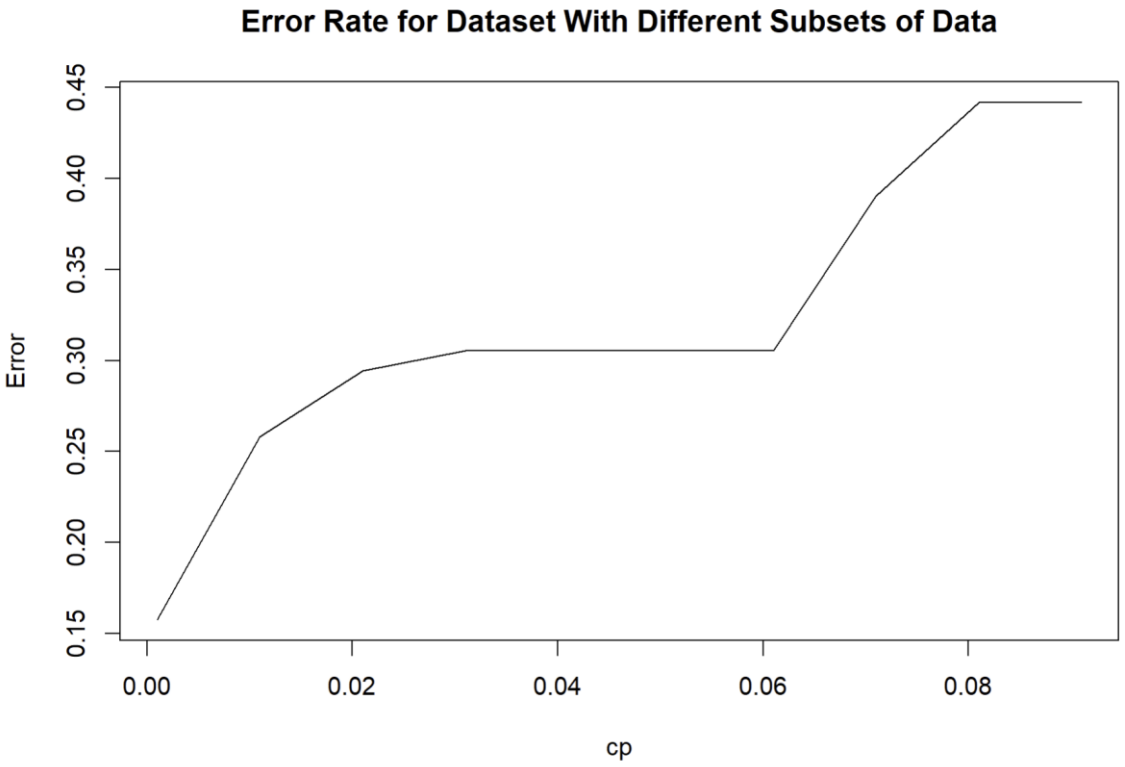
0.071 32746.28 0.6097754 26128.15

0.081 34865.06 0.5582185 28573.25

0.091 34865.06 0.5582185 28573.25

RMSE was used to select the optimal model using the smallest value.

The final value used for the model was cp = 0.001.



With the increase in cp value, the decrease in R-Square and the increase in RMSE, MAE and Error are clearly seen. For this reason, we can say that we obtained the most optimized result at a low cp value and took cp = 0.001 as the best model.

Error Metrics of the best model:

Test R2: 0.8721248 Train R2: 0.8570478

Sum of Squared Residuals (SSE): 1.608365e+12

Mean Squared Error (MSE): 393147084

Root Mean Squared Error (RMSE): 19827.94

Mean Absolute Error (MAE): 14724.77

Random Forest:

"Runtime: 13.198529958725 seconds"

We see that the training time of the model is quite long, approximately 13.2 seconds. This sho
us that the Random Forest algorithm is a slightly more computationally intensive and compl
model.

Test R2: 0.9292307 Train R2: 0.9246744 Validation R2: 0.9148662

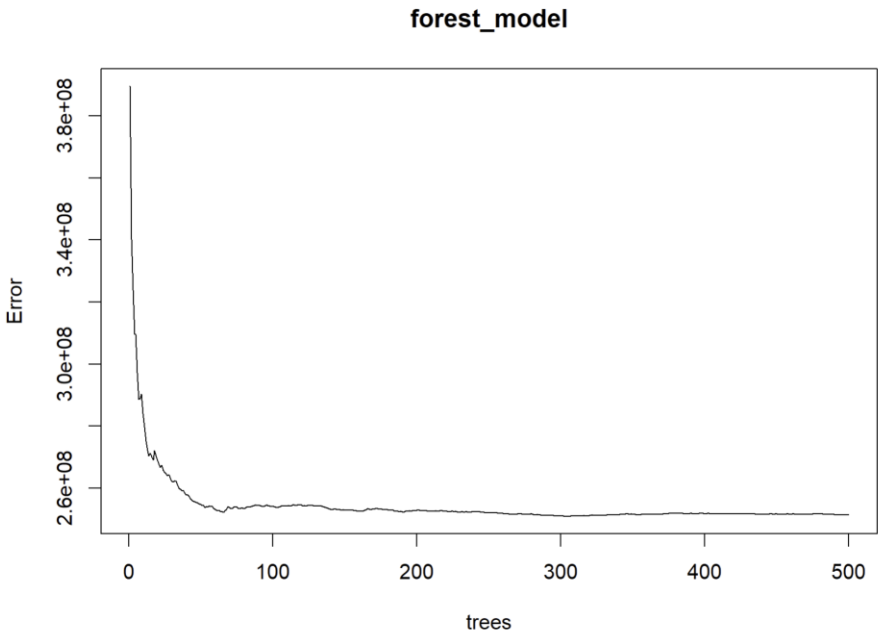
Training Mean Squared Error: 210676221

Validation Mean Squared Error: 233748776

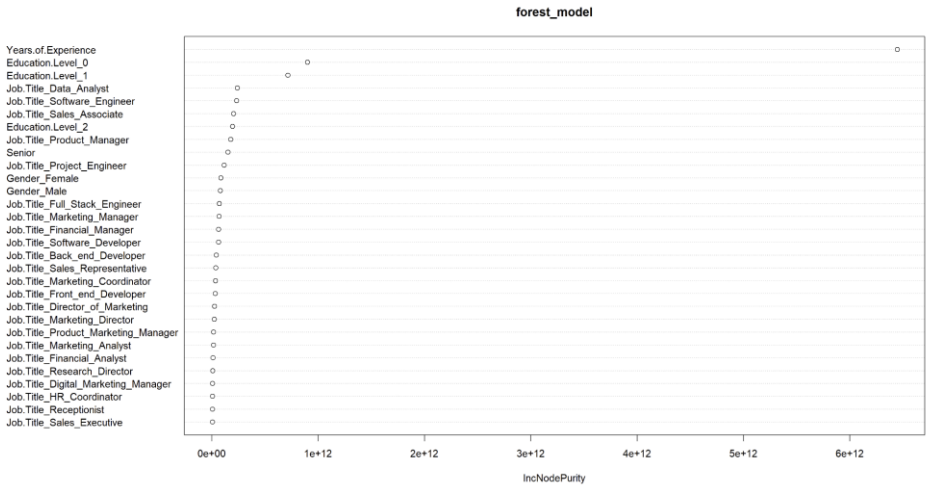
Test Mean Squared Error: 201880952

High R^2 values show us that the model has a high success in explaining the dependent variable.

As a result, our Random Forest model has high accuracy and low error rates.



The error value of the model drops dramatically at the beginning. Afterwards, we can say that the decrease in the error continues with a decreasing acceleration for a while and continues at a constant rate after tree 300.

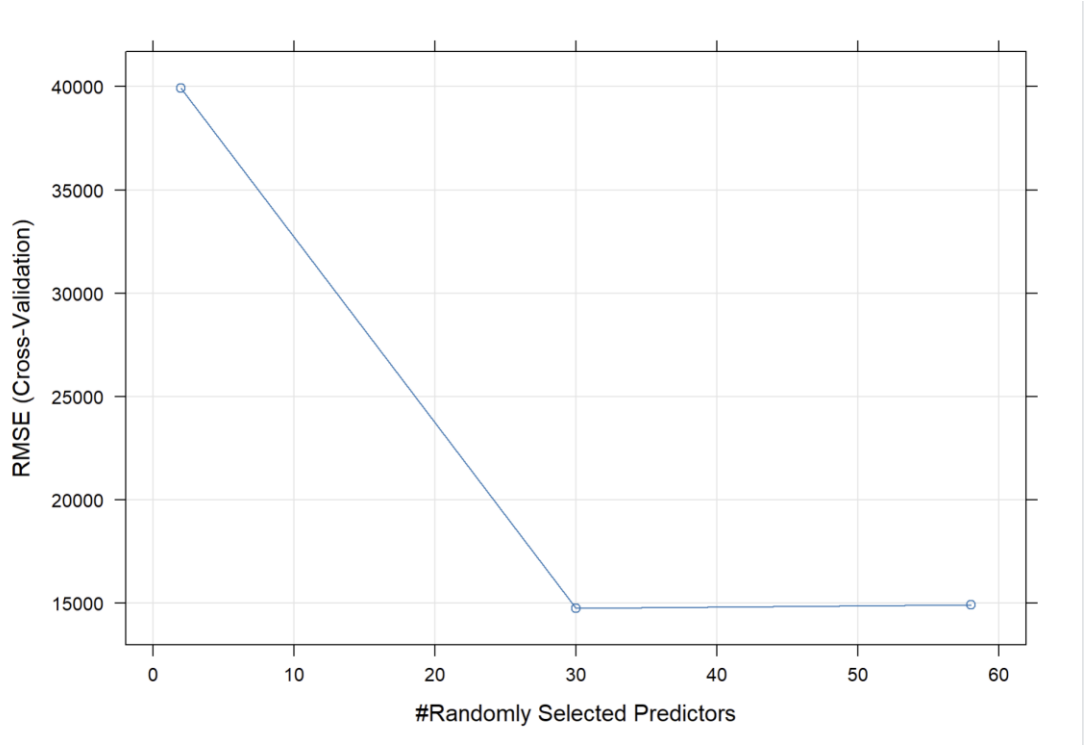


We see in this graph that Years.of.Experience is the most important attribute followed by Education_Level_0.

RMSE of minimum MSE model: 15838.5

Cross-Validation:

```
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 4120, 4118, 4118, 4118, 4118
Resampling results across tuning parameters:

  mtry  RMSE      Rsquared   MAE
   2    39926.70  0.6878333  34428.373
  30    14760.77  0.9209926   9040.680
  58    14936.03  0.9191628   8469.611

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 30.
```



For cross validation, model performance was evaluated at different mtry values and the best model was taken at mtry = 30. While R^2 for vanilla random forest was 0.9148662, the cross validated version of random forest optimal R^2 was measured as 0.9209926. Thus, although it is not a huge performance increase, we still see an increase.

Parameter Tuning:

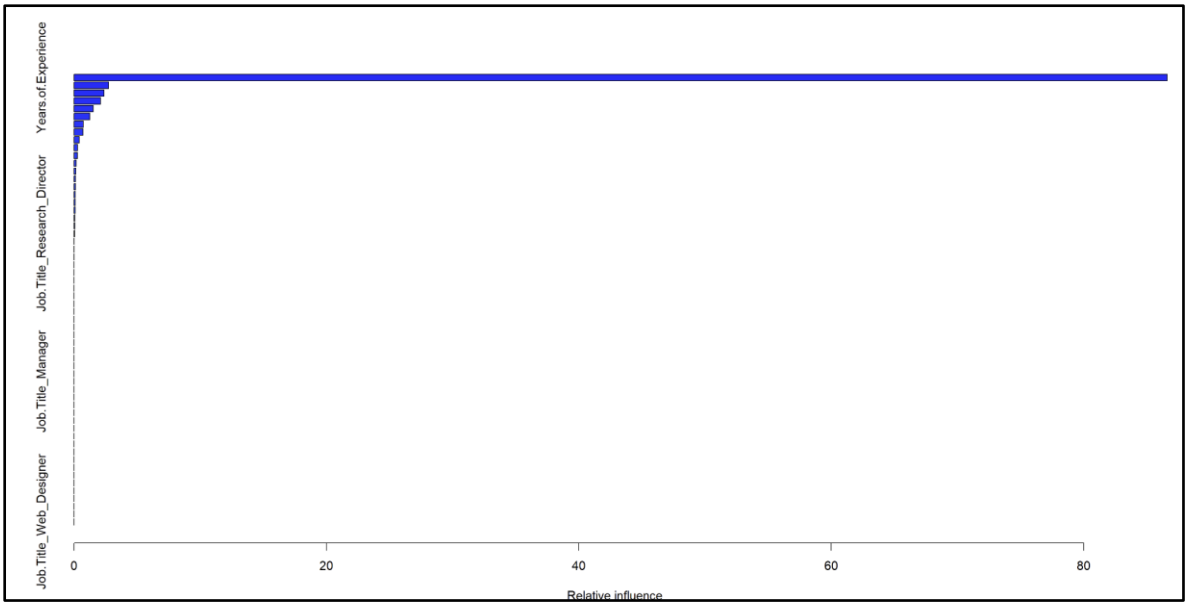
```
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 5148, 5148, 5148, 5148, 5148, 5148, ...
Resampling results across tuning parameters:

  mtry  RMSE      Rsquared   MAE
   1    48457.44  0.6303079  41973.46
   2    40507.05  0.6846522  34964.45
   3    33712.81  0.7305273  28615.93
   4    29310.18  0.7671311  24366.72
   5    26220.13  0.7968653  21367.35
   6    24026.68  0.8191260  19259.16
   7    22360.94  0.8379101  17655.81
   8    21033.58  0.8529249  16354.46
   9    20064.53  0.8636128  15375.42
  10    19209.18  0.8732960  14506.05
  11    18535.20  0.8806867  13792.89
  12    17985.27  0.8867496  13216.69
  13    17498.74  0.8919902  12679.54
  14    17121.67  0.8960023  12255.81
  15    16795.29  0.8994541  11853.75

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 15.
```

In parameter tuning, we observed our mtry value between 1 and 15. It is clearly seen that there is an increase in performance as we increase mtry. When we print out the model after applying 5-fold cross-validation, we can examine our values when mtry is 2, 30 and 58. Based on these two results, it would be appropriate to say that our R2 value increased with the increase of mtry and we obtained the most optimized version when mtry = 30, but after this peak, the model performance decreased.

Gradient Boosting:

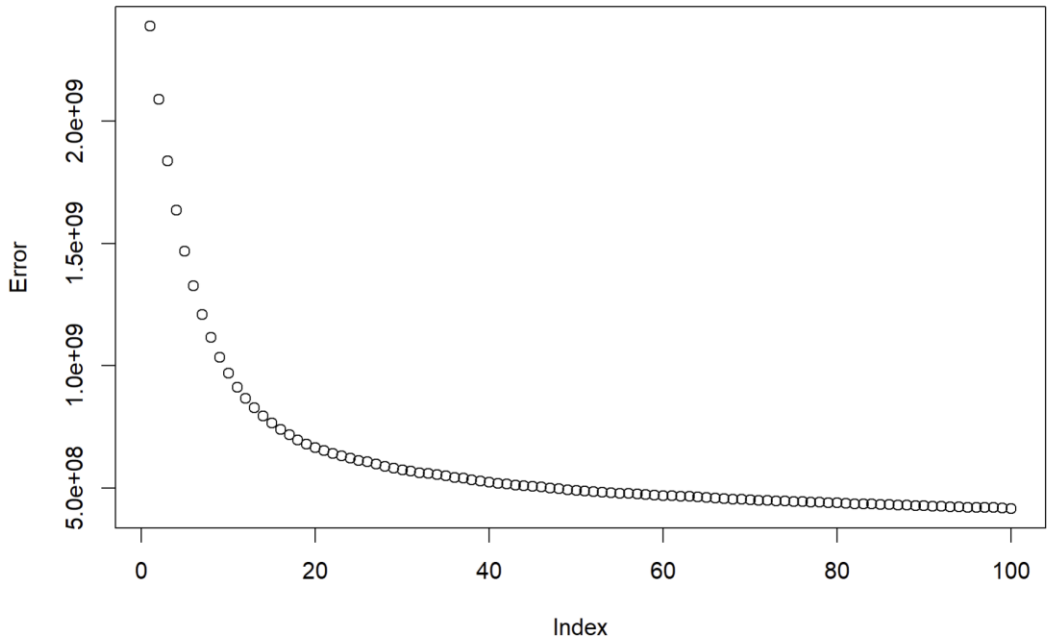
The model was trained with 100 trees on the training data. The training time of the model is quite fast, about 0.6 seconds.

"Runtime: 0.595907926559448 seconds"
Test R2: 0.8661616 Train R2: 0.8508588 Validation R2: 0.8476209

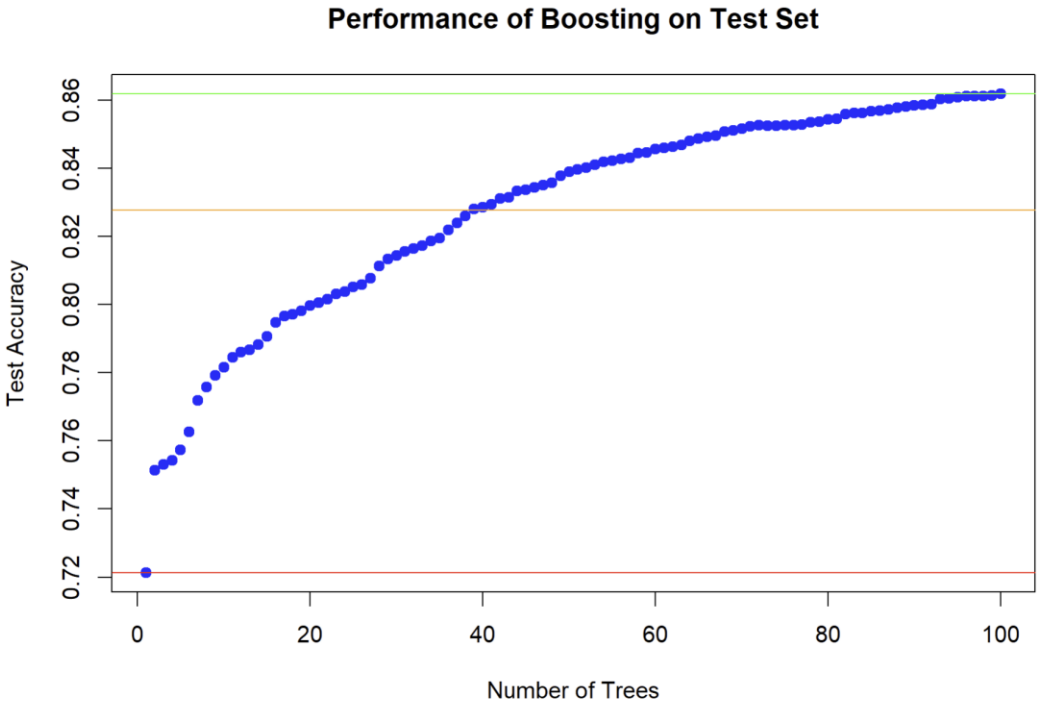


	rel.inf
Years.of.Experience	**86.61114175**
Job.Title_Data_Analyst	**2.73224471**
Job.Title_Software_Engineer	**2.35341376**
Education.Level_0	**2.09390781**
Job.Title_Product_Manager	**1.51536583**
Education.Level_1	**1.25883787**
Job.Title_Project_Engineer	**0.73213786**
Senior	**0.71825723**
Job.Title_Marketing_Manager	**0.41921838**
Job.Title_Marketing_Coordinator	**0.28215001**
…	

As we can observe above, the Years of Experience column has a high influence on the performance of the model with 86.61 relative influence. After that Job title and education level follows with much less influence of 2.73 and 2.09 relative influences.



As the tree index increases, we see an increase in the performance of our model on the Training Set.



Here, the lowest error value of our model on the test set is shown with a red line, the highest error value with a green line, and the mean value error with a yellow line.Also, as the number of trees increases, the accuracy in the test set increases.

Cross-Validation and Parameter Tuning:

We performed 5-fold cross validation repeated 3 times and Grid Search on the Gradient Boosting Model.

method:gbm

	shrinkage	interaction.depth	n.minobsinnode	n.trees	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
1	0.1	1	10	50	25618.31	0.7699290	20413.90	298.8303	0.008870206	271.3034
5	0.1	2	10	50	23585.08	0.8018308	18380.80	324.0611	0.008003499	249.9938
9	0.1	3	10	50	22357.54	0.8208725	17134.93	388.8082	0.008761548	246.1784
13	0.1	4	10	50	21575.72	0.8326400	16325.62	380.3794	0.008147095	301.7956
2	0.1	1	10	100	23855.97	0.7963693	18593.72	328.7421	0.008449972	243.9253
6	0.1	2	10	100	21736.38	0.8297480	16507.04	369.1592	0.007756784	273.7721
10	0.1	3	10	100	20617.75	0.8462423	15311.73	423.0210	0.008092362	282.2032
14	0.1	4	10	100	19854.80	0.8573546	14563.12	408.2768	0.007709284	285.1651
3	0.1	1	10	150	22979.59	0.8099564	17686.30	344.9448	0.007983056	261.8720
7	0.1	2	10	150	20816.13	0.8433217	15520.99	425.2764	0.008333582	291.6947
11	0.1	3	10	150	19730.58	0.8590463	14444.50	398.1639	0.007374528	256.8816
15	0.1	4	10	150	19046.79	0.8686129	13772.78	402.7303	0.007192838	274.9321
4	0.1	1	10	200	22447.51	0.8180469	17112.95	371.8573	0.008140531	291.7128
8	0.1	2	10	200	20220.27	0.8519939	14915.87	426.1900	0.007996840	319.4907
12	0.1	3	10	200	19233.72	0.8659492	13932.60	395.7311	0.007127680	249.4945
16	0.1	4	10	200	18551.30	0.8752625	13291.94	426.9576	0.007294314	238.9087

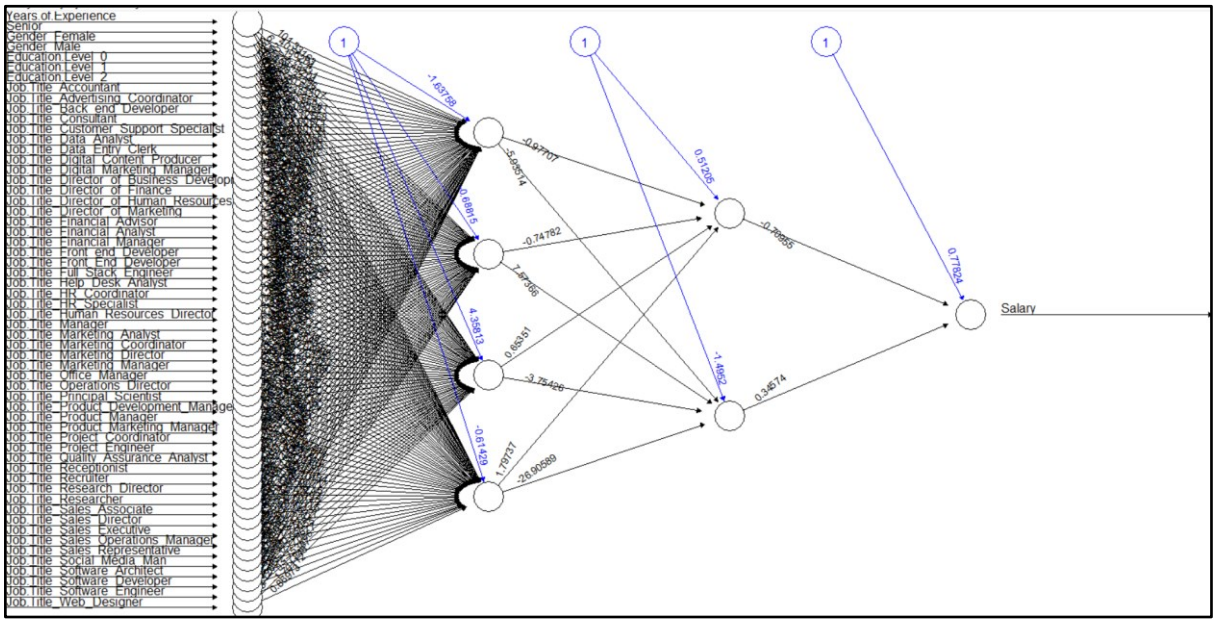
```
Resampling results across tuning parameters:

  interaction.depth  n.trees  RMSE      Rsquared   MAE
  1                   50      25618.31  0.7699290  20413.90
  1                  100      23855.97  0.7963693  18593.72
  1                  150      22979.59  0.8099564  17686.30
  1                  200      22447.51  0.8180469  17112.95
  2                   50      23585.08  0.8018308  18380.80
  2                  100      21736.38  0.8297480  16507.04
  2                  150      20816.13  0.8433217  15520.99
  2                  200      20220.27  0.8519939  14915.87
  3                   50      22357.54  0.8208725  17134.93
  3                  100      20617.75  0.8462423  15311.73
  3                  150      19730.58  0.8590463  14444.50
  3                  200      19233.72  0.8659492  13932.60
  4                   50      21575.72  0.8326400  16325.62
  4                  100      19854.80  0.8573546  14563.12
  4                  150      19046.79  0.8686129  13772.78
  4                  200      18551.30  0.8752625  13291.94

Tuning parameter 'shrinkage' was held constant at a value of 0.1
Tuning parameter 'n.minobsinnode' was held constant at a value of 10
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were n.trees = 200, interaction.depth = 4, shrinkage = 0.1 and n.minobsinnode = 10.
```

The chosen combination of hyperparameters (n.trees = 200, interaction.depth = 4, shrinkage = 0.1, n.minobsinnode = 10) is the one that minimizes the RMSE. And as we observed this parameters slightly increased R^2.

Neural Network:

Neural network is a model used in the fields of machine learning and artificial intelligence. This model has a mathematical structure that imitates the way the human brain works. The data set is divided into 70% training and 30% testing.
Neural network model using the neuralnet package in R and evaluating its performance on both the training and test datasets.

"Runtime: 51.3394901752472 seconds"



There are two hidden layers with 4 and 2 neurons for the first model.

	[,1]
error	8.358283e+00
reached.threshold	9.147647e-03
steps	9.811000e+03

SSE.train
[1] 8.461518

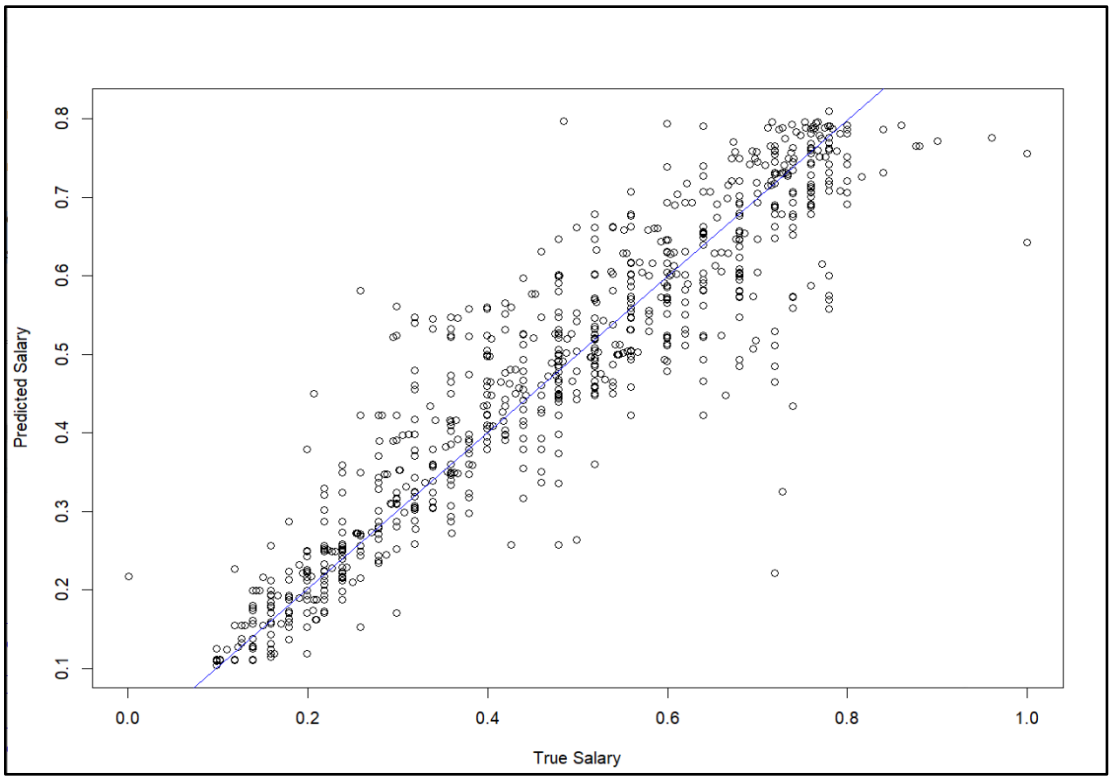
SSE.test
[1] 4.085996

head(error.df)

	df.test.Salary	pr.nn_test.net.result
1	0.3591027	0.5240073
11	0.2990186	0.1706641
12	0.5593831	0.5649762
15	0.1588224	0.1931303

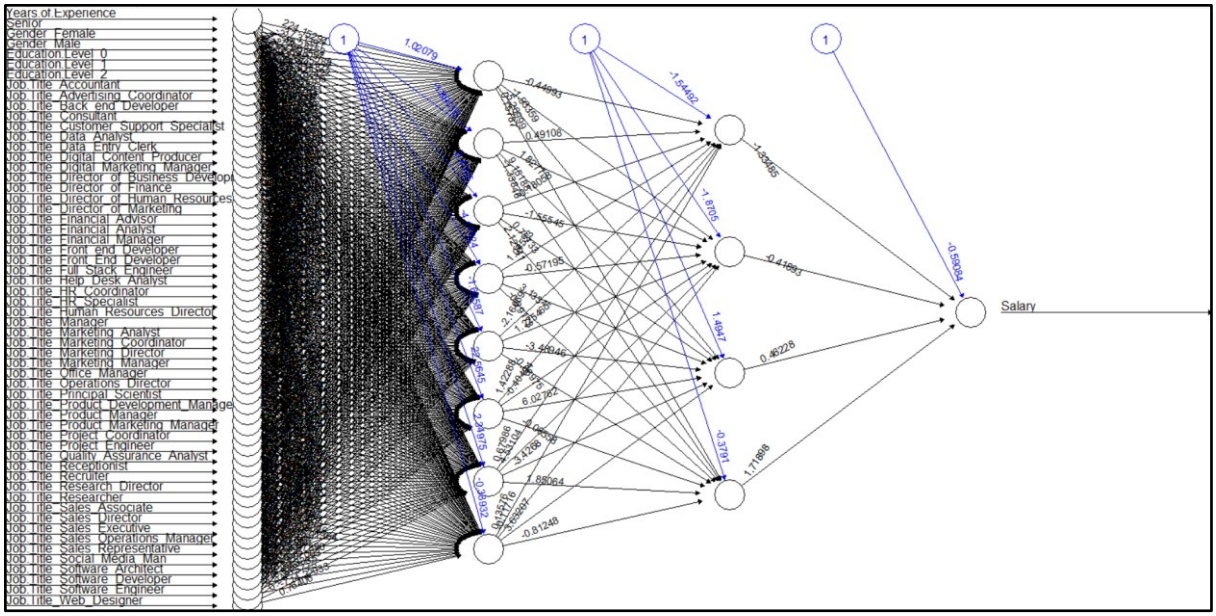
16	0.4992990	0.5523138
18	0.4592429	0.5474419

Our result obtained from rsquare account is 0.8814772 for the first NN.



Above Table shows that our neural network model prediction is quite accurate with a linear prediction line.

Second Neural Network



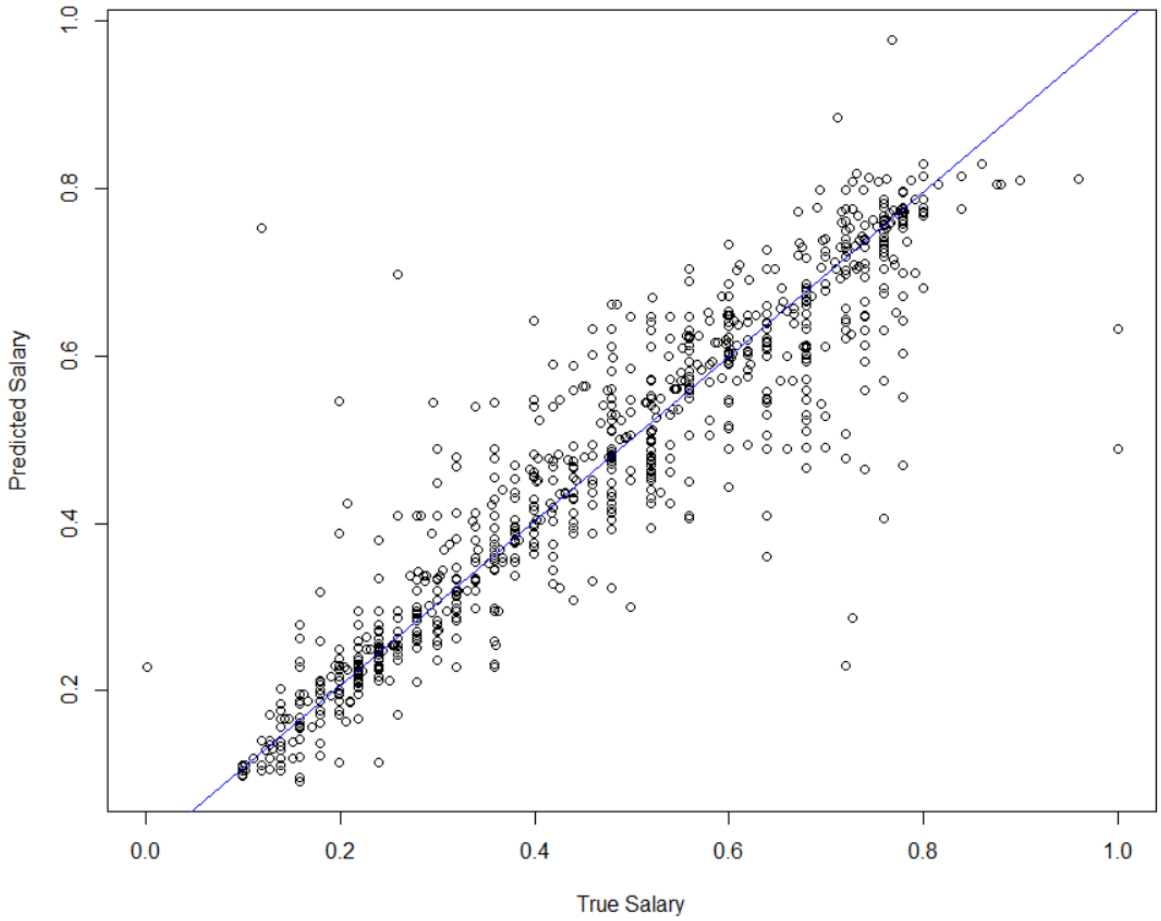
There are two hidden layers with 8 and 4 neurons for the second model.

SSE.train2
[1] 6.131883
SSE.test2
[1] 4.030059
head(error2.df)

	df.test.Salary	pr.nn2_test.net.result
1	0.3591027	0.4887014
11	0.2990186	0.2536704
12	0.5593831	0.5608635
15	0.1588224	0.1953933
16	0.4992990	0.5321240
18	0.4592429	0.6325033

SSE train error decreased from 8.461518 to 6.131883 and SSE test error decreased from 4.085996 to 4.030059 which is a better result. We stopped with these two different neuron numbers but trying this model with more neurons might have helped decreasing the SSE even more.

Our result obtained from rsquare account is 0.897618 for second NN.



We see in this graph that the points are mostly located on the line, so we can see that our model makes accurate predictions.

7. Conclusion

Model Name	SSE	MAE	R^2	RMSE	Runtime
Single Linear Regression	4.732	24314.04	0.693	30318.66	0.006
Multi Linear Regression	2.348	17806.05	0.804	23959.75	0.038
Ridge Regression	1.78811	5116.734	0.987	6670.234	2.133
Decision Tree	1.60836	14724.77	0.872	19827.94	0.186
Random Forest	5.515	9040.680	0.920	14760.77	13.19s
Gradient Boosting	1.322	13291.94	0.875	18551.30	0.5959s
Neural Network First NN	4.085	0.0495	0.881	0.0727	51.339s
Neural Network Second NN	4.030	0.0433	0.897	0.0675	8.452m

We tested 6 different models and examined the error metrics of each. We achieved more optimized results using Cross-Validation and Parameter tuning. The model that gave the most accurate results for our dataset was Ridge Regression, which is the Regularization technique for Linear Regression. Ridge Regression is the method we will prefer in our dataset because it has high performance with a test R2 of 0.987 and it is a method that allows us to avoid overfitting.The Random Forest model follows right after. However, since it is a complex model, we saw that it was the algorithm that worked the longest with 13.19 seconds, after Neural Network (59 seconds) among other models.The Single Linear Regression model we got the lowest performance result from. Even though Cross-Validation was applied, we could not increase the R2 value above 0.69. This is not a performance range desired from these models. Therefore, in our case, the last model we will choose is the Single Linear Regression model. Considering all these results, a choice can be made among the models according to the expectation from the model, time constraint and the size of the dataset.

8. References

[1] "Decision Tree in R Programming." GeeksforGeeks,Available: https://www.geeksforgeeks.org/decision-tree-in-r-programming/.
[2] "Ridge Regression in R." Statology, Available: https://www.statology.org/ridge-regression-in-r/.
[3] "Select Important Variables using Boruta Package." Analytics Vidhya, Available: https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/.
[4] "Random Forest." Wikipedia, Available: https://en.wikipedia.org/wiki/Random_forest.
[5] "Gradient Boosting in R." Data Science+, Available: https://datascienceplus.com/gradient-boosting-in-r/.
[6] "Random Forest in R." Stalogy.org, Available: https://www.statology.org/random-forest-in-r/