DAT-4253 LM 7 - Classification - Summary Project

AUTHOR
Aaron Younger

PUBLISHED
October 12, 2024

Abstract

In this lab four different classification models are explored using a problem presented by Mr. Diaz. Mr. Diaz wants to know what distinguishes and makes a top performing sales rep. The metric used to determine a top performing sales rep is net promoter score. To explore and answer this business problem a dataset was provided which included data from 21990 tech sales rep. Before Modeling Exploratory Data Analysis was used to explore categorical and numeric variables cleaning and exploring potential erros in the data before modeling. The dependent variable, net promoter score (nps), was transformed into a binary to be used in the classification models. Class 1 was a rep with an nps score of 9-10 and a rep with an nps score of 0-8 is in class 0. The first model used was a KNN model where numeric values had to be scaled, this is due to how KNN uses distance to calculate. KNN threshold tuning was found to be the best KNN model. Next Naive Bayes was used where numeric values had to be binned. Among these models Naive Bayes threshold was the best. Next Logistic Regression was explored where the log was taken of salary and years due to their skewness found in EDA. Among the Logistic Regression models, logistic regression using weighting was the best one. Finally a different types of classification trees were made. For these classification tree models, the best pruned tree weighted was the best classification tree model. Among the top models of the classification models logistic regression weighted was chosen as the best model. It had good class discrimination while having the highest balanced accuracy and F1 score. Model evaluation was used to explore this model further continuing to show why this model is fit for predicting high performing tech sales reps. Finally deployment advice was given for this model.

Data Understanding

Correct Version of R Studio

Libraries

Load the Data

```
# A tibble: 21,990 × 11
   sales_rep business   age female years college personality certficates
       <dbl> <fct>    <dbl> <fct>  <dbl> <fct>   <fct>             <dbl>
 1         1 Hardware    59 1          2 Yes     Diplomat              1
 2         2 Hardware    52 0         10 Yes     Diplomat              4
 3         3 Software    47 1          1 Yes     Explorer              1
 4         4 Hardware    61 0          2 Yes     Diplomat              3
```

```
 5         5 Software    39 0          1 No      Diplomat               5
 6         6 Hardware    28 0          6 Yes     Explorer               1
 7         7 Software    25 1          1 Yes     Explorer               5
 8         8 Hardware    51 1         10 No      Explorer               0
 9         9 Hardware    34 0          4 Yes     Diplomat               2
10        10 Hardware    38 1          1 Yes     Explorer               5
# ℹ 21,980 more rows
# ℹ 3 more variables: feedback <dbl>, salary <dbl>, nps <dbl>
```

Comments on Loading in the Data:
It is important to note that all variables that where characters in the dataset have been transformed into factors.

EDA

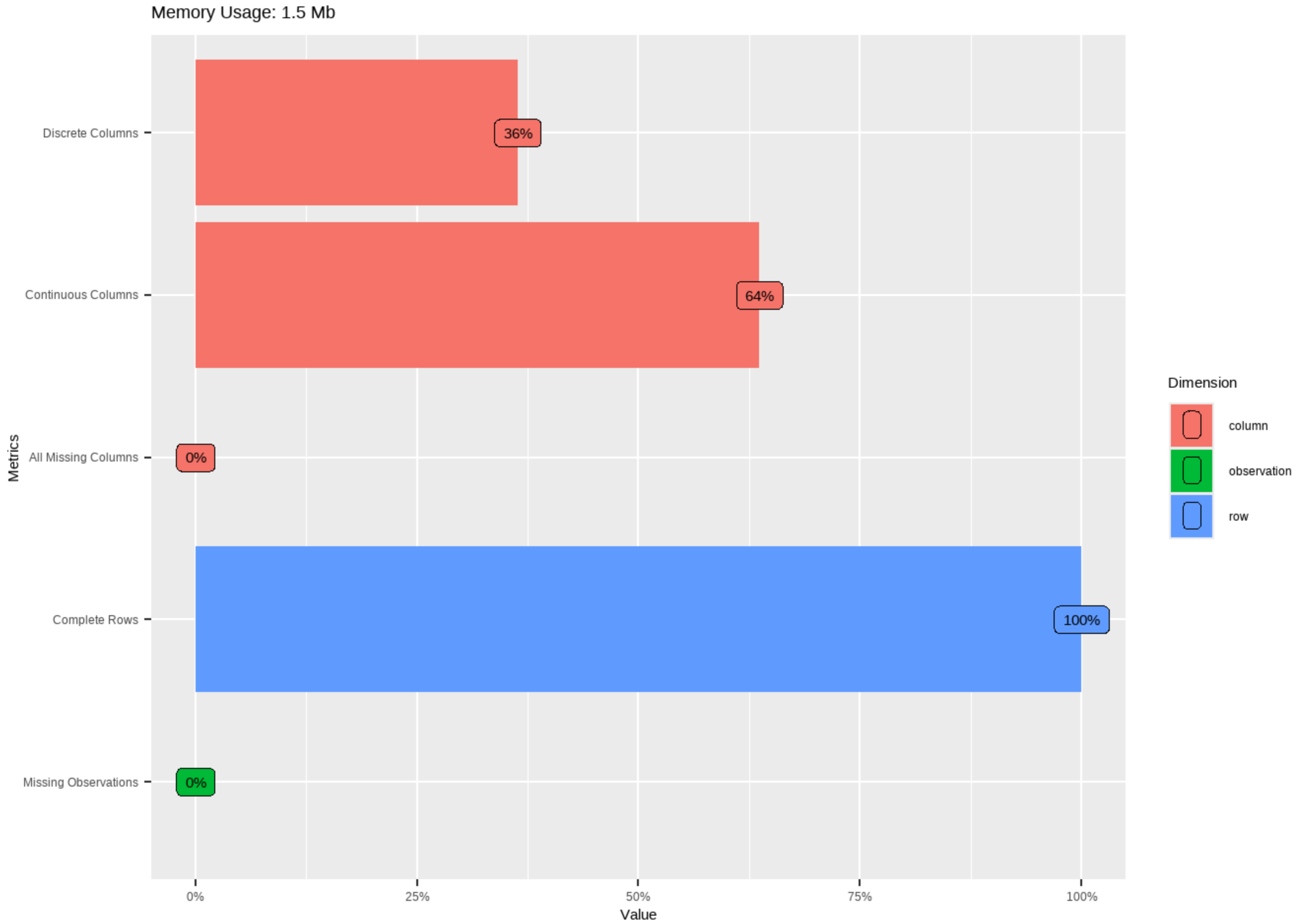
Dataset Exploration

```
# A tibble: 6 × 11
  Sales_Rep Business   Age Female Years College Personality Certficates Feedback
      <dbl> <fct>    <dbl> <fct>  <dbl> <fct>   <fct>             <dbl>    <dbl>
1         1 Hardware    59 1          2 Yes     Diplomat              1     2.01
2         2 Hardware    52 0         10 Yes     Diplomat              4     3.64
3         3 Software    47 1          1 Yes     Explorer              1     3.88
4         4 Hardware    61 0          2 Yes     Diplomat              3     2.7
5         5 Software    39 0          1 No      Diplomat              5     3.44
6         6 Hardware    28 0          6 Yes     Explorer              1     2.43
# ℹ 2 more variables: Salary <dbl>, NPS <dbl>
```

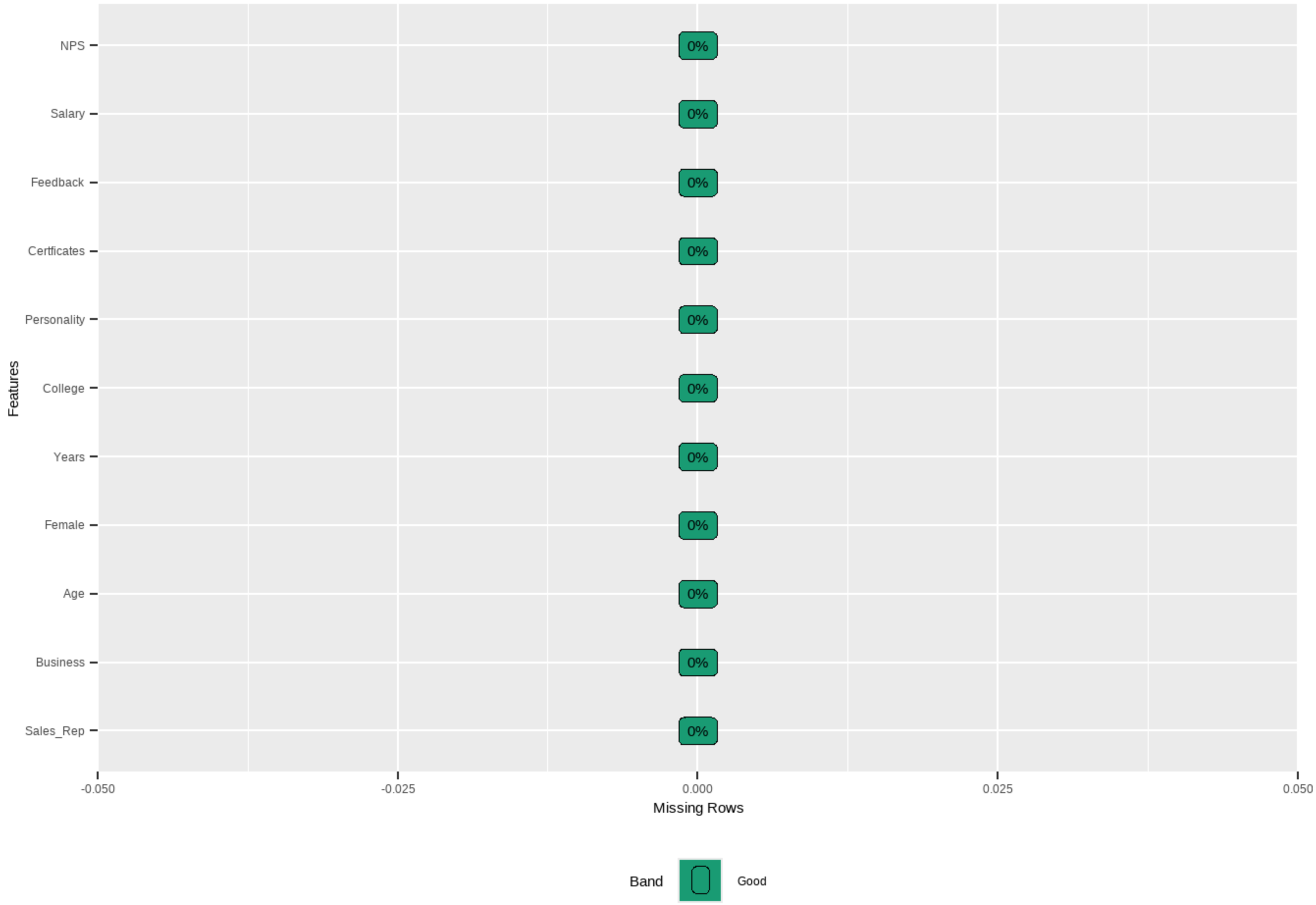
```
# A tibble: 6 × 11
  Sales_Rep Business   Age Female Years College Personality Certficates Feedback
      <dbl> <fct>    <dbl> <fct>  <dbl> <fct>   <fct>             <dbl>    <dbl>
1     21985 Hardware    35 1          8 Yes     Analyst               6     3.3
2     21986 Software    44 0          1 Yes     Diplomat              4     1.8
3     21987 Software    23 1          6 Yes     Analyst               6     1.77
4     21988 Hardware    48 1          4 Yes     Sentinel              0     2.46
5     21989 Software    29 0          4 Yes     Analyst               2     3.68
6     21990 Software    23 1          2 Yes     Explorer              1     2.13
# ℹ 2 more variables: Salary <dbl>, NPS <dbl>
```

```
tibble [21,990 × 11] (S3: tbl_df/tbl/data.frame)
 $ Sales_Rep  : num [1:21990] 1 2 3 4 5 6 7 8 9 10 ...
 $ Business   : Factor w/ 2 levels "Hardware","Software": 1 1 2 1 2 1 2 1 1 1 ...
 $ Age        : num [1:21990] 59 52 47 61 39 28 25 51 34 38 ...
 $ Female     : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 2 2 1 2 ...
 $ Years      : num [1:21990] 2 10 1 2 1 6 1 10 4 1 ...
 $ College    : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 2 1 2 2 ...
 $ Personality: Factor w/ 4 levels "Analyst","Diplomat",..: 2 2 3 2 2 3 3 3 2 3 ...
 $ Certficates: num [1:21990] 1 4 1 3 5 1 5 0 2 5 ...
 $ Feedback   : num [1:21990] 2.01 3.64 3.88 2.7 3.44 2.43 3.3 2.15 2.91 1.23 ...
```

```
 $ Salary      : num [1:21990] 70200 133000 52600 96000 122000 60000 68000 43800 92000 73400 ...
 $ NPS         : num [1:21990] 5 10 8 6 7 6 6 5 7 6 ...
```




```
Rows: 21,990
Columns: 11
$ Sales_Rep   <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…
$ Business    <fct> Hardware, Hardware, Software, Hardware, Software, Hardware…
$ Age         <dbl> 59, 52, 47, 61, 39, 28, 25, 51, 34, 38, 53, 41, 40, 41, 46…
$ Female      <fct> 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0…
$ Years       <dbl> 2, 10, 1, 2, 1, 6, 1, 10, 4, 1, 11, 1, 1, 2, 2, 4, 2, 1, 2…
$ College     <fct> Yes, Yes, Yes, Yes, No, Yes, Yes, No, Yes, Yes, Yes, Yes, …
$ Personality <fct> Diplomat, Diplomat, Explorer, Diplomat, Diplomat, Explorer…
$ Certficates <dbl> 1, 4, 1, 3, 5, 1, 5, 0, 2, 5, 2, 1, 4, 3, 1, 1, 2, 0, 5, 1…
$ Feedback    <dbl> 2.01, 3.64, 3.88, 2.70, 3.44, 2.43, 3.30, 2.15, 2.91, 1.23…
$ Salary      <dbl> 70200, 133000, 52600, 96000, 122000, 60000, 68000, 43800, …
$ NPS         <dbl> 5, 10, 8, 6, 7, 6, 6, 5, 7, 6, 8, 5, 9, 6, 5, 4, 3, 4, 9, …
```



Comments on Dataset Exploration:
This dataset has five numeric variables and five categorical variables. This datset does not contain any missing values.

Variable Exploration

Dependent Variable formatting

```
Cross-Tabulation, Row Proportions
as.factor(TechSales_Data$NPS) * depvar

------------------------------ -------- --------------- -------------- ----------------
                                 depvar               0              1            Total
  as.factor(TechSales_Data$NPS)
                              1              12 (100.0%)      0 (  0.0%)     12 (100.0%)
                              2             426 (100.0%)      0 (  0.0%)    426 (100.0%)
                              3            1817 (100.0%)      0 (  0.0%)   1817 (100.0%)
                              4            3085 (100.0%)      0 (  0.0%)   3085 (100.0%)
                              5            3593 (100.0%)      0 (  0.0%)   3593 (100.0%)
                              6            3188 (100.0%)      0 (  0.0%)   3188 (100.0%)
                              7            2765 (100.0%)      0 (  0.0%)   2765 (100.0%)
                              8            2659 (100.0%)      0 (  0.0%)   2659 (100.0%)
                              9               0 (  0.0%)   2762 (100.0%)   2762 (100.0%)
                             10               0 (  0.0%)   1683 (100.0%)   1683 (100.0%)
```

```
                               Total           17545 ( 79.8%)   4445 ( 20.2%)   21990 (100.0%)
 ------------------------------ -------- --------------- -------------- ---------------
```

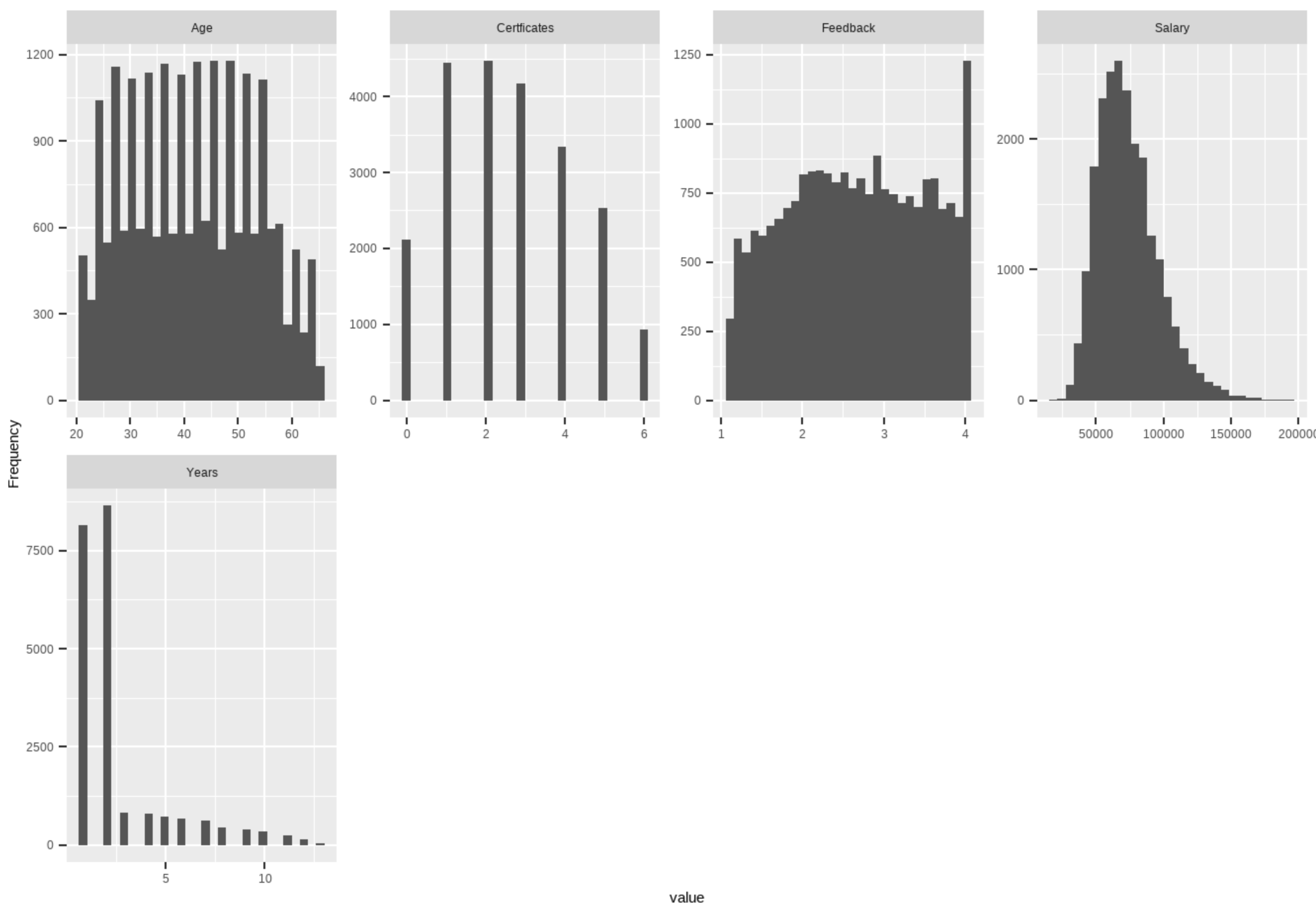
Comments on Dependent Variable Transformation:
For classification I transformed the dependent variable to a binary. The dependent variable for this dataset is NPS which is a net promoter score. This net promoter score is on a scale of 1-10. To make this variable a binary employees who earn a NPS score of 9-10 will be classified into class 1, and any employee who ears a nps score of 1-8 will be classified into class 0. The reason for transforming the NPS variable is to distinguish what makes a top performing tech sales rep (NPS=9-10) compared to other employees.

Proportion of Dependent Variables

```
 Proportion of the Dependent Variable


        0         1 
0.7978627 0.2021373 
```

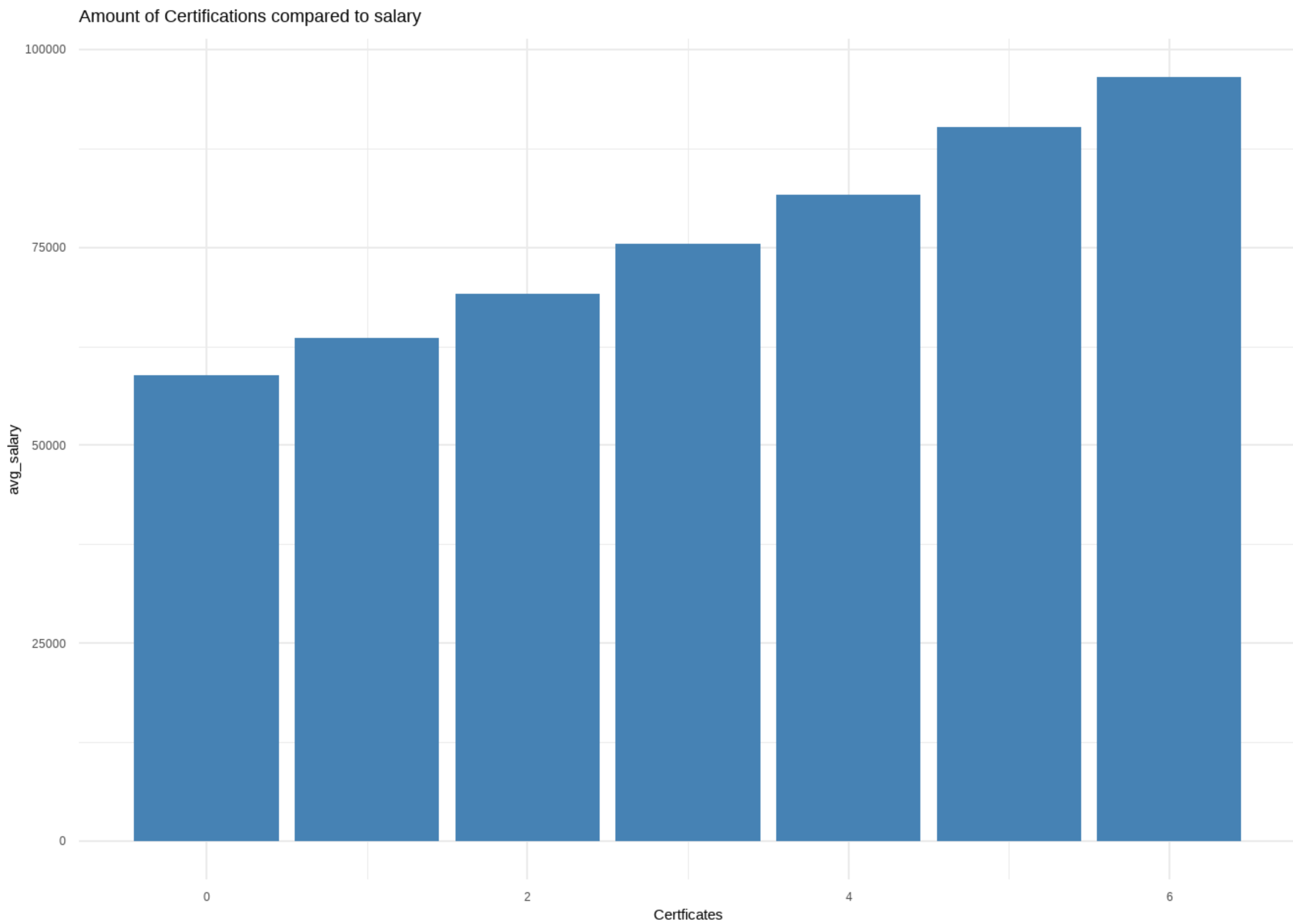
Comments on Proportion of Dependent Variable:
The Dependent Variable is imbalanced in the dataset with the majority class being 0 accounting for approximately 80% of observations with class 1 only accounting for approximately 20% of the dataset.

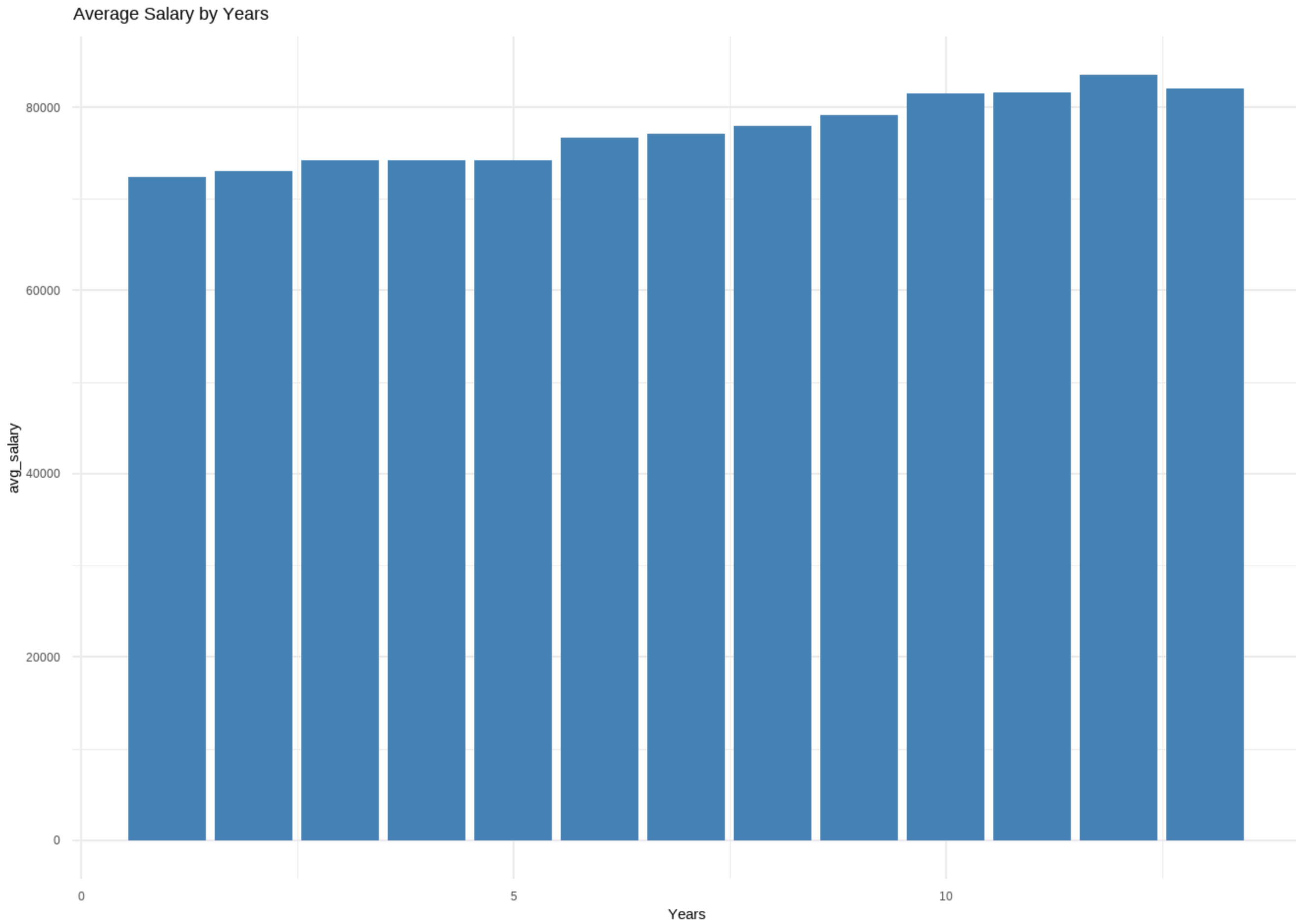
Numeric Varaible Exploration

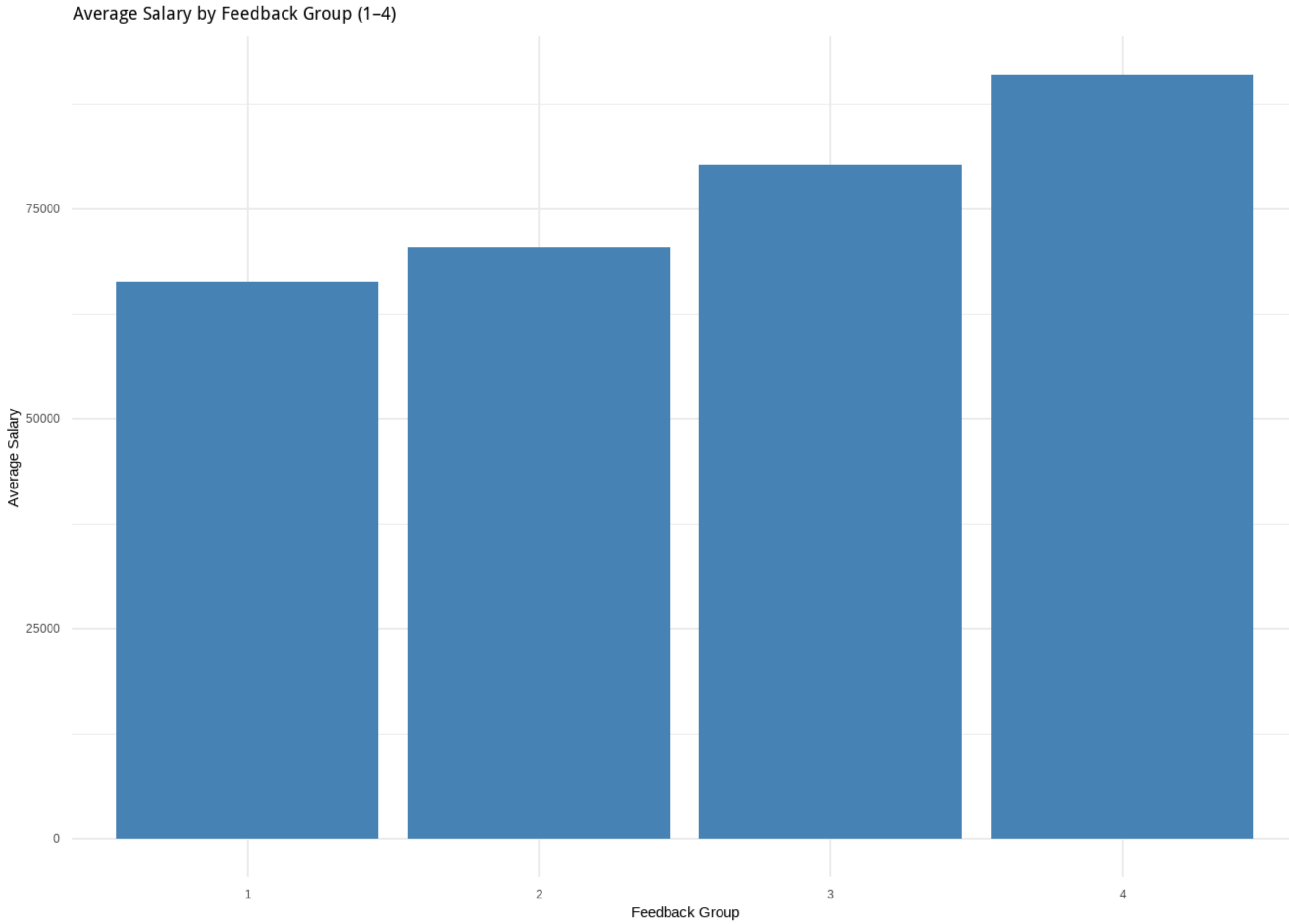


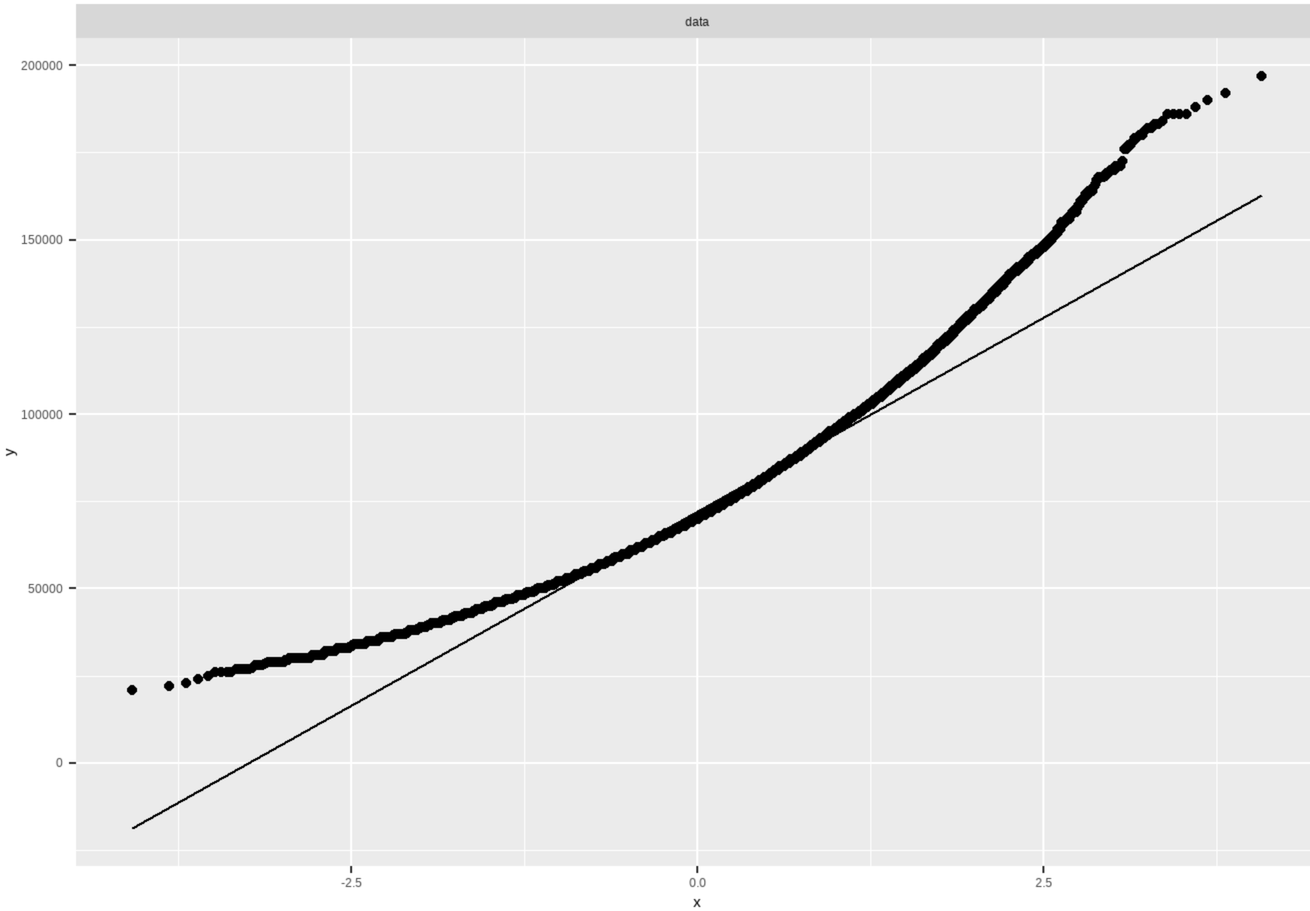

```
[1] 0.9069479
```

```
[1] 2.04887

[1] 0.09155009

[1] 0.2261304

[1] -0.05444711
```




Average Salary by Years
avg_salary
80000
60000
40000
20000
0
0
5
10
Years

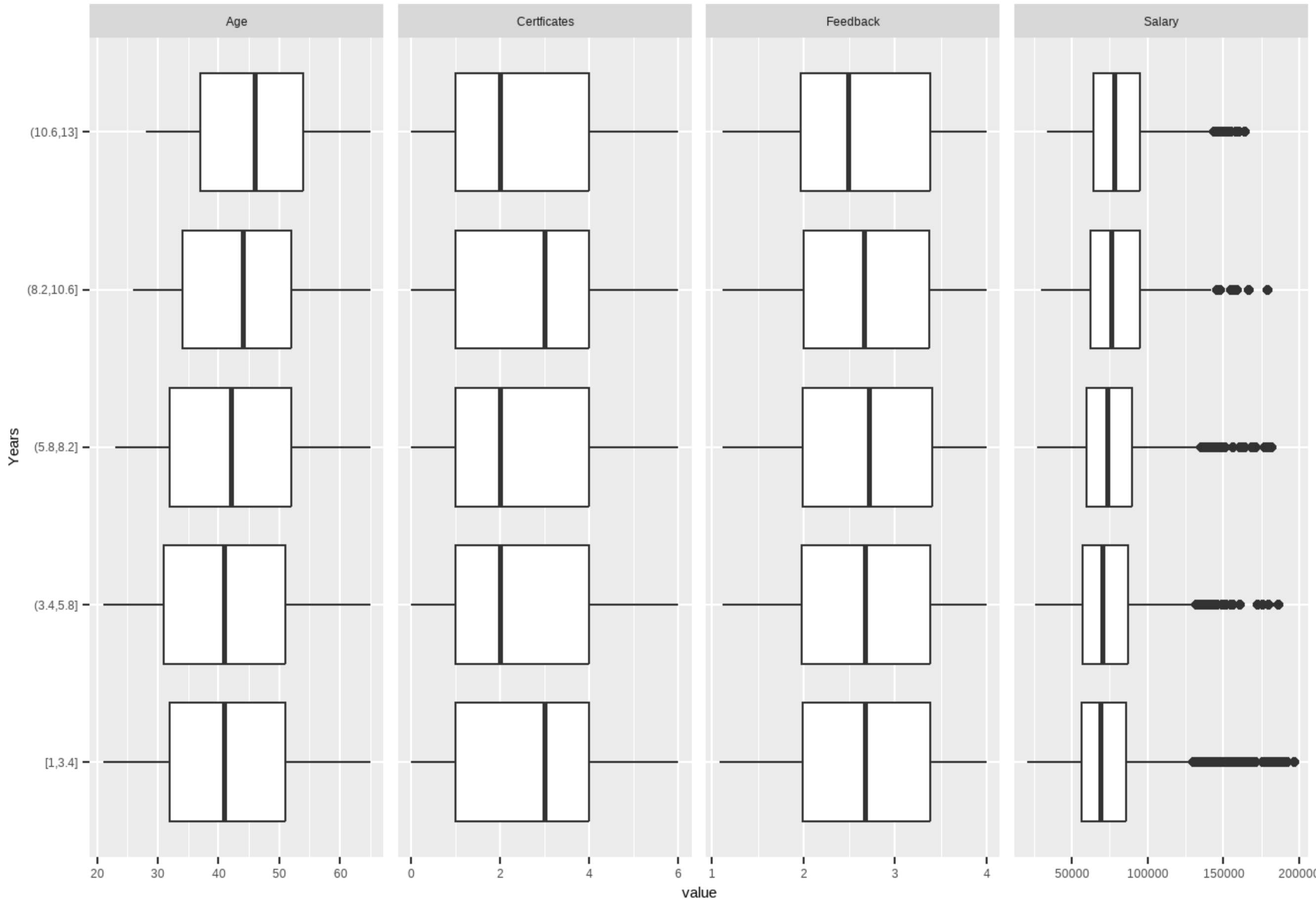

Average Salary by Feedback Group (1–4)
Average Salary
75000
50000
25000
0
1
2
3
4
Feedback Group

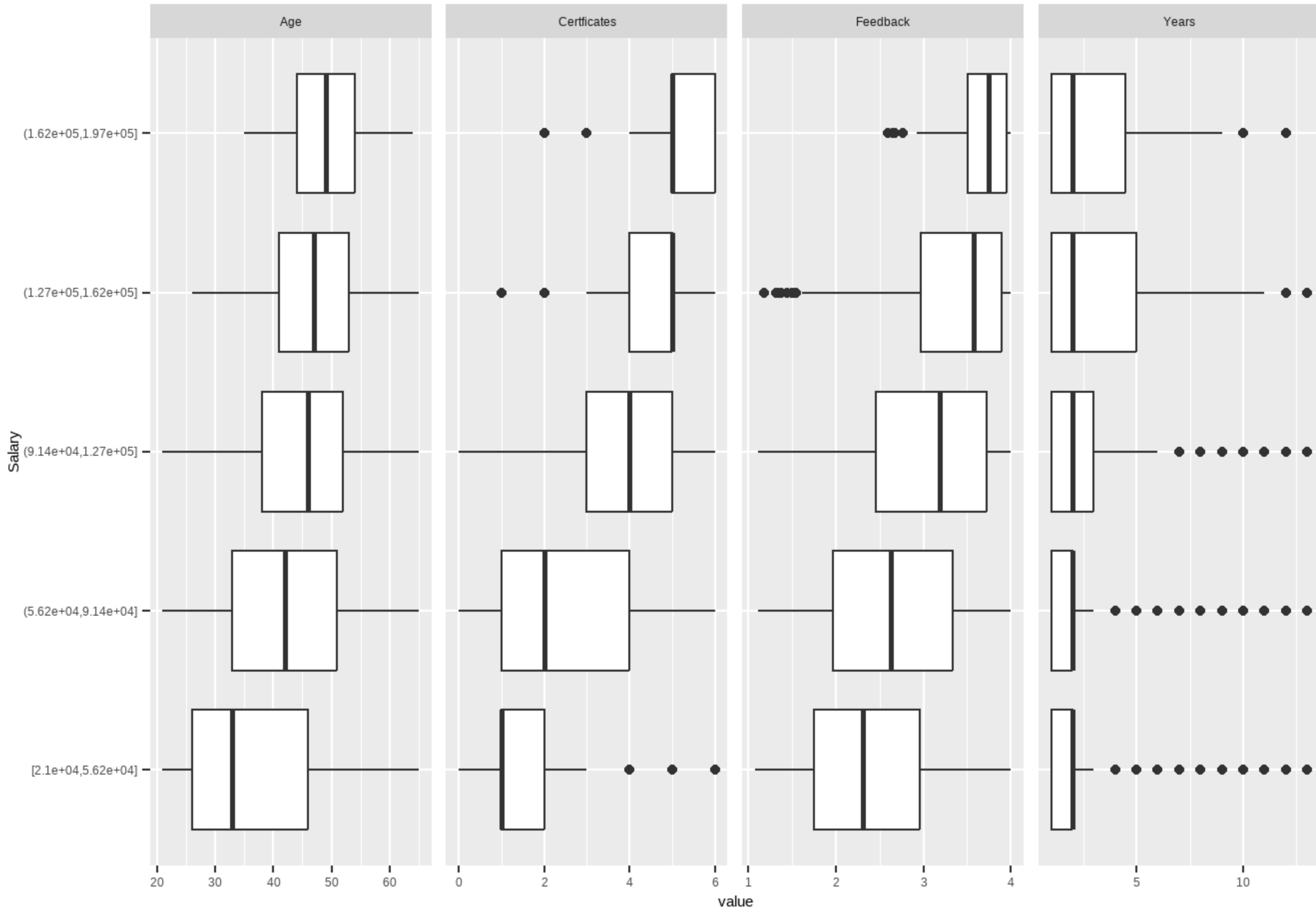


Comments on Numeric Variable Exploration:
All Numeric Variables has some level of skewness but Years and Salary had the largest values for skewness both being skewed to the right. This company has a majority of first and second year workers but significantly less 3-13 year employees. Different numeric variables were graphed with average salary to see if there were any relationship. Number of certifications, Years at company, and feedback all had a positive relationship with salary, meaning as certifications, years at company, and feedback score went up, so did Salary. This is an insight into correlation which will be plotted later. I also plotted a qq plot of salary to see if salary needs to be logged. (Stil deciding)

Check for outliers

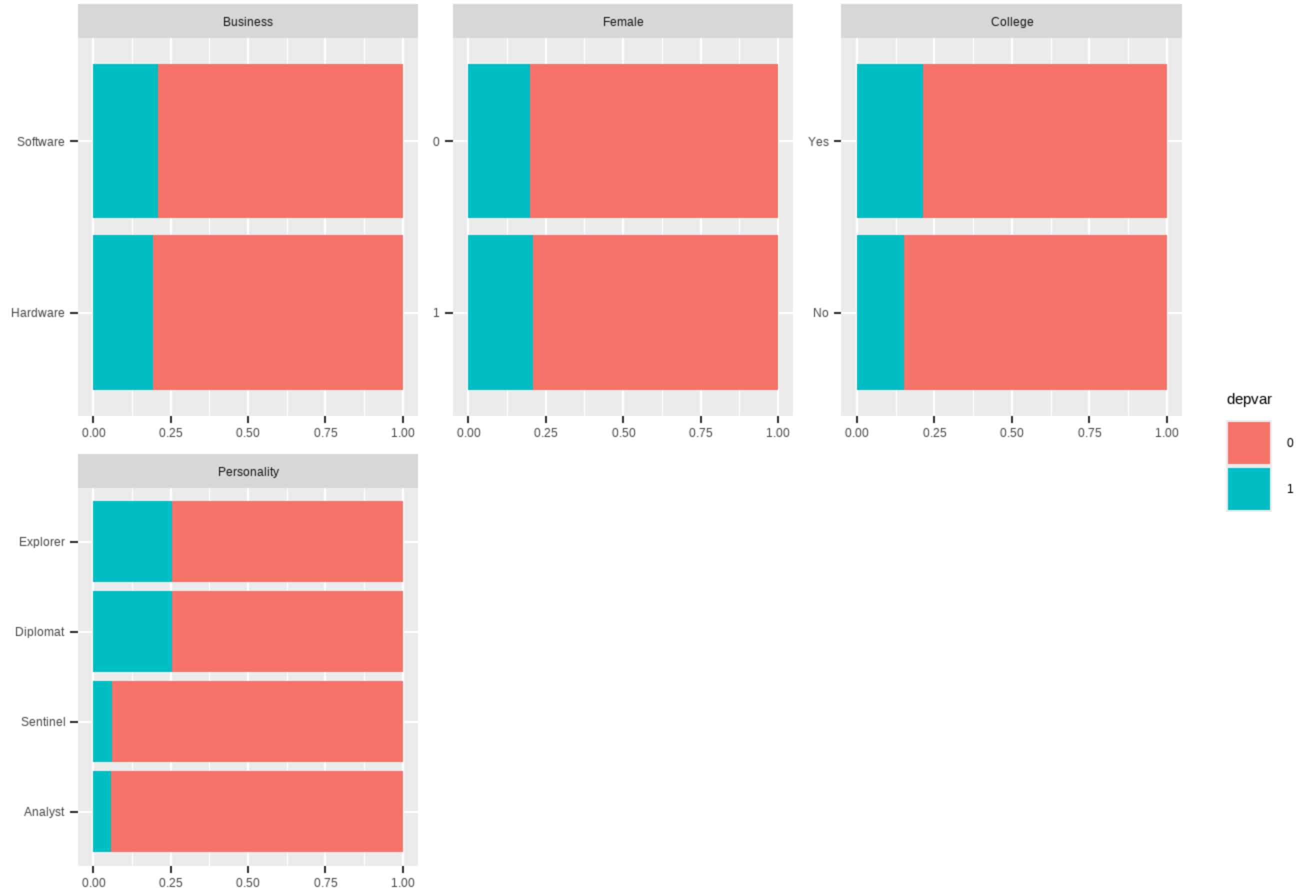
```
# A tibble: 5 × 6
  variables   outliers_cnt outliers_ratio outliers_mean with_mean without_mean
  <chr>              <int>          <dbl>         <dbl>     <dbl>        <dbl>
1 Age                    0           0            NaN        41.5         41.5
2 Years               4377          19.9            6.92      2.65         1.58
3 Certficates            0           0            NaN         2.61         2.61
4 Feedback               0           0            NaN         2.66         2.66
5 Salary               408           1.86     146969.    73674.       72288.
```


Age
Certficates
Feedback
Salary
Years
(10.6,13]
(8.2,10.6]
(5.8,8.2]
(3.4,5.8]
[1,3.4]
20
30
40
50
60
0
2
4
6
1
2
3
4
50000
100000
150000
200000
value



Comments on outlier exploration:
There were only two numeric variables with outliers those being Years and salary. These variables also had the highest skew numbers among other numeric variables. I decided to not remove any outliers as there were no observations that did not make sense. Outliers are present in Years at company due to the massive amount of first and second year employees. I do not thing removing an observation based on the amount of years worked at company is a good reason as I want to see any relationships in the data based on years. Salary also has outliers but this is common in dollar variables due to its nature to be right skewed. The mean with and without outliers is not drastic and therefore no observations will be removed based on salary.

Categorical Variable Exploration



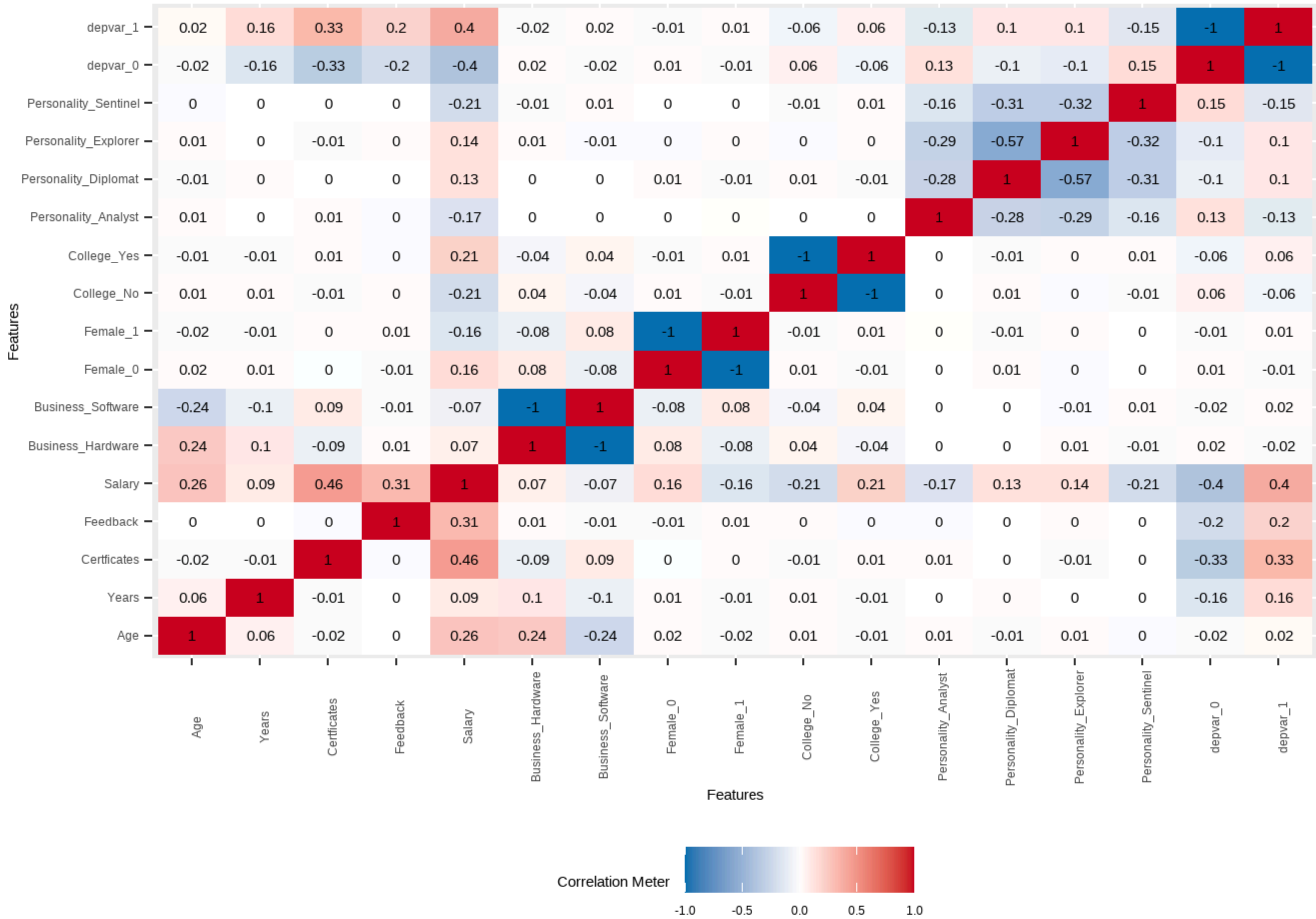

```
# A tibble: 4 × 4
# Groups:   College [2]
  College count prop_0 prop_1
  <fct>   <int>  <dbl>  <dbl>
1 No       3787  0.847 NA
2 No        683 NA      0.153
3 Yes     13758  0.785 NA
4 Yes      3762 NA      0.215
```

```
# A tibble: 8 × 4
# Groups:   Personality [4]
  Personality count prop_0  prop_1
  <fct>       <int>  <dbl>   <dbl>
1 Analyst      2508  0.943 NA
2 Analyst       151 NA      0.0568
3 Diplomat     5851  0.745 NA
4 Diplomat     1998 NA      0.255
5 Explorer     6105  0.745 NA
6 Explorer     2095 NA      0.255
7 Sentinel     3081  0.939 NA
8 Sentinel      201 NA      0.0612
```

Comments on Categorical Variable Exploration:

The Dependent Variable is present in all categorical variables. For both the Business and Female variables, there is minimal variation in the proportion of 1's and 0's across their respective categories. This cannot be said for College and Personality Variables. If an employee went to college they are more likely to receive a 1 than employees that did not go to college. If an employee is either a sentinel or analyst they are significantly more unlikely to recieve 1's compared to if an employee is an explorer or diplomat.

Plot Correlation



Comments on Correlation Plot:
Certificates, Feedback, and Salary have the strongest positive correlation to the dependent variable. This indicates that as the number of certifications, feedback scores, and salary increase, the likelihood of an employee having a Net Promoter Score of 1 also increases.

Modeling

KNN Unweighted

Prepare Data for Unweighted Modeling

Comments on why scaling numeric values is necessary:
Since the KNN model is a distance based model larger numeric scales will skew the distance measurements.

Scaling the numeric variables makes sure each value contributes prportionally to distance and therefore makes the model more accurate.

Partition Datset

```
Proportion of DepVar in Trainset


        0         1
0.7978627 0.2021373

Proportion of DepVar in testset


        0         1
0.7978627 0.2021373
```

Train Unweighted Model

```
k-Nearest Neighbors

13194 samples
    9 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 11874, 11875, 11875, 11874, 11874, 11876, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   1  0.7755798  0.3022994
   2  0.7735357  0.2986292
   3  0.8043813  0.3472163
   4  0.8032448  0.3469803
   5  0.8143085  0.3586508
   6  0.8146894  0.3618825
   7  0.8181754  0.3628415
   8  0.8176441  0.3583381
   9  0.8215847  0.3624889
  10  0.8199181  0.3572135

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 9.
```

Predict Unweighted Model

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
```

```
         0 6523 1134
         1  495  644

               Accuracy : 0.8148
                 95% CI : (0.8065, 0.8229)
    No Information Rate : 0.7979
    P-Value [Acc > NIR] : 0.00003406

                  Kappa : 0.3369

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.36220
            Specificity : 0.92947
         Pos Pred Value : 0.56541
         Neg Pred Value : 0.85190
             Prevalence : 0.20214
         Detection Rate : 0.07322
   Detection Prevalence : 0.12949
      Balanced Accuracy : 0.64584

       'Positive' Class : 1

F1 Score: 0.44155
```

KNN Weighted Oversampling

Prepare Data for oversampling

Train Weighted Model

```
k-Nearest Neighbors

13194 samples
    9 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 11874, 11875, 11875, 11874, 11874, 11876, ...
Addtional sampling using up-sampling

Resampling results across tuning parameters:

  k   Accuracy   Kappa
   1  0.7756557  0.3026426
   2  0.7332892  0.3158765
   3  0.7038817  0.3122761
   4  0.6853128  0.2984047
```

```
   5  0.6813748  0.3023370
   6  0.6784929  0.2940408
   7  0.6853900  0.3069886
   8  0.6894852  0.3121770
   9  0.6979708  0.3234616
  10  0.7035040  0.3313727

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 1.
```

Predict Weighted Model

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 6004 1054
         1 1014  724

               Accuracy : 0.7649
                 95% CI : (0.7559, 0.7737)
    No Information Rate : 0.7979
    P-Value [Acc > NIR] : 1.0000

                  Kappa : 0.2649

 Mcnemar's Test P-Value : 0.3911

            Sensitivity : 0.40720
            Specificity : 0.85551
         Pos Pred Value : 0.41657
         Neg Pred Value : 0.85067
             Prevalence : 0.20214
         Detection Rate : 0.08231
   Detection Prevalence : 0.19759
      Balanced Accuracy : 0.63136

       'Positive' Class : 1


F1 Score: 0.41183
```

KNN Weighted Threshold Tuning

KNN with Treshold Tuning

```
OPTIMAL CUTOFF VALUE OF: 0.1555556
```

Predict KNN with Threshold Tuning

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4789  384
         1 2229 1394

               Accuracy : 0.7029
                 95% CI : (0.6933, 0.7125)
    No Information Rate : 0.7979
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3362

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.7840
            Specificity : 0.6824
         Pos Pred Value : 0.3848
         Neg Pred Value : 0.9258
             Prevalence : 0.2021
         Detection Rate : 0.1585
   Detection Prevalence : 0.4119
      Balanced Accuracy : 0.7332

       'Positive' Class : 1


F1 Score:  0.5162007
```

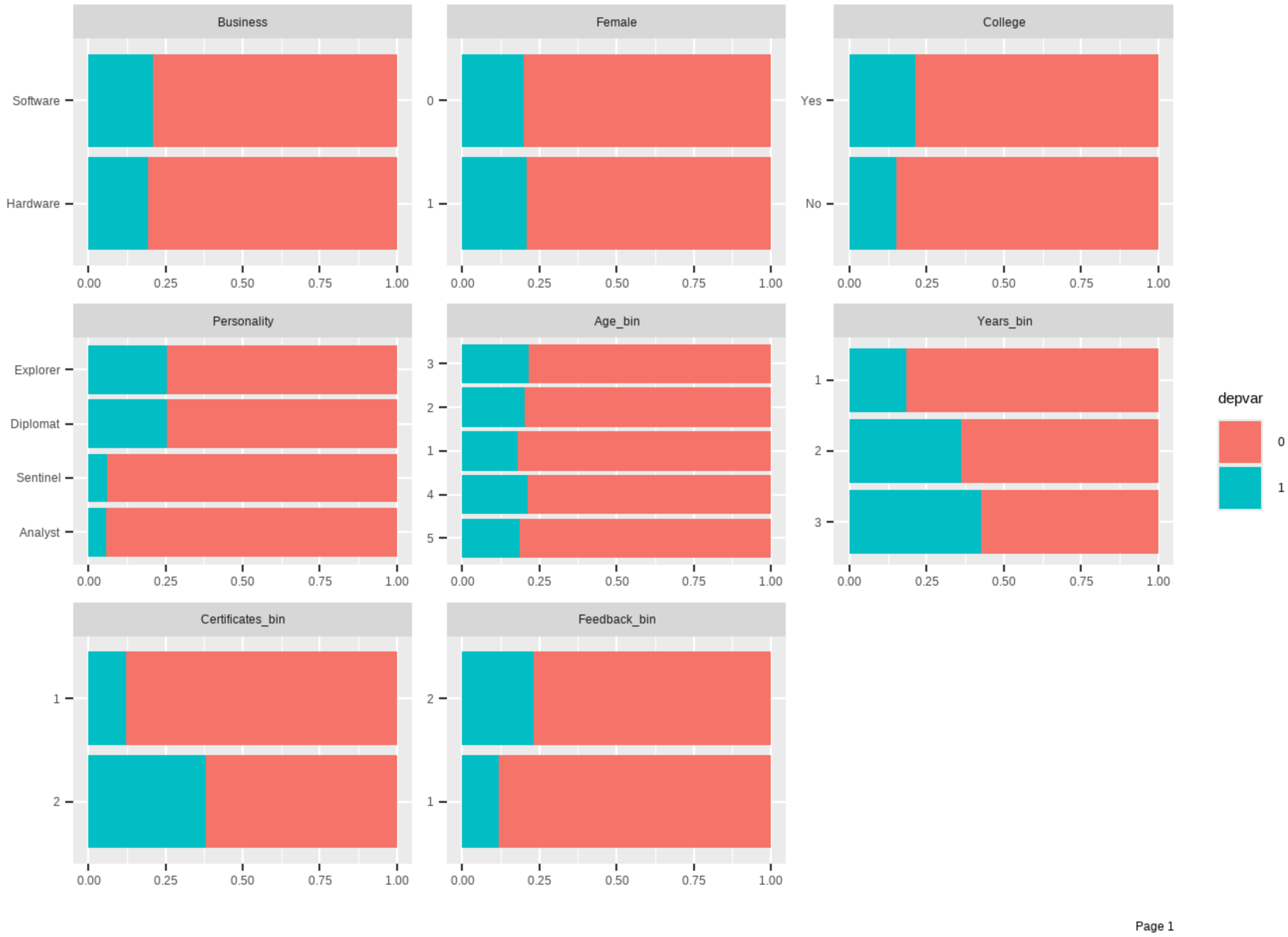
Comments on best model for KNN:
The best model out of the KNN models is the KNN model adjusted for threshold tuning.

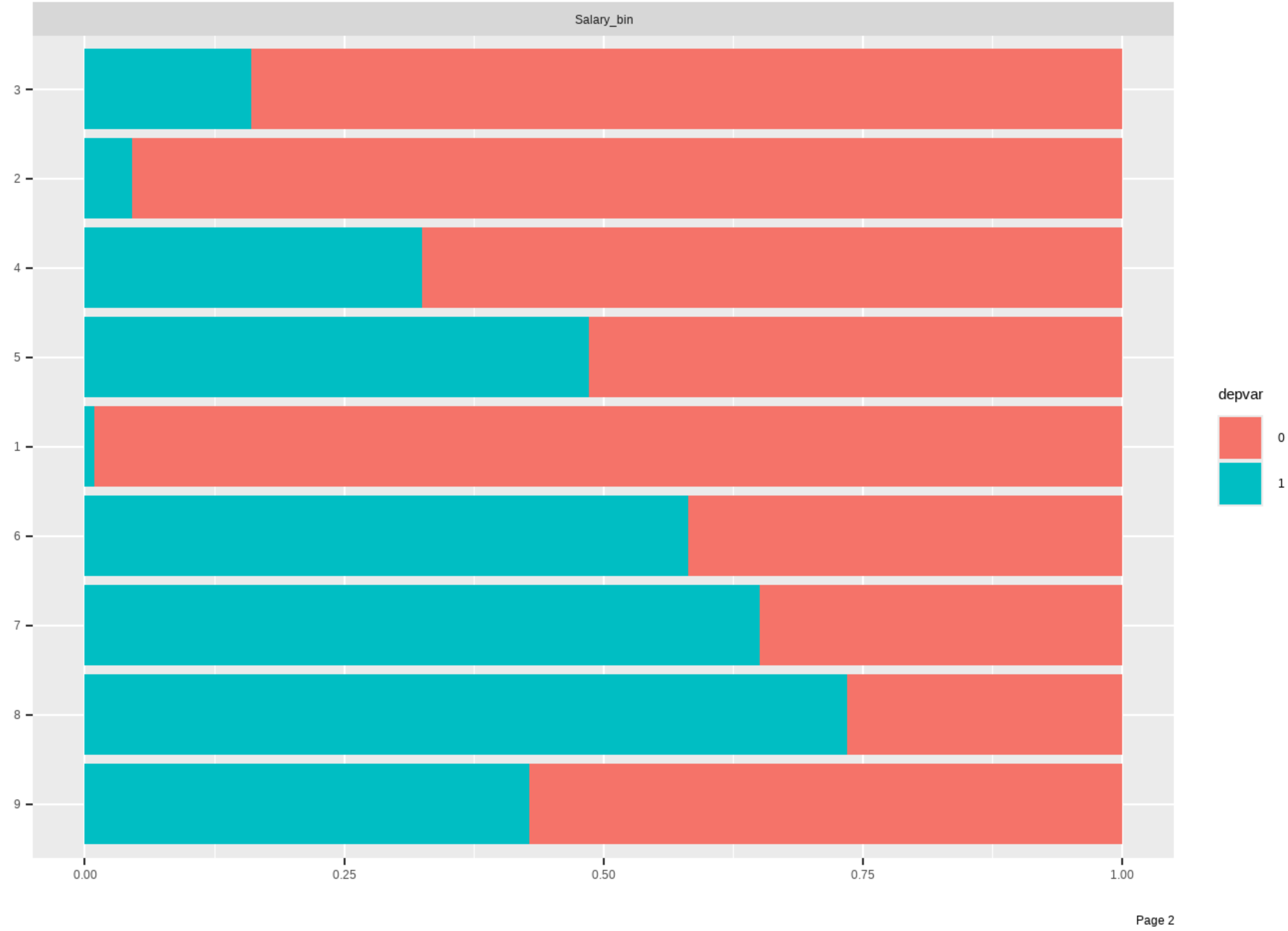
Naive Bayes

Prepare Data for Naive Bayes

```
 [1] "Business"         "Age"              "Female"           "Years"
 [5] "College"          "Personality"      "Certficates"      "Feedback"
 [9] "Salary"           "depvar"           "Age_bin"          "Years_bin"
[13] "Certificates_bin" "Feedback_bin"     "Salary_bin"

tibble [21,990 × 10] (S3: tbl_df/tbl/data.frame)
 $ Business        : Factor w/ 2 levels "Hardware","Software": 1 1 2 1 2 1 2 1 1 1 ...
 $ Female          : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 2 2 1 2 ...
 $ College         : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 2 1 2 2 ...
 $ Personality     : Factor w/ 4 levels "Analyst","Diplomat",..: 2 2 3 2 2 3 3 3 2 3 ...
 $ depvar          : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
 $ Age_bin         : Factor w/ 5 levels "1","2","3","4",..: 4 4 3 5 2 1 1 4 2 2 ...
```

```
 $ Years_bin       : Factor w/ 3 levels "1","2","3": 1 2 1 1 1 1 1 2 1 1 ...
 $ Certificates_bin: Factor w/ 2 levels "1","2": 1 2 1 1 2 1 2 1 1 2 ...
 $ Feedback_bin    : Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 1 ...
 $ Salary_bin      : Factor w/ 9 levels "1","2","3","4",..: 3 6 2 4 6 2 3 2 4 3 ...
```





Comments on data preparation for Naive Bayes:
I manually put in the bins, below is a key to what each number represents for the bins.

Key for Binned Variables:

- Age_bin: 1 = 20–29, 2 = 30–39, 3 = 40–49, 4 = 50–59, 5 = 60–69.
- Years_bin: 1 = 1–5 years, 2 = 6–10 years, 3 = 11–15 years.
- Certificates_bin: 1 = 0–3 certificates, 2 = 4–6 certificates.
- Feedback_bin: 1 = Feedback score 1–2, 2 = Feedback score 3–4.
- Salary_bin: 1 = 20,000–39,999, 2 = 40,000–59,999, 3 = 60,000–79,999, 4 = 80,000–99,999, 5 = 100,000–119,999, 6 = 120,000–139,999, 7 = 140,000–159,999, 8 = 160,000–179,999, 9 = 180,000–199,999.

Partition Dataset

```
Proportion of Depvar for Trainset


        0         1 
0.7978627 0.2021373 

Proportion of Depvar for Testset


        0         1 
```

```
0.7978627 0.2021373
```

Train Unweighted NB Model

```
Naive Bayes

13194 samples
    9 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 11874, 11875, 11875, 11874, 11874, 11876, ...
Resampling results:

  Accuracy   Kappa
  0.7978627  0

Tuning parameter 'fL' was held constant at a value of 1
Tuning
 parameter 'usekernel' was held constant at a value of TRUE
Tuning
 parameter 'adjust' was held constant at a value of 1
```

Comments on Unweighted NB model:
The Accuracy is around 80% but the kappa = 0, which means the models predictions are no better than always guessing the majority class.

Predict Unweighted NB Model

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7018 1778
         1    0    0

               Accuracy : 0.7979
                 95% CI : (0.7893, 0.8062)
    No Information Rate : 0.7979
    P-Value [Acc > NIR] : 0.5063

                  Kappa : 0

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.0000
            Specificity : 1.0000
         Pos Pred Value :    NaN
         Neg Pred Value : 0.7979
             Prevalence : 0.2021
```

```
         Detection Rate : 0.0000
   Detection Prevalence : 0.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : 1
```

Comments on unweighted Naive Bayes Model:
This model has no minority class predictive power making this model not usable.

Weighted Naive Bayes

Threshold Tuning Naive Bayes

```
OPTIMAL CUTOFF VALUE OF: 0.00001470381
```

The cutoff value is unusually low, this supports the fact that there is class imbalance in the dataset however the classes are not so severely imbalanced to produce a cutoff value that low.

Predict Threshold Tuning Naive Bayes

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4994  399
         1 2024 1379

               Accuracy : 0.7245
                 95% CI : (0.7151, 0.7339)
    No Information Rate : 0.7979
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3632

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.7756
            Specificity : 0.7116
         Pos Pred Value : 0.4052
         Neg Pred Value : 0.9260
             Prevalence : 0.2021
         Detection Rate : 0.1568
   Detection Prevalence : 0.3869
      Balanced Accuracy : 0.7436

       'Positive' Class : 1


F1 Score:  0.5323297
```

Comments on threshold Tuning NB model:
This model is a big improvement from the unweighted Naive Bayes model. It has good class discrimination, higher F1 score and moderately high Balanced Accuracy. This is the best model from Naive Bayes.

Logistic Regression Unweighted

Prepare Data For Logistic Regression

```
[1] 0.9069479

[1] 2.04887
```

Partition Data For Logistic Regression

```
        0         1 
0.7978627 0.2021373 


        0         1 
0.7978627 0.2021373 
```

Train Unweighted Model

```
Call:
NULL

Coefficients:
                      Estimate Std. Error z value             Pr(>|z|)    
(Intercept)         -26.427080   1.380489 -19.143 < 0.0000000000000002 ***
BusinessSoftware      0.117289   0.054811   2.140             0.032364 *  
Age                  -0.004490   0.002595  -1.730             0.083549 .  
Female1               0.232930   0.055372   4.207            0.0000259 ***
CollegeYes            0.249086   0.072458   3.438             0.000587 ***
PersonalityDiplomat   1.992017   0.130227  15.296 < 0.0000000000000002 ***
PersonalityExplorer   1.983991   0.130009  15.260 < 0.0000000000000002 ***
PersonalitySentinel   0.184237   0.157930   1.167             0.243382    
Certficates           0.554867   0.020494  27.075 < 0.0000000000000002 ***
Feedback              0.685823   0.036201  18.945 < 0.0000000000000002 ***
log_Salary            1.635568   0.133321  12.268 < 0.0000000000000002 ***
log_Years             0.995327   0.049789  19.991 < 0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 13282.4  on 13193  degrees of freedom
Residual deviance:  9452.4  on 13182  degrees of freedom
```

```
AIC: 9476.4

Number of Fisher Scoring iterations: 6
   BusinessSoftware                  Age              Female1           CollegeYes
           1.099062             1.254189             1.092274             1.108266
PersonalityDiplomat PersonalityExplorer PersonalitySentinel          Certficates
           6.211930             6.242053             2.213472             1.477071
           Feedback           log_Salary            log_Years
           1.262741             1.746404             1.079247

                      Overall
BusinessSoftware     2.139890
Age                  1.730456
Female1              4.206677
CollegeYes           3.437690
PersonalityDiplomat 15.296466
PersonalityExplorer 15.260432
PersonalitySentinel  1.166574
Certficates         27.074505
Feedback            18.944750
log_Salary          12.267917
log_Years           19.990836
```

Potential multicollinearity in Diplomat and Explorer. Certificates and Feedback have most variable importance when it comes to the dependent variable.

Predict Unweighted Model

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 6625 1104
         1  393  674

               Accuracy : 0.8298
                 95% CI : (0.8218, 0.8376)
    No Information Rate : 0.7979
    P-Value [Acc > NIR] : 0.00000000000001447

                  Kappa : 0.3798

 Mcnemar's Test P-Value : < 0.0000000000000022

            Sensitivity : 0.37908
            Specificity : 0.94400
         Pos Pred Value : 0.63168
         Neg Pred Value : 0.85716
             Prevalence : 0.20214
         Detection Rate : 0.07663
   Detection Prevalence : 0.12131
```

```
       Balanced Accuracy : 0.66154

        'Positive' Class : 1

F1 Score:  0.4738137
```

This model overclassifies the 0 class and underclassifies the 1 class as seen in the difference between sensitivity and specificity. The balanced accuracy is also moderately low.

Logistic Regression Weighted

Train Using Weights

```
Call:
NULL

Coefficients:
                      Estimate Std. Error z value             Pr(>|z|)    
(Intercept)         -25.894698   1.212293 -21.360 < 0.0000000000000002 ***
BusinessSoftware      0.183805   0.046975   3.913           0.00009123 ***
Age                  -0.003857   0.002217  -1.740               0.0819 .  
Female1               0.217119   0.047476   4.573           0.00000480 ***
CollegeYes            0.275291   0.061399   4.484           0.00000734 ***
PersonalityDiplomat   2.082243   0.101179  20.580 < 0.0000000000000002 ***
PersonalityExplorer   2.077988   0.101118  20.550 < 0.0000000000000002 ***
PersonalitySentinel   0.135314   0.119920   1.128               0.2592    
Certficates           0.588869   0.017904  32.890 < 0.0000000000000002 ***
Feedback              0.714886   0.030847  23.176 < 0.0000000000000002 ***
log_Salary            1.671443   0.117290  14.251 < 0.0000000000000002 ***
log_Years             1.071837   0.043955  24.385 < 0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 18291  on 13193  degrees of freedom
Residual deviance: 12345  on 13182  degrees of freedom
AIC: 15058

Number of Fisher Scoring iterations: 5

   BusinessSoftware                 Age             Female1          CollegeYes 
           1.098278            1.260680            1.087572            1.120000 
PersonalityDiplomat PersonalityExplorer PersonalitySentinel         Certficates 
           5.051532            5.089706            2.128732            1.545891 
           Feedback          log_Salary           log_Years 
           1.310502            1.743951            1.102991 
```

```
                        Overall
BusinessSoftware     3.912793
Age                  1.739738
Female1              4.573267
CollegeYes           4.483631
PersonalityDiplomat 20.579872
PersonalityExplorer 20.550067
PersonalitySentinel  1.128369
Certficates         32.890051
Feedback            23.175507
log_Salary          14.250526
log_Years           24.384605
```

Still potential multicollinearity in Diplomat and Explorer. The most important variables are still certificates and feedback in realtion to the dependent variable.

Predict using Weights

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5343  405
         1 1675 1373

               Accuracy : 0.7635
                 95% CI : (0.7545, 0.7724)
    No Information Rate : 0.7979
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4212

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.7722
            Specificity : 0.7613
         Pos Pred Value : 0.4505
         Neg Pred Value : 0.9295
             Prevalence : 0.2021
         Detection Rate : 0.1561
   Detection Prevalence : 0.3465
      Balanced Accuracy : 0.7668

       'Positive' Class : 1


F1 Score:  0.5690012
```

A much better model compared to the unweighted logistic regression model. Sensitivity and specificity are a lot more balanced in this model. The F1 score and Balanced accuracy are also higher in this weighted model as well.

Logistic Regression Threshold Tuning + weighted

Train Using Threshold Tuning + weighted

```
OPTIMAL CUTOFF VALUE OF: 0.4021295

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4867  253
         1 2151 1525

               Accuracy : 0.7267
                 95% CI : (0.7172, 0.736)
    No Information Rate : 0.7979
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3941

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.8577
            Specificity : 0.6935
         Pos Pred Value : 0.4149
         Neg Pred Value : 0.9506
             Prevalence : 0.2021
         Detection Rate : 0.1734
   Detection Prevalence : 0.4179
      Balanced Accuracy : 0.7756

       'Positive' Class : 1


F1 Score:  0.5592226
```

Not as good of a model as just weighting, although it has higher sensitivity which represents the minority class, specificity drops and the model loses the balance of specificity and sensitivity the weighted model had. The F1 score is slightly lower in this model and the Balanced accuracy is slightly above the balanced accuracy in the weighted model. The Best model for Logistic Regression is the Logistic Model with just weighting.

Classification Trees

Partition Data

```
        0         1
```

```
0.7978433 0.2021567

        0         1 
0.7979078 0.2020922 
```

Full Tree

```
Full Tree cp table

           CP nsplit rel error xerror   xstd
1  0.0269923      0  1.000000  1.000 0.0160
2  0.0131748      4  0.891388  0.904 0.0154
3  0.0059447      5  0.878213  0.888 0.0153
4  0.0041774      7  0.866324  0.880 0.0153
5  0.0040167     10  0.851864  0.870 0.0152
6  0.0024904     12  0.843830  0.866 0.0151
7  0.0019280     18  0.821979  0.860 0.0151
8  0.0013496     19  0.820051  0.862 0.0151
9  0.0012853     26  0.810411  0.871 0.0152
10 0.0011247     37  0.793380  0.875 0.0152
11 0.0010711     54  0.774100  0.876 0.0152
12 0.0009640     57  0.770887  0.878 0.0152
13 0.0008837     72  0.756105  0.881 0.0153
14 0.0008569     76  0.752571  0.880 0.0153
15 0.0008033     82  0.747429  0.881 0.0153
16 0.0007498     89  0.741645  0.888 0.0153
17 0.0007230    114  0.721401  0.888 0.0153
18 0.0006427    125  0.710154  0.899 0.0154
19 0.0005623    200  0.659383  0.904 0.0154
20 0.0005356    206  0.655527  0.915 0.0155
21 0.0005021    247  0.632391  0.929 0.0156
22 0.0004820    270  0.619859  0.929 0.0156
23 0.0004499    378  0.562661  0.934 0.0156
24 0.0004284    394  0.553985  0.937 0.0156
25 0.0004131    414  0.545308  0.938 0.0156
26 0.0004017    439  0.533419  0.941 0.0156
27 0.0003856    453  0.527635  0.940 0.0156
28 0.0003749    464  0.523136  0.942 0.0157
29 0.0003213    483  0.514781  1.003 0.0160
30 0.0002892    989  0.348972  1.013 0.0161
31 0.0002812   1000  0.345758  1.027 0.0162
32 0.0002754   1050  0.329692  1.027 0.0162
33 0.0002678   1060  0.326799  1.030 0.0162
34 0.0002571   1120  0.303663  1.039 0.0162
35 0.0002472   1266  0.255463  1.045 0.0163
36 0.0002410   1284  0.250964  1.047 0.0163
37 0.0002142   1342  0.235219  1.068 0.0164
38 0.0001964   1519  0.194409  1.079 0.0165
39 0.0001928   1576  0.180591  1.083 0.0165
40 0.0001691   1610  0.172879  1.083 0.0165
41 0.0001607   1630  0.169344  1.144 0.0168
```

```
42 0.0001500   2272  0.060733  1.152 0.0168
43 0.0001428   2318  0.052057  1.157 0.0169
44 0.0001377   2360  0.043702  1.167 0.0169
45 0.0001285   2370  0.042095  1.169 0.0169
46 0.0001205   2401  0.037596  1.169 0.0169
47 0.0001168   2409  0.036632  1.195 0.0171
48 0.0001071   2420  0.035347  1.195 0.0171
49 0.0000964   2653  0.007069  1.198 0.0171
50 0.0000918   2663  0.006105  1.198 0.0171
51 0.0000803   2698  0.002249  1.202 0.0171
52 0.0000000   2718  0.000643  1.203 0.0171
```

```
Unweighted Variable Importance

               Overall
Age          2194.4497
Business      564.5281
Certficates  1552.8733
College       462.4661
Feedback     2865.2052
Female        531.2570
Personality  1214.6814
Salary       3108.4842
Years        1524.8041
```

Predict Full Tree

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4470  758
         1  793  575

               Accuracy : 0.7649
                 95% CI : (0.7544, 0.775)
    No Information Rate : 0.7979
    P-Value [Acc > NIR] : 1.000

                  Kappa : 0.278

 Mcnemar's Test P-Value : 0.388

            Sensitivity : 0.43136
            Specificity : 0.84933
         Pos Pred Value : 0.42032
         Neg Pred Value : 0.85501
             Prevalence : 0.20209
         Detection Rate : 0.08717
```

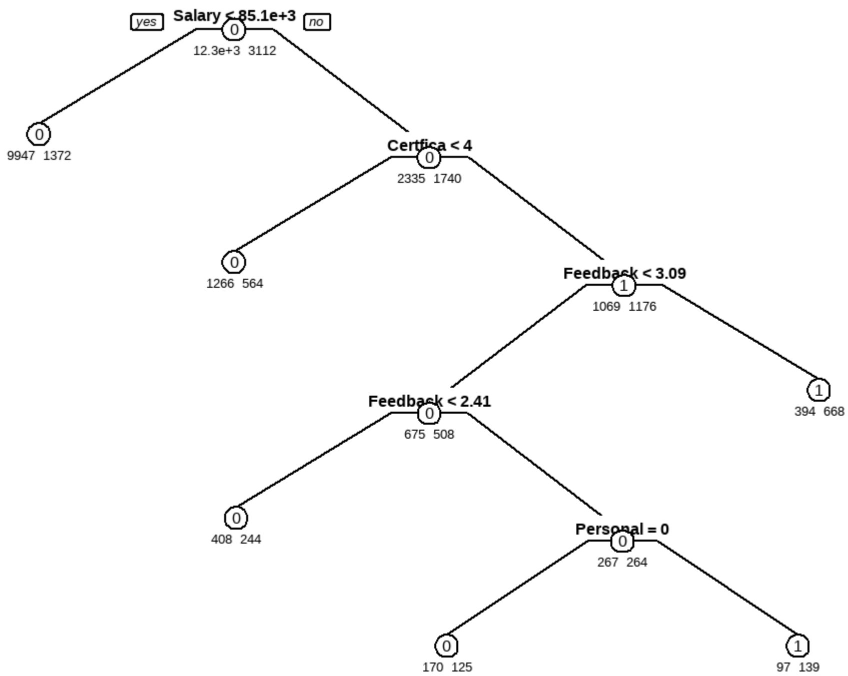
```
     Detection Prevalence : 0.20740
        Balanced Accuracy : 0.64034

         'Positive' Class : 1

F1 Score:  0.4257682
```

The Full Tree struggles with the class imbalance in the dataset as the specificity value is much higher than the sensitivity value.

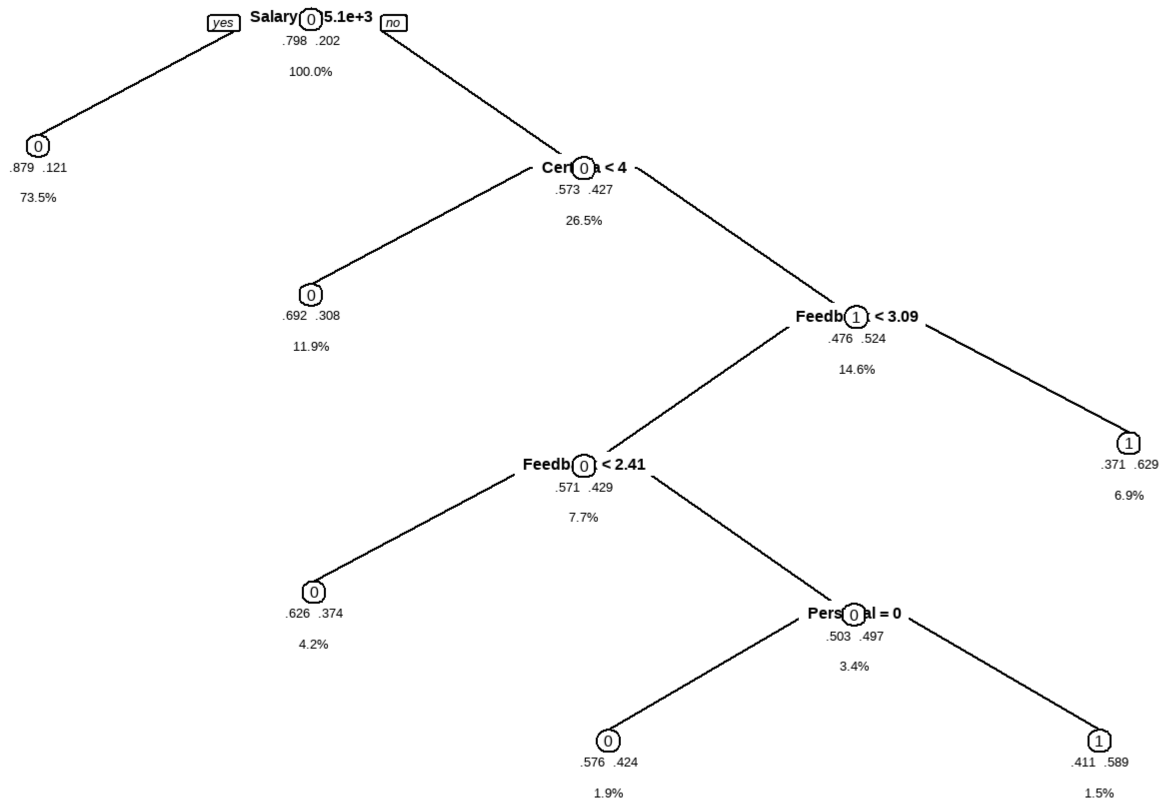
Best Pruned Tree

Best Pruned Tree Unweighted

```
unweighted best pruned cptable

           CP nsplit rel error
1 0.029348757      0 1.0000000
2 0.006748072      3 0.9119537
3 0.005944730      5 0.8984576

Variable Importance

rpart variable importance

                     Overall
Salary               100.000
Certficates           79.131
Feedback              43.372
Years                 24.696
PersonalitySentinel   22.220
PersonalityDiplomat    3.850
Female1                1.661
BusinessSoftware       0.000
PersonalityExplorer    0.000
CollegeYes             0.000
Age                    0.000


TREE DIAGRAM WITH NODE COUNTS
```



TREE DIAGRAM SHOWING PROBABILTIES AND NODE PROPORTION



Predict Best Pruned Unweighted Tree

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5059  987
         1  204  346

               Accuracy : 0.8194
                 95% CI : (0.8099, 0.8287)
    No Information Rate : 0.7979
    P-Value [Acc > NIR] : 0.000005512

                  Kappa : 0.2828

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.25956
            Specificity : 0.96124
         Pos Pred Value : 0.62909
         Neg Pred Value : 0.83675
             Prevalence : 0.20209
         Detection Rate : 0.05246
```

```
    Detection Prevalence : 0.08338
       Balanced Accuracy : 0.61040

        'Positive' Class : 1

F1 Score:  0.3674987
```

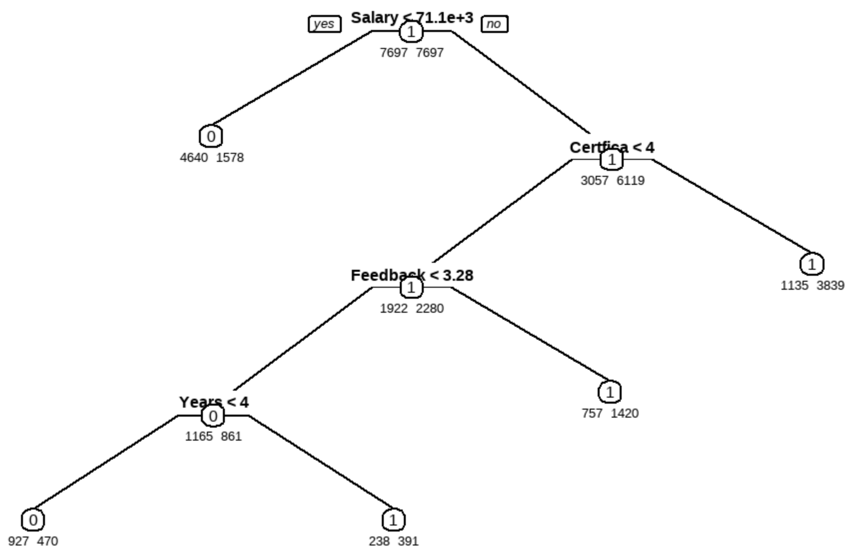
Comments on best pruned unweighted tree:
This model is showing problems representing the minority class. This can be seen in the very low sensitivity value. This model is also overaclassifying the majority class as can be seen in the very high specificity number. This model also has a lower f1 score than the full tree. It is important to note that the full tree having higher accuracy or f1 score can be expected as it can overfit to the data it is being trained on.

Best Pruned Tree Weighted

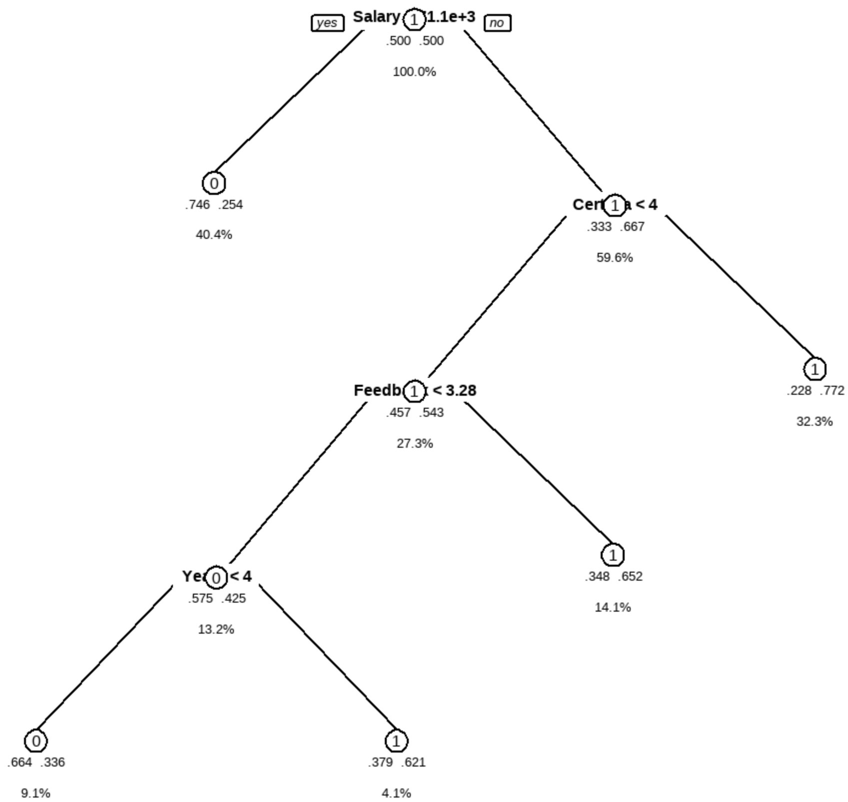
Train Best Pruned Tree Weighted

```
Class Counts (n) for Admitted from the Training Dataset



    0     1
12282  3112



Class Total: 15394

Weighted Best Pruned Tree cp Table

           CP nsplit rel error
1 0.397820561      0 1.0000000
2 0.019767260      1 0.6021794
3 0.006748072      4 0.5428133


Weighted Variable Importance

rpart variable importance

                    Overall
Salary               100.00
Certficates           80.62
Feedback              38.21
PersonalitySentinel   29.79
Years                 29.47
PersonalityExplorer    0.00
CollegeYes             0.00
PersonalityDiplomat    0.00
BusinessSoftware       0.00
Age                    0.00
Female1                0.00
```

TREE DIAGRAM WITH NODE COUNTS



TREE DIAGRAM SHOWING PROBABILTIES AND NODE PROPORTION



Predict Best Pruned Tree Weighted

```
CONFUSION MATRIX AT DEFAULT CUTOFF VALUE

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 3824  333
         1 1439 1000

               Accuracy : 0.7314
                 95% CI : (0.7205, 0.742)
    No Information Rate : 0.7979
    P-Value [Acc > NIR] : 1

                  Kappa : 0.364

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.7502
            Specificity : 0.7266
         Pos Pred Value : 0.4100
```

```
         Neg Pred Value : 0.9199
             Prevalence : 0.2021
         Detection Rate : 0.1516
   Detection Prevalence : 0.3698
      Balanced Accuracy : 0.7384

       'Positive' Class : 1

F1 Score:  0.5302227
```

Comments on Weighted Best Pruned Tree:
This model is the best model among the different classification tree models. The class distinction is very good as specificity and sensitivity are very close together. This model also has the highest balanced accuracy and F1 score.

Evaluation

Model Comparison: Choosing Best Model

MODEL PERFORMANCE COMPARISON

Metric	KNN_Threshold	NB_Threshold	LR_Weights	BP_weighted
Accuracy	0.703	0.725	0.764	0.731
Kappa	0.336	0.363	0.421	0.364
Sensitivity	0.784	0.776	0.772	0.750
Specificity	0.682	0.712	0.761	0.727
Pos Pred Value	0.385	0.405	0.450	0.410
Prevalence	0.202	0.202	0.202	0.202
Detection Rate	0.158	0.157	0.156	0.152
Balanced Accuracy	0.733	0.744	0.767	0.738
F1	0.516	0.532	0.569	0.530

Comments on Best Model:
Based on the confusion matrix metrics and F1 score of the best models from the different classification models used, the chosen best model to futher evaluate is the Logistic Regression model using weighting.

Logistic Regression Deep Dive into Confusion Matrix and F1 Score

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5343  405
         1 1675 1373
```

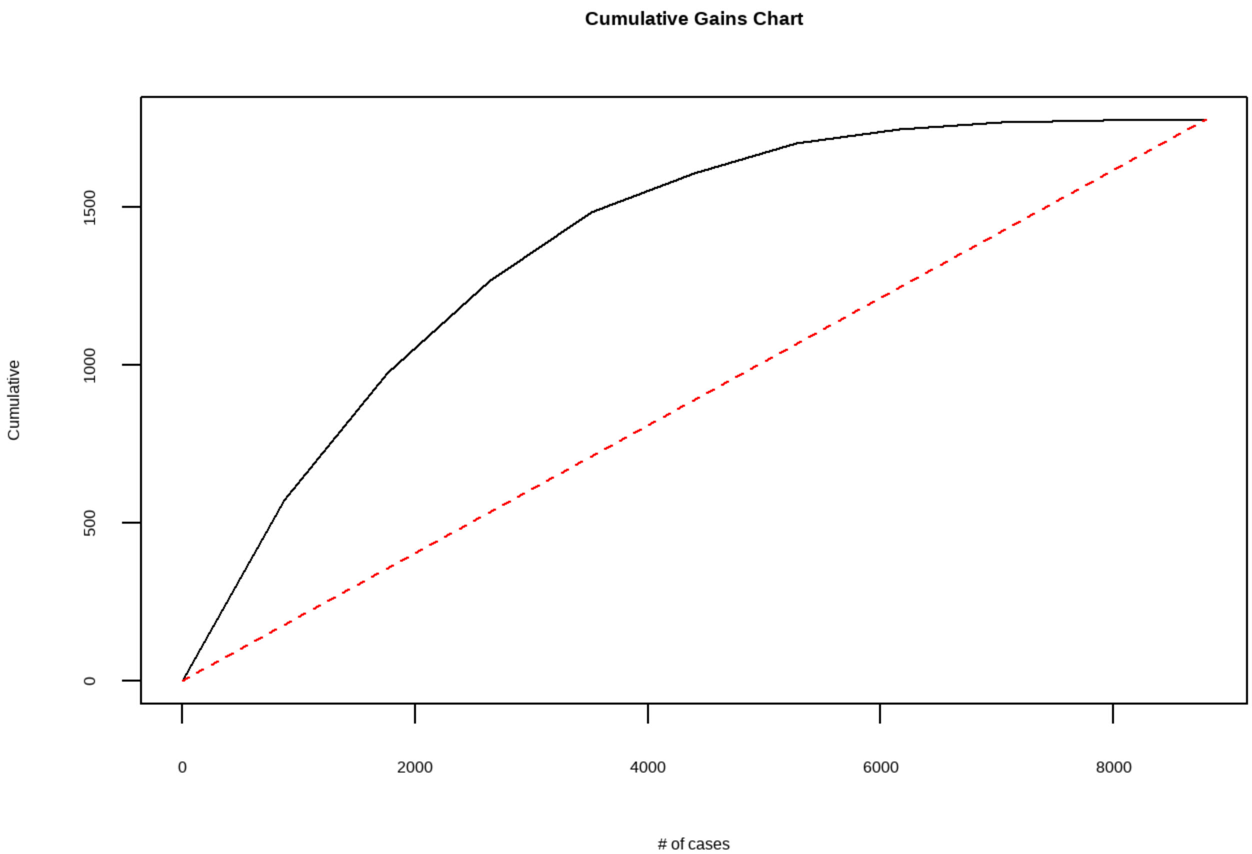
```
               Accuracy : 0.7635
                 95% CI : (0.7545, 0.7724)
    No Information Rate : 0.7979
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4212

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.7722
            Specificity : 0.7613
         Pos Pred Value : 0.4505
         Neg Pred Value : 0.9295
             Prevalence : 0.2021
         Detection Rate : 0.1561
   Detection Prevalence : 0.3465
      Balanced Accuracy : 0.7668

       'Positive' Class : 1

F1 Score:  0.5690012
```

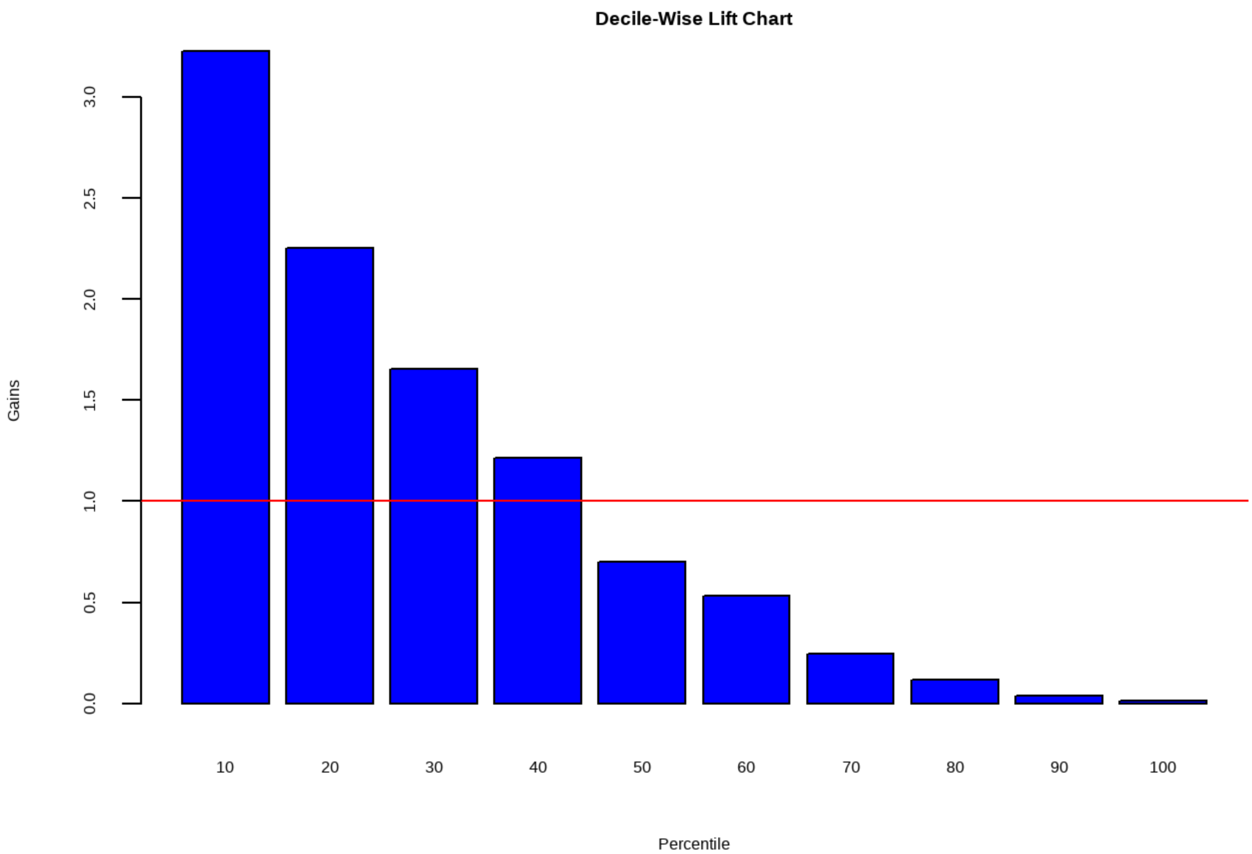
- Accuracy (0.7635) – 76.35% of predictions were correct.
- 95% CI (0.7545 – 0.7724) – The true accuracy likely lies between 75.45% and 77.24%.
- No Information Rate (0.7979) – The majority (non-positive) class makes up 79.79% of the data, meaning a model that always predicts “0” would already achieve about 80% accuracy.
- P-Value (1) – The model’s accuracy is not statistically better than simply predicting the majority class every time.
- Kappa (0.4212) – Shows moderate agreement between predicted and actual outcomes beyond random chance.
- McNemar’s Test (p < 0.0000000000000000002) – Indicates a significant difference between the types of errors the model makes (false positives vs. false negatives).
- Sensitivity (0.7722) – The model correctly identified 77.22% of the positive (1) cases. This means it successfully detects most of the positives.
- Specificity (0.7613) – The model correctly identified 76.13% of the negative (0) cases, showing good ability to distinguish between the two classes.
- Pos Pred Value (Precision, 0.4505) – Of all cases predicted as positive, only 45.05% were actually positive. This suggests the model produces a fair number of false positives.
- Neg Pred Value (0.9295) – Of all cases predicted as negative, 92.95% were actually negative, indicating strong reliability when the model predicts “0.”

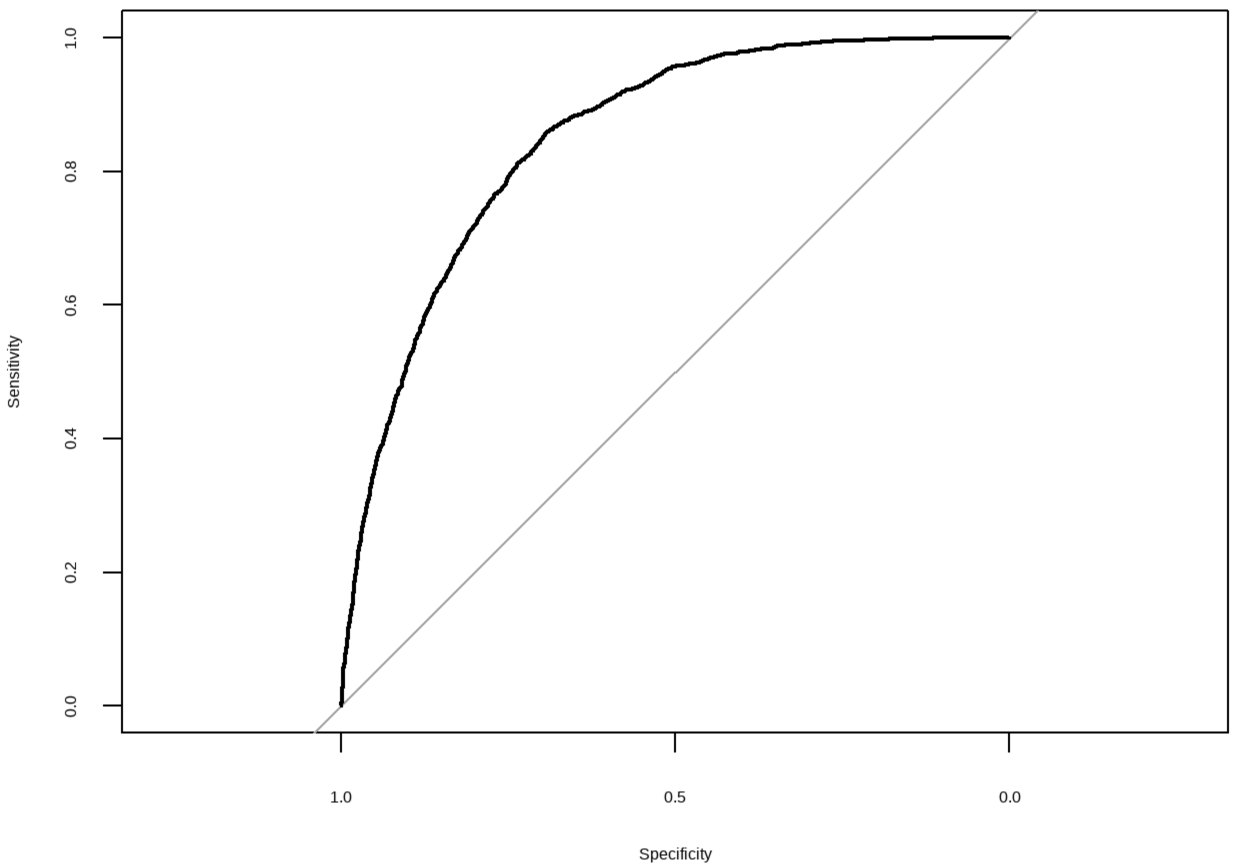
- Prevalence (0.2021) – The positive class makes up 20.21% of the dataset, confirming the data are imbalanced.
- Detection Rate (0.1561) – About 15.61% of all samples were correctly identified as belonging to the positive class.
- Detection Prevalence (0.3465) – Roughly 34.65% of cases were predicted as positive, regardless of correctness, showing the model predicts more positives than truly exist.
- Balanced Accuracy (0.7668) – Averaging sensitivity and specificity, the model correctly identifies both classes about 76.7% of the time, showing strong overall balance.
- F1 Score (0.5690) – Indicates moderate balance between precision and recall. The model finds many of the positive cases but still mislabels some negatives as positives.

Model Evaluation Charts

```
Depth                              Cume   Cume Pct                     Mean
 of           Cume      Mean       Mean   of Total    Lift   Cume     Model
File     N       N      Resp       Resp       Resp   Index   Lift     Score
---------------------------------------------------------------------------
  10   879    879       0.65       0.65      32.2%     322    322      0.91
  20   880   1759       0.46       0.55      54.8%     225    274      0.77
  30   879   2638       0.33       0.48      71.3%     165    238      0.63
  40   880   3518       0.25       0.42      83.5%     121    209      0.49
  50   880   4398       0.14       0.37      90.5%      70    181      0.37
  60   879   5277       0.11       0.32      95.8%      53    160      0.26
  70   880   6157       0.05       0.28      98.3%      25    140      0.18
  80   879   7036       0.02       0.25      99.5%      12    124      0.11
  90   880   7916       0.01       0.22      99.9%       4    111      0.06
 100   880   8796       0.00       0.20     100.0%       1    100      0.02
```


Cumulative Gains Chart
Cumulative
0
500
1000
1500
0
2000
4000
6000
8000
# of cases


Decile-Wise Lift Chart
Gains
3.0
2.5
2.0
1.5
1.0
0.5
0.0
10
20
30
40
50
60
70
80
90
100
Percentile




```
Area under the curve: 0.8494
```

Comments on Model Evaluation Charts:
The evaluation results demonstrate that the model effectively ranks cases by their likelihood of belonging to the positive class. The gains table indicates that most of the model’s predictive strength is concentrated in the top-ranked cases, meaning the model identifies a large share of positive outcomes early on.

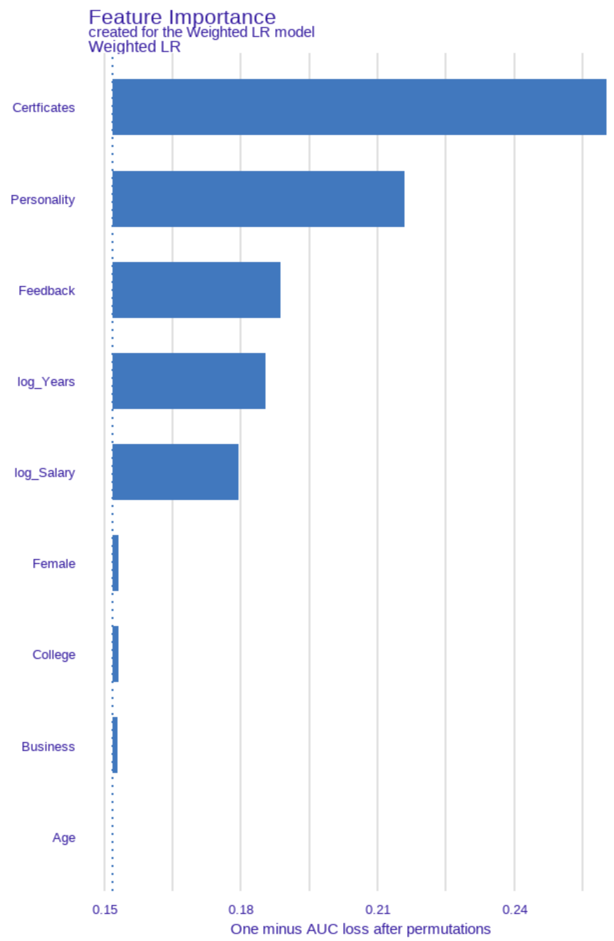
The Cumulative Gains Chart reinforces this, as the model’s curve (black line) rises well above the baseline (red dashed line), confirming that it performs substantially better than random guessing. Similarly, the Decile-Wise Lift Chart shows that the first few deciles have lift values greater than 1, meaning these top portions of the data contain a disproportionately high number of actual positive cases.

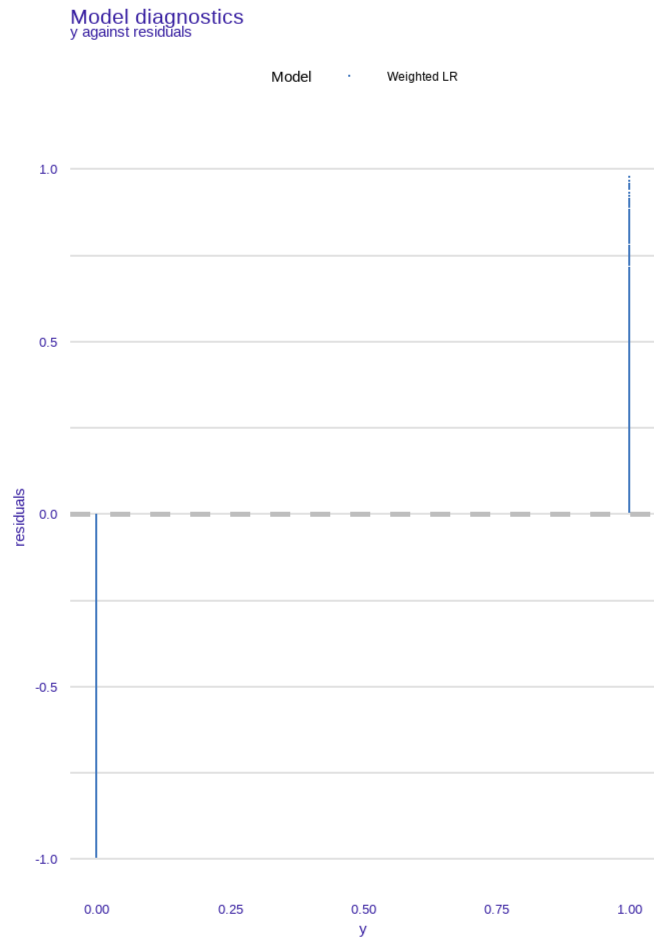
Finally, the ROC Curve and AUC score of 0.8494 indicate that the model has strong discriminatory power—it can reliably distinguish between positive and negative outcomes across various thresholds. Overall, these results confirm that the model performs effectively in identifying and ranking likely positive cases.

Dalex Graph

```
Preparation of a new explainer is initiated
  -> model label       :  Weighted LR
  -> data              :  8796  rows  9  cols
  -> target variable   :  8796  values
  -> predict function  :  pfun
  -> predicted values  :  No value for predict function target column. (  default  )
```

```
-> model_info        :  package caret , ver. 6.0.94 , task classification (  default  )
-> predicted values  :  numerical, min =  0.001460258 , mean =  0.3792872 , max =  0.994452
-> residual function :  difference between y and yhat (  default  )
-> residuals         :  numerical, min =  -0.994452 , mean =  -0.1771498 , max =  0.9781912
A new explainer has been created!
```





Comments on Dalex Chart:
The Feature Importance chart shows that the model most important variables are Certificates and Personality when distinguishing between the two outcome classes. These predictors contribute the most to improving classification accuracy when permuted. In contrast, variables such as Age, Business, and Female have minimal influence on the model's predictions, indicating they add little explanatory power once the stronger predictors are considered.

From the Model Diagnostics (Residuals) plot, we see that most residuals cluster near 0, with a few points extending toward the upper limit (+1). This pattern suggests that while the model predicts many observations accurately, there are some cases it systematically misclassifies. This aligns with the dataset's class imbalance, where class 0 dominates the other.

Deployment

The model shows strong overall discrimination Auc = 0.85 and performs better than random chance, indicating it's suitable for limited deployment or pilot testing. Key predictors such as Certificates and Personality drive most of the model's predictive power, providing useful insights for decision-making.

However, residual bias toward the majority class suggests the model may underperform on minority outcomes, so adjustments like class weighting or threshold tuning are recommended. Given its interpretability and stability, the model is deployment-ready but should be monitored and retrained to help mitigate class imbalance.

Citations

```
R Version Information:

[1] "R version 4.4.3 (2025-02-28 ucrt)"

      Package Version
         xfun    0.53
       readxl   1.4.5
    tidyverse   2.0.0
        dplyr   1.1.4
      ggplot2   3.5.2
 DataExplorer   0.8.3
       dlookr   0.6.3
        caret  6.0.94
         pROC  1.18.5
        gains     1.2
    gridExtra     2.3
      janitor   2.2.0
 summarytools   1.0.1
        psych  2.4.12
        e1071  1.7.16
    scorecard   0.4.5
   woeBinning   0.1.6
         klaR   1.7.3
        rpart  4.1.24
   rpart.plot   3.1.3
        DALEX   2.5.2


Source Citation:

ChatGPT (GPT-5, OpenAI). (2025). Assistance with R coding and model interpretation. Retrieved
from https://chat.openai.com/
```