Prediction of Job Offer Acceptance: A HR Data Analysis

ABSTRACT

This project aims to analyze a human resources dataset to predict the likelihood of a person joining a company based on various factors. The target variable in the study, which indicates whether a person will join the company or not, is framed as a binary classification problem. The analysis creates and assesses predictive models using a variety of classification techniques. The study's results are presented in the report, along with information on the models' precision and the significance of various variables in outcome prediction.

The dataset used in this study includes details on applicants who submitted job applications to a company, such as their notice period, gender, and offered salaries. The project investigates these variables' ability to predict a candidate's likelihood of accepting a job offer. Based on the dataset used in this study, the study also determines the best classification algorithm for predicting candidate acceptance.

The project has implications for HR recruitment procedures by shedding light on the variables that affect a person's choice to work for a particular organization. The results of this study can be applied to enhance hiring practices, draw in qualified applicants, and ultimately raise the success rate of job offers.

INTRODUCTION AND OVERVIEW

The dataset consists of 8,996 observations. The dataset has been filtered to include only the required columns for analysis. The dependent variable for this dataset is the "Joining status," which can be considered as the outcome variable that will be predicted based on the independent variables. There are ten independent variables that have been selected for this analysis. The independent variables are age, gender, bonus offered or not etc.

To prepare the dataset for analysis, non-numerical categorical variables have been converted to numerical variables. We have used Linear regression, KNN, Random forest, Boosting and SVM to determine which technique provides a better fit for this particular dataset.

EXPLORATORY DATA ANALYSIS

In this project, we have a dataset containing both numerical and categorical variables.

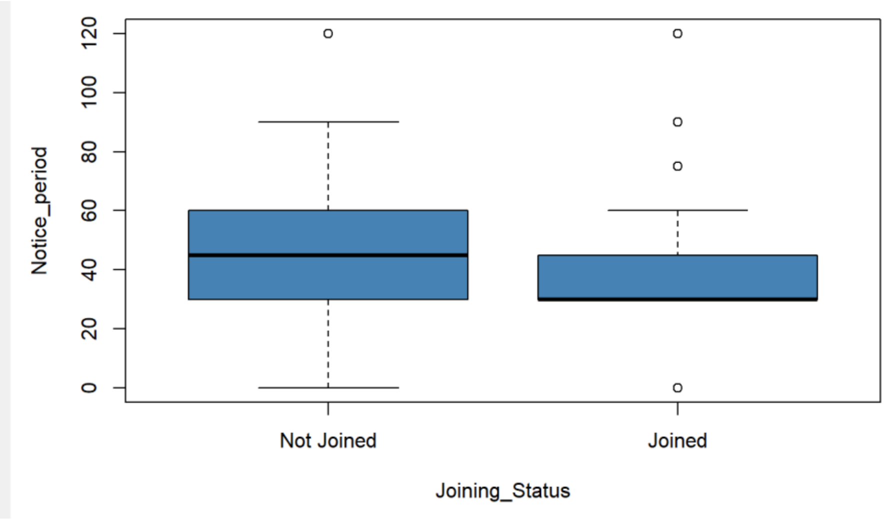
Initially, we looked for missing values in the dataset and found none. As a result, we could proceed with the analysis without having to replace or remove missing data.

The categorical variables in the dataset were then selected and transformed to factors. This step allows us to consider categorical variables as factors and do categorical data analysis.

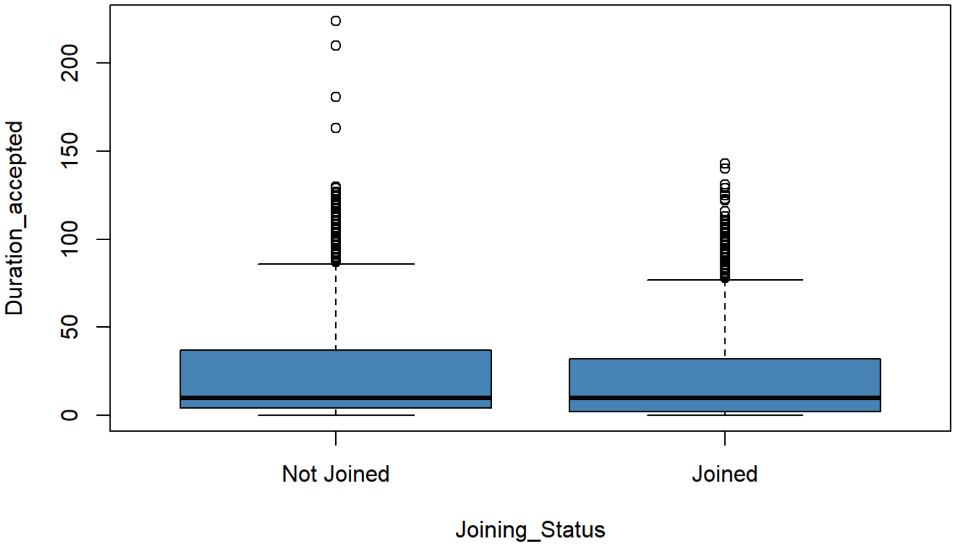
For the numerical variables, we created boxplots to visualize the distribution of the variables. The boxplots provide information about the variables' range, median, quartiles, and outliers. We may use these visualizations to discover and examine any odd observations.

We generated bar graphs to represent the frequency of each category for the categorical variables. The bar graphs provide information about the distribution of the variables and assist us in identifying any imbalances or unusual categories.

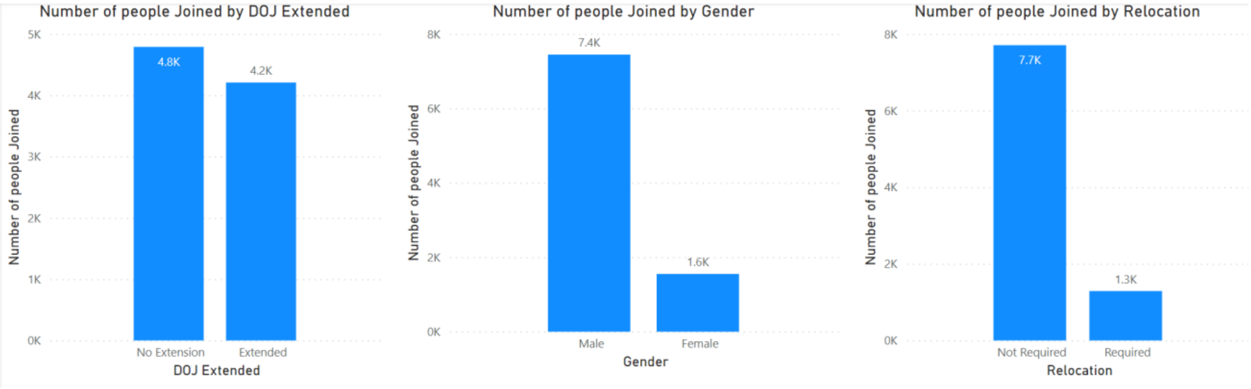
Lastly, we divided the dataset into two parts: training and testing. The training data accounts for 80% of the dataset, whereas the testing data accounts for 20%.



By examining the boxplot, it is evident that candidates who had a shorter notice period had a higher likelihood of joining compared to candidates with a longer notice period.



It can be observed that candidates who were offered acceptance later did not end up joining the company.



Additionally, a significant proportion of candidates who postponed their joining date did not ultimately join the company. A majority of female candidates did not end up joining the company, this may be due to the less female candidates overall. A higher number of candidates who did not need to relocate joined the company compared to those who required relocation.

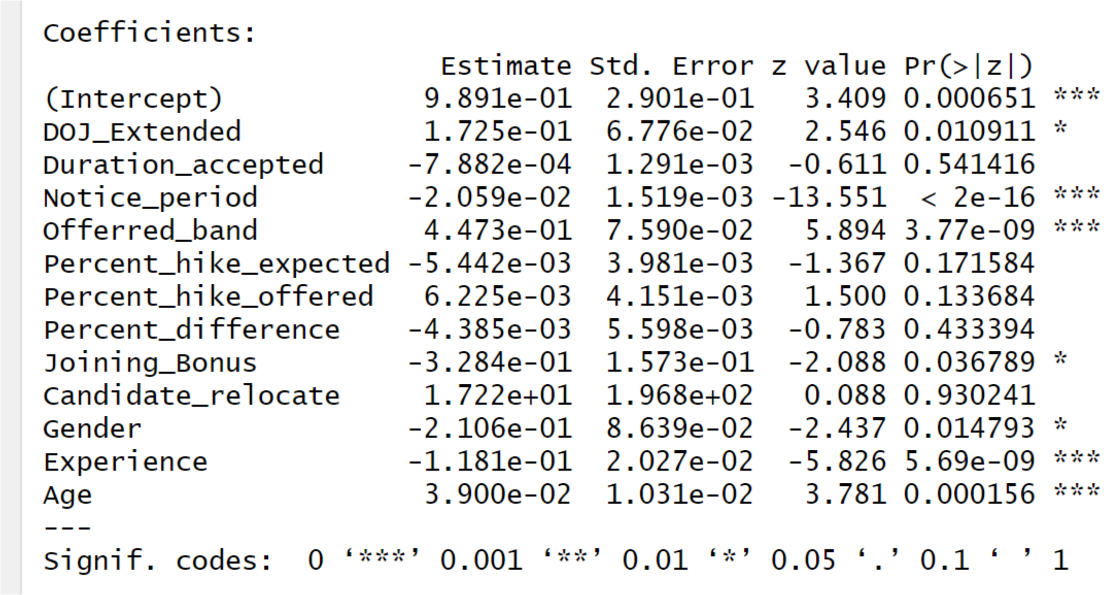
LOGISTIC REGRESSION

In order to model the relationship between our binary response variable and predictor variables, we utilized logistic regression. One of the reasons we chose this method is because our data contains outliers, and logistic regression is known to be less sensitive to outliers than other classification methods such as Linear Discriminant Analysis (LDA).

Furthermore, another reason for choosing logistic regression over LDA is that logistic regression provides an easy interpretation of the results, making it a popular choice for binary classification tasks.

Logistic regression is a popular statistical method for modeling binary dependent variables, and is particularly useful when the goal is to predict the probability of a certain event occurring. The method involves fitting a logistic function to the data in order to estimate the probability of the event of interest. The logistic function is a mathematical function that maps any input value to a value between 0 and 1, which is ideal for modeling probabilities.

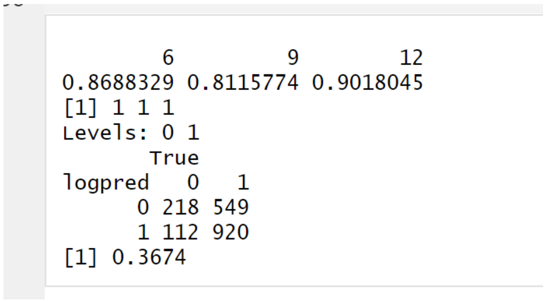
Overall, given the presence of outliers in our data and the ease of interpretation offered by logistic regression, we concluded that this method was the most suitable choice for our analysis.



```
Coefficients:
                          Estimate Std. Error z value Pr(>|z|)    
(Intercept)              9.891e-01  2.901e-01   3.409 0.000651 ***
DOJ_Extended             1.725e-01  6.776e-02   2.546 0.010911 *  
Duration_accepted       -7.882e-04  1.291e-03  -0.611 0.541416    
Notice_period           -2.059e-02  1.519e-03 -13.551  < 2e-16 ***
Offerred_band            4.473e-01  7.590e-02   5.894 3.77e-09 ***
Percent_hike_expected -5.442e-03  3.981e-03  -1.367 0.171584    
Percent_hike_offered     6.225e-03  4.151e-03   1.500 0.133684    
Percent_difference      -4.385e-03  5.598e-03  -0.783 0.433394    
Joining_Bonus           -3.284e-01  1.573e-01  -2.088 0.036789 *  
Candidate_relocate       1.722e+01  1.968e+02   0.088 0.930241    
Gender                  -2.106e-01  8.639e-02  -2.437 0.014793 *  
Experience              -1.181e-01  2.027e-02  -5.826 5.69e-09 ***
Age                      3.900e-02  1.031e-02   3.781 0.000156 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

From the logistic regression, we can observe that the variables Notice period,Offered band, Experience and age are highly significant.

Tthe model had an accuracy rate of 63%. However, the error rate was 36%, which indicates that a significant proportion of our predictions were incorrect.



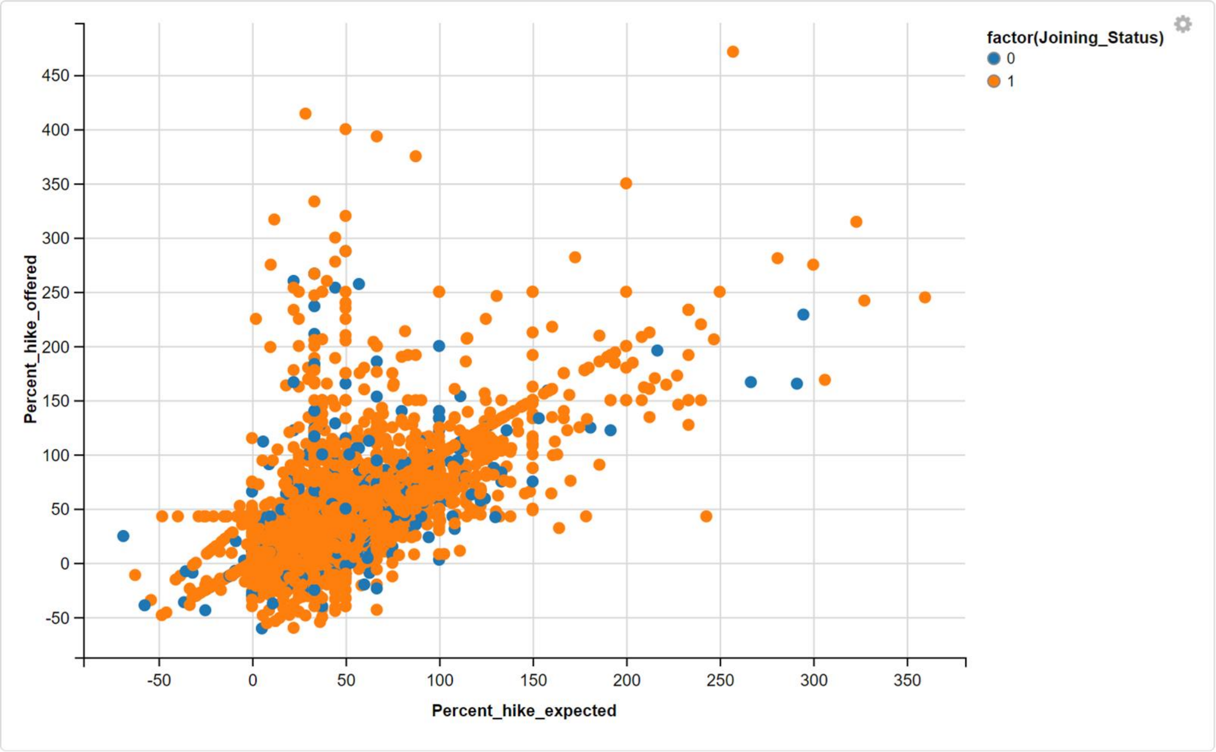
```
        6         9        12 
0.8688329 0.8115774 0.9018045 
[1] 1 1 1
Levels: 0 1
       True
logpred   0   1
      0 218 549
      1 112 920
[1] 0.3674
```

In this case, the confusion matrix shows that the model predicted 330 observations as positive (1) and 767 observations as negative (0). However, the actual labels reveal that there are 330 true positives (predicted positive and actually positive), 549 false negatives (predicted negative but actually positive), 218 true negatives (predicted negative and actually negative), and 920 false positives (predicted positive but actually negative).

This means that the model made incorrect predictions for over one-third of the observations, which indicates that it may not be a highly accurate model.

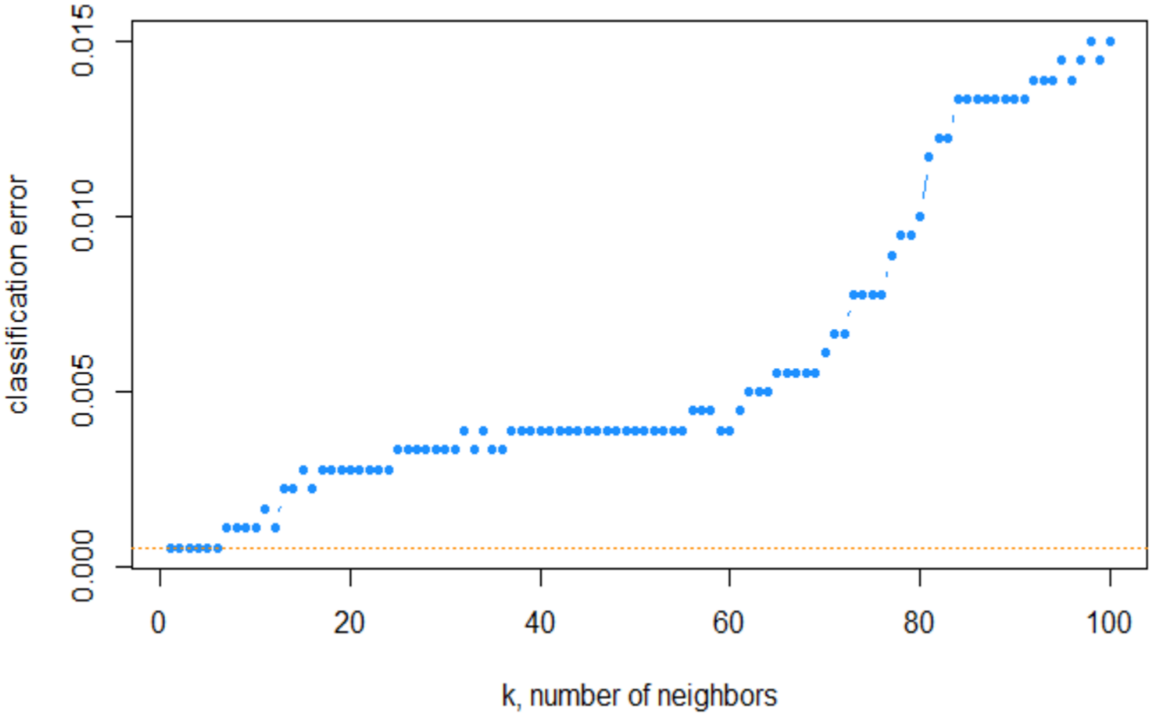
KNN CLASSIFICATION

K- nearest neighbhors classification(KNN) is a supervised and non parametric model. It doens't consider any model parameters and the distribution of the data. It assumes that similar data points are found near each other by calculating the distance between the datapoints. The class labels are assigned by calculating this distance to predict the model.

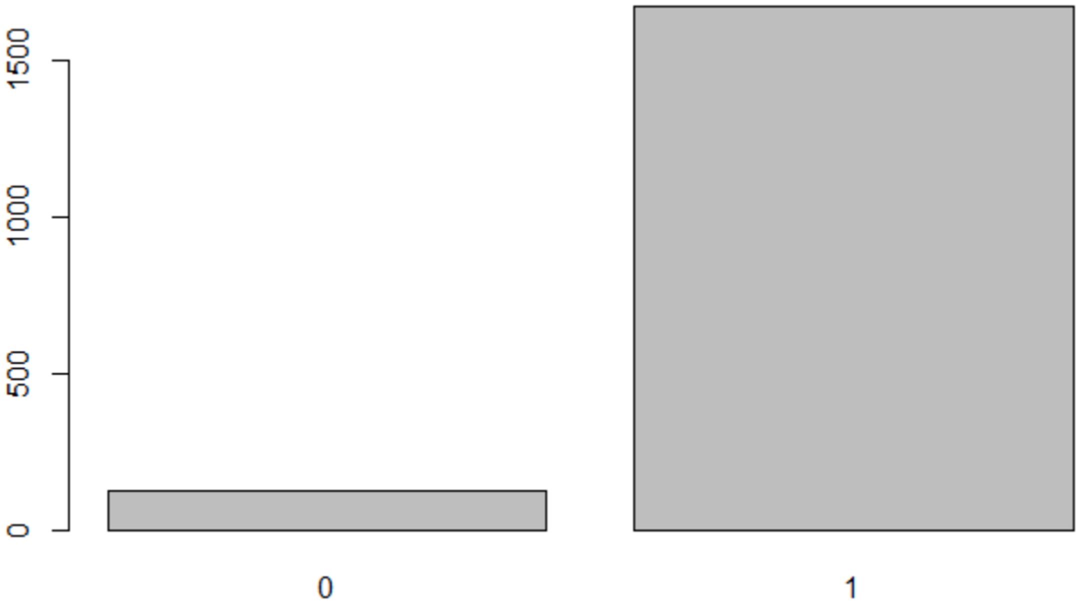


It has a tuning parameter "k" which can be adjusted for the effective performance of the model. Choosing k value is one of the key role since taking higher k value than required can lead to overfitting of the data. These values are always assumed to be a odd number. These are chosen using several techniques. We can either use trial and error method and compoare the accuracy and error rate and adjust the k value. Or we can use Grid search or the cross validation technique.

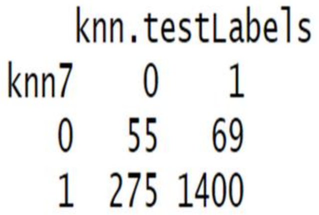
We used cross validation technique to choose the k value and the optimal k value obtained is k=7.



The above graph describes the k values and their respective error rates. We can observe that the lowest values of k has the least error. Using cross validation that lowest k value we obtained is 7.



If we observe the above graph of KNN between two variables, there are many points in orange color than in the blue. Hence we can say that our data is more inclined towards 1. It is an imbalance data.



```
     knn.testLabels
knn7    0    1
   0   55   69
   1  275 1400
```

Above is the confusion matrix obtained from the KNN model.

The confusion matrix describes that:

55- There are 55 values which are correctly predicted to be 0.

1400- There are 1400 values which are correctly predicted to be 1.

275- There are 275 values which are falsely predicted to be 0.

69- There are 69 values which are falsely predicted to be 1.

By looking at these values, we can observe that the percentage of TN and TP varies too much. Because of the imabalnced data and KNN model not predicted the imbalance data and hence the difference is occurred.

For this model our error rate obtained is 19.21%

The accuracy obtained is 80.87%.

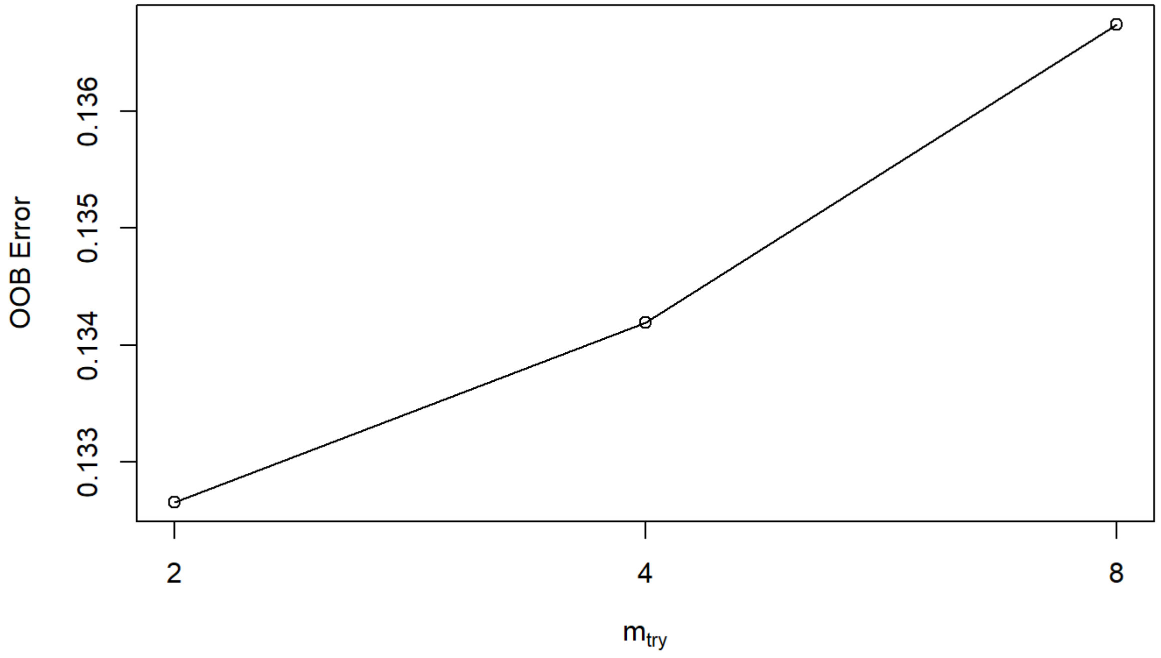
Here, even though we can observe the high accuracy and low error rate, it is falsely predicting the values. Hence, we can conclude that KNN model is poorly performing on our data.

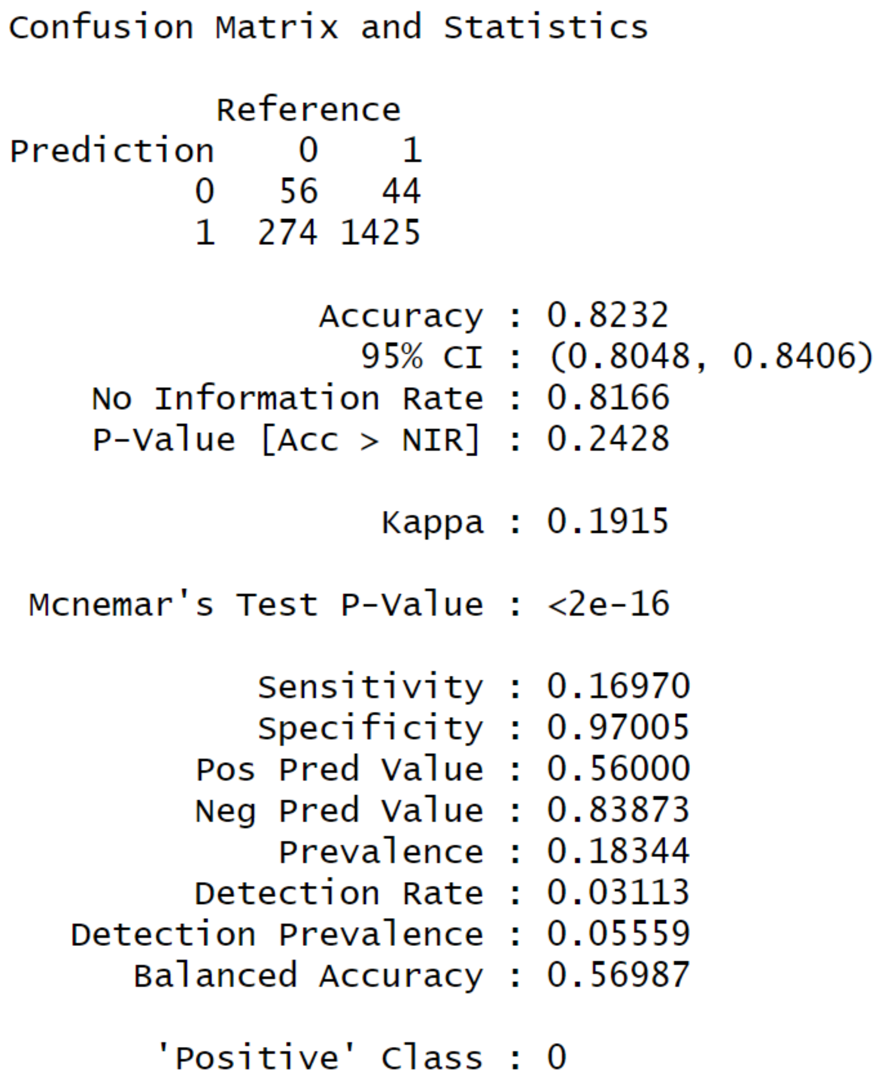
RANDOM FOREST MODEL:

Since our data is very imbalanced we tried to use Random forest method, as it performs well with the imbalanced data.

In random forest, each decision tree is constructed randomly by creating multiple trees and and combines to form predictions.

```
Call:
 randomForest(formula = factor(Joining_Status) ~ ., data = train.df,      importance =
TRUE, proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 17.77%
Confusion matrix:
    0    1 class.error
0 235 1117  0.82618343
1 162 5682  0.02772074
mtry = 4  OOB error = 0.1341913
Searching left ...
mtry = 2        OOB error = 0.1326594
0.01141584 0.05
Searching right ...
mtry = 8        OOB error = 0.1367379
-0.01897707 0.05
  mtry  OOBError
2    2 0.1326594
4    4 0.1341913
8    8 0.1367379
2
2
```





```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0   56   44
         1  274 1425

               Accuracy : 0.8232
                 95% CI : (0.8048, 0.8406)
    No Information Rate : 0.8166
    P-Value [Acc > NIR] : 0.2428

                  Kappa : 0.1915

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.16970
            Specificity : 0.97005
         Pos Pred Value : 0.56000
         Neg Pred Value : 0.83873
             Prevalence : 0.18344
         Detection Rate : 0.03113
   Detection Prevalence : 0.05559
      Balanced Accuracy : 0.56987

       'Positive' Class : 0
```

Above is the confusion matrix obtained from the Random forest model.

The confusion matrix describes that:

56- There are 56 values which are correctly predicted to be 0.

1425- There are 1425 values which are correctly predicted to be 1.

274- There are 274 values which are falsely predicted to be 0.

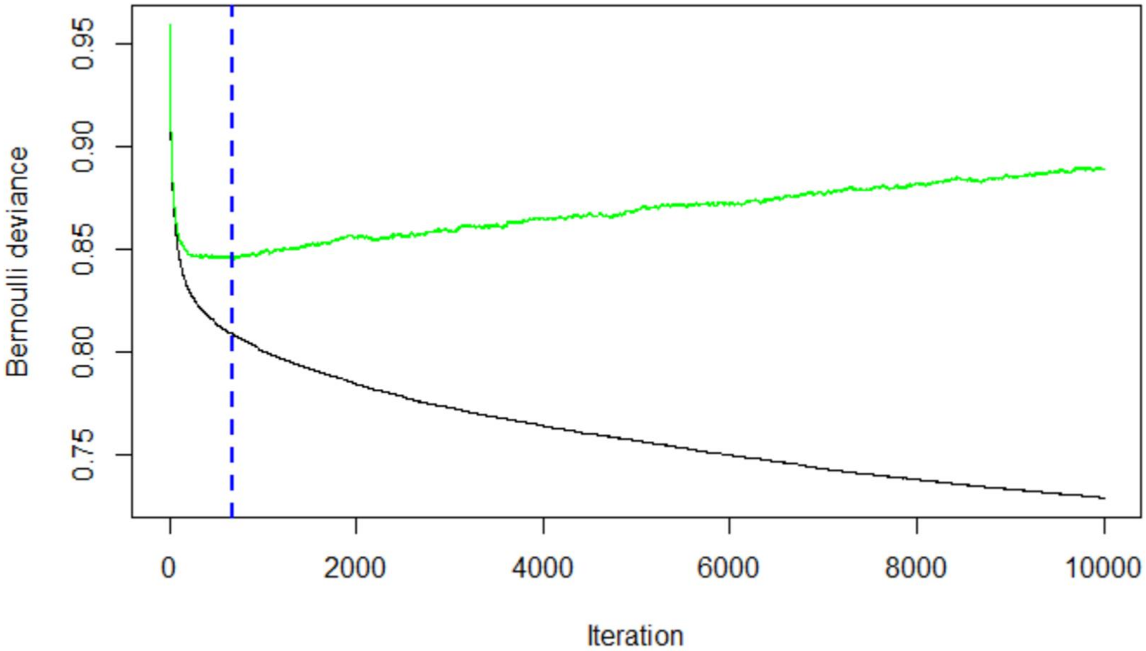
44- There are 44 values which are falsely predicted to be 1.

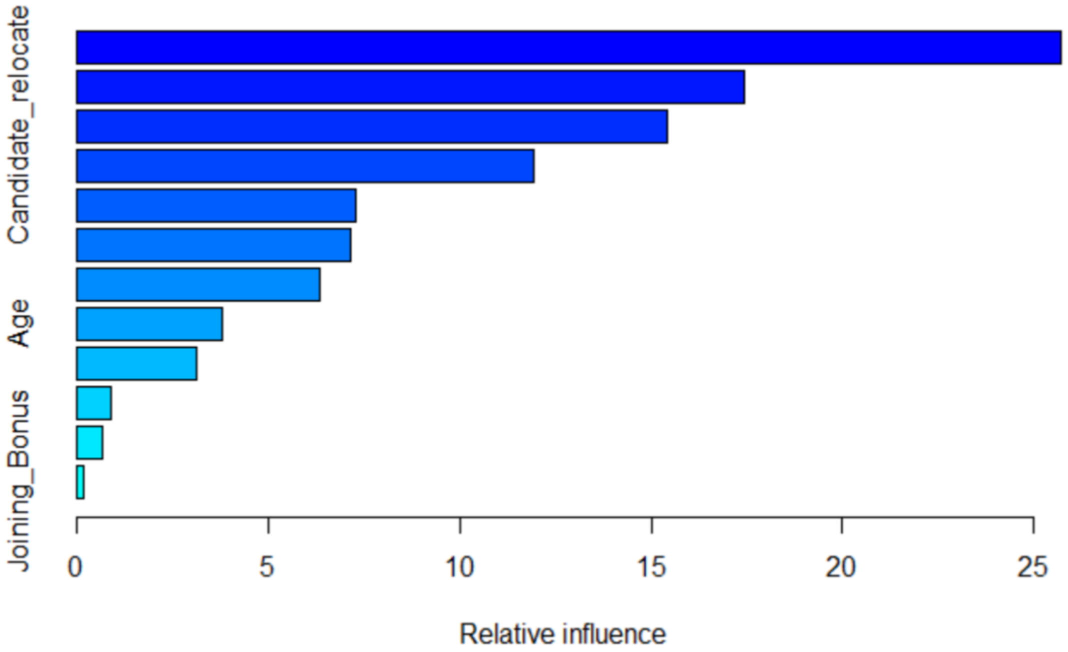
Using Random forest model, the error rate is obtained as 0.179 and the accuracy is 82%.

BOOSTING MODEL

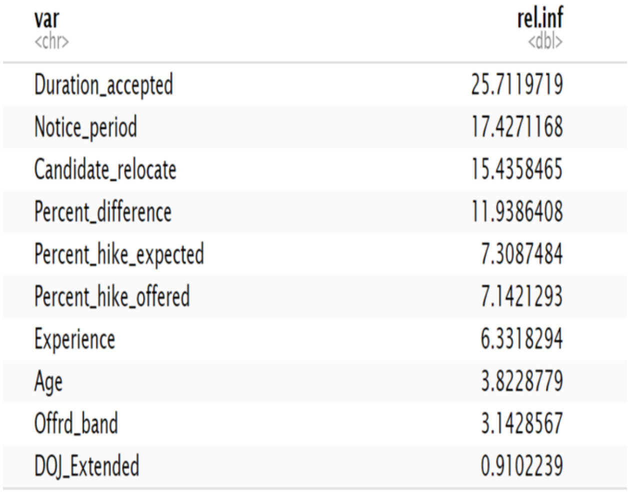
Boosting is a classification technique where multiple nodes are combined to forma single model that can make better predictions.

We choose boosting because it performs well with imbalanced datasets by stabilizing the data. In Boosting we have hyper parameters, where adjusting their values and using optimal values of hyperparameters gives us the optimal model. One of the hyper parameter is ntree which is number of trees used in building the model. Choosing optimal ntree values is crucial because using higher order of ntree can overfit the data. We choose cross validation technique to choose optimal ntree value. From the below graph it is observed that the optimal ntree lies in between 500 to 1000. Thus by using cross validation exact ntree value is given as 678.

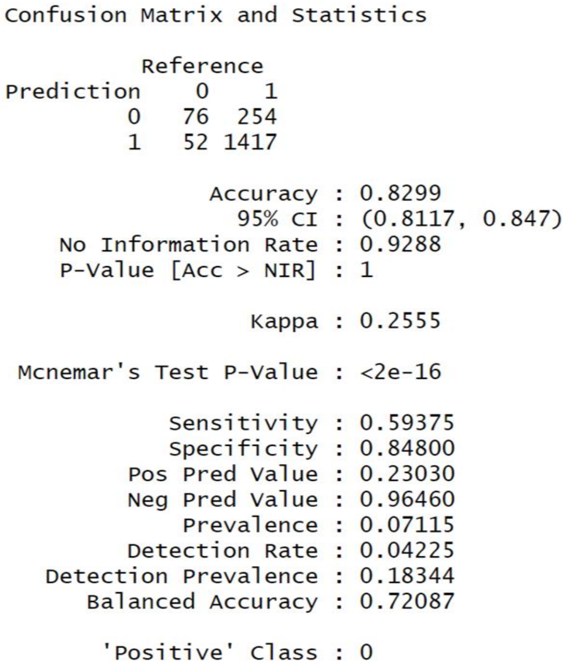




We did obtained variable of importance which gives all the variables that has most effect on the predictor. In our model, duration accepted, notice period and candidate relocation. It is very important to know our important variables if the analysis is all about analysing which factors are effecting the predictor more.



var	rel.inf
<chr>	<dbl>
Duration_accepted	25.7119719
Notice_period	17.4271168
Candidate_relocate	15.4358465
Percent_difference	11.9386408
Percent_hike_expected	7.3087484
Percent_hike_offered	7.1421293
Experience	6.3318294
Age	3.8228779
Offrd_band	3.1428567
DOJ_Extended	0.9102239



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0   76  254
         1   52 1417

               Accuracy : 0.8299
                 95% CI : (0.8117, 0.847)
    No Information Rate : 0.9288
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2555

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.59375
            Specificity : 0.84800
         Pos Pred Value : 0.23030
         Neg Pred Value : 0.96460
             Prevalence : 0.07115
         Detection Rate : 0.04225
   Detection Prevalence : 0.18344
      Balanced Accuracy : 0.72087

       'Positive' Class : 0
```

Above is the confusion matrix obtained from the Boosting model.

The confusion matrix describes that:

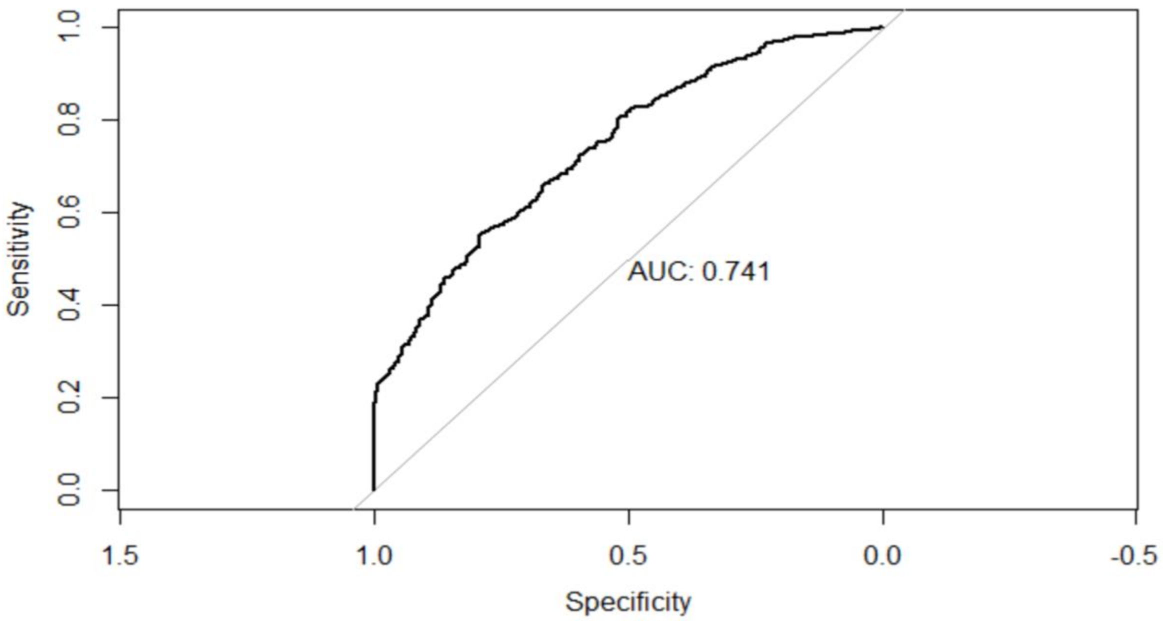
76- There are 76 values which are correctly predicted to be 0.

1417- There are 1417 values which are correctly predicted to be 1.

52- There are 52 values which are falsely predicted to be 0.

254- There are 254 values which are falsely predicted to be 1.

SVM Model

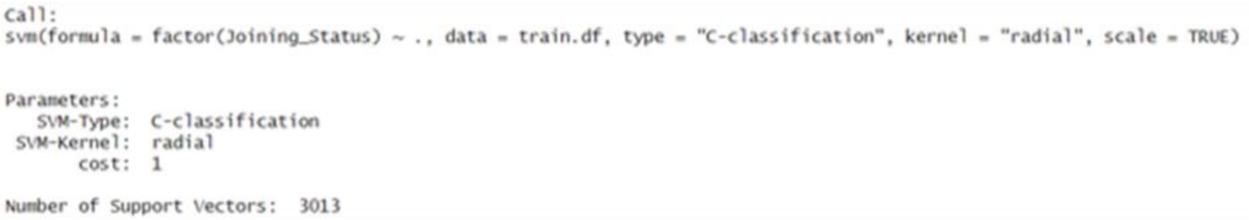


ROC curve represents that the data fit is almost equal to 1, The area under curve is about 0.741. Hence with the accuracy, error rate, ROC and other factors we are considering Boosting is the best m odel obtained till now.

SVM Model

SVM Model is a machine learning algorithm used for classification and regression analysis. It is mainly used for classification problems where the goal is to predict the class of an observation based on its feature.

SVM (Support Vector Machine) Model is particularly useful in situation where data is non-linearly separable. SVM uses a kernel function to transfer data into higher- dimensional space, where data can be linearly separable.



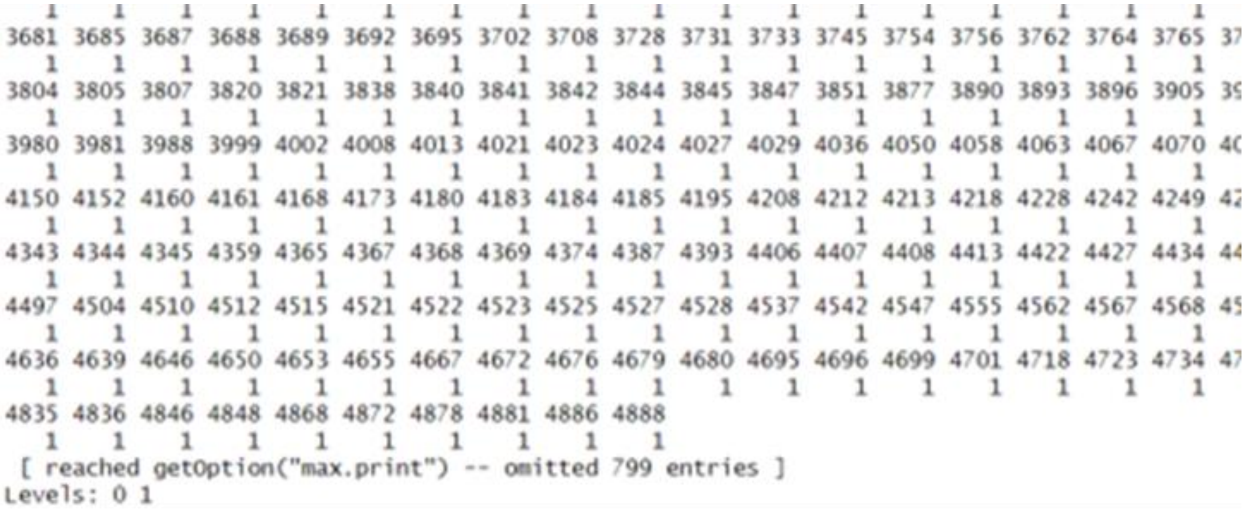
```
Call:
svm(formula = factor(Joining_Status) ~ ., data = train.df, type = "C-classification", kernel = "radial", scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  3013
```

As the data is imbalanced and it is also non-linear to perform SMV models. So we perform SVM model with kernel as radial, where the kernel is used to transform the input data into higher dimensional, where it can be separated into classes. Radial is usually used for non-linear data. The SVM method with joining_ status gives:

Cost: 1

Number of Support: 3013

The predictions for the SVM method states with their classes either the candidates do not join or join, where the levels are 0 and 1 respectively.



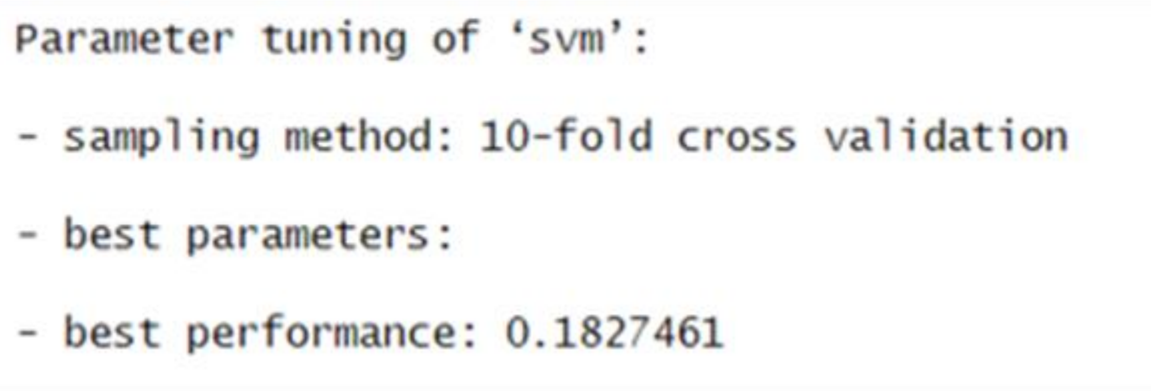
```
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
3681 3685 3687 3688 3689 3692 3695 3702 3708 3728 3731 3733 3745 3754 3756 3762 3764 3765 37
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
3804 3805 3807 3820 3821 3838 3840 3841 3842 3844 3845 3847 3851 3877 3890 3893 3896 3905 39
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
3980 3981 3988 3999 4002 4008 4013 4021 4023 4024 4027 4029 4036 4050 4058 4063 4067 4070 40
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
4150 4152 4160 4161 4168 4173 4180 4183 4184 4185 4195 4208 4212 4213 4218 4228 4242 4249 42
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
4343 4344 4345 4359 4365 4367 4368 4369 4374 4387 4393 4406 4407 4408 4413 4422 4427 4434 44
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
4497 4504 4510 4512 4515 4521 4522 4523 4525 4527 4528 4537 4542 4547 4555 4562 4567 4568 45
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
4636 4639 4646 4650 4653 4655 4667 4672 4676 4679 4680 4695 4696 4699 4701 4718 4723 4734 47
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
4835 4836 4846 4848 4868 4872 4878 4881 4886 4888
   1    1    1    1    1    1    1    1    1    1
 [ reached getOption("max.print") -- omitted 799 entries ]
Levels: 0 1
```

The accuracy of SVM model is 81.93% and Error rate is 18.07%

```
0.8193441
```

Tuning method is a technique to optimize the hyperparameters of the model by searching for the best combination of the parameter values which maximize the model's performance.

The hyperparameter of the model, such as the cost parameters, kernel function parameters, can greatly affect the models performance.

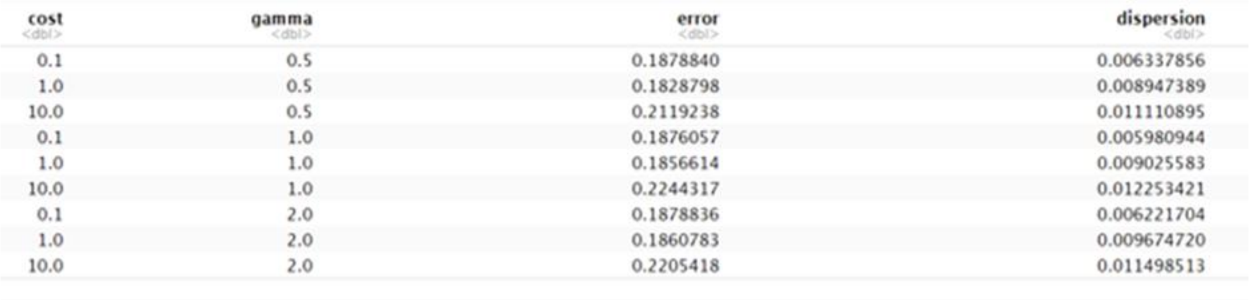


```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:

- best performance: 0.1827461
```

The performance involves tuning the SVM model with different combinations of hyperparameters values and evaluating the model's performance on a cross validation.

The hyperparameters involved in the tune method are cost and gamma, where cost trades off misclassification of training against simplicity and of decision and the values of cost are 0.1, 1, 10. Gamma value defines how far the influence of boundaries is and it is defined as 0.5, 1, 2.

The tune method undergoes each value of cost and gamma and inferences its best performance.

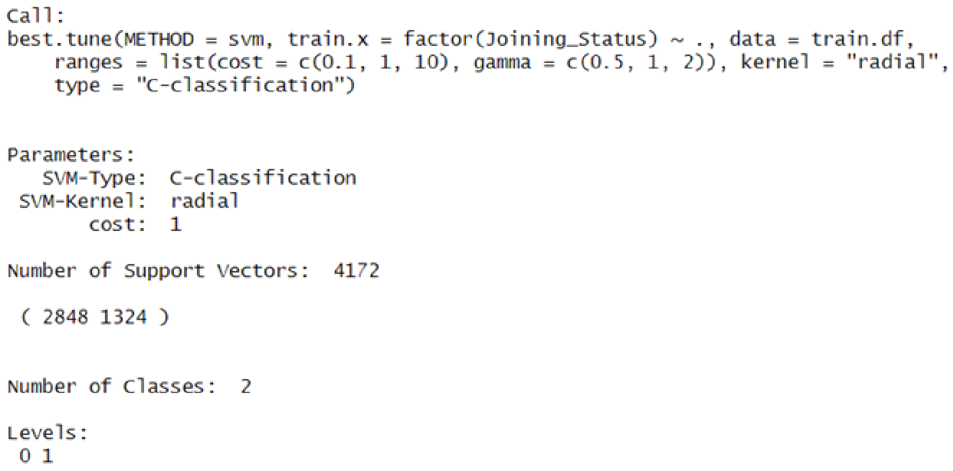


cost	gamma	error	dispersion
<dbl>	<dbl>	<dbl>	<dbl>
0.1	0.5	0.1878840	0.006337856
1.0	0.5	0.1828798	0.008947389
10.0	0.5	0.2119238	0.011110895
0.1	1.0	0.1876057	0.005980944
1.0	1.0	0.1856614	0.009025583
10.0	1.0	0.2244317	0.012253421
0.1	2.0	0.1878836	0.006221704
1.0	2.0	0.1860783	0.009674720
10.0	2.0	0.2205418	0.011498513

When the cost is 1.0 and gamma is 0.5 had the validation set and error is 0.1828

The best model that achieved the highest performance on a given validation set. The hyperparameters that result in the highest performance on the validation set are selected as the optimal hyperparameters for this model.

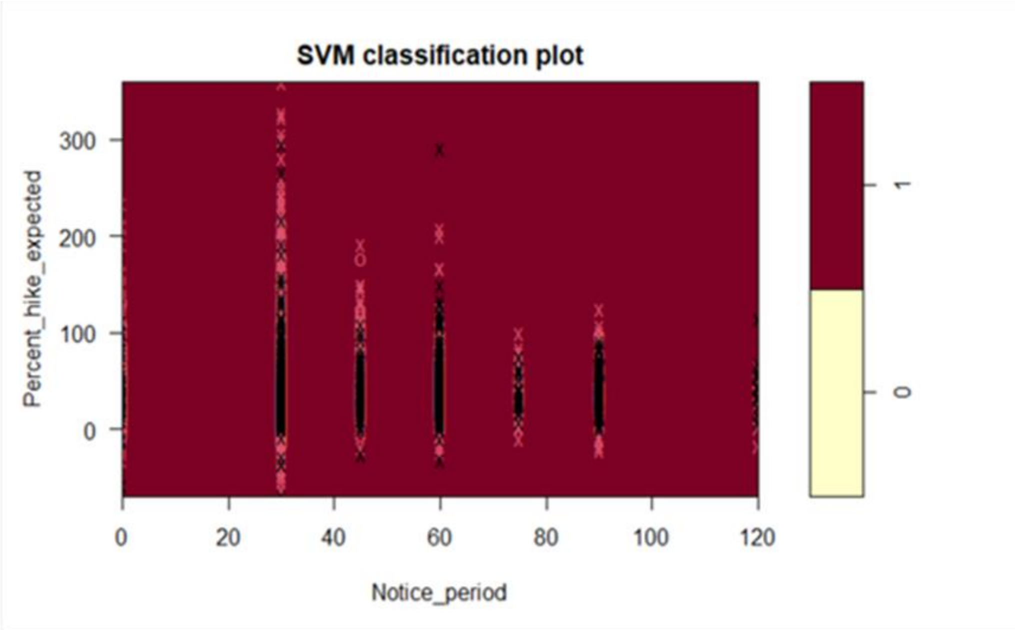
Performing the best model on the tune method achieves the highest performance on a validation set. The hyperparameters result in the highest performance.



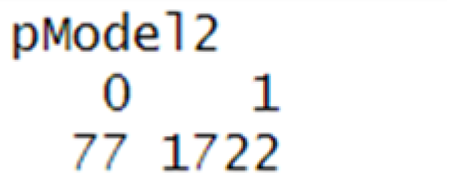
```
Call:
best.tune(METHOD = svm, train.x = factor(Joining_Status) ~ ., data = train.df,
    ranges = list(cost = c(0.1, 1, 10), gamma = c(0.5, 1, 2)), kernel = "radial",
    type = "C-classification")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  4172

 ( 2848 1324 )


Number of Classes:  2

Levels:
 0 1
```

The Number of Vectors increased by tuning the SVM model is 4172.

Using the training data of the best model performance using tune method over radial basis. The classification plot is created for predicting the joining_status of an employee which is grouped with Notice Period and Percent hike expected.

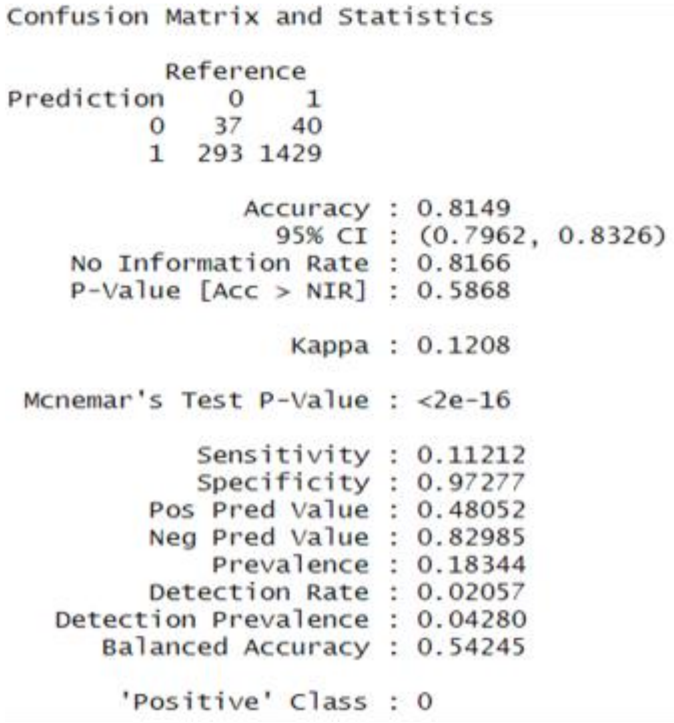


The prediction of the best model on the testing data says below



```
pModel2
   0    1
  77 1722
```

Where Not joined is 77 and Joined is 1722 from the test dataset.



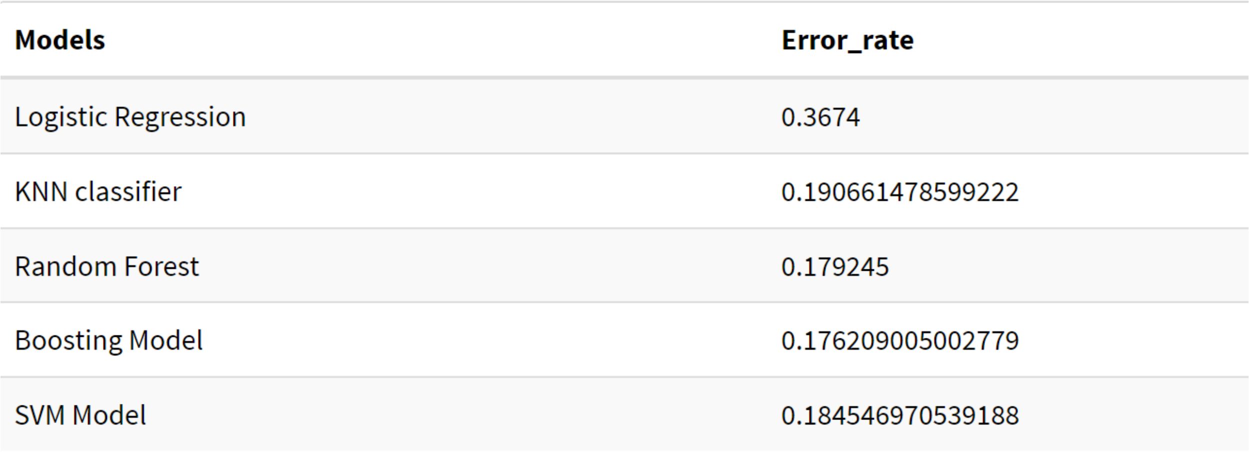
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0   37   40
         1  293 1429

               Accuracy : 0.8149
                 95% CI : (0.7962, 0.8326)
    No Information Rate : 0.8166
    P-Value [Acc > NIR] : 0.5868

                  Kappa : 0.1208

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.11212
            Specificity : 0.97277
         Pos Pred Value : 0.48052
         Neg Pred Value : 0.82985
             Prevalence : 0.18344
         Detection Rate : 0.02057
   Detection Prevalence : 0.04280
      Balanced Accuracy : 0.54245

       'Positive' Class : 0
```

The Confusion matrix on the testing data set on prediction on best model performance scored accuracy with 81.49% and the error rate is estimated to 18.51%.

COMPARING ALL THE MODELS

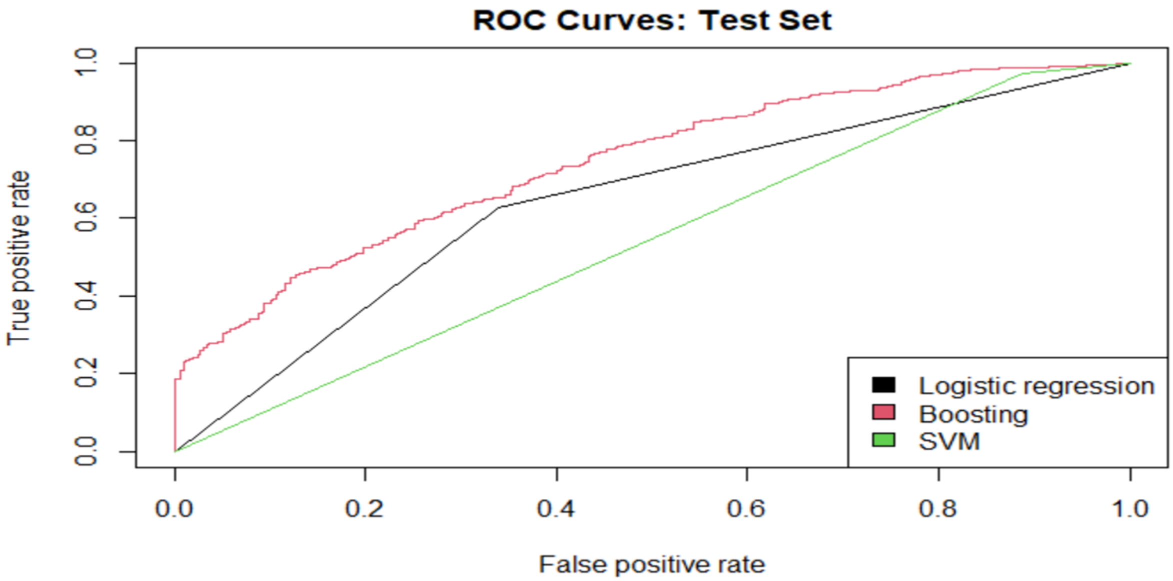
Human Resource (HR) datasets are widely used for predictive modeling and analysis to support important business decisions such as joining status, Notice period, joining bonus, and more. In this analysis, we compared different statistical models such as exploratory analysis, logistic regression, KNN classification, Boosting model, and SVM model to predict employee joining status.

The error rate evaluates the performance on the model's accuracy on making the prediction on each model. The lowest error rate says the model performed better among the comparison of other models.



Models	Error_rate
Logistic Regression	0.3674
KNN classifier	0.190661478599222
Random Forest	0.179245
Boosting Model	0.176209005002779
SVM Model	0.184546970539188

The different models with all the error rates stated and the lowest error rate estimated at Boosting model with 17.62% than other model, but random forest model is estimated as 17.92%.



The test dataset over the ROC Curve shows the performance of a classification model. And the graph eventually indicates that the Boosting model shows higher performance than SVM logistic regression.

CONCLUSION AND FUTURE RESEARCH

On the basis of the HR dataset, we analyzed the error rates of the various models to find which one is most effective at forecasting employee joining status.

We discovered that Boosting beat the other models in terms of accuracy and the capacity to datasets with high-dimensional features, and it had the lowest error rate when we compared the error rates of the various models. To identify the optimal model for the given problem, exploratory analysis and model selection are crucial. It is vital to remember that the choice of model will rely on the specific dataset and job.