Project Initial Data Analysis

Class: DSA-5103

Group 16 Members(M):

Karen Ochie(O), David Nnamdi(O), Tobenna Anyaezu(I), Oluwatobi Oyebanji(I)

Project Understanding

Over the recent past, Company X has seen an increased rate of employee attrition. The dataset contains information relevant to employee engagement, satisfaction, and turnover. Features available include subjective user input like “satisfaction level”, and other details of the work such as time, hazard, and compensation.

This project plans to assess and identify the most notable features that affect employee attrition. The objective of this project is to build a minimally viable product model to predict the outcome of An employee’s attrition. This is a binary Classification problem, predicting whether an employee will leave the company or not and the goal is to identify and improve these factors to prevent the loss of good people. The deliverable of the work is this report which includes the exploratory data analysis, modelling process and further recommendations. The success criteria for the project would be a high classifier probabilty for probability predictions.

For this project, the Cross Industry Process for Data Mining, CRISP-DM was followed. The first step was to understand the project, it's objectives and deliverables which are listed above. The next step was to load the data and understand the data. Data properties, statistics and visualizations helped achieve this. After this the next step was to prepare the data for modelling. Missing data, outliers, duplicates and redundant data were handled as a part of the exploratory data analysis. The next step was modelling the data. For this work, six models were considered: Logistic Regression, K-Nearest Neighbours (kNN), Decision Trees (DT), Random Forest (RF), Extreme Stochastic Gradient Boosted Trees (XgbTree) and Support Vector Machines (SVM).

Exploratory Data Analysis

After loading the required libraries and datasets, the next step was to visualize the dataset. We first converted all blank spaces to missing values and then glimpsed the dataset. The data contained 26 numeric variables and 9 character variables. On glimpsing the data, we decided to convert numeric data with less than 10 unique values to factor variables. We decided on this number based on our understanding of the dataset given. After ensuring all character variables were converted to factor variables, our dataset had become modified to 14 numeric variables and 21 factor variables. From viewing the missing value in the dataset using the MICE package, the missing values in the dataset is shown below. The numeric data had missing values in three variables – Age, DistanceFromHome and DailyRate. The factor data had missing values in two variables – BusinessTravel and MaritalStatus. The target variable had no missing value and we removed it from the dataset before commencing the missing value imputation.

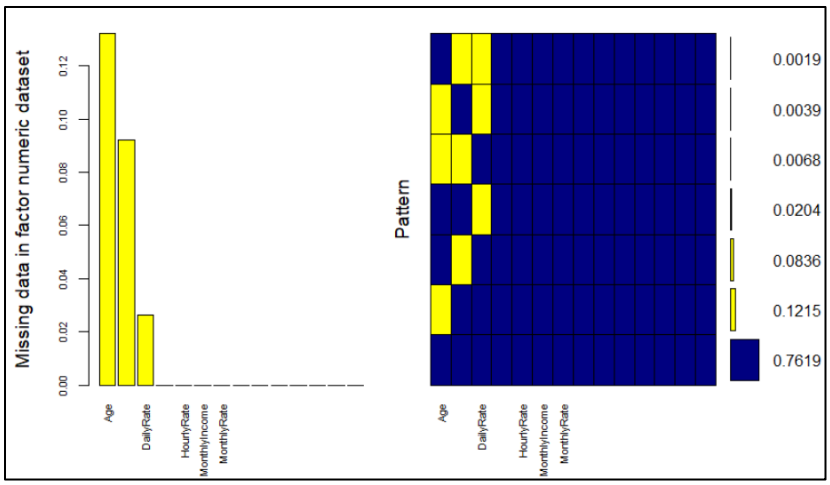


Figure 1: Missingness map of numeric data

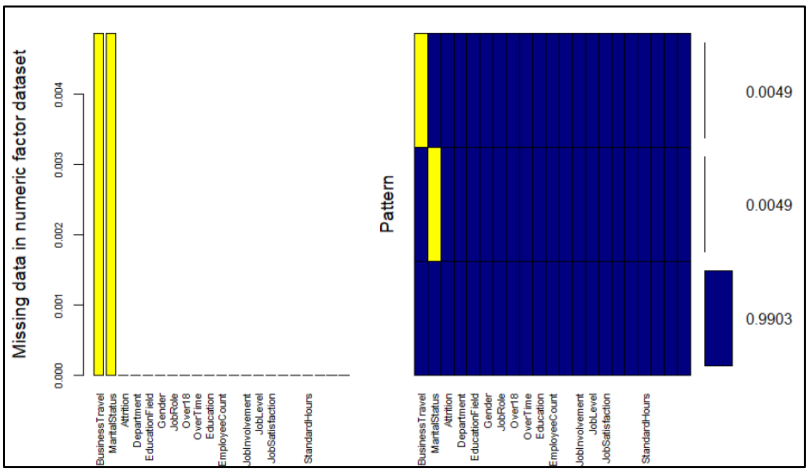


Figure 2: Missingness map of factor data

Using the predictive mean matching for the numeric data and the polytomous logistic regression for the factor data, we computed the missing value imputation. We then combined the data and the combined dataset with no missing values is shown in the figure below.

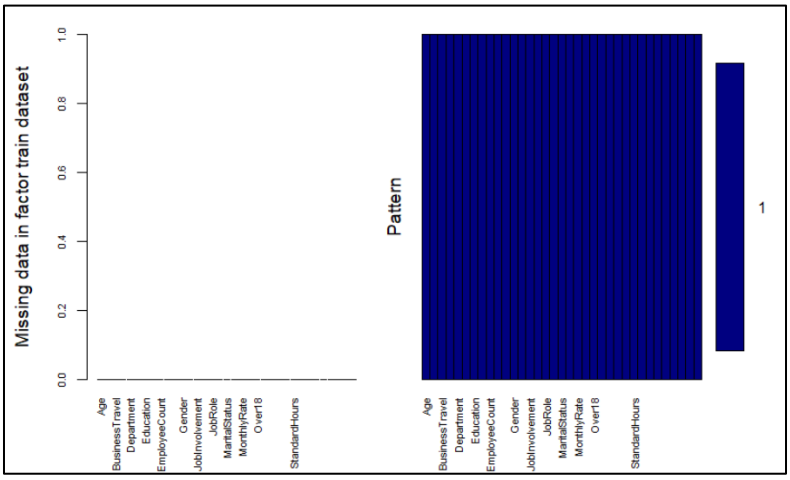


Figure 3: Clean missingness map

The next step was to commence feature engineering. We first plotted the distributions for each feature to see the data trends as shown below. From the density plot, we noticed some of the numeric data were skewed and would need to undergo data transformation. We also ran the correlation to check for linear related features and replicas. Feature interaction is crucial to model performance provided the selected important features can be randomly paired and that they are not strongly correlated with other features, as it can mar interpretability of the model. Guided by the information from the correlation plot in figure 6, we can attempt to compute features interactions on the important attributes in the dataset depending on performance and needs for the model built.

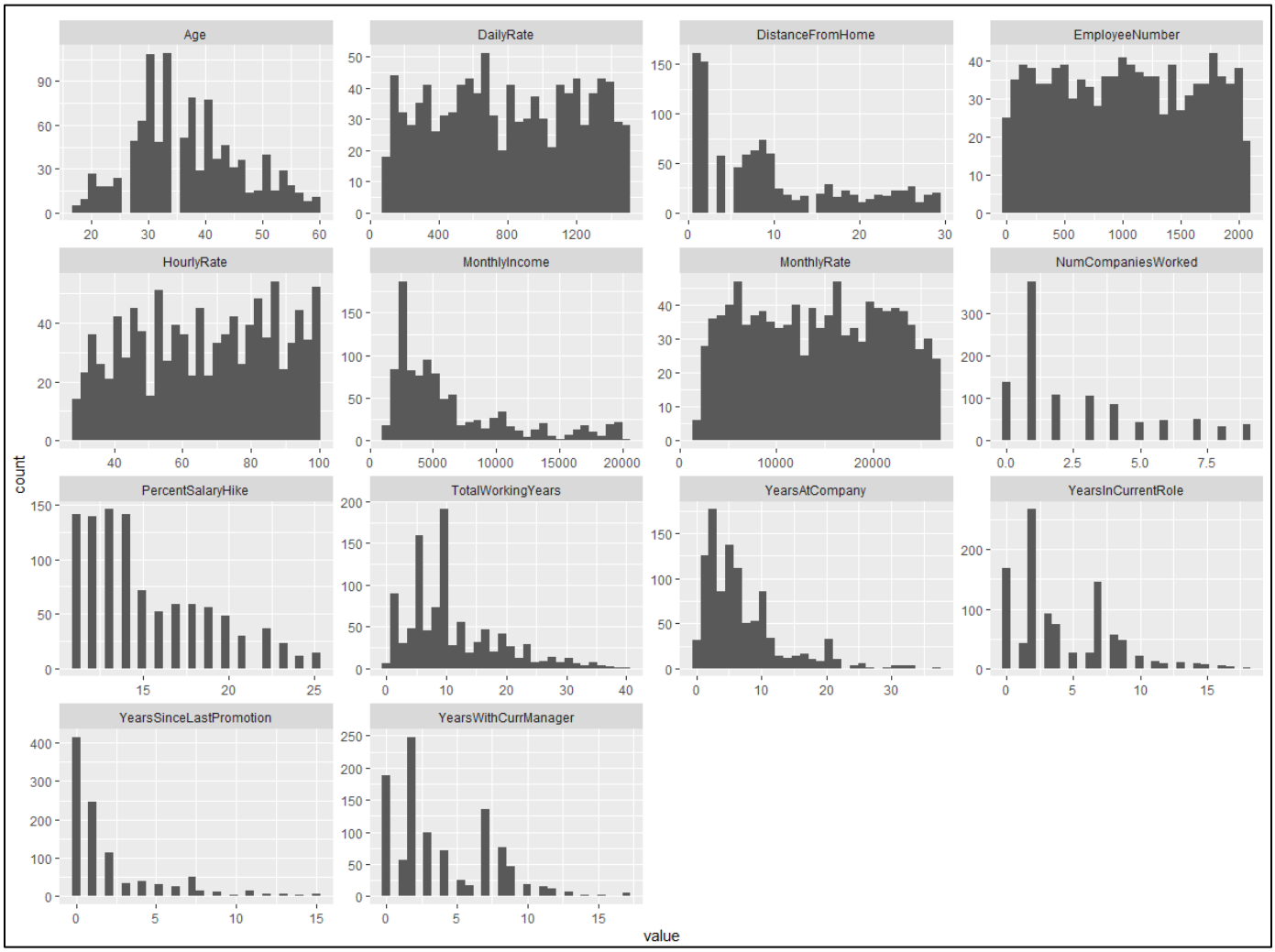


Figure 4: Data distribution

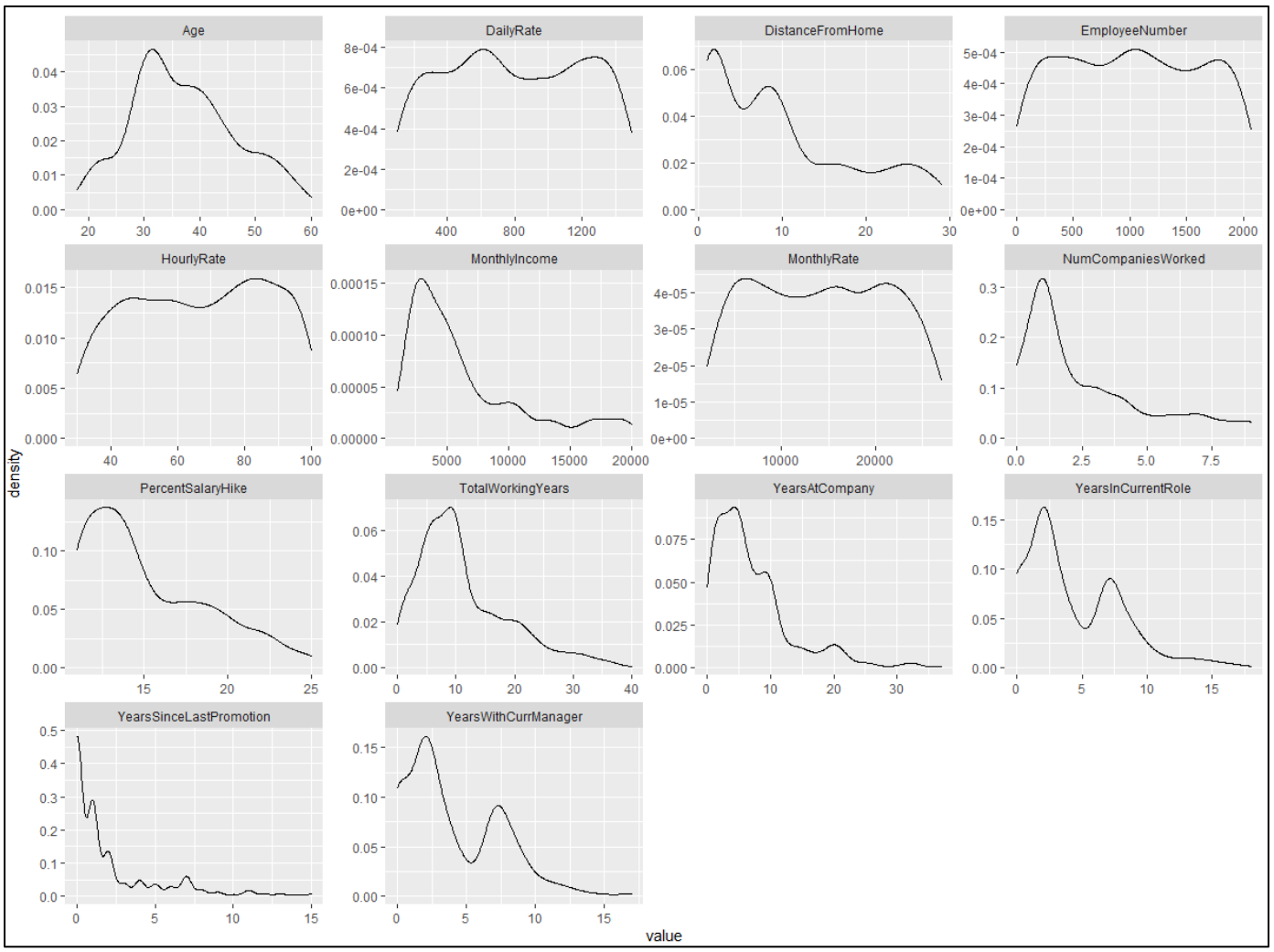


Figure 5: Data density plot

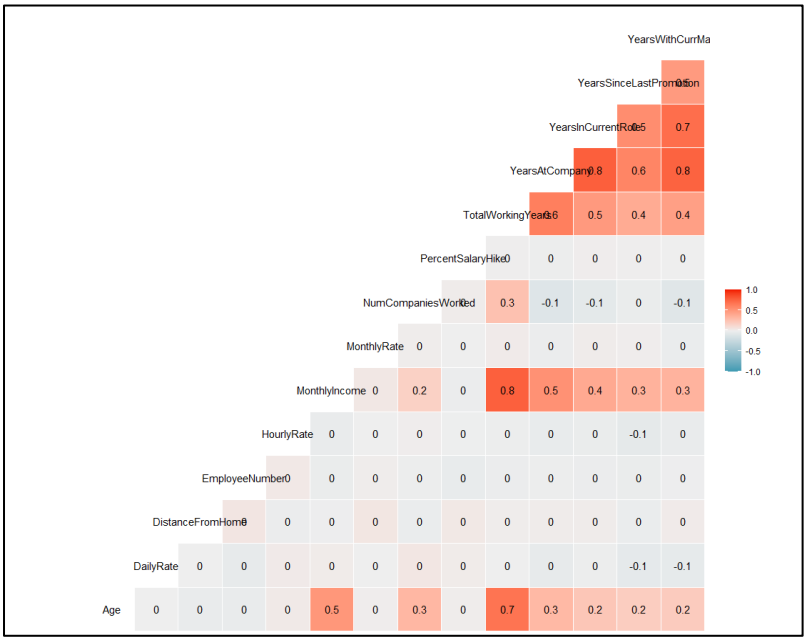


Figure 6: Correlation plot

From the correlation plot, we observed some variables were highly correlated. YearsWithCurrManager was highly correlated with YearsSinceLastPromotion, YearsInCurrentRole and TotalWorkingYears hence we decided to drop it. TotalWorkingYears was also highly correlated with Age, MonthlyIncome, YearsAtCompany and YearsAtCurrentRole, hence we also dropped it.

Next, we identified and removed factor features with just one level for the data as they could not provide us with any additional insight. We dropped EmployeeCount, Over18 and StandardHours.

We then proceeded to check for outliers in numeric dataset . Outliers are values that are distant from other observations and differ significantly from other points. Outliers might be important or might not be. This is where thorough data understanding and ample Exploration Data Analysis (EDA) can be helpful. For this project, we looked at multiple ways to characterize outliers in the data including:

1. Boxplots
2. Statitiscal tests (Grubb's test)
3. Density-Based Spatial Clustering of Applications with Noise (DBSCAN): is a density-based clustering algorithm that groups points that are close together (neighbors), marking points that lie alone in low-density regions.
4. Expectation Maximisation (EM) algorithm: is an iterative unsupervised clustering algorithm that tries to find similar clusters based on their orientation and variance. It is used to find maximum likelihood of statisticalmethods.

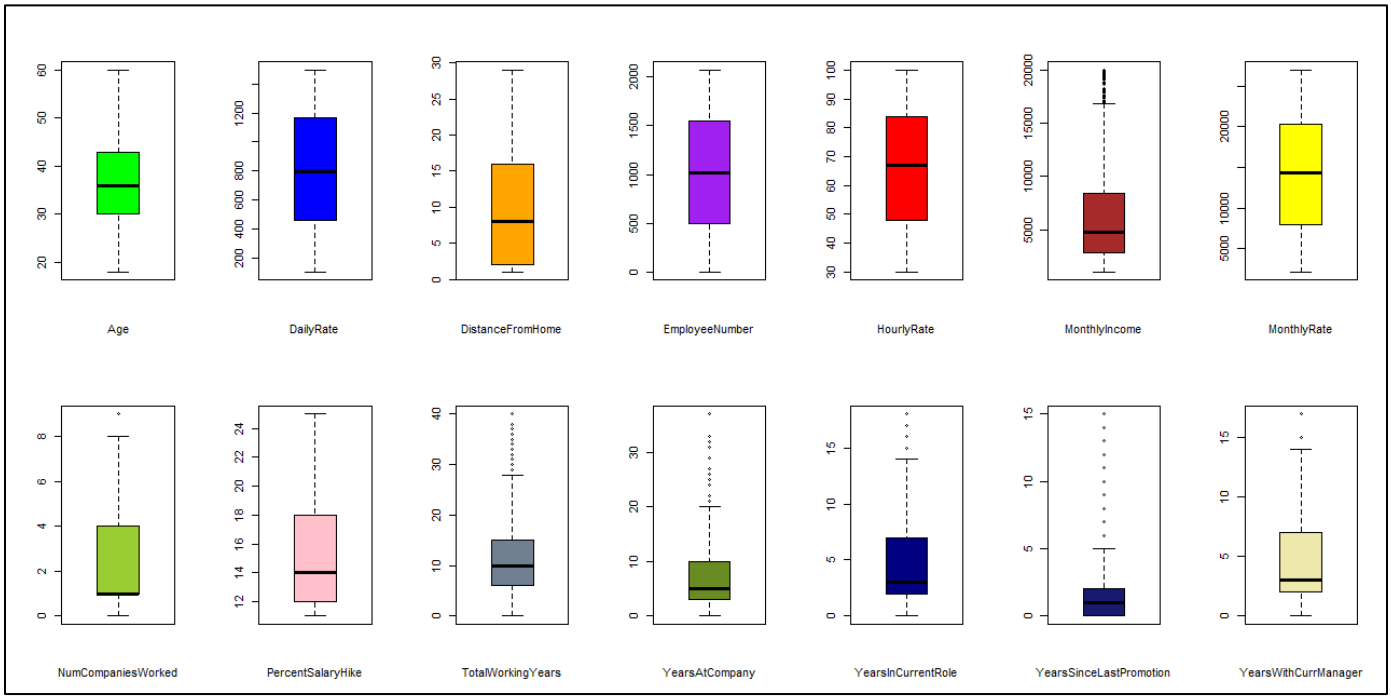


Figure 7: Boxplot of all numerical variables

The boxplot reveals the numerical variables with outliers includes: MonthlyIncome, NumCompaniesWorked, TotalWorkingYears, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, and YearsWithCurrManager. A closer look at these variables plotted against the target variable shows some interesting results.

According to the Grubbs test, two variables had outliers. The variables with outliers were YearsAtCompany and YearsSinceLastPromotion. Based on our domain knowledge of the project understanding, we decided not to drop these values as it was possible to stay more than 37 years in a company or to stay 15 years without a promotion especially for low and high level jobs. Based on the data, in the plot below older employees with high job levels tend to have stayed longer years at the company.

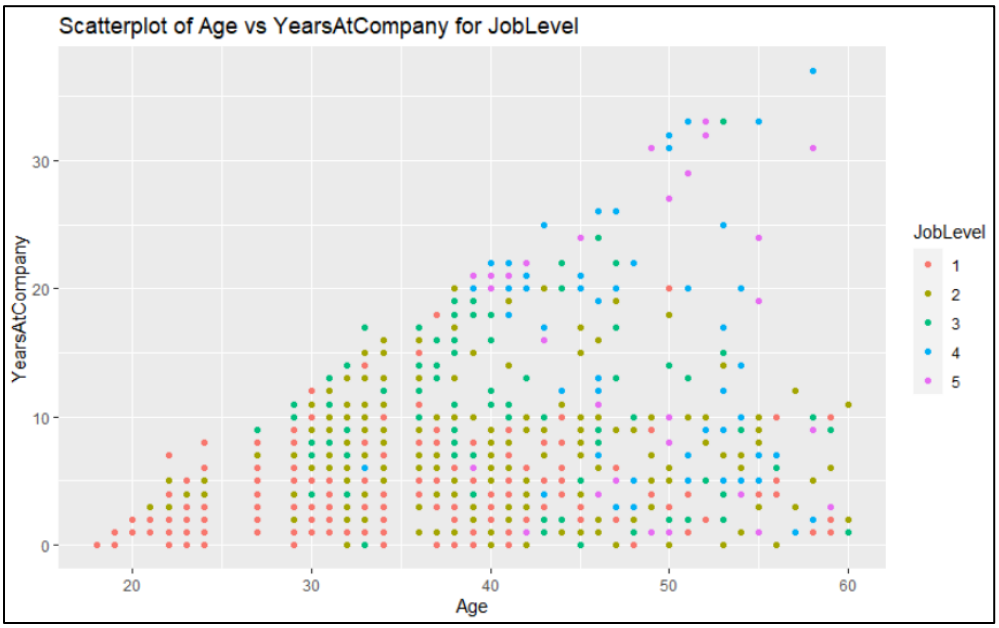


Figure 8: Scatterplot of Years at Company vs Age

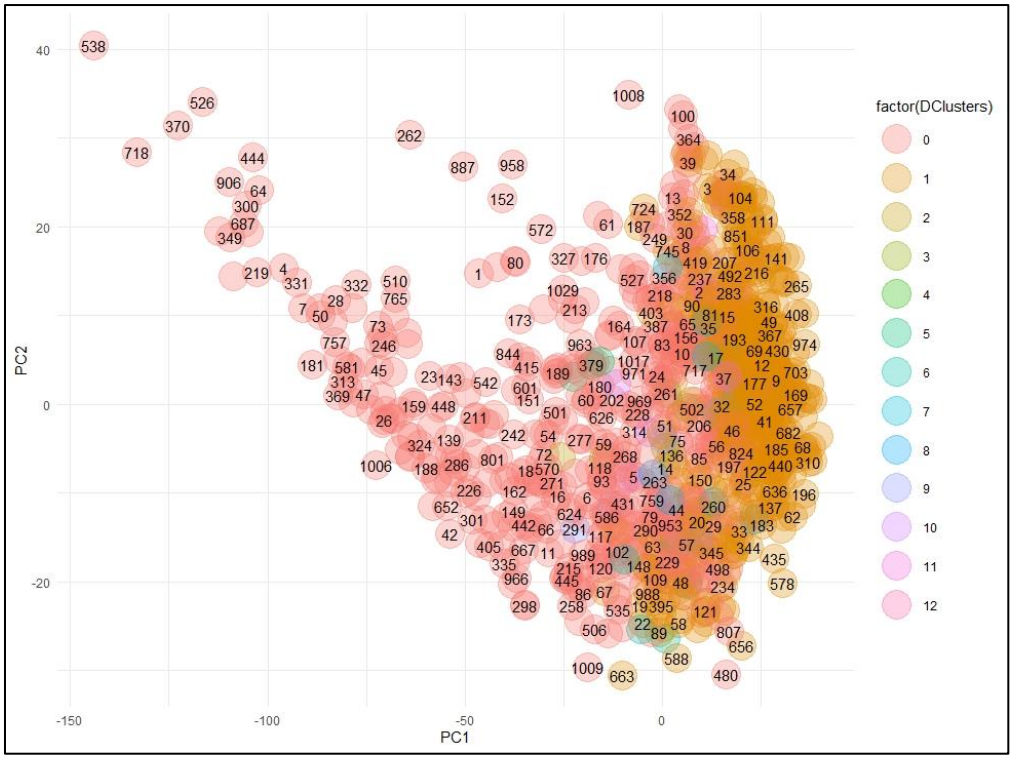


Figure 9: Result from dbscan

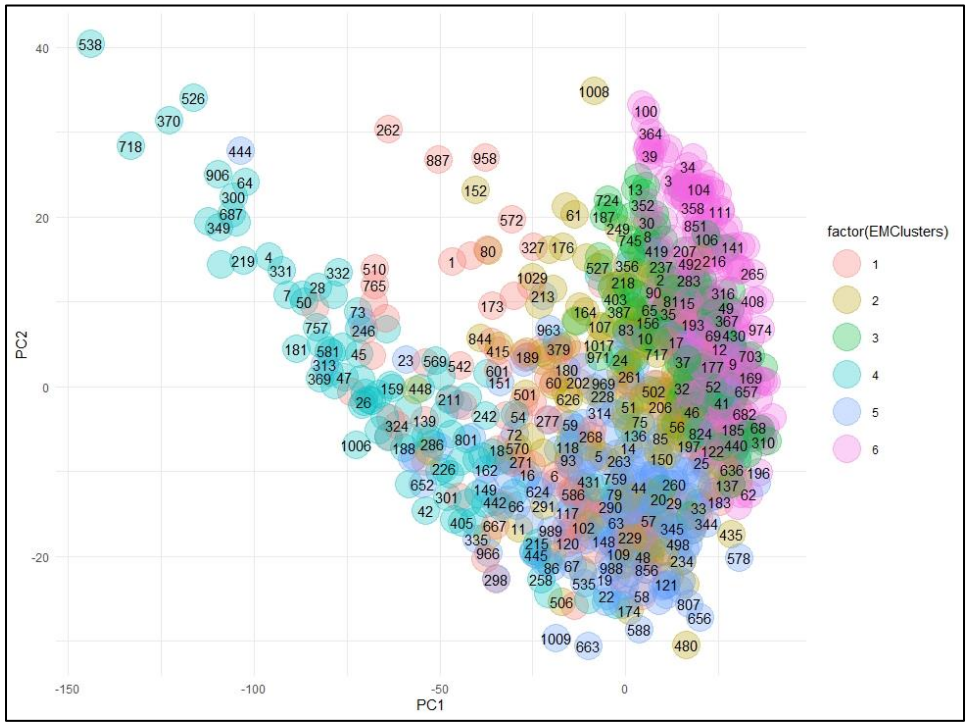


Figure 10: Result from Mclustering

The plot from the dbscan algorithm automatically clusters the data into 13 separate groups while the plot from the Mclust algorithm was manually set to 6 clusters. The x-axis and y-axis are the first and second principal components on the numerical data. Together, PC1 and PC2 represents 99% of the explained variance. Both plots supports the hypothesis that row numbers 370, 526, 538, 718 are most likely to be outliers.

We then checked the feature importance in order to highlight features of high value in the dataset. We see how the variance of each feature estimates the partial dependence of the target feature, Attrition. The Boruta algorithm in the Boruta library checks for the variable importance measure based on Random Forest. This method trains a random forrest classifier on several copies of the features. Based on the Mean decrease accuracy, it ranks features with higher means as important. Increase in the P-values of the means or the number of run of the classifier helps to properly rank the features. From the Boruka plot in figure 11, 4 of the 12 numeric features have feature importance greater than 10, while 6 of 12 features are greater than 5.

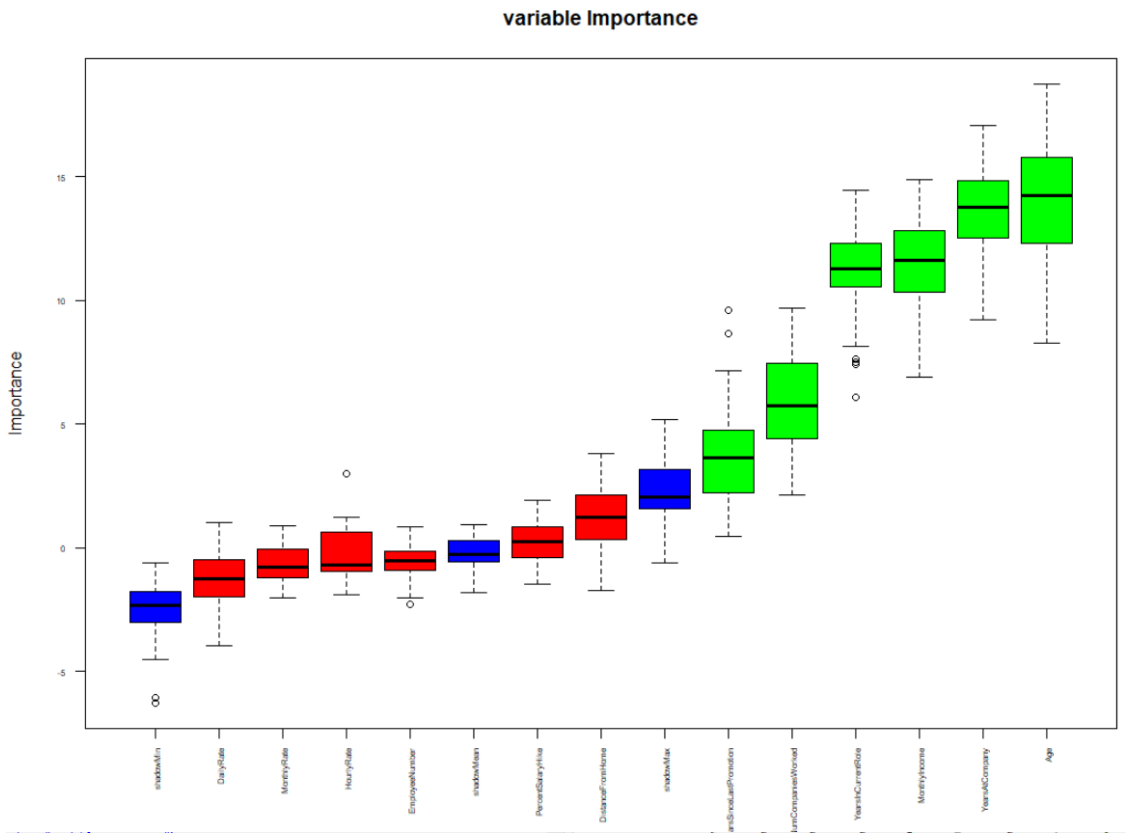


Figure 11: Boruka plot

The Varrank Approach is a filter-based approach that combines model learning steps to measure mutual information gained between two variables. It finds variable that maximizes the Mutual Information (MI) gained score between redundant and relevant variables. This approach uses the minimum redundancy maximum relevance algorithm. From the Varrank plot in figure 12, the blue color represents redundancy and red color for relevance and. At each stage, the variables with the highest relevance is selected. Monthlyincome feature have the highest MI value of 0.05 as information (relevance) is gained at each steps. Monthlyincome feature is followed by monthlyrate feature. The top five (5) relevant features selected are monthlyIncome, monthlyrate, WorkfromHome, Age, and Daily rate .

A key legend with the distribution of scores density is provided. Infering from this, a large percentage of the features are redundant. The monthly rate variable is ranked based on the Mutual Information. Negative score represents redundancy final trade of information while positive score is the relevancy final trade of information. The score matrix is displayed using both numerical values and color code.

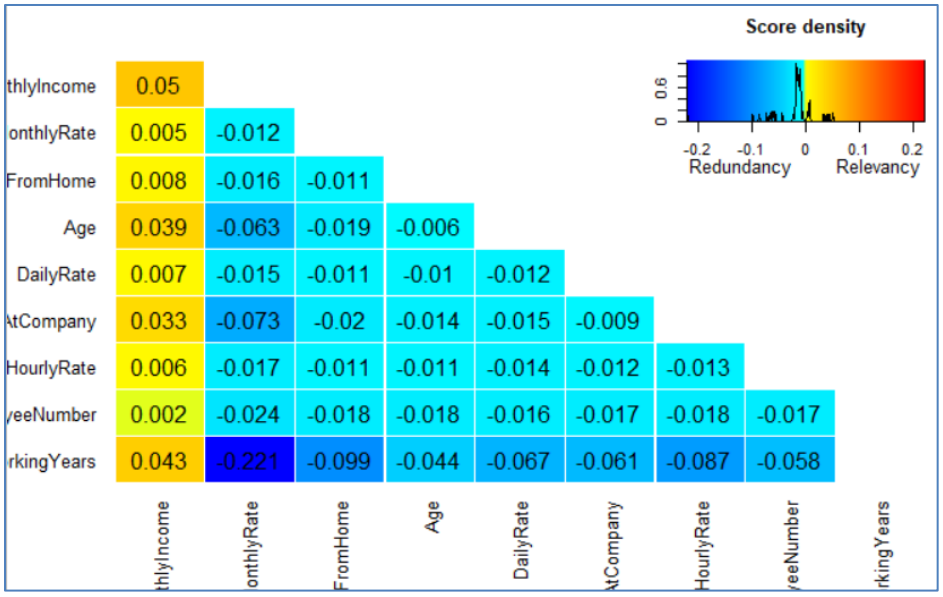


Figure 12: Plot of Varrank analysis of relevant features from the dataset

We finally merged the data in preparation for modelling. Our final data quality report in preparation for modeling for the numeric and categorical data is shown below. The final dataset contained 12 numerical and 17 factor variables.

variable	n	missing	missing_pct	unique	unique_pct	mean	min	Q1	median	Q3	max	sd
Age	1029	0	0	39	3.790	37	18	30	36	43	60	9
DailyRate	1029	0	0	692	67.250	801	102	458	798	1168	1496	409
DistanceFromHome	1029	0	0	27	2.624	10	1	2	8	16	29	8
EmployeeNumber	1029	0	0	1029	100.000	1024	1	496	1019	1553	2068	606
HourlyRate	1029	0	0	71	6.900	67	30	48	67	84	100	20
MonthlyIncome	1029	0	0	963	93.586	6450	1009	2814	4735	8446	19999	4795
MonthlyRate	1029	0	0	1010	98.154	14251	2094	7950	14295	20392	26999	7089
NumCompaniesWorked	1029	0	0	10	0.972	3	0	1	1	4	9	3
PercentSalaryHike	1029	0	0	15	1.458	15	11	12	14	18	25	4
YearsAtCompany	1029	0	0	32	3.110	7	0	3	5	10	37	6
YearsInCurrentRole	1029	0	0	19	1.846	4	0	2	3	7	18	4
YearsSinceLastPromotion	1029	0	0	16	1.555	2	0	0	1	2	15	3

Figure 13: Data Quality Report for Numeric Data

variable	n	missing	missing_pct	unique	unique_pct	freqRatio	1st mode	1st mode freq	2nd mode	2nd mode freq	least common	least common freq
BusinessTravel	1029	0	0	3	0.292	3.63	Travel_Rarely	726	Travel_Frequently	200	Non-Travel	103
Department	1029	0	0	3	0.292	2.17	Research & Development	676	Sales	311	Human Resources	42
EducationField	1029	0	0	6	0.583	1.3	Life Sciences	426	Medical	328	Human Resources	17
Gender	1029	0	0	2	0.194	1.5	Male	617	Female	412	Female	412
JobRole	1029	0	0	9	0.875	1.01	Sales Executive	217	Research Scientist	214	Human Resources	34
MaritalStatus	1029	0	0	3	0.292	1.49	Married	477	Single	320	Divorced	232
OverTime	1029	0	0	2	0.194	2.45	No	731	Yes	298	Yes	298
Education	1029	0	0	5	0.486	1.38	3	386	4	279	5	36
EnvironmentSatisfaction	1029	0	0	4	0.389	1.03	3	310	4	300	1	207
JobInvolvement	1029	0	0	4	0.389	2.25	3	606	2	269	1	60
JobLevel	1029	0	0	5	0.486	1.15	1	403	2	350	5	47
JobSatisfaction	1029	0	0	4	0.389	1.05	4	315	3	301	1	198
PerformanceRating	1029	0	0	2	0.194	5.27	3	865	4	164	4	164
RelationshipSatisfaction	1029	0	0	4	0.389	1.09	3	318	4	293	1	194
StockOptionLevel	1029	0	0	4	0.389	1.04	0	432	1	417	3	69
TrainingTimesLastYear	1029	0	0	7	0.680	1.11	2	385	3	346	0	39
WorkLifeBalance	1029	0	0	4	0.389	2.5	3	626	2	250	1	54
Attrition	1029	0	0	2	0.194	4.85	No	853	Yes	176	Yes	176

Figure 14: Data Quality Report for Factor Data

Data Preparation

Based on the twenty-nine predictors (12 numerical and 17 factor) extracted from the provided dataset, an attempt to build a minimally viable product (MVP) model was made. One key observation was made about the dataset and target variable – there was significant imbalance in the dataset. This is illustrated in the plot below.

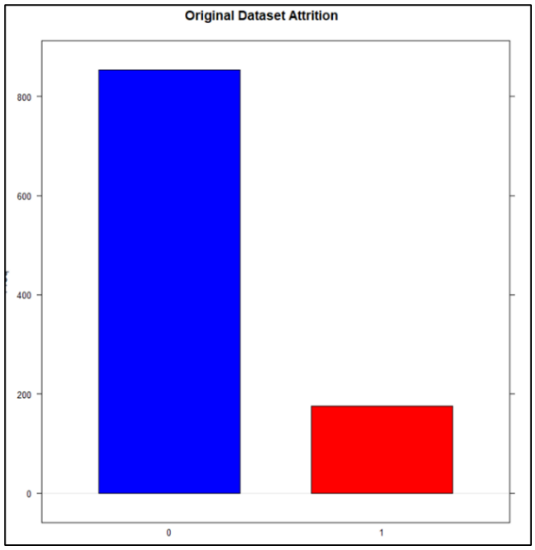


Figure 15: Imbalaced target variable

Based on the plot above, the minority class is ~17% of the entire dataset. Based on this knowledge, we defined key modelling objectives to be achieved in building the MVP. These objectives are:

- Select appropriate resampling method to deal with dataset imbalance
- Build MVP model with good prediction accuracy – explore different models
- The MVP must be robust to reduce with false negatives – increased focus on sensitivity (or recall) due to heavy imbalance

The subsequent sections outline steps taken in model building and calibration.

Train-Test Split

The dataset set was split 70:30 into training and test data using a stratified sampling approach. This approach was adopted to maintain class imbalance of original dataset in train and test splits. The plots below show the staff attrition of the train and test data splits.

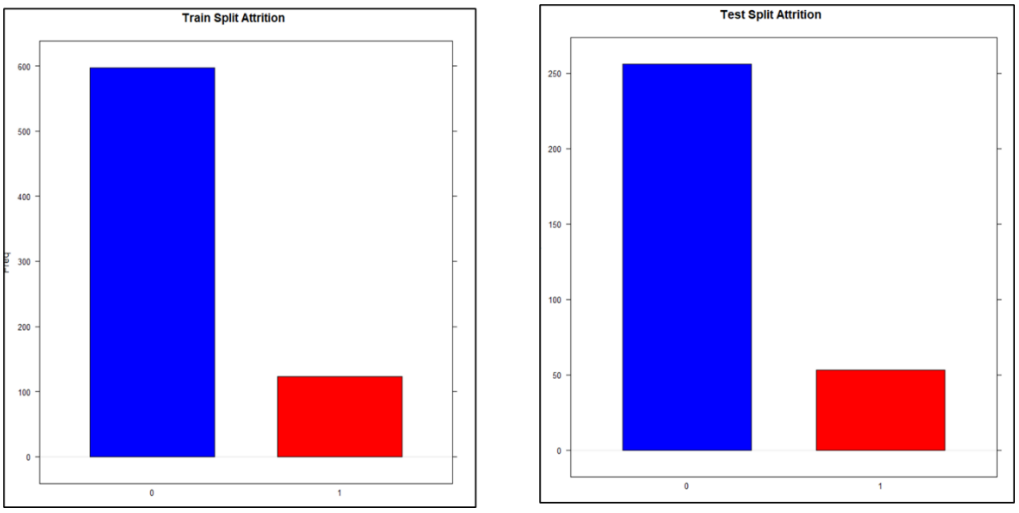


Figure 16: Train-Test data split

Feature Transformation

We observed some positive skew in certain numeric variables which inherently manifested as outliers but based on our understanding of the dataset were deemed to not be outliers (e.g., an employee spending 37 years with a company was deemed to be an outlier but this is very possible in reality).

To remove this skewness, the Yeo-Johnson transformation was used on the train-split data and then applied to the test split data. The results of the Yeo-Johnson transformation resulted in more gaussian like data as can be seen from the sample plot below.

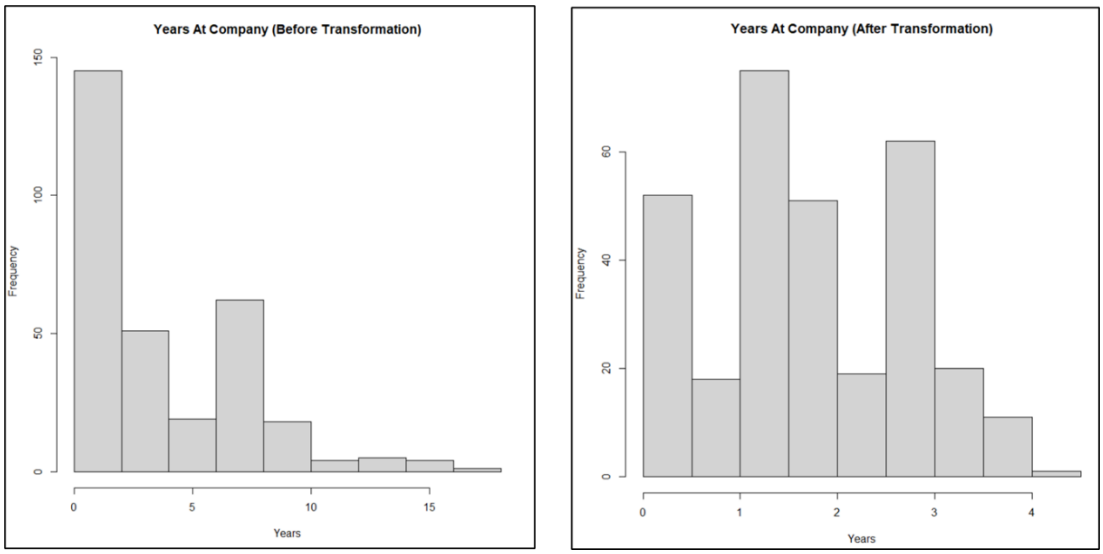


Figure 17: The results of the Yeo-Johnson transformation

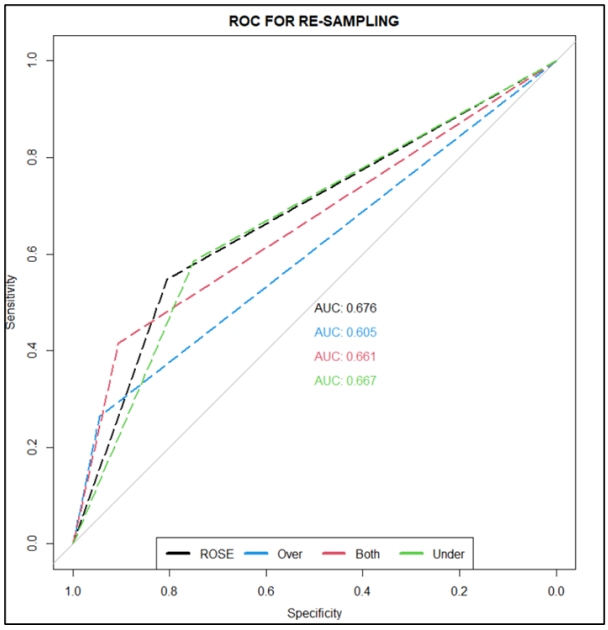
Re-sampling

When dealing with data with high class imbalance, re-sampling methods are utilized to improve on the samples of the minority class available for training. Doing this improves model's ability to predict minority class accurately, which in our case is the staff leaving the company (an attrition of 1).

Using the Random-Over-Sampling-Examples (ROSE) sampling package in R, four distinct types of sampling techniques were implemented. They are:

- *Over Sampling*: Here the minority class is oversampled, and a more balanced dataset is created for training
- *Under Sampling*: Here the majority class is under sampled to create a more balanced dataset. Has the drawback of reducing the amount of information available to train the model
- *Both*: Combines both above techniques
- *ROSE*: Here synthetic samples from the minority class are generated to improve dataset balance

After re-sampling using the above techniques, a simple random forest model with no cross-validation was done to evaluate what sampling technique will be used to train our models subsequently. The result of this analysis is listed in the table below.



Re-sampling method	Accuracy	Sensitivity (Recall)	AUROC
ROSE	0.761	0.547	0.676
Oversampling	0.828	0.264	0.605
Both	0.822	0.415	0.661
Undersampling	0.722	0.585	0.667

Figure 18: Results of resampling

Based on the table below, we compare three different metrics – Accuracy, Sensitivity and Area Under Receiver Operating Characteristic (AUROC) curve. The ROSE sampling technique is chosen for further modelling because it has the highest AUROC, and a good balance of accuracy and sensitivity when compared to the other methods. The under-sampling method also performed very well but due to information loss when using under sampling we decided to stick with the ROSE sample.

MVP model Building and Performance Evaluation

Six classification ML models were utilized for MVP model building. They are:

- Logistic Regression
- K-Nearest Neighbours (kNN)

- Decision Trees (DT)
- Random Forest (RF)
- Extreme Stochastic Gradient Boosted Trees (XgbTree)
- Support Vector Machines (SVM)

Training of each of these models was done using the Caret R package with 5-fold cross-validation to minimize overfitting issues. The metrics used to assess the performance of these models on test data include the accuracy, sensitivity, specificity, F1-score, Kappa and AUROC.

The results of these models are detailed in the table below.

Classification Model	Accuracy	Sensitivity (Recall)	Specificity	F1-score	Kappa	AUROC	RANK
Log Regression	0.738	0.547	0.777	0.831	0.260	0.662	2
Knn	0.631	0.415	0.676	0.752	0.065	0.545	6
Decision Trees	0.715	0.566	0.746	0.813	0.238	0.656	4
Random Forest	0.683	0.472	0.727	0.791	0.151	0.599	5
Xtreme Grad Boosted Trees	0.754	0.528	0.801	0.844	0.276	0.665	1
SVM (Radial)	0.754	0.509	0.805	0.844	0.266	0.657	3

Figure 19: Results from Models

The rank column in the table above was generated manually post inspection of the performance metrics. Based on the 6-performance metrics, XgbTree is ranked as the best performing model having the highest score in all but the sensitivity where it is the second highest and is chosen as our MVP

The tuned parameters for the XgbTree that resulted in the above metrics are shown in figure 20 below.

```
The final values used for the model were nrounds = 100, max_depth = 3, eta = 0.4, gamma =
 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample = 1.
```

Figure 20: XgbTree Metrics

Logistic regression model is chosen as the second-best performer based on the sensitivity which is the highest. LR also performs well based on accuracy, F1 and AUROC. Following at a close third is the SVM with a radial kernel. The ROC curve for all six models is shown below.

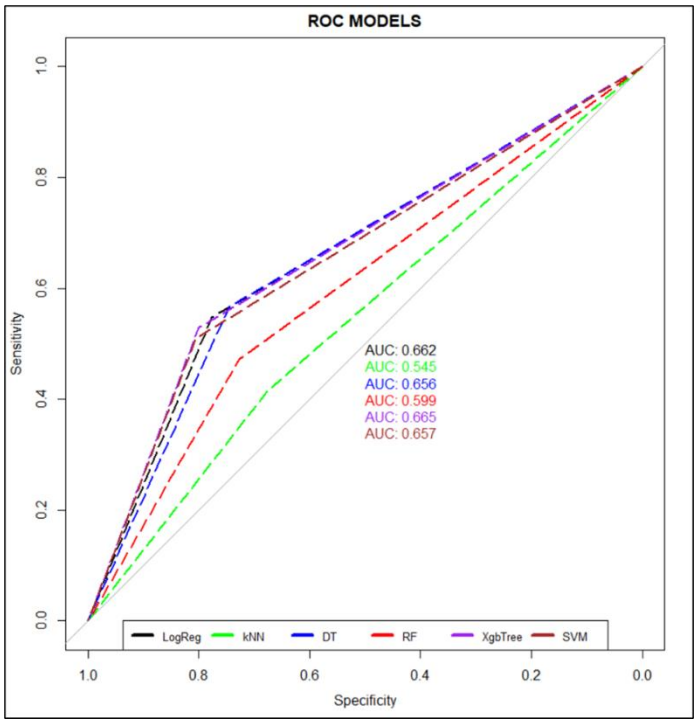


Figure 21: ROC Curve for all models

In summary, we were able to meet the set objectives for building our MVP as the XgbTree model offers an optimal performance overall, with reasonable sensitivity thus reducing false negatives.

Limitations & Areas for Improvement

We recognize that higher performance may be realized from available classification models, as we did not exhaust all the possible classification models in building our MVP. Also, using other re-sampling methods such as SMOTE (the version that can work with categorical data) may improve training performance however, an exhaustive search showed that the categorical SMOTE is more readily available in Python than in R.

Finally, adequate feature engineering may lead to creation of features that are better predictors of staff attrition and this has been identified as an area for further improvement.