Predicting Employee Retention

- ## Objective

The objective of this assignment is to develop a Logistic Regression model. You will be using this model to analyse and predict binary outcomes based on the input data. This assignment aims to enhance understanding of logistic regression, including its assumptions, implementation, and evaluation, to effectively classify and interpret data.

- ## Business Objective

A mid-sized technology company wants to improve its understanding of employee retention to foster a loyal and committed workforce. While the organization has traditionally focused on addressing turnover, it recognises the value of proactively identifying employees likely to stay and understanding the factors contributing to their loyalty.

In this assignment you'll be building a logistic regression model to predict the likelihood of employee retention based on the data such as demographic details, job satisfaction scores, performance metrics, and tenure. The aim is to provide the HR department with actionable insights to strengthen retention strategies, create a supportive work environment, and increase the overall stability and satisfaction of the workforce.

- # Assignment Tasks

You need to perform the following steps to complete this assignment:

1. Data Understanding
2. Data Cleaning
3. Train Validation Split
4. EDA on training data
5. EDA on validation data [Optional]
6. Feature Engineering
7. Model Building
8. Prediction and Model Evaluation

- # Data Dictionary

The data has 24 Columns and 74610 Rows. Following data dictionary provides the description for each column present in dataset:

Column Name	Description
Employee ID	A unique identifier assigned to each employee.
Age	The age of the employee, ranging from 18 to 60 years.
Gender	The gender of the employee.
Years at Company	The number of years the employee has been working at the company.
Monthly Income	The monthly salary of the employee, in dollars.
Job Role	The department or role the employee works in, encoded into categories such as Finance, Healthcare, Technology, Education, and Media.
Work-Life Balance	The employee's perceived balance between work and personal life (Poor, Below Average, Good, Excellent).
Job Satisfaction	The employee's satisfaction with their job (Very Low, Low, Medium, High).
Performance Rating	The employee's performance rating (Low, Below Average, Average, High).
Number of Promotions	The total number of promotions the employee has received.
Overtime	Number of overtime hours.
Distance from Home	The distance between the employee's home and workplace, in miles.
Education Level	The highest education level attained by the employee (High School, Associate Degree, Bachelor’s Degree, Master’s Degree, PhD).
Marital Status	The marital status of the employee (Divorced, Married, Single).
Number of Dependents	Number of dependents the employee has.
Job Level	The job level of the employee (Entry, Mid, Senior).
Company Size	The size of the company the employee works for (Small, Medium, Large).
Company Tenure (In Months)	The total number of years the employee has been working in the industry.
Remote Work	Whether the employee works remotely (Yes or No).
Leadership Opportunities	Whether the employee has leadership opportunities (Yes or No).
Innovation Opportunities	Whether the employee has opportunities for innovation (Yes or No).
Company Reputation	The employee's perception of the company's reputation (Very Poor, Poor, Good, Excellent).
Employee Recognition	The level of recognition the employee receives(Very Low, Low, Medium, High).
Attrition	Whether the employee has left the company.

1. Data Understanding

In this step, load the dataset and check basic statistics of the data, including preview of data, dimension of data, column descriptions and data types.

1.0 Import Libraries

1.1 Load the Data

	Employee ID	Age	Gender	Years at Company	Job Role	Monthly Income	Work-Life Balance	Job Satisfaction	Performance Rating	Num Prom
0	8410	31	Male	19	Education	5390	Excellent	Medium	Average	
1	64756	59	Female	4	Media	5534	Poor	High	Low	
2	30257	24	Female	10	Healthcare	8159	Good	High	Low	
3	65791	36	Female	7	Education	3989	Good	High	High	
4	65026	56	Male	41	Education	4821	Fair	Very High	Average	

5 rows × 24 columns

1.2 Check the basic statistics

1.3 Check the data type of columns

```
 1   Age                          int64
 2   Gender                       object
 3   Years at Company             int64
 4   Job Role                     object
 5   Monthly Income               int64
 6   Work-Life Balance            object
 7   Job Satisfaction             object
 8   Performance Rating           object
 9   Number of Promotions         int64
 10  Overtime                     object
 11  Distance from Home           float64
 12  Education Level              object
 13  Marital Status               object
 14  Number of Dependents         int64
 15  Job Level                    object
 16  Company Size                 object
 17  Company Tenure (In Months)   float64
 18  Remote Work                  object
 19  Leadership Opportunities     object
 20  Innovation Opportunities     object
 21  Company Reputation           object
 22  Employee Recognition         object
 23  Attrition                    object
dtypes: float64(2), int64(6), object (16)
```

```
memory usage: 13.7+ MB
```

2. Data Cleaning

2.1 Handle the missing values

2.1.1 Check the number of missing values

2.1.2 Check the percentage of missing values

2.1.3 Handle rows with missing values

2.1.4 Check percentage of remaning data after missing values are removed

2.2 Identify and handle redundant values within categorical columns (if any)

Examine the categorical columns to determine if any value or column needs to be treated

Drop redundant columns

3. Train-Validation Split

3.1 Import required libraries

- Import seaborn

3.2 Define feature and target variables

3.3 Split the data

- Split the data into 70% train data and 30% validation data

4. EDA on training data

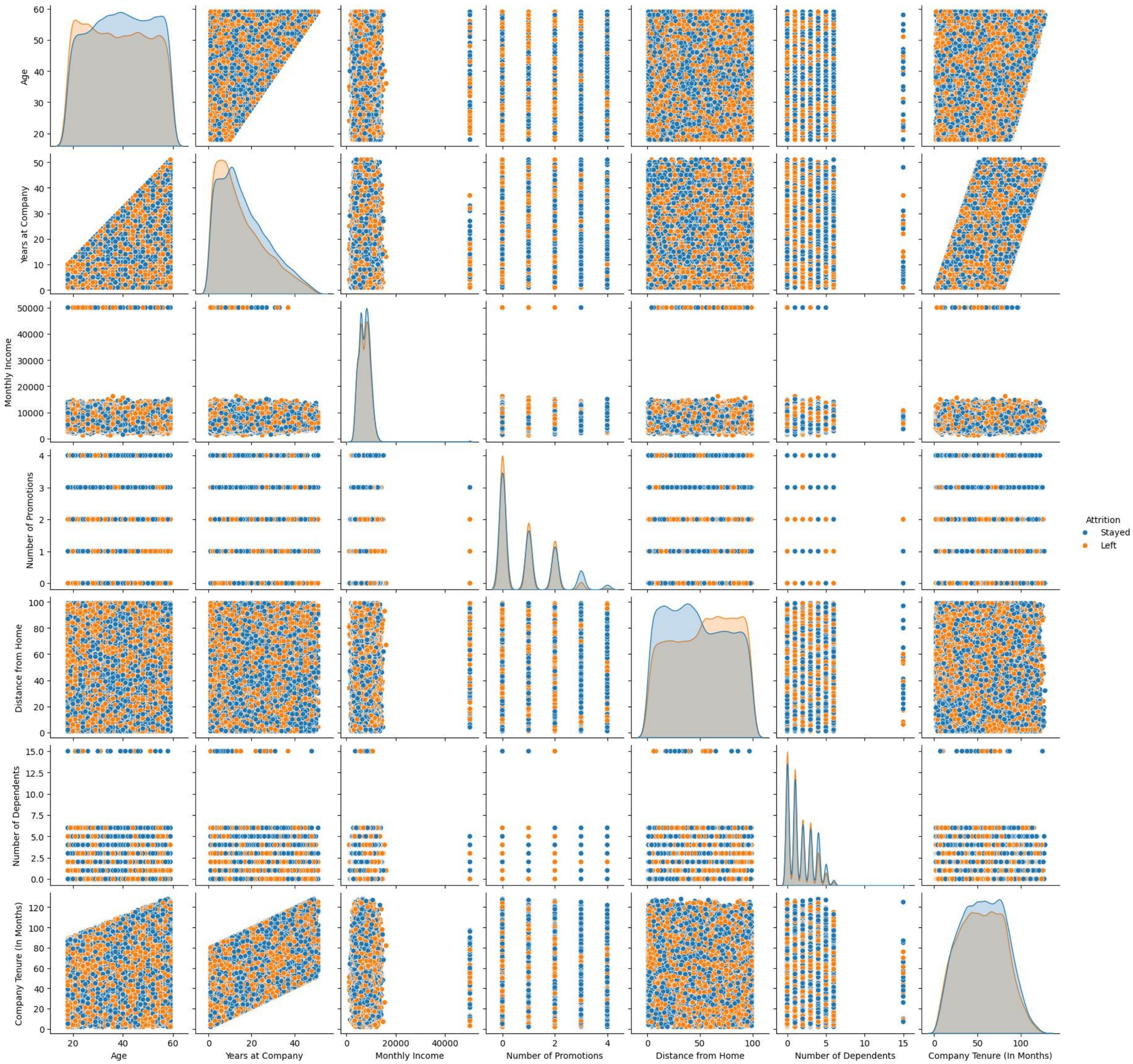
4.1 Perform univariate analysis

Perform univariate analysis on training data for all the numerical columns.

4.1.1 Select numerical columns from training data

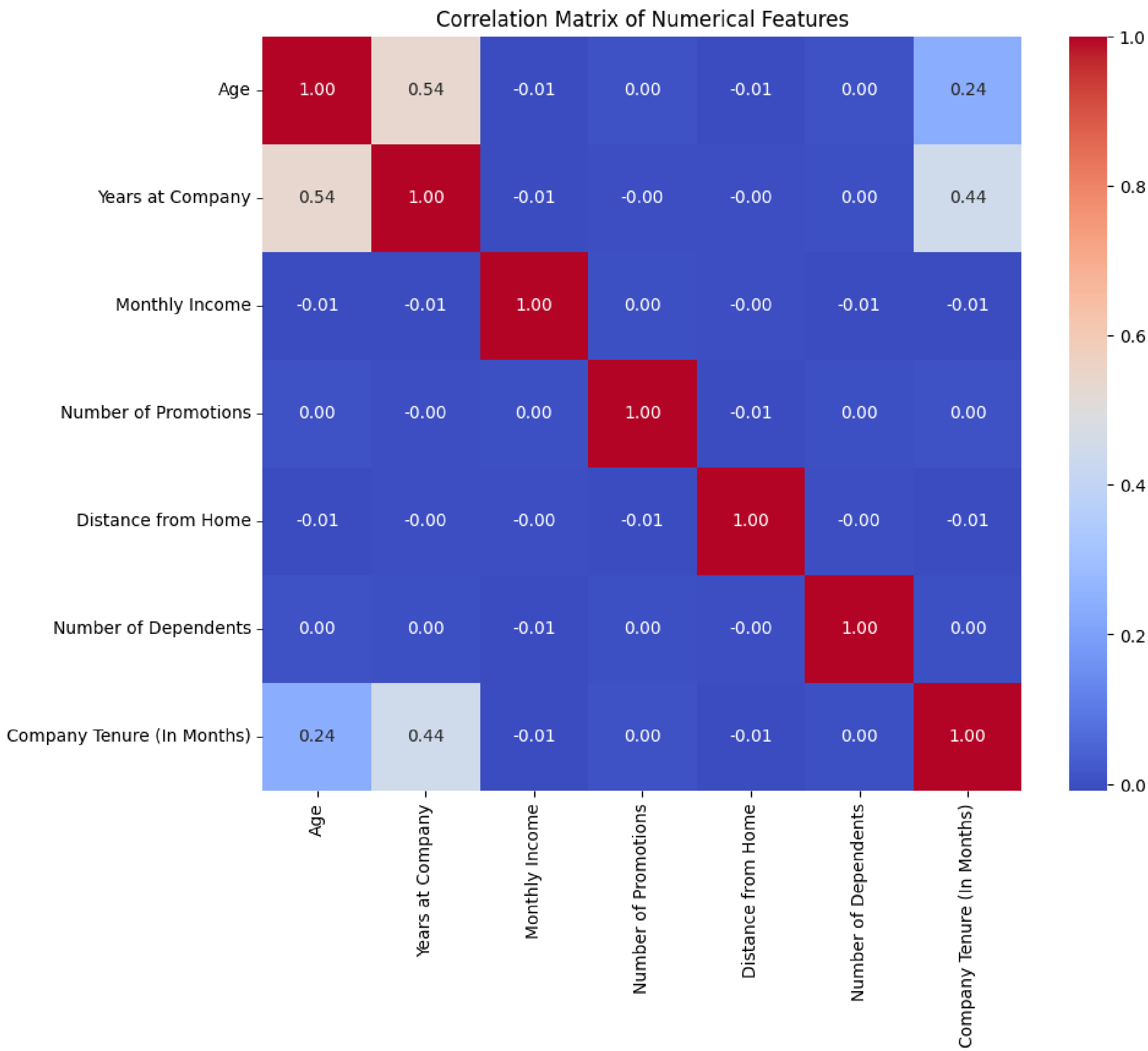
4.1.2 Plot distribution of numerical columns Plot all the numerical columns to understand their distribution

    Import necessary libraries
    import seaborn as sns
    import matplotlib.pyplot as plt
    import matplotlib.pyplot as plt



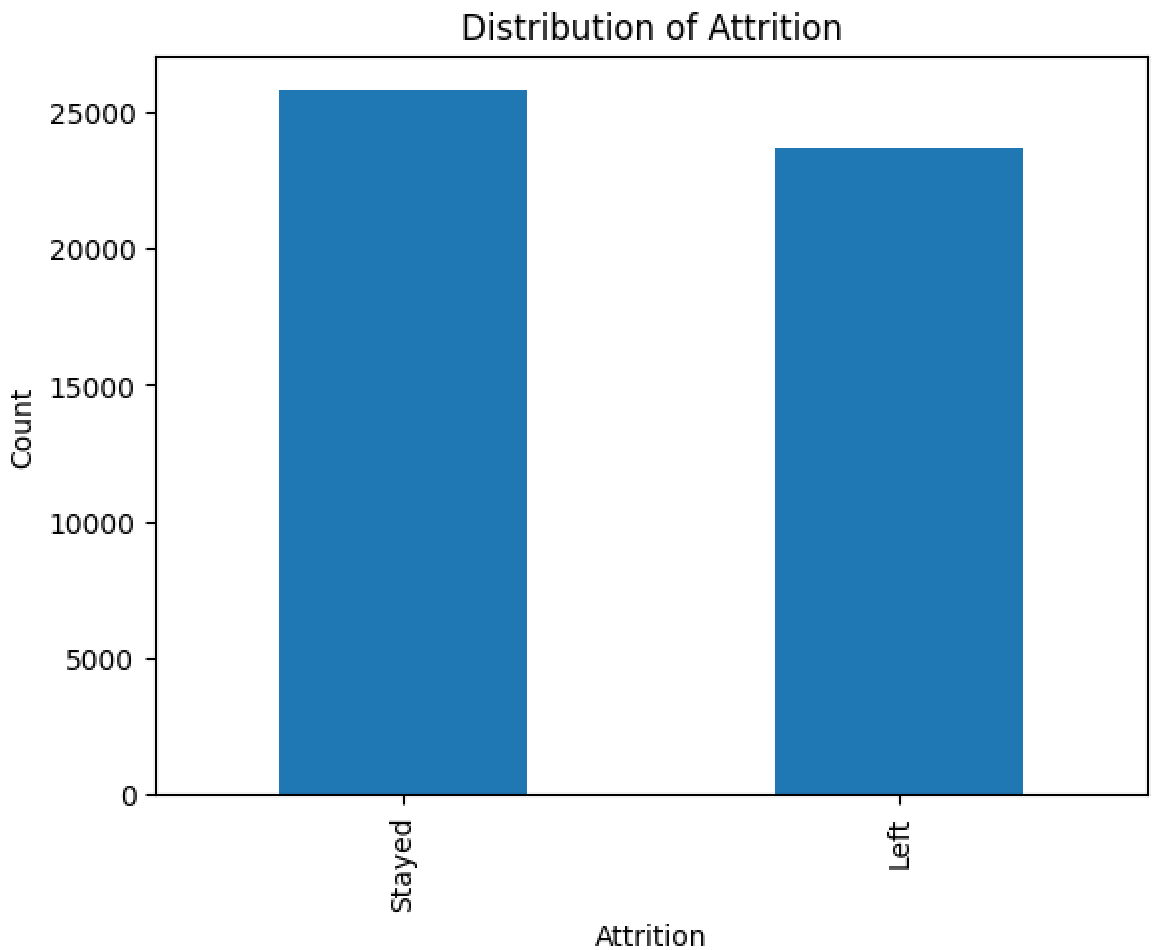
4.2 Perform correlation analysis

Check the correlation among different numerical variables.



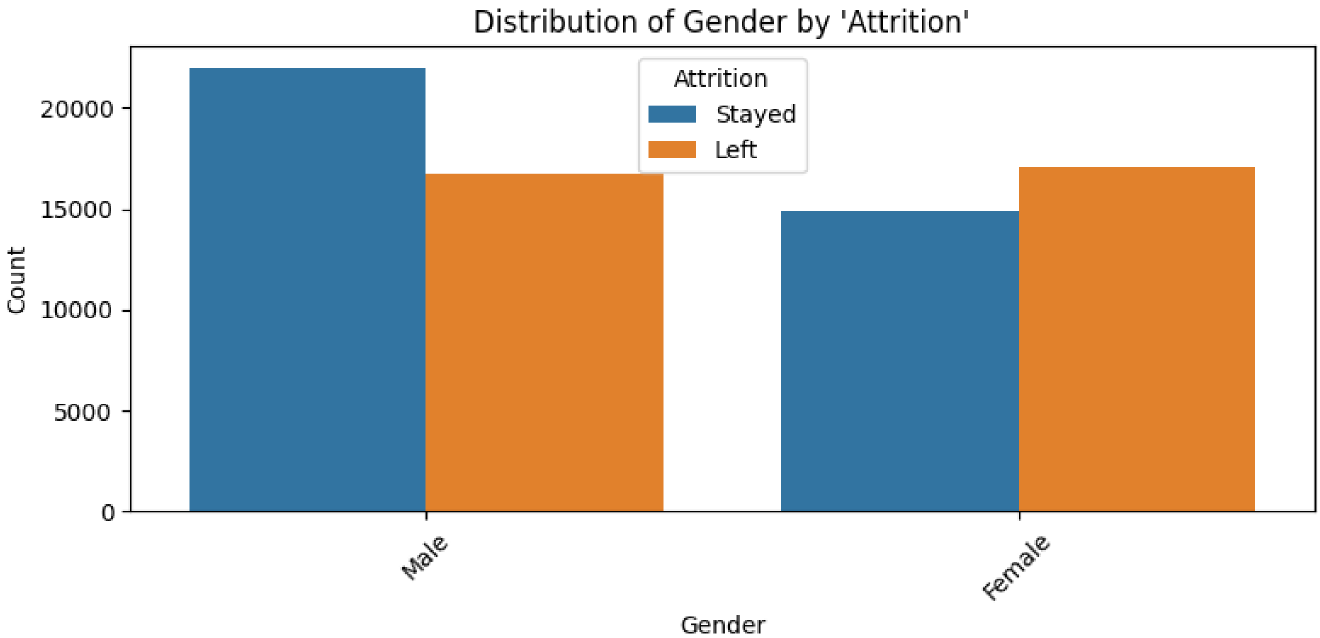
4.3 Check class balance

Check the distribution of target variable in training set to check class balance.

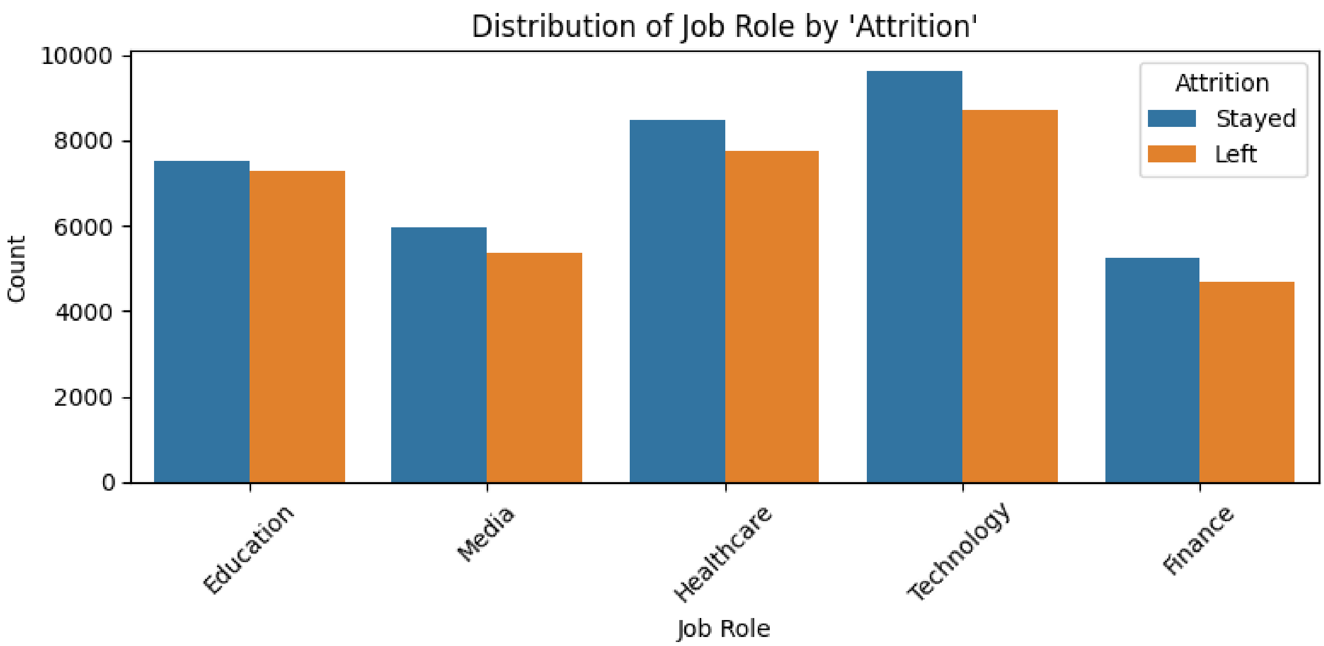


4.4 Perform bivariate analysis

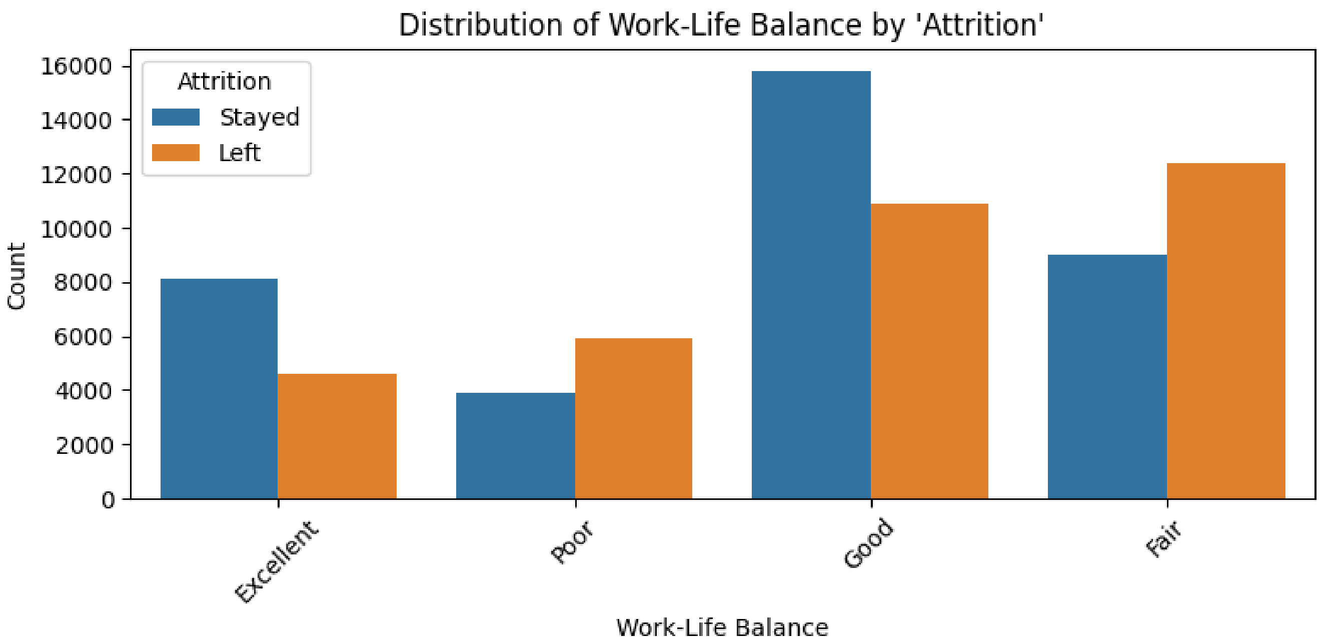
Perform bivariate analysis on training data between all the categorical columns and target variable to analyse how the categorical variables influence the target variable.

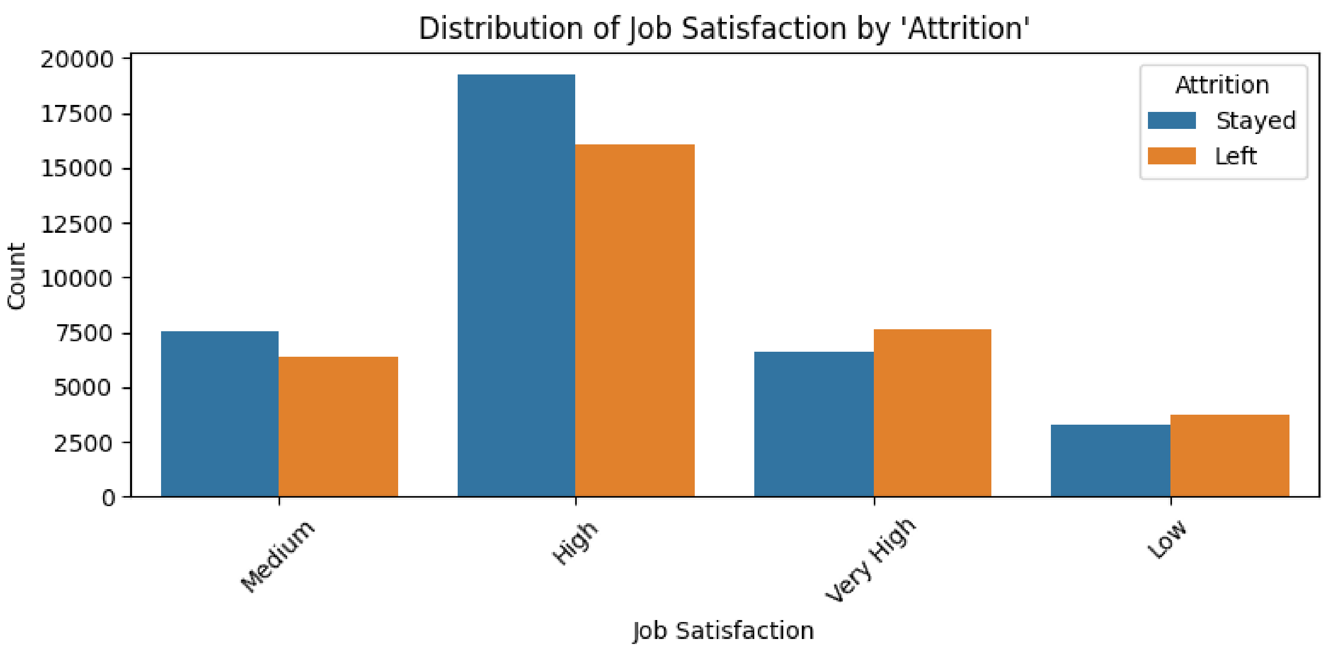


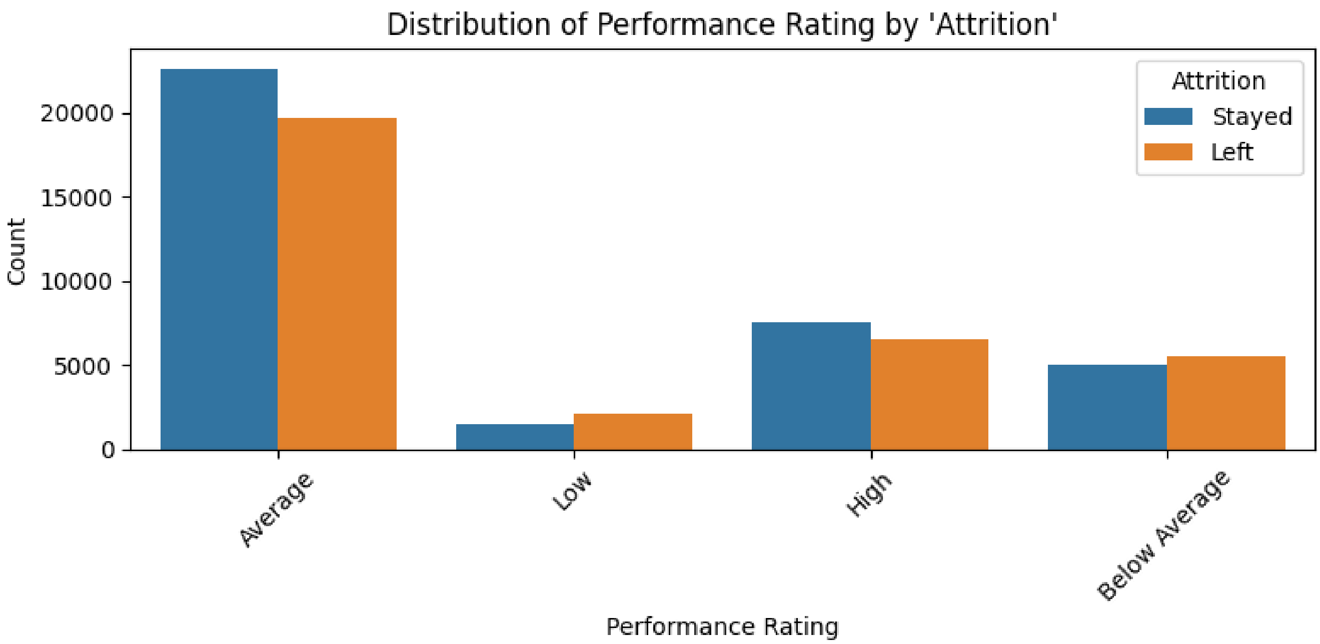
Distribution of Gender by 'Attrition'
Attrition
Stayed
Left
Count
0
5000
10000
15000
20000
Male
Female
Gender

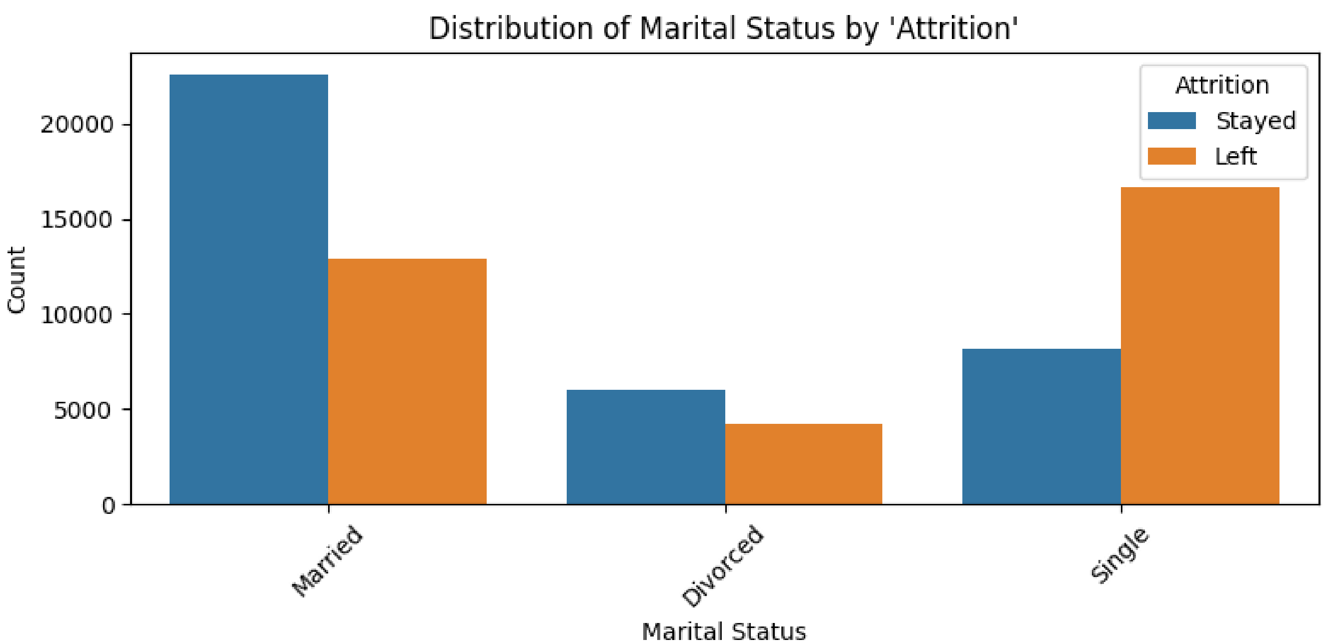


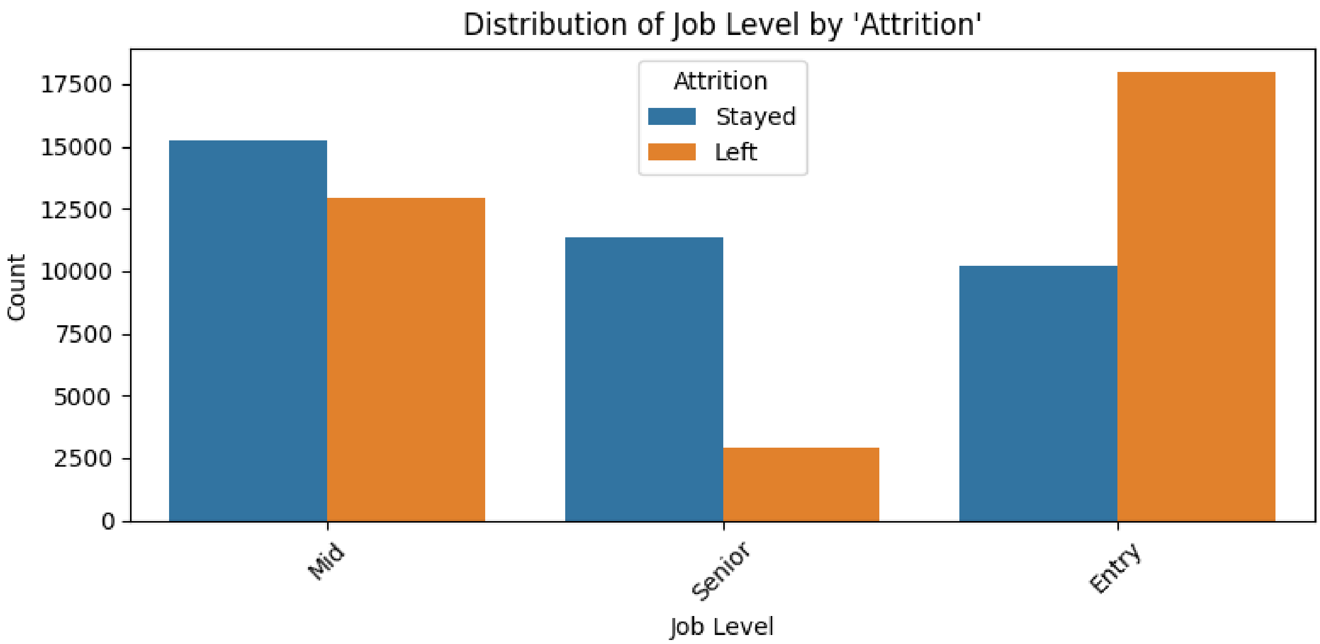
Distribution of Job Role by 'Attrition'
Attrition
Stayed
Left
Count
0
2000
4000
6000
8000
10000
Education
Media
Healthcare
Technology
Finance
Job Role

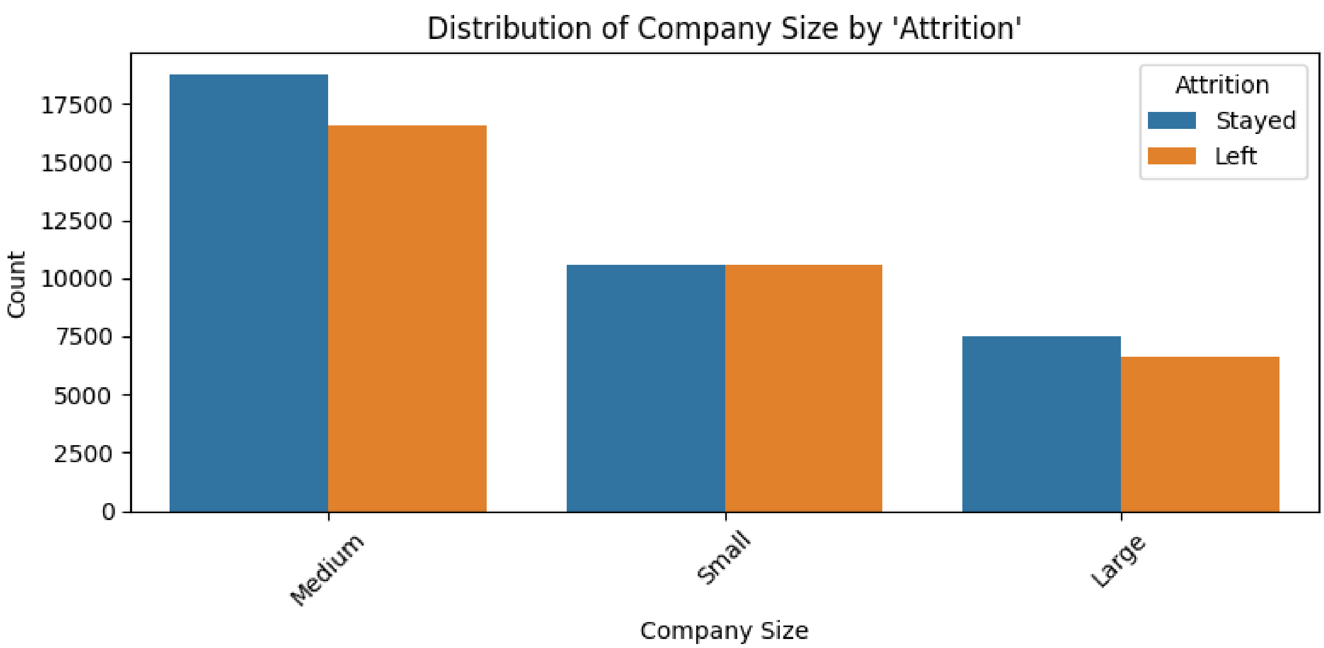

Distribution of Work-Life Balance by 'Attrition'
Attrition
Stayed
Left
Count
16000
14000
12000
10000
8000
6000
4000
2000
0
Excellent
Poor
Good
Fair
Work-Life Balance

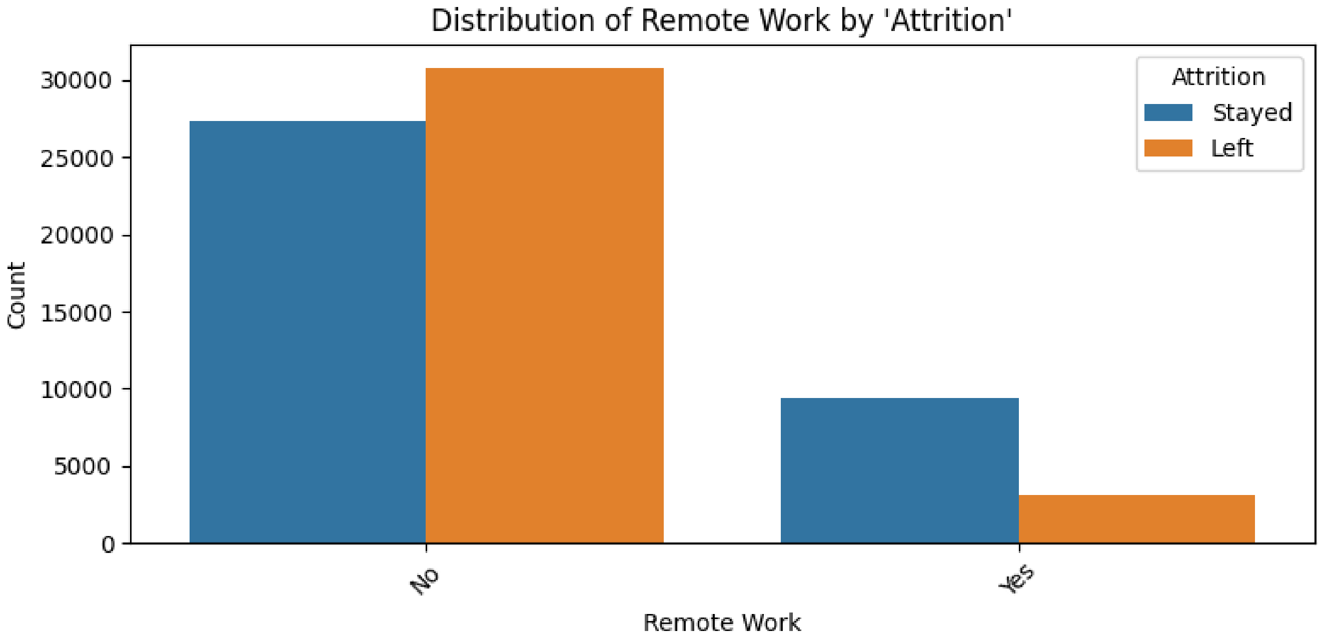

Distribution of Job Satisfaction by 'Attrition'
Attrition
Stayed
Left
Count
20000
17500
15000
12500
10000
7500
5000
2500
0
Medium
High
Very High
Low
Job Satisfaction

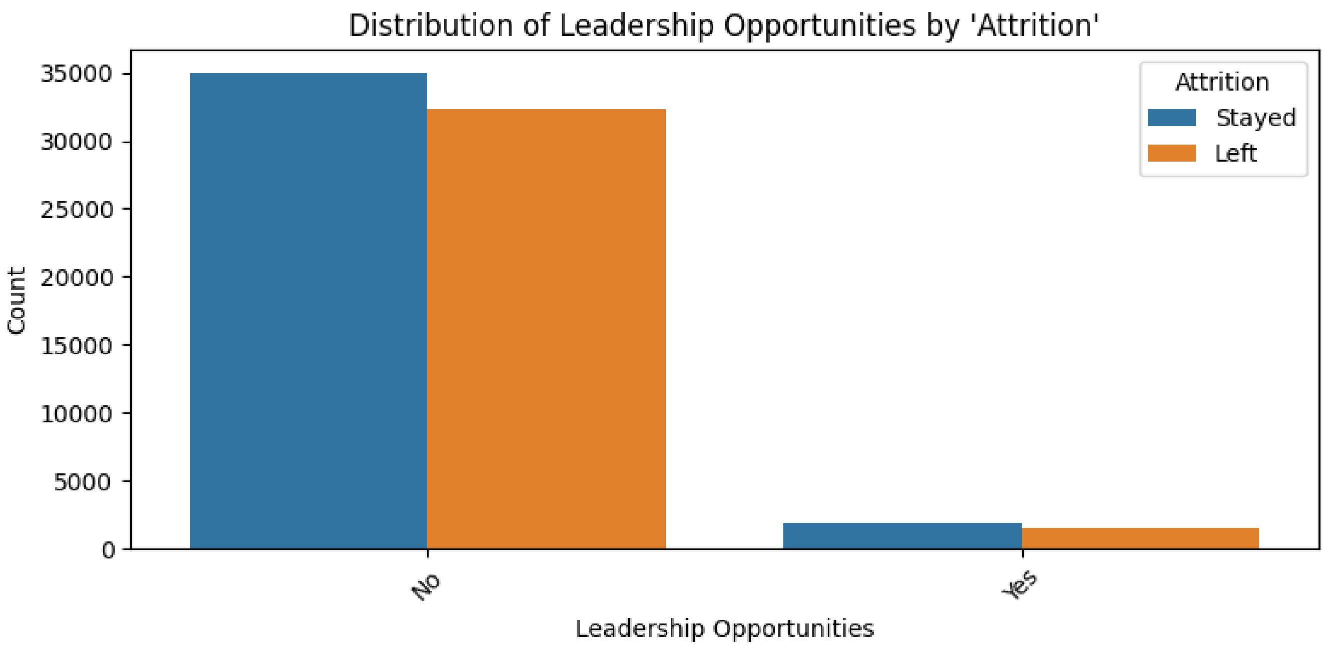

Distribution of Performance Rating by 'Attrition'
Attrition
Stayed
Left
Count
0
5000
10000
15000
20000
Average
Low
High
Below Average
Performance Rating

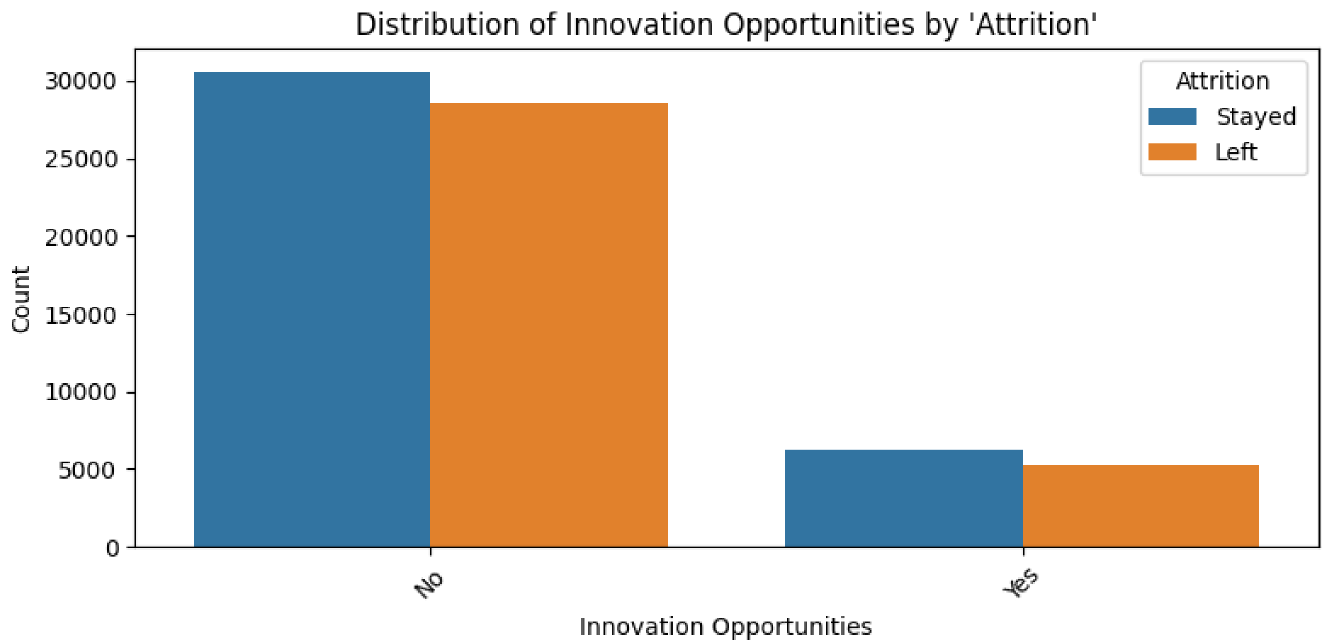

Distribution of Marital Status by 'Attrition'
Attrition
Stayed
Left
Count
0
5000
10000
15000
20000
Married
Divorced
Single
Marital Status


Distribution of Job Level by 'Attrition'
Attrition
Stayed
Left
Count
17500
15000
12500
10000
7500
5000
2500
0
Mid
Senior
Entry
Job Level

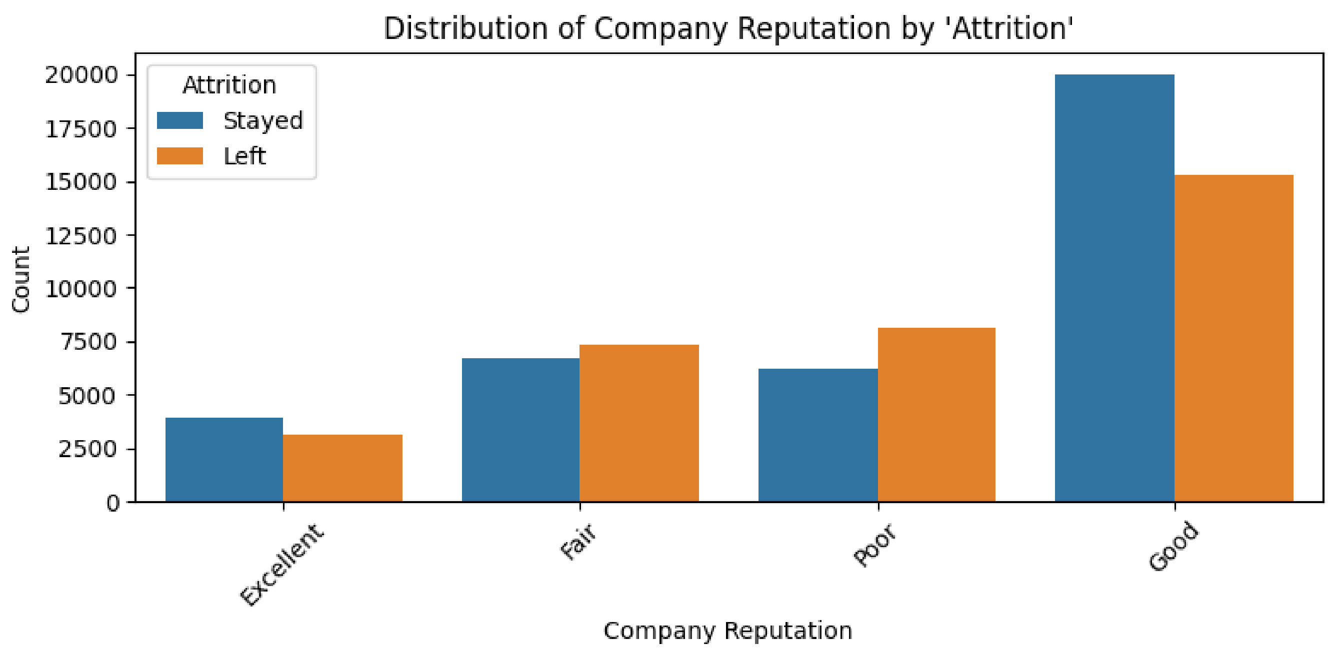

Distribution of Company Size by 'Attrition'
Attrition
Stayed
Left
Count
17500
15000
12500
10000
7500
5000
2500
0
Medium
Small
Large
Company Size


Distribution of Remote Work by 'Attrition'
Attrition
Stayed
Left
Count
Remote Work
No
Yes
0
5000
10000
15000
20000
25000
30000

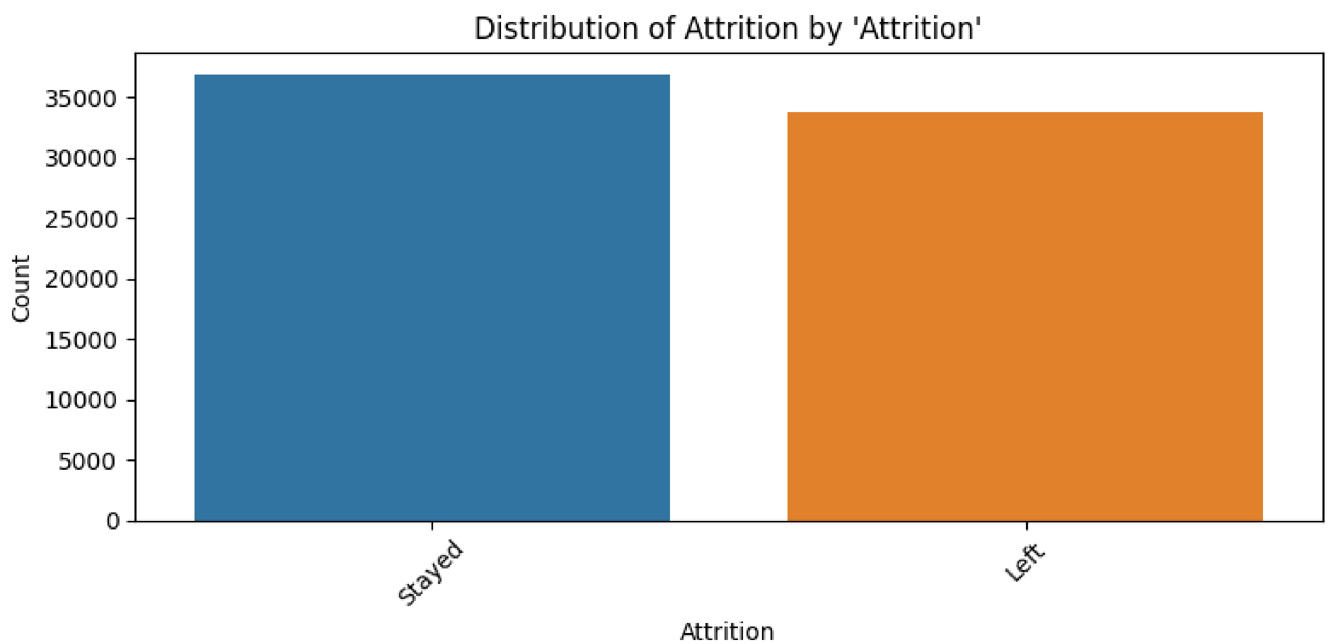

Distribution of Leadership Opportunities by 'Attrition'
Attrition
Stayed
Left
Count
Leadership Opportunities
No
Yes
0
5000
10000
15000
20000
25000
30000
35000



Distribution of Innovation Opportunities by 'Attrition'
Attrition
Stayed
Left
Count
30000
25000
20000
15000
10000
5000
0
No
Yes
Innovation Opportunities



Distribution of Company Reputation by 'Attrition'
Attrition
Stayed
Left
Count
20000
17500
15000
12500
10000
7500
5000
2500
0
Excellent
Fair
Poor
Good
Company Reputation



5. EDA on validation data

5.1 Perform univariate analysis

Perform univariate analysis on validation data for all the numerical columns.

5.1.1 Select numerical columns from validation data

5.1.2 Plot distribution of numerical columns

5.2 Perform correlation analysis

Check the correlation among different numerical variables.

5.3 Check class balance

Check the distribution of target variable in validation data to check class balance.

5.4 Perform bivariate analysis

Perform bivariate analysis on validation data between all the categorical columns and target variable to analyse how the categorical variables influence the target variable.

6. Feature Engineering

6.1 Dummy variable creation

The next step is to deal with the categorical variables present in the data.

6.1.1 Identify categorical columns where dummy variables are required

Work-Life Balance	
Good	18668
Fair	15041
Excellent	8867
Poor	6868

6.1.2 Create dummy variables for independent columns in training set

Now, drop the original categorical columns and check the DataFrame

6.1.3 Create dummy variables for independent columns in validation set

6.1.4 Create DataFrame for dependent column in both training and validation set

6.1.5 Create dummy variables for dependent column in training set

6.1.6 Create dummy variable for dependent column in validation set

6.1.7 Drop redundant columns

6.2 Feature scaling

Apply feature scaling to the numeric columns to bring them to a common range and ensure consistent scaling.

6.2.1 Import required libraries

- from sklearn.preprocessing import StandardScaler

6.2.2 Scale the numerical features

Model Building

7.1 Feature selection

As there are a lot of variables present in the data, Recursive Feature Elimination (RFE) will be used to select the most influential features for building the model.

7.1.1 Import required libraries

7.1.2 Import RFE and select 15 variables

- from sklearn.feature_selection import RFE

7.1.3 Store the selected features

7.2 Building Logistic Regression Model

Now that you have selected the variables through RFE, use these features to build a logistic regression model with statsmodels. This will allow you to assess the statistical aspects, such as p-values and VIFs, which are important for checking multicollinearity and ensuring that the predictors are not highly correlated with each other, as this could distort the model's coefficients.

7.2.1 Select relevant columns on training set

7.2.2 Add constant to training set

7.2.3 Fit logistic regression model

Model Interpretation

The output summary table will provide the features used for building model along with coefficient of each of the feature and their p-value. The p-value in a logistic regression model is used to assess the statistical significance of each coefficient. Lesser the p-value, more significant the feature is in the model.

A positive coefficient will indicate that an increase in the value of feature would increase the odds of the event occurring. On the other hand, a negative coefficient means the opposite, i.e, an increase in the value of feature would decrease the odds of the event occurring.

7.2.4 Evaluate VIF of features

	Feature	VIF
0	const	0.000000
1	Gender	1.000448
2	Remote Work	1.000395
3	Work-Life Balance_Excellent	inf
4	Work-Life Balance_Fair	inf
5	Work-Life Balance_Good	inf
6	Work-Life Balance_Poor	inf
7	Job Satisfaction_Low	1.028979
8	Job Satisfaction_Very High	1.028978
9	Performance Rating_Low	1.000078
10	Education Level_PhD	1.000259
11	Marital Status_Single	1.000207
12	Job Level_Entry	1.202031
13	Job Level_Senior	1.202087
14	Company Reputation_Excellent	1.123323
15	Company Reputation_Good	1.123006

Proceed to the next step if p-values and VIFs are within acceptable ranges. If you observe high p-values or VIFs, create new cells to drop the features and retrain the model.

	Feature	VIF
0	const	12.697819
1	Gender	1.000448
2	Remote Work	1.000395
3	Work-Life Balance_Excellent	1.880679
4	Work-Life Balance_Fair	2.220539
5	Work-Life Balance_Good	2.314942
6	Job Satisfaction_Low	1.028994
7	Job Satisfaction_Very High	1.028982
8	Performance Rating_Low	1.000487
9	Education Level_PhD	1.000259
10	Marital Status_Single	1.000224
11	Job Level_Entry	1.202077
12	Job Level_Senior	1.202088
13	Company Reputation_Excellent	1.123324
14	Company Reputation_Good	1.123008

7.2.5 Make predictions on training set

7.2.6 Format the prediction output

```
(49444, 1)
```

7.2.7 Create a DataFrame with the actual stayed flag and the predicted probabilities

```
       Actual_Attrition  Predicted_Probability
41465                 1               0.943048
69350                 1               0.830942
28247                 1               0.820431
3217                  1               0.140556
73636                 1               0.895176
```

7.2.8 Create a new column 'Predicted' with 1 if predicted probabilities are greater than 0.5 else 0

```
       Actual_Attrition  Predicted_Probability  Predicted
41465                 1               0.943048          1
69350                 1               0.830942          1
28247                 1               0.820431          1
3217                  1               0.140556          0
73636                 1               0.895176          1
```

Evaluation of performance of Model

Evaluate the performance of the model based on the predictions made on the training set.

7.2.9 Check the accuracy of the model based on the predictions made on the training set

```
Training Accuracy: 0.7527
```

7.2.10 Create a confusion matrix based on the predictions made on the training set

```
Confusion Matrix:
 [[19745  6040]
 [ 6186 17473]]
```

7.2.11 Create variables for true positive, true negative, false positive and false negative

```
True Negatives (TN): 19745
False Positives (FP): 6040
False Negatives (FN): 6186
True Positives (TP): 17473
```

7.2.12 Calculate sensitivity and specificity of model

```
Sensitivity (Recall): 0.7385

Specificity: 0.7658
```

7.2.13 Calculate precision and recall of model

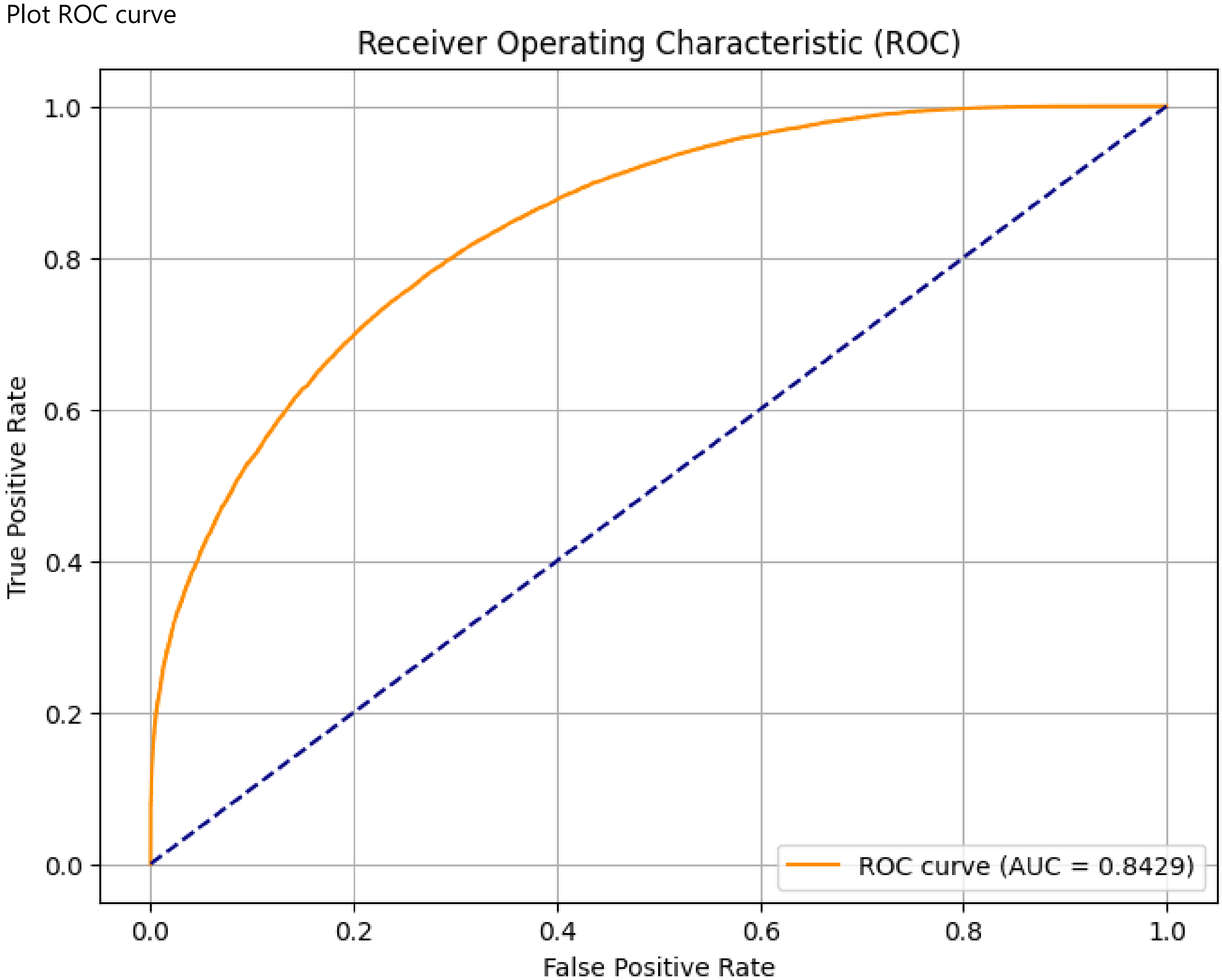
```
Precision: 0.7431

Recall: 0.7385
```

7.3 Find the optimal cutoff

Find the optimal cutoff to improve model performance. While a default threshold of 0.5 was used for initial evaluation, optimising this threshold can enhance the model's performance.

First, plot the ROC curve and check AUC.

7.3.1 Plot ROC curve




```
AUC Score: 0.8429
```

Sensitivity and Specificity tradeoff

Check sensitivity and specificity tradeoff to find the optimal cutoff point.

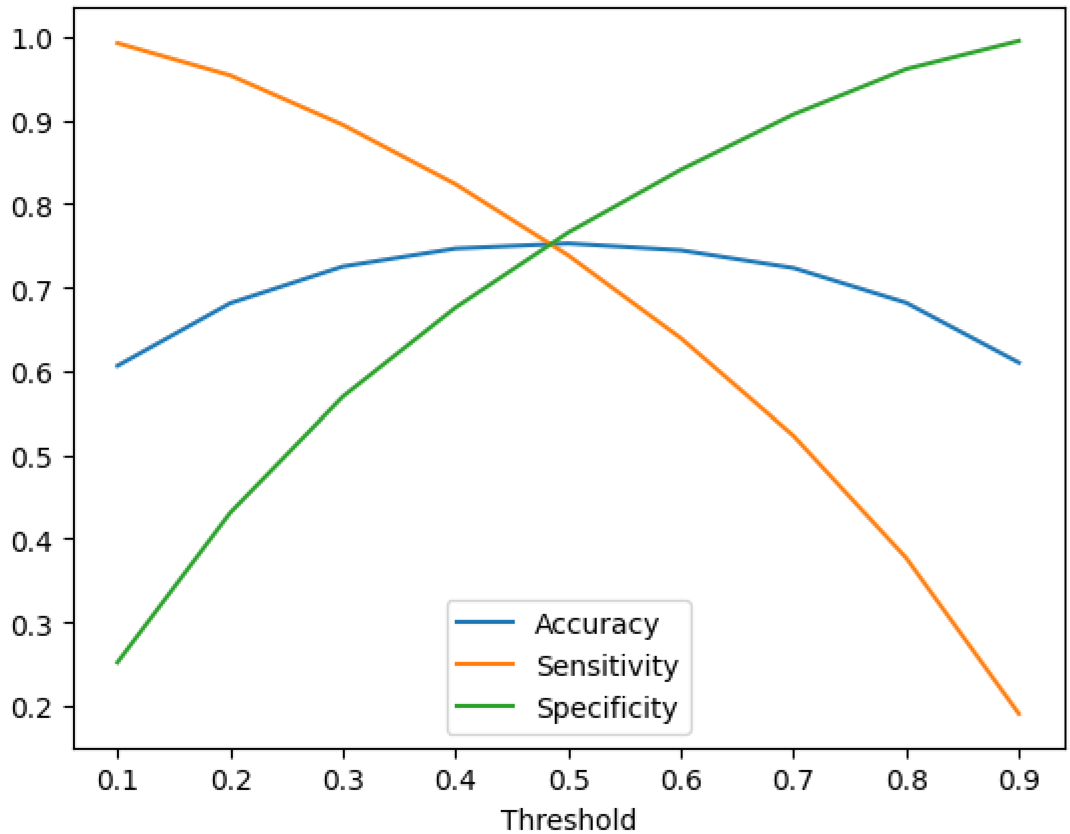
7.3.2 Predict on training set at various probability cutoffs

7.3.3 Plot for accuracy, sensitivity, specificity at different probability cutoffs

Metrics at Different Thresholds:

	Threshold	Accuracy	Sensitivity	Specificity
0	0.1	0.606302	0.992434	0.252007
1	0.2	0.681195	0.953886	0.430987
2	0.3	0.725123	0.894628	0.569595
3	0.4	0.746622	0.823661	0.675936

4	0.5	0.752730	0.738535	0.765755
5	0.6	0.744519	0.639418	0.840954
6	0.7	0.723323	0.522972	0.907155
7	0.8	0.681721	0.376939	0.961373
8	0.9	0.610084	0.190498	0.995075



7.3.4 Create a column for final prediction based on the optimal cutoff

7.3.5 Calculate model's accuracy

Training Accuracy with Optimal Cutoff: 0.7527

7.3.6 Create confusion matrix

7.3.7 Create variables for true positive, true negative, false positive and false negative

```
True Negatives (TN): 19745
False Positives (FP): 6040
False Negatives (FN): 6186
True Positives (TP): 17473
```

7.3.8 Calculate sensitivity and specificity of the model

Sensitivity (Recall): 0.8346

Specificity: 0.7658

7.3.9 Calculate precision and recall of the model

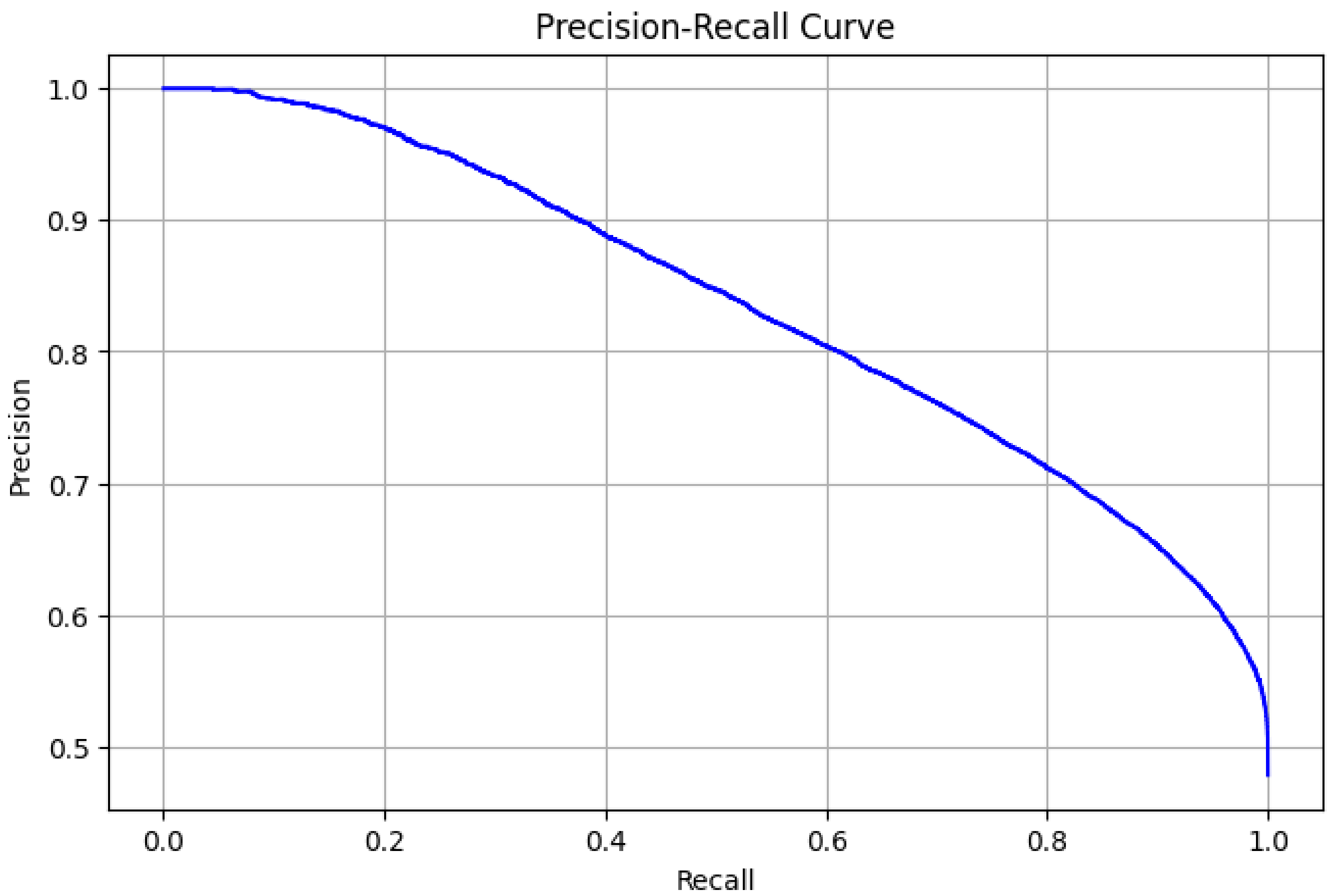
Precision: 0.7431

Recall: 0.7385

Precision and Recall tradeoff

Check optimal cutoff value by plotting precision-recall curve, and adjust the cutoff based on the precision and recall tradeoff if required.

7.3.10 Plot precision-recall curve



7. Prediction and Model Evaluation

Use the model from the previous step to make predictions on the validation set with the optimal cutoff. Then evaluate the model's performance using metrics such as accuracy, sensitivity, specificity, precision, and recall.

8.1 Make predictions over validation set

8.1.1 Select relevant features for validation set

8.1.2 Add constant to X_validation
8.1.3 Make predictions over validation set

8.1.4 Create DataFrame with actual values and predicted values for validation set

8.1.5 Predict final prediction based on the cutoff value

8.2 Calculate accuracy of the model

Accuracy of the Model on Validation Set: 0.72852

8.3 Create confusion matrix and create variables for true positive, true negative, false positive and false negative

```
True Negative (TN): 8179
False Positive (FP): 2846
False Negative (FN): 2907
True Positive (TP): 7259
```

8.4 Calculate sensitivity and specificity

```
Sensitivity (Recall): 0.71405

Specificity: 0.74186
```

8.5 Calculate precision and recall

```
Precision: 0.71836

Recall: 0.71405
```

Conclusion

This notebook builds an end-to-end pipeline for predicting employee attrition using logistic regression. It follows key data science steps:

- **Cleaning**
- **Preprocessing**
- **Modeling**
- **Evaluation**
- **Interpretation**

The logistic regression model helps HR or management understand which features (like income, overtime, or job satisfaction) influence whether an employee is likely to leave.

Submitted by

Apurv Goyal

Yamini Kodali