Predicting Employee Retention Case Study Report

Group: *Pranav Jadhav* & *Shiva Ashwardh M*

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:

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.

Import the libraries

```
import numpy as np
import pandas as pd
from rapidfuzz import fuzz
pd.setoption('display.maxcolumns', None)
pd.setoption('display.maxcolwidth', None)
```

Load the dataset

	Employee	Age	Gender	Years at C	Job Role	Monthly I	Work-Life	Job Satisfa	Performar	Number o	Overtime	Distance f	Education	Marital Sta	Number o	Job Level	Company	Company	Remote W	Leadershi	Innovatio	Company	Employee	Attrition
2	8410	31	Male	19	Education	5390	Excellent	Medium	Average	2	No	22	Associate	Married	0	Mid	Medium	89	No	No	No	Excellent	Medium	Stayed
3	64756	59	Female	4	Media	5534	Poor	High	Low	3	No	21	Masterâ€™	Divorced	3	Mid	Medium	21	No	No	No	Fair	Low	Stayed
4	30257	24	Female	10	Healthcare	8159	Good	High	Low	0	No	11	Bachelorâ	Married	3	Mid	Medium	74	No	No	No	Poor	Low	Stayed
5	65791	36	Female	7	Education	3989	Good	High	High	1	No	27	High Scho	Single	2	Mid	Small	50	Yes	No	No	Good	Medium	Stayed
6	65026	56	Male	41	Education	4821	Fair	Very High	Average	0	Yes	71	High Scho	Divorced	0	Senior	Medium	68	No	No	No	Fair	Medium	Stayed
7	24368	38	Female	3	Technolog	9977	Fair	High	Below Ave	3	No	37	Bachelorâ	Married	0	Mid	Medium	47	No	No	Yes	Fair	High	Left
8	64970	47	Male	23	Education	3681	Fair	High	High	1	Yes	75	High Scho	Divorced	3	Entry	Small	93	No	No	No	Good	Medium	Left

Shape of the dataset : (74610, 24)

Check the info to see the types of the feature variables and the null values present

- emp_data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74610 entries, 0 to 74609
Data columns (total 24 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Employee ID                 74610 non-null  int64
 1   Age                         74610 non-null  int64
 2   Gender                      74610 non-null  object
 3   Years at Company            74610 non-null  int64
 4   Job Role                    74610 non-null  object
 5   Monthly Income              74610 non-null  int64
 6   Work-Life Balance           74610 non-null  object
 7   Job Satisfaction            74610 non-null  object
 8   Performance Rating          74610 non-null  object
 9   Number of Promotions        74610 non-null  int64
 10  Overtime                    74610 non-null  object
 11  Distance from Home          72698 non-null  float64
 12  Education Level             74610 non-null  object
 13  Marital Status              74610 non-null  object
 14  Number of Dependents        74610 non-null  int64
 15  Job Level                   74610 non-null  object
 16  Company Size                74610 non-null  object
 17  Company Tenure (In Months)  72197 non-null  float64
 18  Remote Work                 74610 non-null  object
 19  Leadership Opportunities    74610 non-null  object
 20  Innovation Opportunities    74610 non-null  object
 21  Company Reputation          74610 non-null  object
 22  Employee Recognition        74610 non-null  object
 23  Attrition                   74610 non-null  object
dtypes: float64(2), int64(6), object(16)
memory usage: 13.7+ MB
```

2. Data Cleaning

2.1 Handling the missing values

	0
Employee ID	0.000000
Age	0.000000
Gender	0.000000
Years at Company	0.000000
Job Role	0.000000
Monthly Income	0.000000
Work-Life Balance	0.000000
Job Satisfaction	0.000000
Performance Rating	0.000000
Number of Promotions	0.000000
Overtime	0.000000
Distance from Home	2.562659
Education Level	0.000000
Marital Status	0.000000
Number of Dependents	0.000000
Job Level	0.000000
Company Size	0.000000
Company Tenure (In Months)	3.234151
Remote Work	0.000000
Leadership Opportunities	0.000000
Innovation Opportunities	0.000000
Company Reputation	0.000000
Employee Recognition	0.000000
Attrition	0.000000

As seen above there are **1912(2.56%)** missing values in ‘Distance from Home’ and **2413(3.23%)** missing values in ‘Company Tenure (In Months)’

2.2. Handling redundant values

Unique values of the categorical Columns:

Gender :	['Male' 'Female']
Job Role:	['Education' 'Media' 'Healthcare' 'Technology' 'Finance']
Work-Life Balance:	['Excellent' 'Poor' 'Good' 'Fair']
Job Satisfaction:	['Medium' 'High' 'Very High' 'Low']
Performance Rating:	['Average' 'Low' 'High' 'Below Average']
Overtime:	['No' 'Yes']
Education Level:	['Associate Degree' 'Masterâ€™s Degree' 'Bachelorâ€™s Degree' 'High School' 'PhD']
Marital Status:	['Married' 'Divorced' 'Single']
Job Level:	['Mid' 'Senior' 'Entry']
Company Size:	['Medium' 'Small' 'Large']
Remote Work:	['No' 'Yes']
Leadership Opportunities:	['No' 'Yes']
Innovation Opportunities:	['No' 'Yes']
Company Reputation:	['Excellent' 'Fair' 'Poor' 'Good']
Employee Recognition:	['Medium' 'Low' 'High' 'Very High']
Attrition:	['Stayed' 'Left']

2.3 Drop redundant columns

Dropping the Employee ID and Company Tunure (In Months) columns form the dataframe

- emp_data_copy.drop(['Employee ID', 'Company Tenure (In Months)'], axis = 1, inplace = True)

3. Train-Validation Split

Splitting the data into train and validation datasample in the ratio 70:30, By using model_selection in sklearn Library.

Put all the feature variables in X

- **X = emp_data_copy.drop('Attrition', axis = 1)**

Put the target variable in y

- **y = emp_data_copy['Attrition']**

Splitting the data using train_test_split method from sklearn library

- **x_train, x_val, y_train, y_val = train_test_split(x,y,test_size = 0.3, random_state = 42)**

This method return 4 variables

1. **x_train**
2. **x_val**
3. **y_train**
4. **y_val**

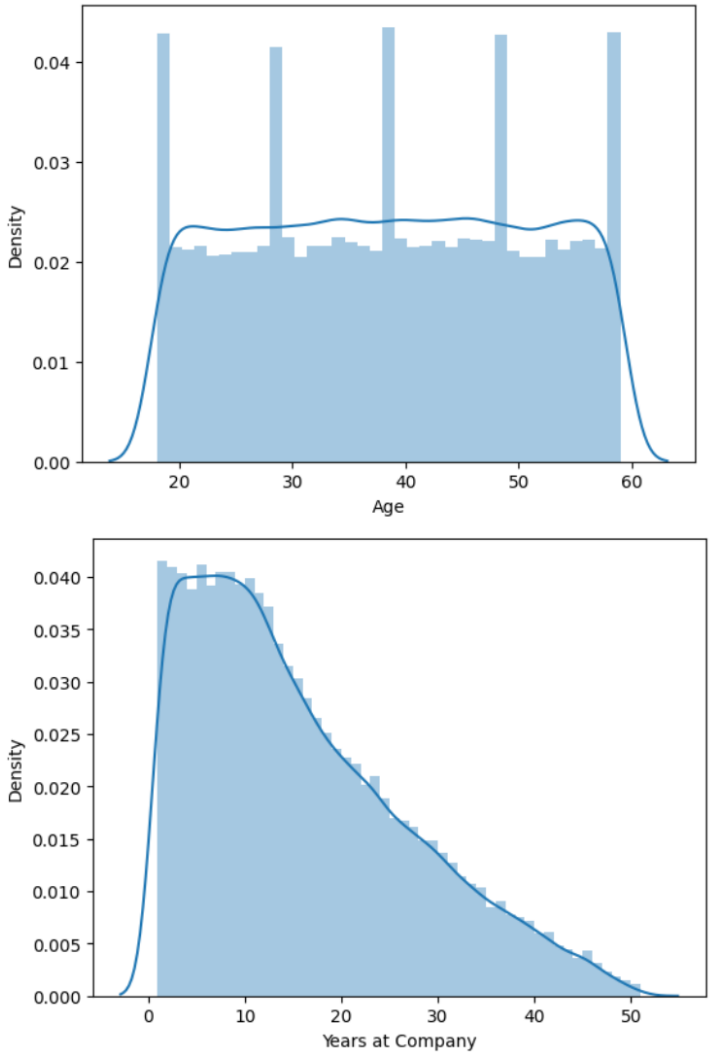
4.EDA on training data

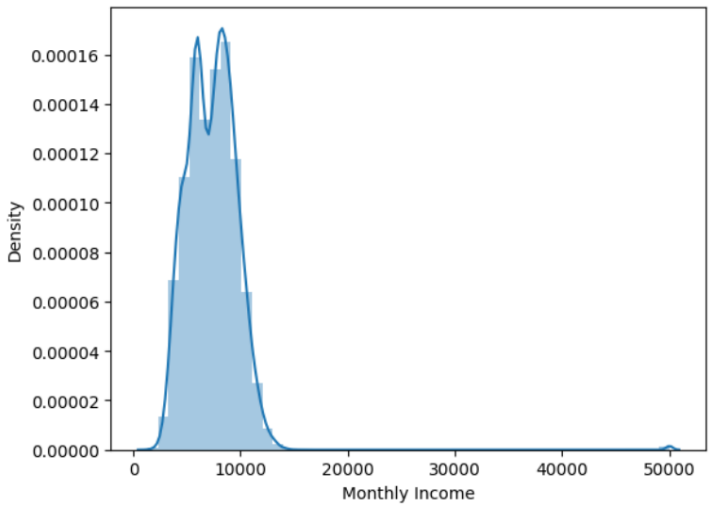
Performing Univariate Analysis

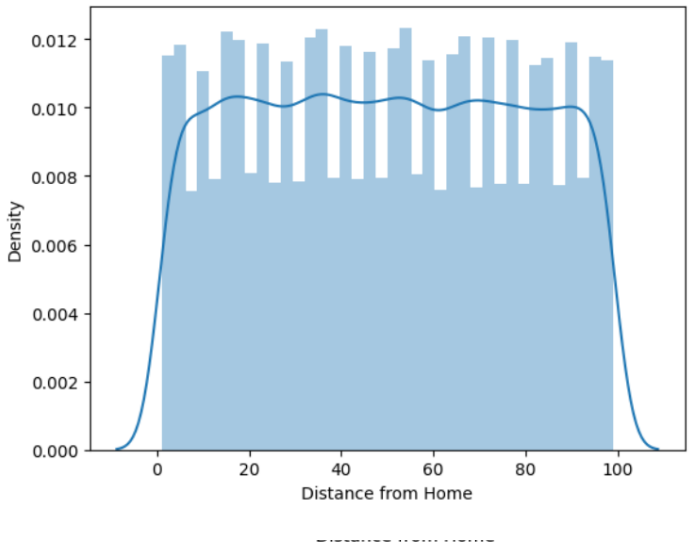
Libraries required are:

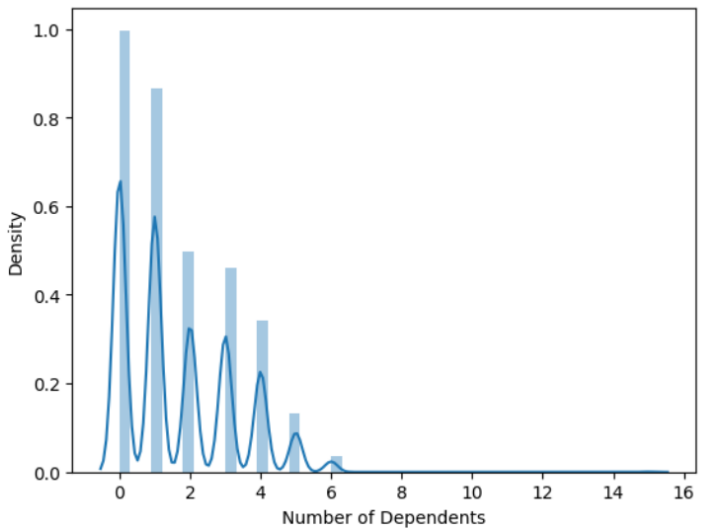
- **import seaborn as sns**
- **import matplotlib.pyplot as plt**
- **%matplotlib inline**

Plotting the distribution of all the Numeric columns

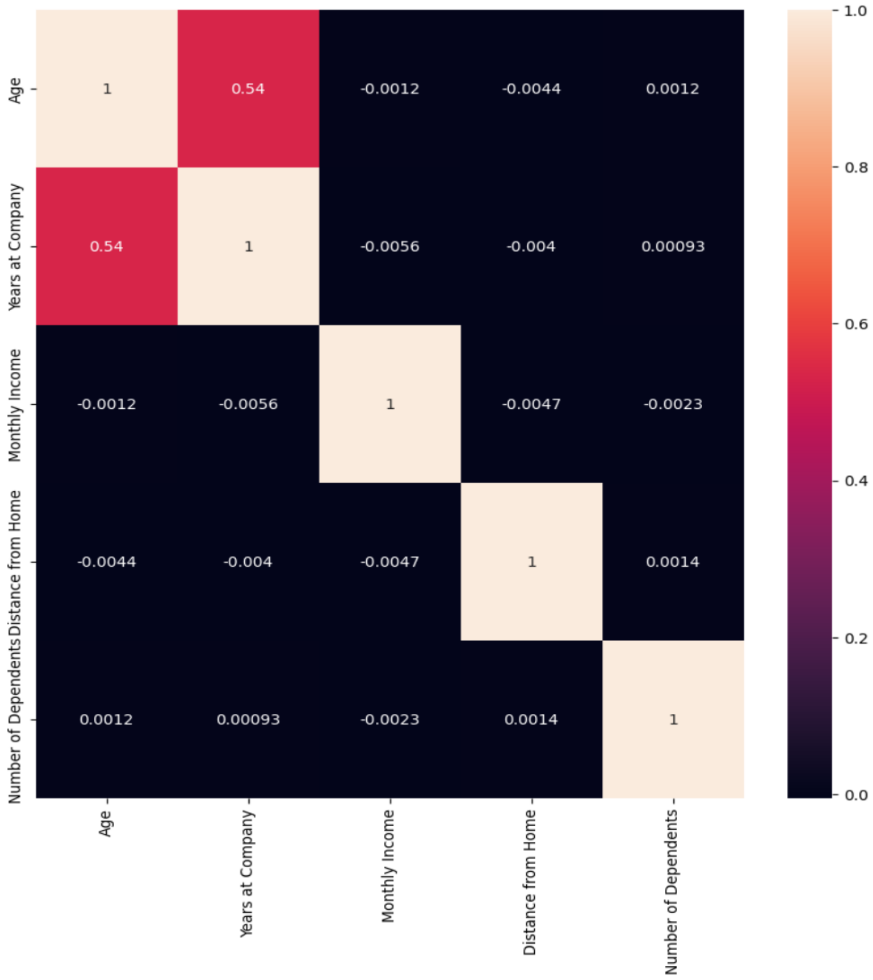



0.00016
0.00014
0.00012
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
Density
0
10000
20000
30000
40000
50000
Monthly Income


0.012
0.010
0.008
0.006
0.004
0.002
0.000
Density
0
20
40
60
80
100
Distance from Home


1.0
0.8
0.6
0.4
0.2
0.0
Density
0
2
4
6
8
10
12
14
16
Number of Dependents

Correlation among all the Numerical Variables are given below:

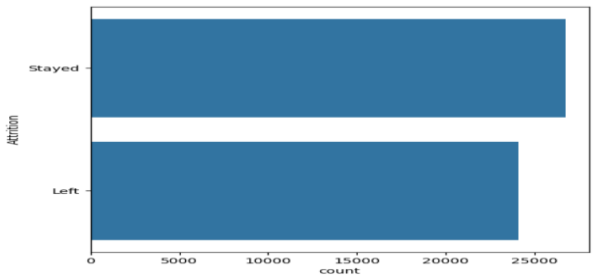


From the above Heatmap we can clearly say that,

- Correlation between 'Years at Company' and 'Age' is greatest i.e., 0.54
- All the other have way less correlation values.

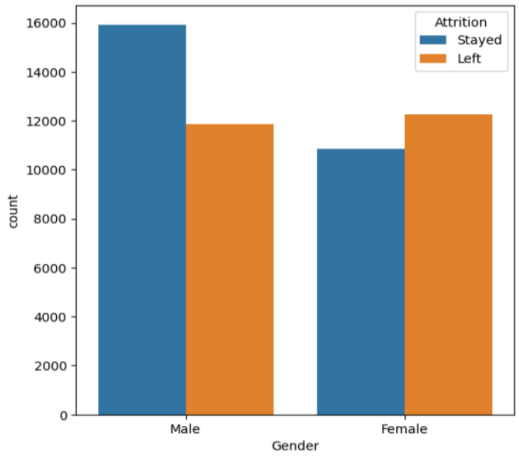
Checking Class Balance

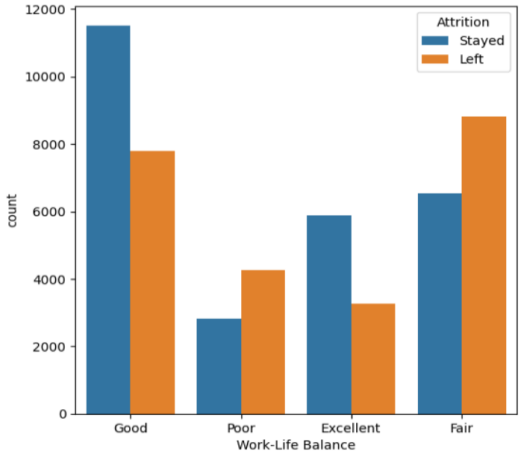
Check the distribution of target variable in training set to check class balance.

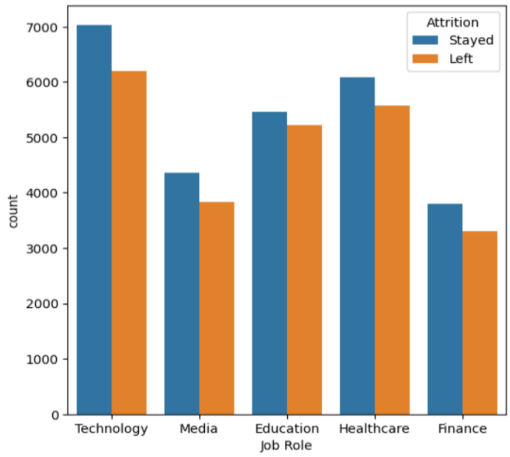


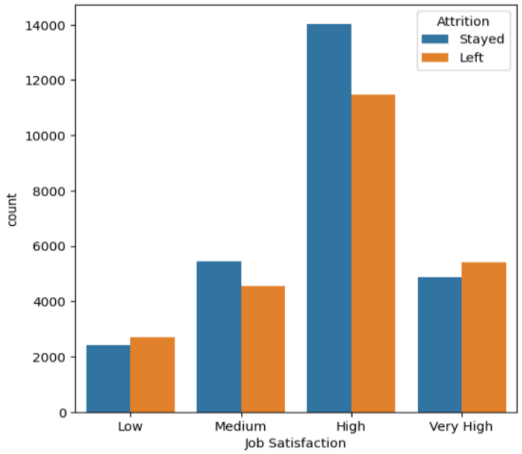
From the above balance we can say the target variable is well balance.

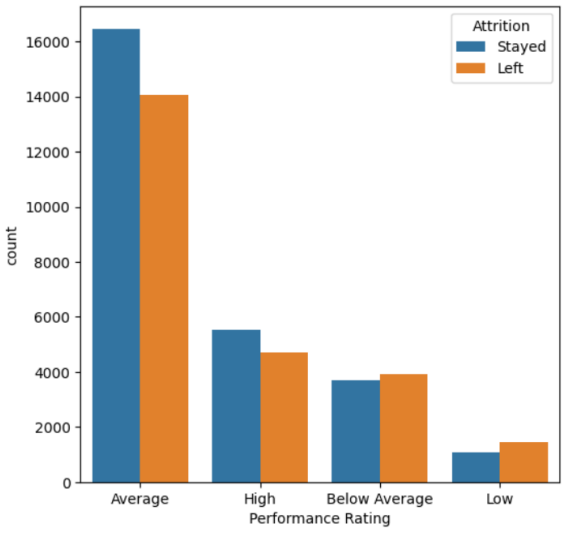
Performing Bivariate Analysis

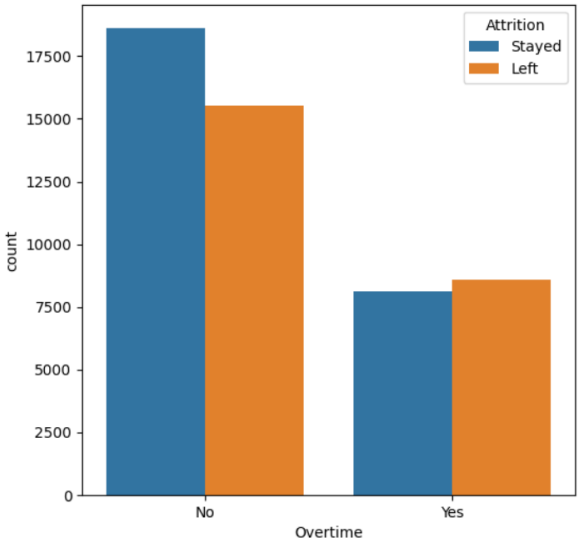


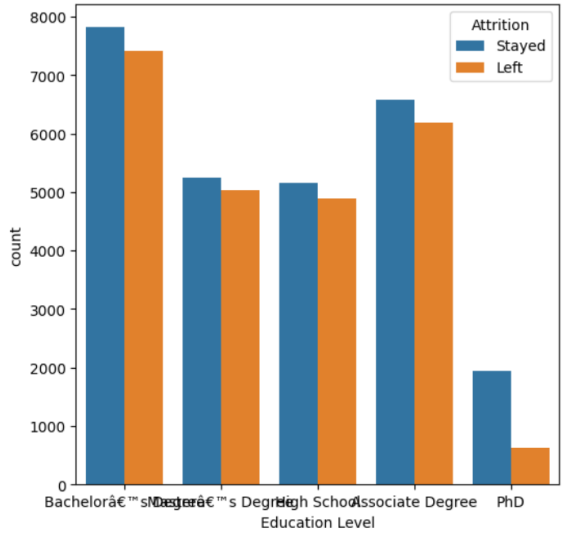


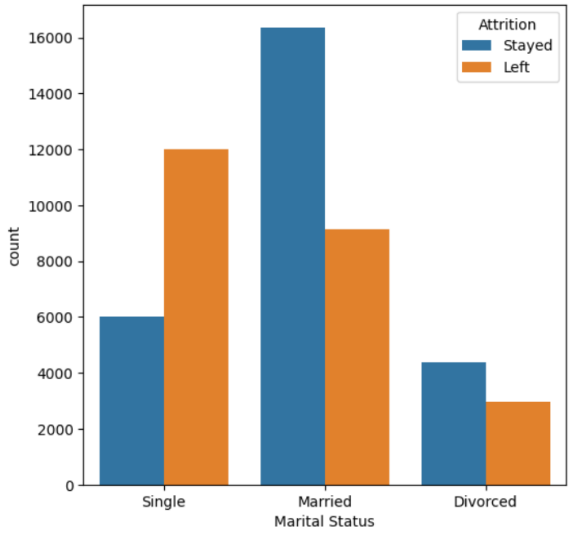


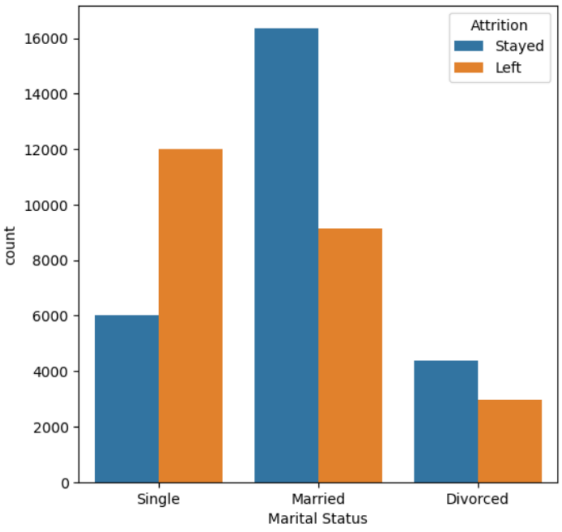


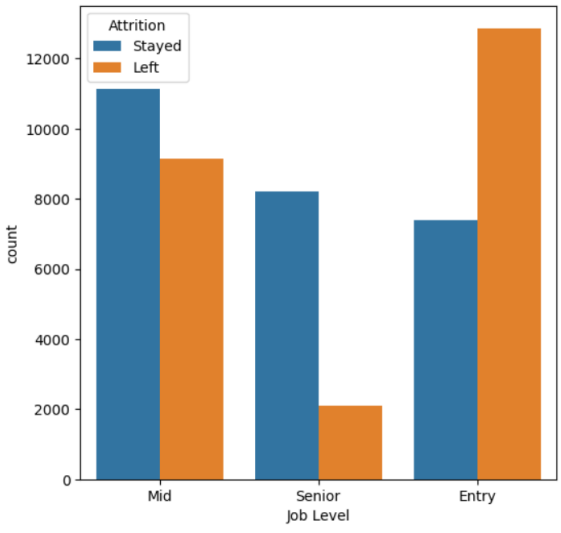

Attrition
Stayed
Left
count
Average
High
Below Average
Low
Performance Rating

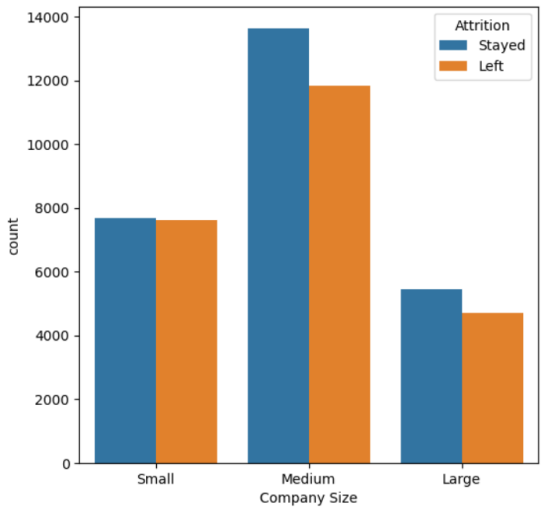

Attrition
Stayed
Left
count
No
Yes
Overtime

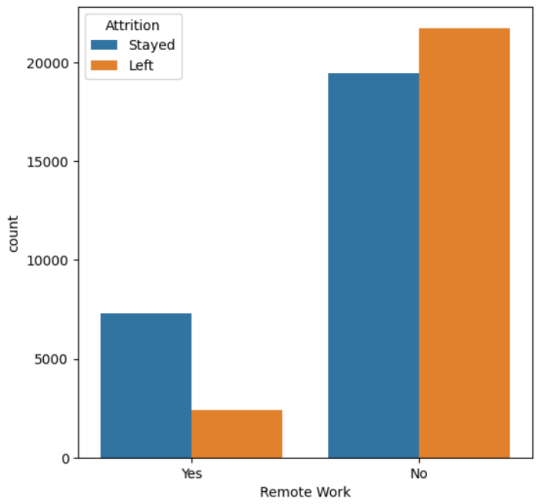

Attrition
Stayed
Left
count
Education Level

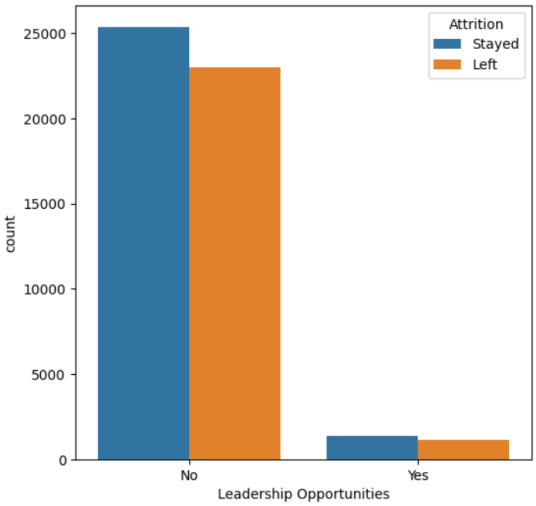

Attrition
Stayed
Left
count
Single
Married
Divorced
Marital Status

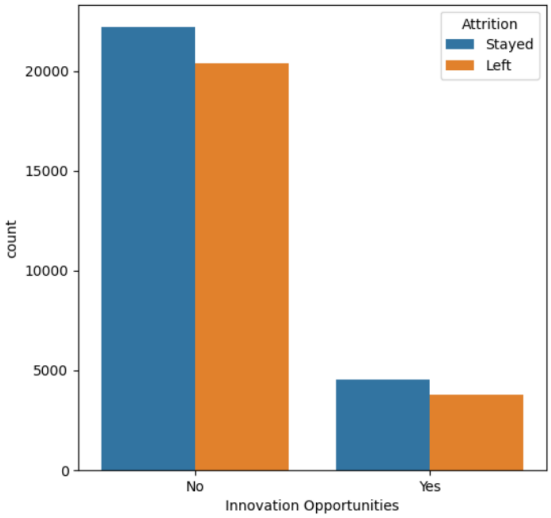

Attrition
Stayed
Left
count
Single
Married
Divorced
Marital Status

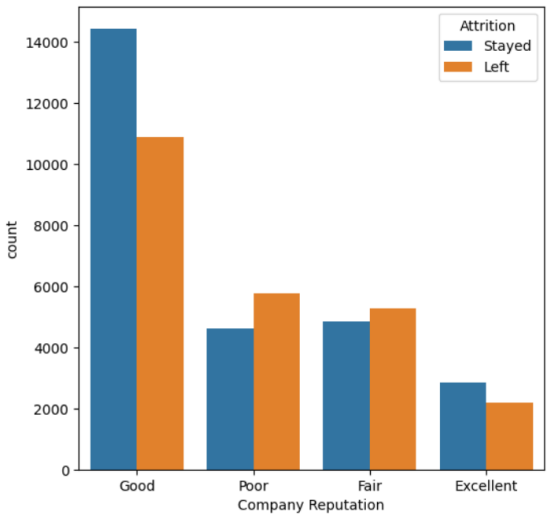

Attrition
Stayed
Left
count
Mid
Senior
Entry
Job Level

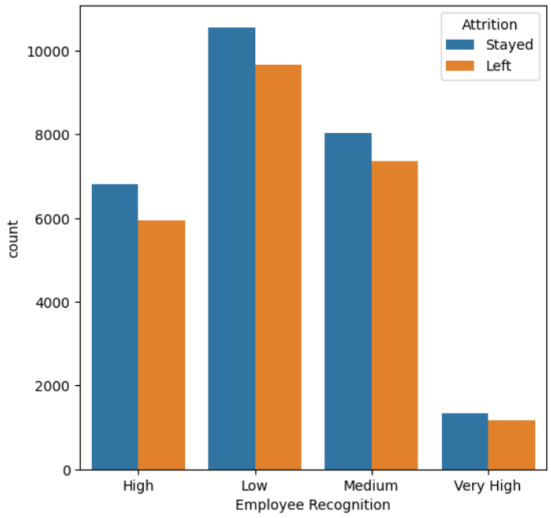

Attrition
Stayed
Left
count
14000
12000
10000
8000
6000
4000
2000
0
Small
Medium
Large
Company Size


Attrition
Stayed
Left
count
20000
15000
10000
5000
0
Yes
No
Remote Work


Attrition
Stayed
Left
count
25000
20000
15000
10000
5000
0
No
Yes
Leadership Opportunities


Attrition
Stayed
Left
count
20000
15000
10000
5000
0
No
Yes
Innovation Opportunities


Attrition
Stayed
Left
count
14000
12000
10000
8000
6000
4000
2000
0
Good
Poor
Fair
Excellent
Company Reputation


Attrition
Stayed
Left
count
10000
8000
6000
4000
2000
0
High
Low
Medium
Very High
Employee Recognition

- **Job Role vs Attrition**: Employees in **Media** and **Technology** roles show higher attrition compared to others like **Finance** or **Healthcare**.
- **Gender vs Attrition**: **Male** employees show slightly higher attrition rates than females.
- **Work-Life Balance vs Attrition**: Poor work-life balance is associated with higher attrition; those with **Excellent** balance are more likely to stay.
- **Job Satisfaction vs Attrition**: Employees with **Low** job satisfaction tend to leave more often than those with **High** or **Very High** satisfaction.
- **Performance Rating vs Attrition**: Employees with **Below Average** or **Low** ratings are more likely to leave.
- **Overtime vs Attrition**: Employees working **Overtime** have significantly higher attrition rates.
- **Education Level vs Attrition**: No major difference across education levels, though **High School** and **Associate Degree** holders show slightly higher attrition.
- **Marital Status vs Attrition**: **Single** employees have higher attrition compared to **Married** or **Divorced** ones.
- **Job Level vs Attrition**: **Entry-level** employees are more likely to leave than **Mid** or **Senior** levels.
- **Company Size vs Attrition**: Employees in **Small** companies show higher attrition rates than those in **Medium** or **Large** firms.
- **Remote Work vs Attrition**: Those **not working remotely** have slightly higher attrition.
- **Leadership Opportunities vs Attrition**: Lack of leadership opportunities correlates with higher attrition.
- **Innovation Opportunities vs Attrition**: Absence of innovation opportunities is linked with higher attrition.
- **Company Reputation vs Attrition**: Companies rated **Poor** or **Fair** in reputation see more attrition than those rated **Good** or **Excellent**.
- **Employee Recognition vs Attrition**: Employees with **Low** or **Medium** recognition are more likely to leave compared to those with **High** or **Very High** recognition.

6 Feature Engineering

Dummy Variable Creation

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

Identifying categorical columns where dummy variables are required

1. 'Gender'
2. 'Job Role'
3. 'Work-Life Balance'
4. 'Job Satisfaction'
5. 'Performance Rating'
6. 'Overtime'
7. 'Education Level'
8. 'Marital Status'
9. 'Job Level'
10. 'Company Size'
11. 'Remote Work'
12. 'Leadership Opportunities'
13. 'Innovation Opportunities'
14. 'Company Reputation'
15. 'Employee Recognition'

Using **pd.get_dummies** we have replaces all the category variables with dummy variables.

Initially we created dummy variables in Training data then we have done the same in Validation data. Also we removed/dropped categorical variables which are further useless.

Feature scaling

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

We have used the tools **MinMaxScaler** of **preprocessing** from **Scikit- learn** Library

```
# Scale the numeric features present in the training set
MMS = MinMaxScaler()
x_train[num_col] = MMS.fit_transform(x_train[num_col])
# Scale the numerical features present in the validation set
x_val[val_num_col] = MMS.transform(x_val[val_num_col])
```

7. Model Building

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.

Importanted necessary tools and libraries

We have used **LogisticRegression** from **Scikit- learn** Library

- Log_reg = LogisticRegression()
- Creted **Log_reg** object of LogisticRegression

Evaluate VIF and p-values of features:

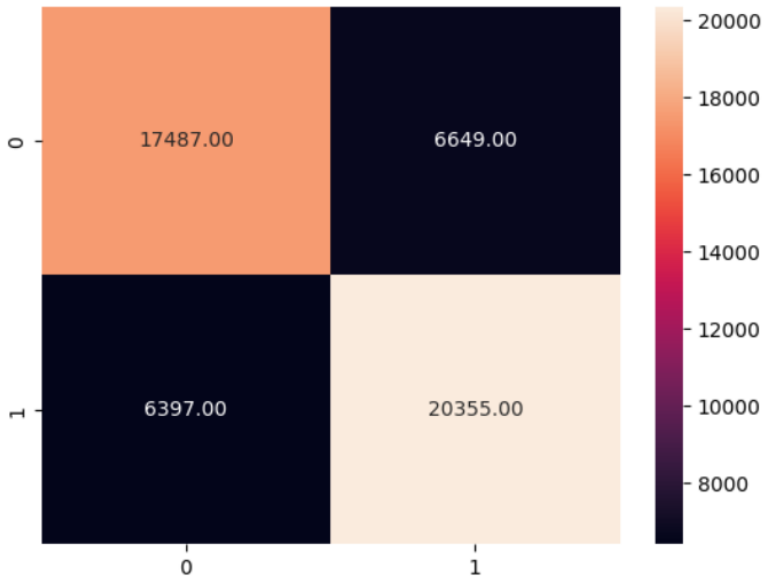
	Features	VIF	p_value
0	const	11.930415	3.296281e-01
8	Performance Rating_Low	1.000317	5.317120e-26
7	Job Satisfaction_Low	1.000317	5.414360e-28
1	Years at Company	1.000263	3.258197e-68
14	Company Reputation_Fair	1.068317	8.211632e-78
3	Number of Dependents	1.000178	5.371759e-87
2	Distance from Home	1.000295	2.892847e-138
15	Company Reputation_Poor	1.068066	1.308312e-151
4	Gender_Male	1.000433	1.571906e-161
9	Education Level_PhD	1.000300	1.782508e-176
6	Work-Life Balance_Poor	1.075553	0.000000e+00
5	Work-Life Balance_Fair	1.075288	0.000000e+00
11	Job Level_Mid	1.203851	0.000000e+00
10	Marital Status_Single	1.000295	0.000000e+00
13	Remote Work_Yes	1.000249	0.000000e+00
12	Job Level_Senior	1.203743	0.000000e+00

Evaluation of performance of Model

Checking the accuracy of the model based on the predictions made on the training set

Accurary of the model is : **0.7436330765602892**

Creating a confusion matrix based on the predictions made on the training set:



Sensitivity of Model : 0.7321637916596885

Specificity of Model : 0.7537772181898978

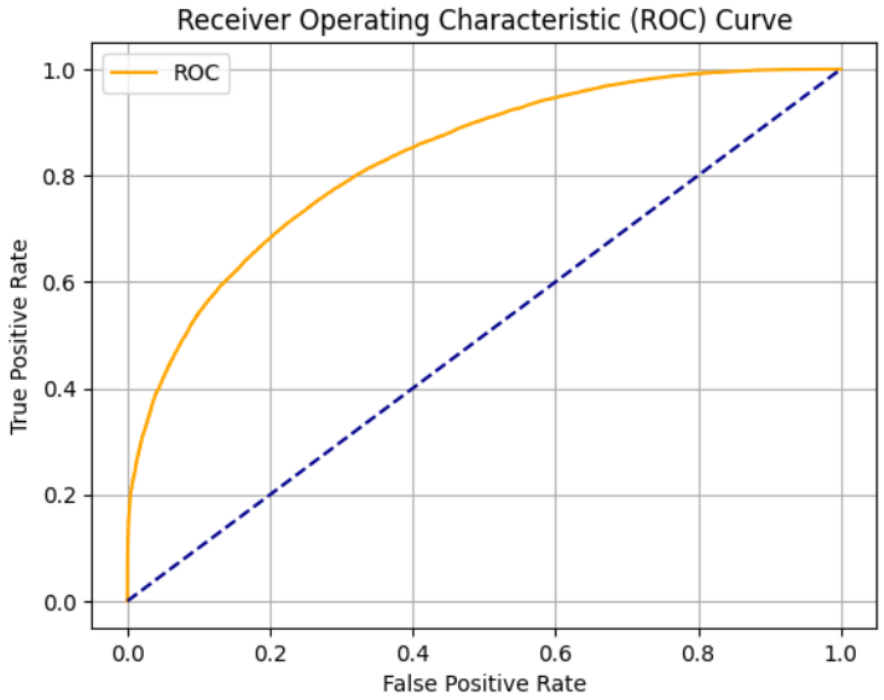
Precision of Model : 0.7537772181898978

Recall of Model : 0.7608776913875598

Finding the Optimal Cutoff:

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

Plotting the ROC curve and checking AUC.



The ROC curve plots the True Positive Rate (Sensitivity) against the False Positive Rate at various thresholds.
The orange curve lies well above the diagonal baseline, indicating that the logistic regression model performs significantly better than random guessing.

A curve that bows toward the top-left corner, as seen here, reflects a strong classifier with good discriminatory ability. The area under this curve (AUC) would be high, confirming the model's effectiveness in distinguishing between employees who stay and those who leave.

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

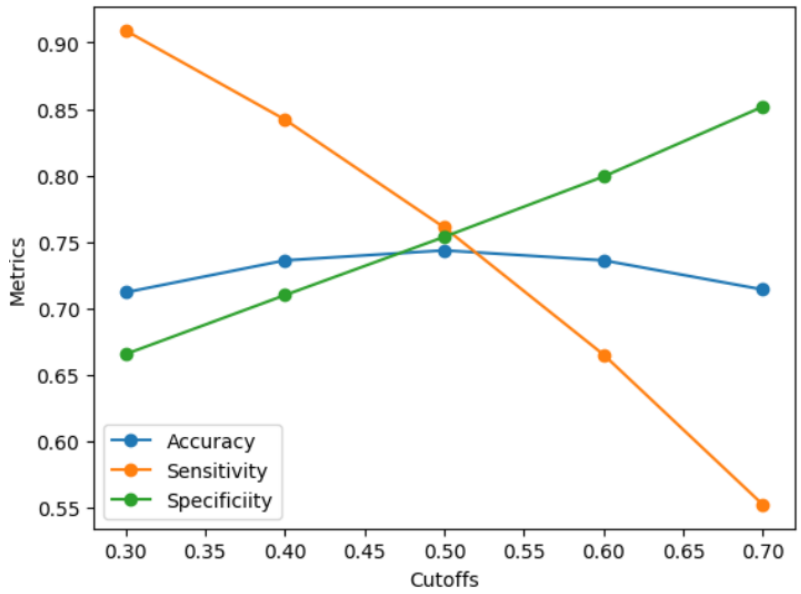
Predicting on training set at various probability cutoffs

	Actual	Predicted_prob	Predicted	cutoff0.3	cutoff0.4	cutoff0.6	cutoff0.7	cutoff0.5
40355	1	0.739138	1	1	1	1	1	1
8360	1	0.281596	0	0	0	0	0	0
41040	1	0.939555	1	1	1	1	1	1
50228	0	0.173144	0	0	0	0	0	0
60999	1	0.578070	1	1	1	0	0	1
...	...	...	...	...	...	...	...	...
38179	1	0.744340	1	1	1	1	1	1
6675	1	0.970114	1	1	1	1	1	1
56240	0	0.046831	0	0	0	0	0	0
1270	0	0.034551	0	0	0	0	0	0
16205	0	0.470791	0	1	1	0	0	0

50888 rows × 8 columns

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

	cutoffs	Accuracy	Sensitivity	Specificiity
0	0.3	0.712015	0.909054	0.665526
1	0.4	0.736107	0.842068	0.709937
2	0.5	0.743633	0.760878	0.753777
3	0.6	0.736126	0.665184	0.799201
4	0.7	0.714137	0.552295	0.851830



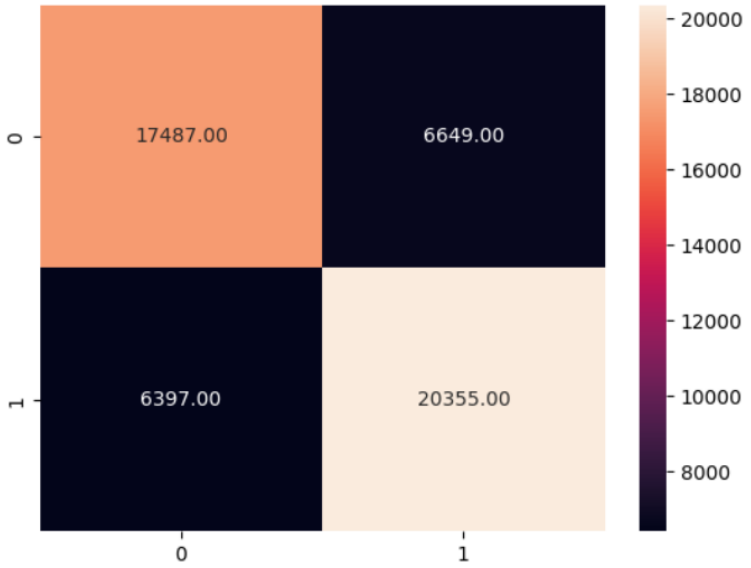
The graph illustrates how accuracy, sensitivity, and specificity vary with different classification cutoffs.

- As the **cutoff increases** from 0.3 to 0.7:
    - **Sensitivity** (ability to detect employees who leave) **decreases sharply**.

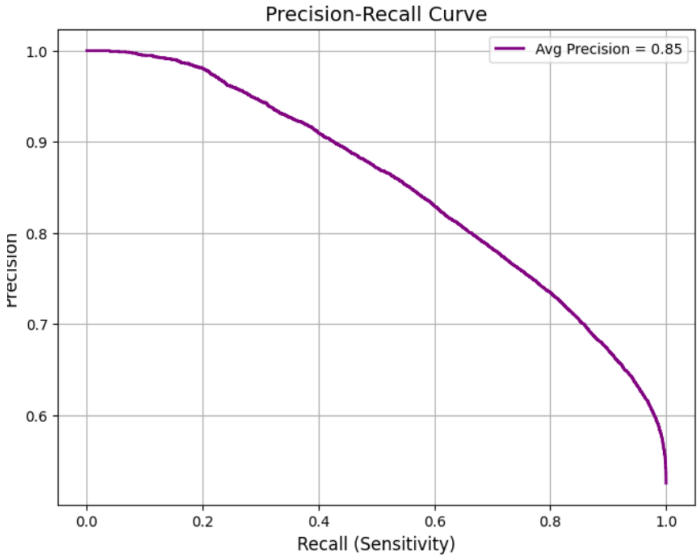
- **Specificity** (ability to detect employees who stay) **increases steadily**.
- **Accuracy** peaks around **0.5**, suggesting it as the optimal cutoff.

This trade-off highlights the balance between catching true leavers (high sensitivity) and avoiding false alarms among stayers (high specificity). A cutoff of **0.5** provides the best compromise between the three metrics, making it a suitable threshold for this model.

Confusion Matrix:



Checking the optimal cutoff value by plotting precision-recall curve



As we can see the Avg Precision is 85% which is good.

The sensitivity-specificity trade-off graph was used to visualize how these two metrics change as we adjust the classification threshold:

- **Sensitivity (Recall)**: The ability of the model to correctly identify employees who left.
- **Specificity**: The ability of the model to correctly identify employees who stayed.

As we decrease the threshold:

- **Sensitivity increases** (more employees who actually left are predicted correctly),
- But **Specificity decreases** (more employees who stayed are falsely predicted to leave).

The graph helps identify an **optimal cutoff** where both sensitivity and specificity are reasonably high. This threshold balances the need to catch potential attrition risks without flagging too many false positives.

8. Making Predictions over Validation Set

We have created Dataframe with actual values and predicted Values for Validation Set

	Actual	Predicted_prob
0	1	0.962509
1	0	0.179952
2	1	0.467769
3	1	0.906619
4	0	0.178586
...	...	...
21805	0	0.048784
21806	1	0.829293
21807	1	0.218272
21808	0	0.273091
21809	1	0.696605

21810 rows × 2 columns

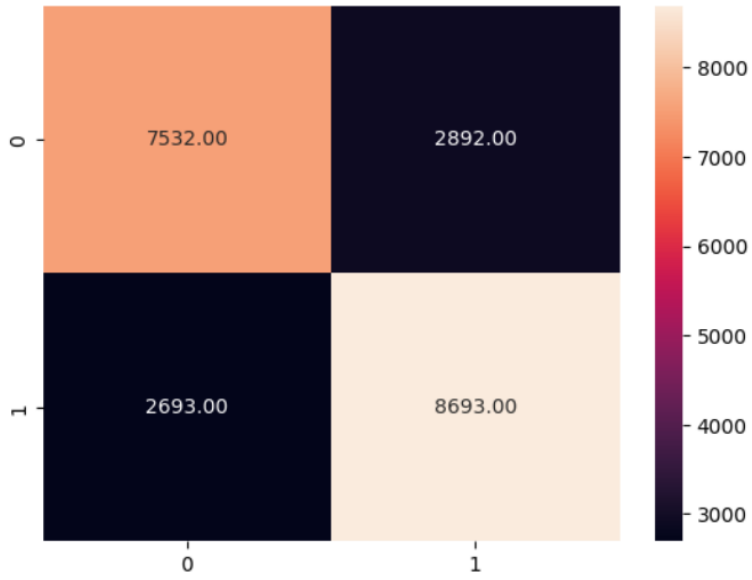
Make predictions on the validation set using the optimal cutoff and store it in a column 'final_prediction'

	Actual	Predicted_prob	final_prediction
0	1	0.962509	1
1	0	0.179952	0
2	1	0.467769	0
3	1	0.906619	1
4	0	0.178586	0
...	...	...	...
21805	0	0.048784	0
21806	1	0.829293	1
21807	1	0.218272	0
21808	0	0.273091	0
21809	1	0.696605	1

21810 rows × 3 columns

Calculating Accuracy of the model on Validation set:

Creating confusion matrix and create variables for true positive, true negative, false positive and false negative



Sensitivity of Model: 0.7366259168704157

Specificity of Model: 0.7503668536901166

Precision of Model: 0.7503668536901166

Recall Score of Model: 0.7634814684700509

Model Summary

- **Model Used**: Logistic Regression
- **Feature Selection**: Recursive Feature Elimination (RFE)
- **Scaling Method**: MinMaxScaler
- **Accuracy on Training Set**: 74.36%
- **Accuracy on Validation Set**: ~74%
- **Precision**: ~75%
- **Recall (Sensitivity)**: ~76%
- **Specificity**: ~75%
- **Average Precision (from Precision-Recall Curve)**: 85%
- **ROC AUC**: High (value not specified, but ROC was plotted)

These metrics show that the model is well-balanced, capturing both those who are likely to stay and those likely to leave, with reasonably high precision and recall.

Conclusion:

The logistic regression model developed for predicting employee retention has demonstrated reliable performance with balanced sensitivity and specificity on both training and validation datasets. After thorough data cleaning, feature engineering, and model tuning (including optimal cutoff selection), the model achieved an accuracy of approximately 74%, with a recall of ~76% and a precision of ~75% on the validation set.

These results suggest that the model is effective in identifying employees likely to leave, enabling HR teams to proactively address attrition risks. The insights derived from significant features—such as tenure, job satisfaction, and performance ratings—can guide targeted retention strategies and foster a more stable workforce.