INSTITUTE FOR ADVANCED COMPUTING AND
SOFTWARE DEVELOPMENT AKURDI, PUNE

Documentation On

# "Employee Performance Prediction, Based On The Visualisation and Analysis of Past Data of Employee Performance"

PG-DBDA SEPT 2023

*Submitted By:*

**Group No: 25**


| Roll No. | Name: |
|---|---|
| **239509** | **Anuja Chavan** |
| **239519** | **Gautamee Jakinkar** |

**Mr. Shantanu Pathak**
**Project Guide**

**Mr. Rohit Puranik**
**Centre Coordinator**

## Abstract

The strength of any organization is dependent on the Employees performance. The evaluation of employee is the most important function of the HR. Evaluation of employee is a continuous process of measuring employee performance against the company's goal. The effective evaluation system of employee performance is help to take decision like development of employee, training, promotion and behavioral aspect. For this purpose we have build classification model to predict the employee performance by adopting the technologies Like J48 decision tree, SVM and Naïve Bayes Classification technique

# ACKNOWLEDGEMENT

We would like to express our sincere gratitude to everyone who has contributed to the completion of our project.
First and foremost, we would like to thank our project guide **Mr. Shantanu Pathak** Sir for their constant guidance and support throughout the project. We extend our sincere thanks to our respected Centre Co-Ordinator, **Mr. Rohit Puranik**, for allowing us to use the facilities available.
We would also like to express our appreciation to the faculty members of our department for their constructive
feedback and encouragement. Their insights and suggestions have helped us to refine our ideas and enhance
the quality of our work.
Furthermore, we would like to thank our families and friends for their unwavering support and encouragement throughout our academic journey. Their love and support have been a constant source of motivation and inspiration for us.
Thank you all for your valuable contributions to our project,

Anuja Chavan (239509)
Gautamee Jakinkar (239519)

# Contents

# Introduction

## 1.1 PROBLEM STATEMENT

Build a model to predict performance of employees, based on the visualisation and analysis of past data of employee performance

.

## 1.2 Product Scope

The main use of this classification models is to check the quality of the question posted by the user in our web interface. The user will input the question's title and body and press the Predict Question Button after this the model will predict the quality of the question and send that quality back to the user.

## 1.3 Aims & Objectives

The main objective of our project is to create an analysis model that makes performance management at individual employee level simpler. We aim to use the Binary Classification technique for the extraction of knowledge significant for predicting and monitoring employee performance using previous appraisal records and other employee related data such as experience, age, academic qualification, professional training, previous year rating and gender. We will be using three different ML models along with different data scaling functions and determine the best Scaling Function and best model for our dataset by comparing the accuracy using Confusion Matrices.

# 2. Overall Description

## 2.1 Workflow of Project:

The diagram below shows the workflow of this project.

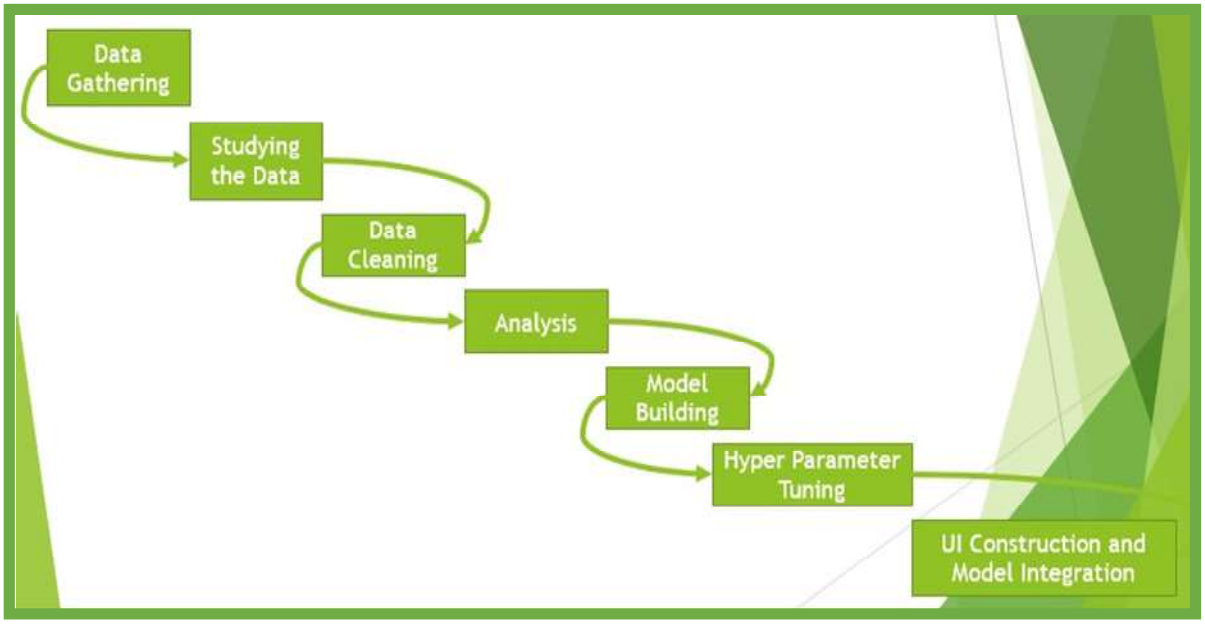



*Figure 1Workflow Diagram*

## 2.2 Data Preprocessing and Cleaning:

### 2.2.1 Data Cleaning:

The data can have many irrelevant, missing parts, HTML tags, links. To handle this part, data cleaning is done.

*1.* **Treating NULL values**

The NULL values have to be treated by either removing them or by replacing the NULL values by some other relevant values

### *2.*. Removing Duplicate Data

The Duplicate Data present in our dataset doesn't have any significance when wetrain our model. Removing duplicate data will make the size of our dataset small and this reduces training time.

## 2.3 Exploratory Data Analysis:

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

Following are some plots we used to extract some useful information

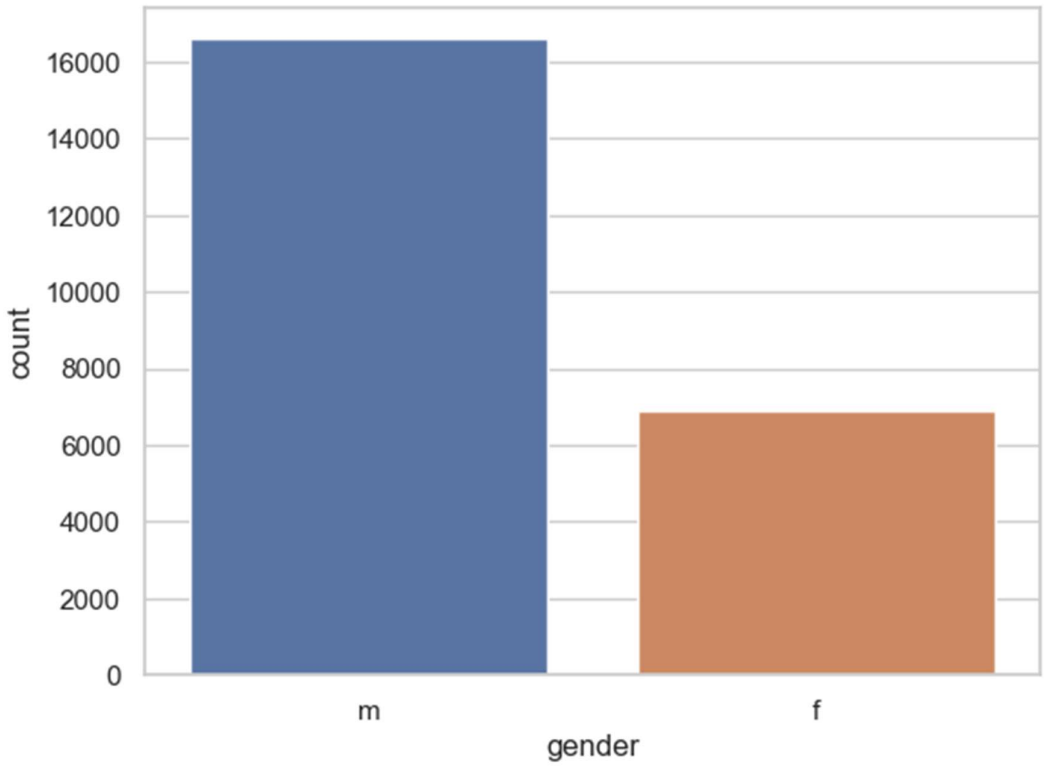



*Figure 2 A Bar Chart showing the number of males and females in the organization*

This graph shows that the total workforce comprises 23490 employees in which 16596 are Males and 6894 are Females.

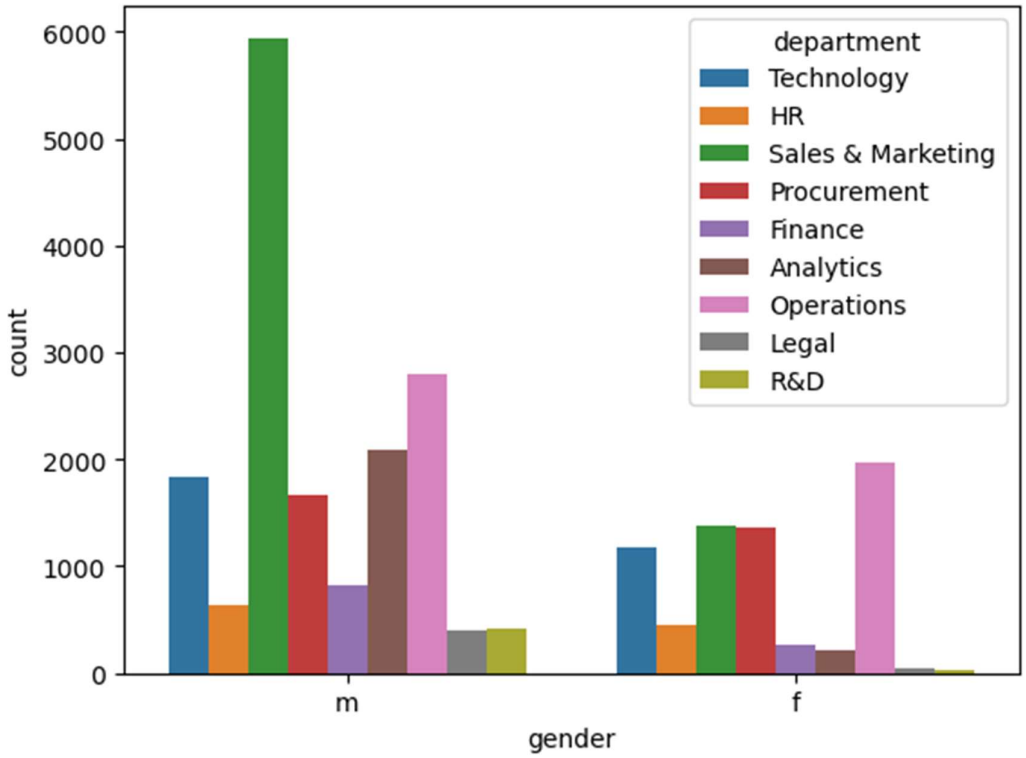
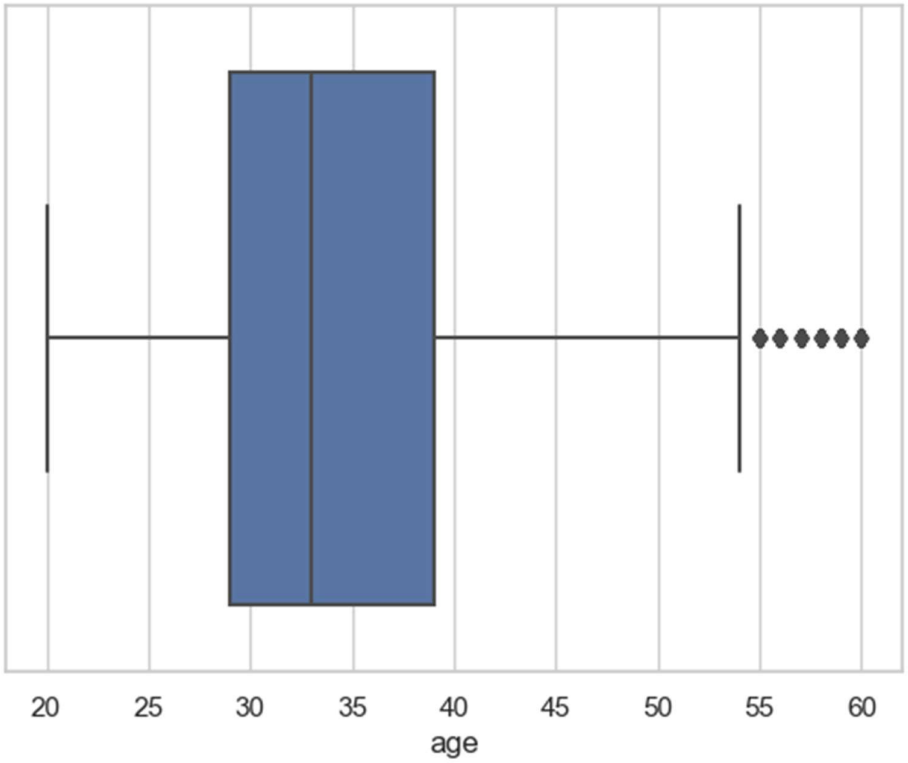



*Figure 3 A Bar Chart showing the number of males and females working in different departments*

We can see that higher number of Males work in Sales & Marketing Department and a higher number of Females work in Operations Department.
In the Procurement & HR Departments the number of Males and Females are balanced.




.0

*Figure 4 A Boxplot showing the distribution of Age Column*

This boxplot reveals that most employees fall within the interquartile range of 28 to 39 and the mean age is 34.

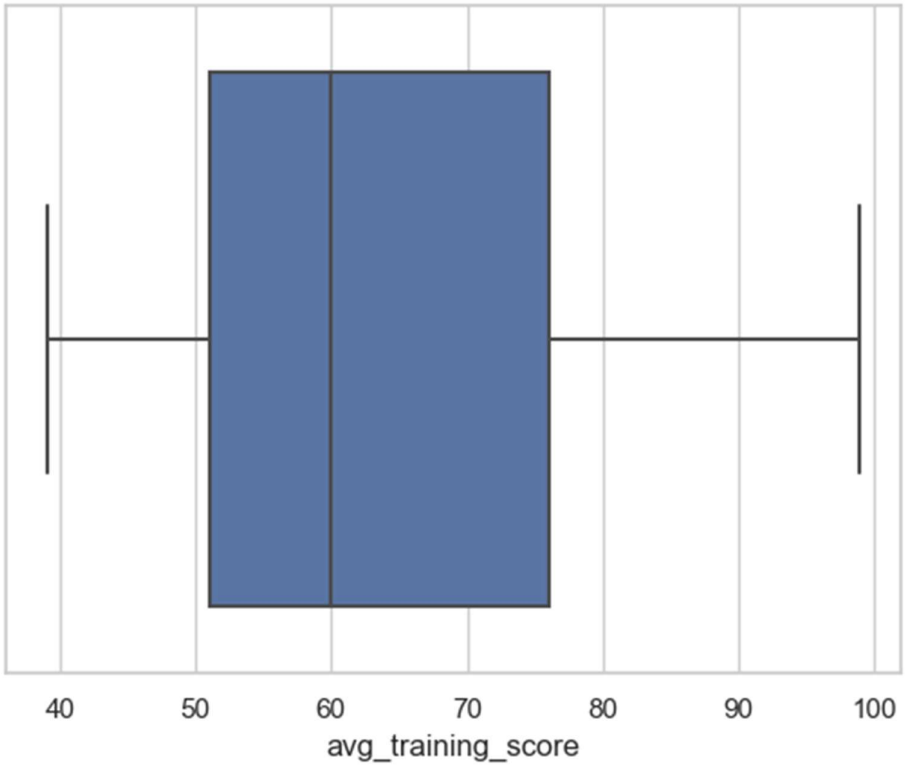



*Figure 5 A Boxplot showing the distribution of Average Training Score of all Employees*

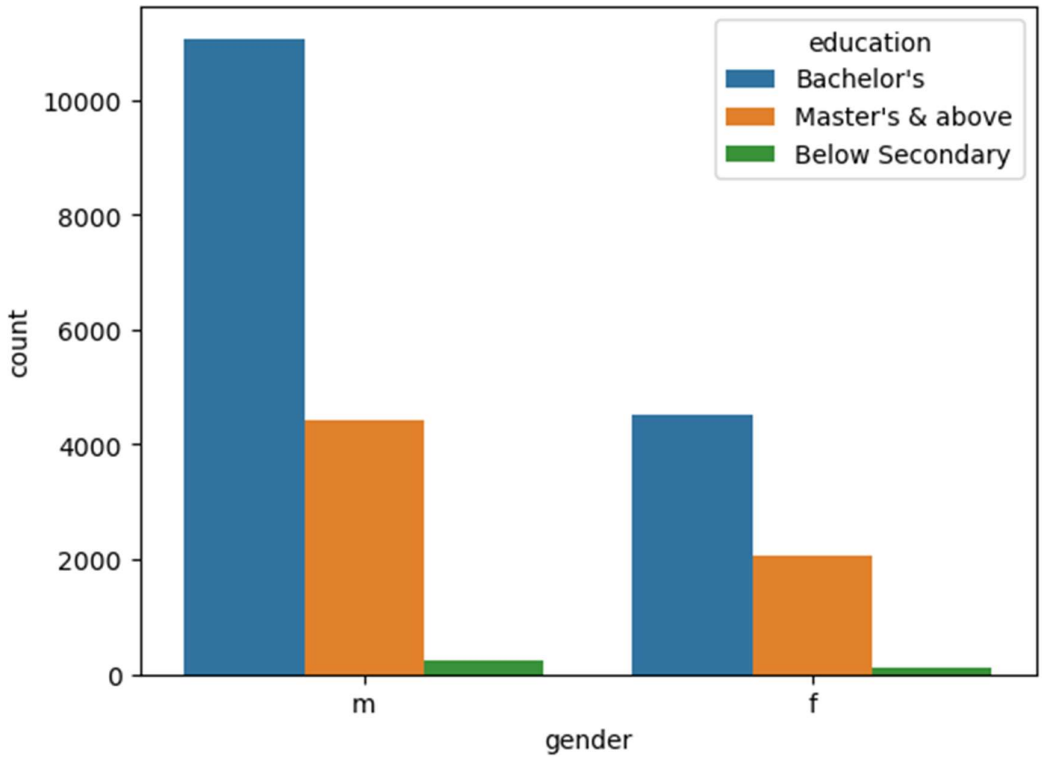



*Figure 5 A Bar chart showing the Gender wise comparison of Education of all Employees*

## 2.4 Model Building:

1. Train/Test split:

One important aspect of all machine learning models is to determine their accuracy. Now, in order to determine their accuracy, one can train the model using the given dataset and then predict the response values for the same dataset using that modeland hence, find the accuracy of the model.

A better option is to split our data into two parts: first one for training our machine learning model, and second one for testing our model.

- Split the dataset into two pieces: a training set and a testing set.
- Train the model on the training set.
- Test the model on the testing set, and evaluate how well our model did.

Advantages of train/test split:

- Model can be trained and tested on different data than the one used for training.
- Response values are known for the test dataset, hence predictions can be evaluated
- Testing accuracy is a better estimate than training accuracy of out-of-sample performance.

2. Data Scaling:

**Scaling** is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing.
In machine learning, scalers are preprocessing techniques used to standardize or normalize the features of a dataset before feeding them into a machine learning model. We have used the below-mentioned Scalers to check their effect on our data and have chosen the most appropriate Scaler based on the results.

1. RobustScaler:

- RobustScaler scales features using statistics that are robust to outliers.
- It removes the median and scales the data according to the interquartile range (IQR), so it's not influenced by outliers.
- It's particularly useful when the data contains outliers and you don't want them to skew the scaling process.

2. StandardScaler:

- StandardScaler standardizes features by removing the mean and scaling to unit variance.
- It transforms the data such that it has a mean of 0 and a standard deviation of 1.
- It's suitable when the features are normally distributed.

3. MinMaxScaler :

- MinMaxScaler scales features to a specified range, typically between 0 and 1.
- It transforms the data by shifting and rescaling each feature to a given range.
- It's useful when the features have varying scales and you want to scale them to a uniform range.

4. MaxAbsScaler :

- MaxAbsScaler scales features to the range [-1, 1] by dividing through the maximum absolute value of each feature.
- It scales each feature by dividing it by its maximum absolute value, resulting in a range of [-1, 1].
- It's useful when the data contains both positive and negative values and you want to preserve the signs of the features.

These scalers help in preprocessing the data to ensure that features are on a similar scale, which can improve the performance and convergence of many machine learning algorithms. The choice of scaler depends on the distribution and characteristics of the data and the requirements of the specific algorithm being used.

3. ML Model for Binary Classification Problem:

1. CatBoost
   - CatBoost is a powerful open-source gradient boosting library developed by Yandex. It is designed to handle categorical features natively and provides state-of-the-art performance on many tabular data problems.
   - One of the key features of CatBoost is that it can handle categorical features automatically. It can learn the best way to convert categorical features into numerical values during training, without requiring manual encoding. This is achieved using a technique called ordered boosting, which learns the optimal ordering of categorical values based on their relationship to the target variable.
   - Another powerful feature of CatBoost is the ability to handle missing values. This can improve the performance of the model and reduce the need for preprocessing.
   - In conclusion, CatBoost is a powerful and easy-to-use gradient boosting library that can handle categorical features and missing values. It provides state-of-the-art performance on many tabular data problems and is a valuable tool for any data scientist or machine learning practitioner.

2. XGBoost
    - XGBoost (Extreme Gradient Boosting) is a popular implementation of the gradient boosting algorithm, known for its speed and performance in handling large-scale datasets. It was developed by Tianqi Chen and is now maintained by the Distributed (Deep) Machine Learning Community.
    - One of the key advantages of XGBoost is its ability to handle missing values in datasets. It does this by assigning a score to missing values and then using that score to split the data. Another advantage is its ability to handle both regression and classification problems. It also includes built-in regularization techniques such as L1 and L2 regularization to prevent overfitting.
    - In conclusion, XGBoost is a powerful tool for improving the performance of machine learning models, especially when dealing with large-scale datasets. Its ability to handle missing values and built-in regularization techniques make it a popular choice among data scientists and machine learning practitioners.

3. LightGBM
    - LightGBM is a powerful gradient boosting framework that is designed to be efficient and scalable, making it an excellent choice for large datasets. It is optimized for both speed and accuracy, and its key features include:
    - Gradient-based One-Side Sampling (GOSS) for faster training on large datasets
    - Exclusive Feature Bundling (EFB) for feature transformation and dimension reduction
    - Histogram-based Gradient Boosting (HGB) for faster and more efficient gradient calculation
    - LightGBM, a powerful gradient boosting framework that is designed to be efficient and scalable. We've also demonstrated how it can be used in practice with a real-world example of predicting California housing prices. With its impressive speed and accuracy, LightGBM is a valuable tool in any data scientist's toolkit.

## Function To Evaluate Each Combination of Scaling Function and Model Using Confusion Matrices

```
def evaluate_scaling_models(X_train, X_test, Y_train, Y_test, scaler_names, models):
    results = {}

    for scaler_name in scaler_names:
        for model_name, model in models.items():
            # Scale the data
            X_train_scaled = scale_dataset(X_train, [scaler_name])[scaler_name]
            X_test_scaled = scale_dataset(X_test, [scaler_name])[scaler_name]

            # Train the model
            model.fit(X_train_scaled, Y_train)

            # Predict
            Y_pred = model.predict(X_test_scaled)

            # Calculate confusion matrix
            cm = confusion_matrix(Y_test, Y_pred)
            results[(scaler_name, model_name)] = cm

    return results
```

## Define The Models and The Scaling Techniques

```
models = {
    "XGBoost": xgb.XGBClassifier(),
    "CatBoost": CatBoostClassifier(verbose=False),
    "LightGBM": lgb.LGBMClassifier()
}

scaler_names = ['RobustScaler', 'StandardScaler', 'MinMaxScaler', 'MaxAbsScaler']
```

## Evaluate The Models

```
results = evaluate_scaling_models(X_train, X_test, Y_train, Y_test, scaler_names, models)
```

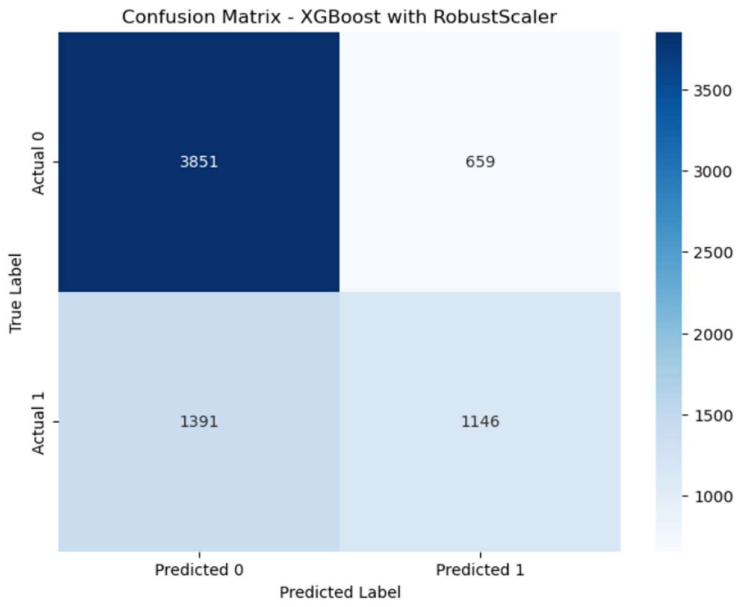
## To Determine The Best Scaling Function and Model Based on Accuracy

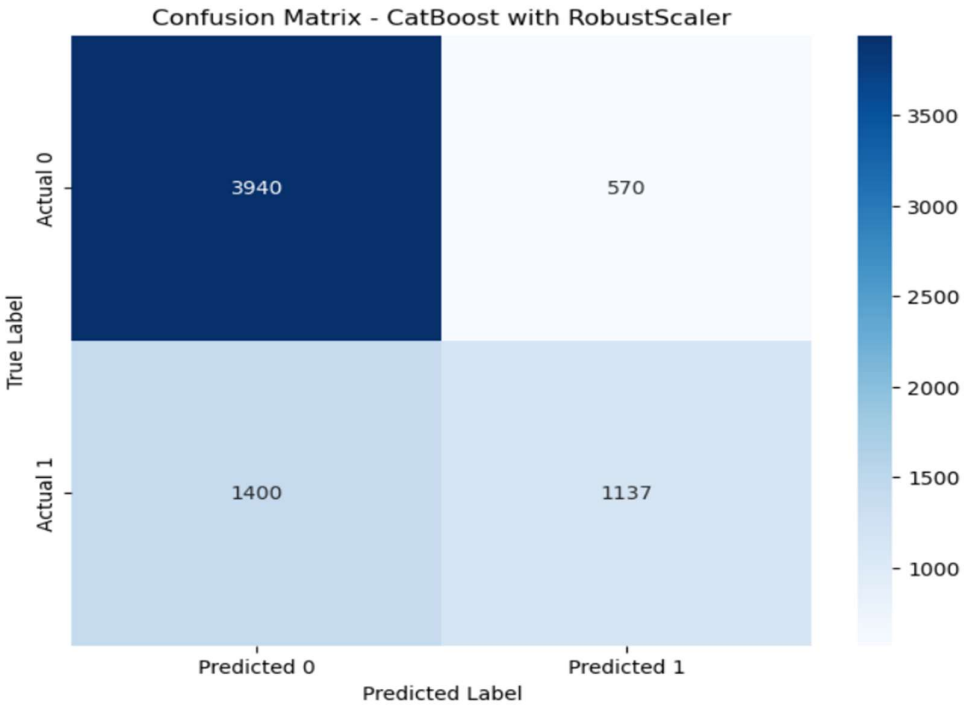
```
best_accuracy = 0
best_combination = None

for (scaler_name, model_name), cm in results.items():
    accuracy = (cm[0,0] + cm[1,1]) / cm.sum()
    print('Scaling Function: ', scaler_name)
    print('Model Name: ', model_name)
    print('Accuracy: ', accuracy)
    if accuracy > best_accuracy:
        best_accuracy = accuracy
        best_combination = (scaler_name, model_name)

print("\nBest Scaling Function and Model Combination:")
print("Scaling Function:", best_combination[0])
print("Model:", best_combination[1])
print("Accuracy:", best_accuracy)
```

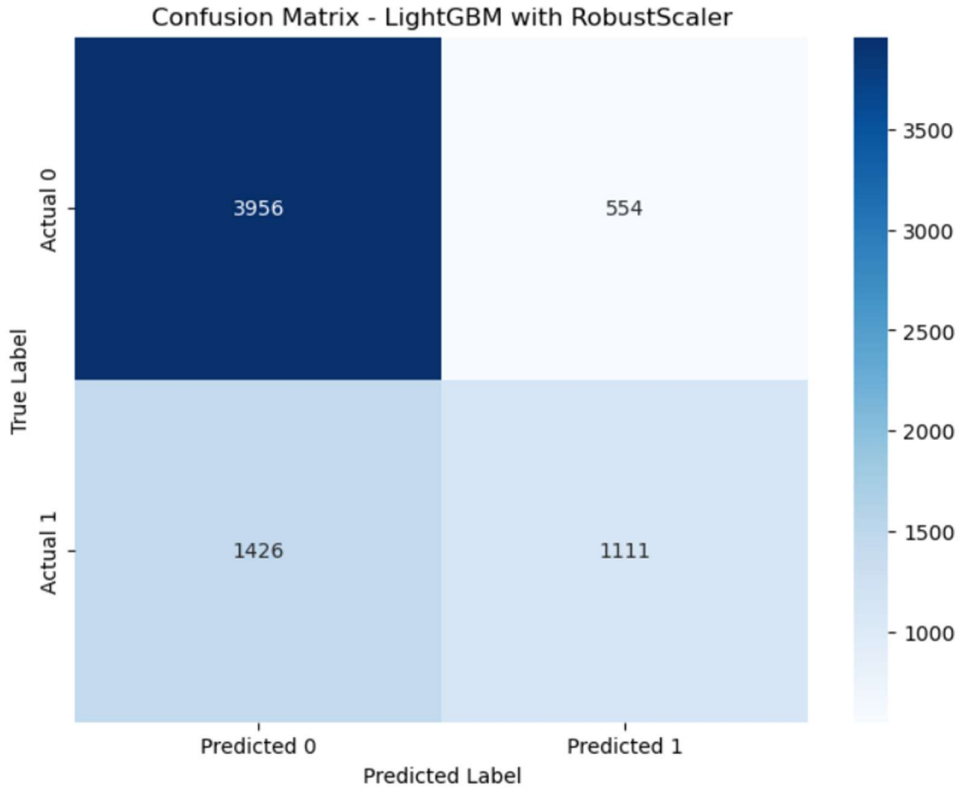
# Accuracy Determined By Using Combination of Scaling Function and Model Algorithm

## 1. Robust Scaler and XGBoost/CATBoost/LightGBM

```
Scaling Function:  RobustScaler
Model Name:  XGBoost
Accuracy:  0.709096069249326
Scaling Function:  RobustScaler
Model Name:  CatBoost
Accuracy:  0.7204484177664254
Scaling Function:  RobustScaler
Model Name:  LightGBM
Accuracy:  0.719029374201788
```

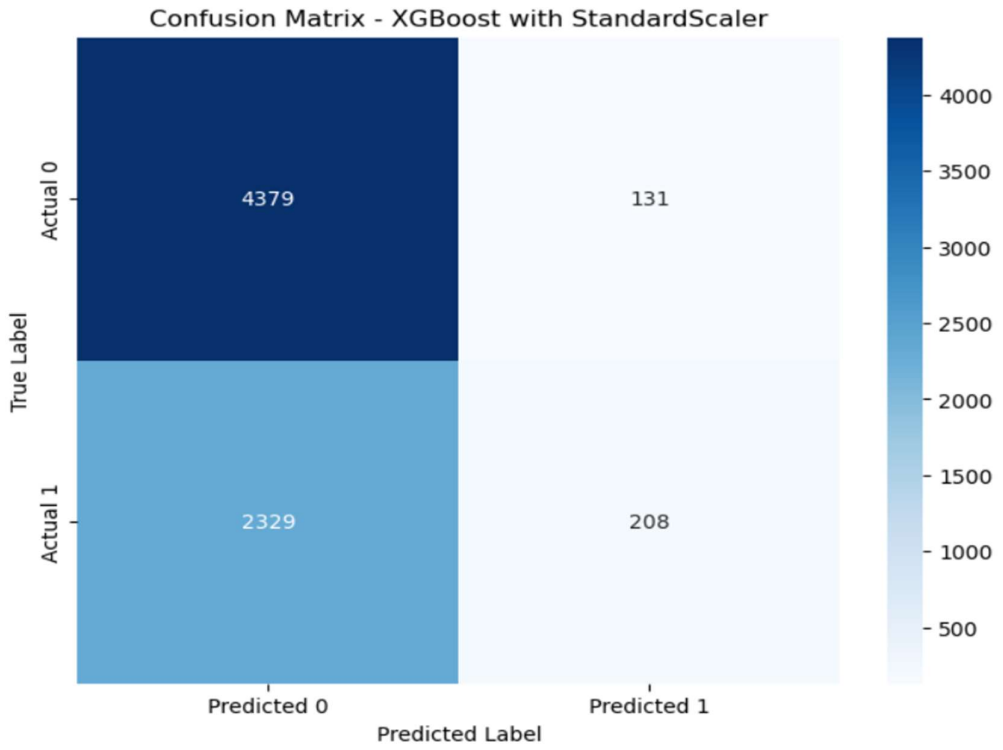


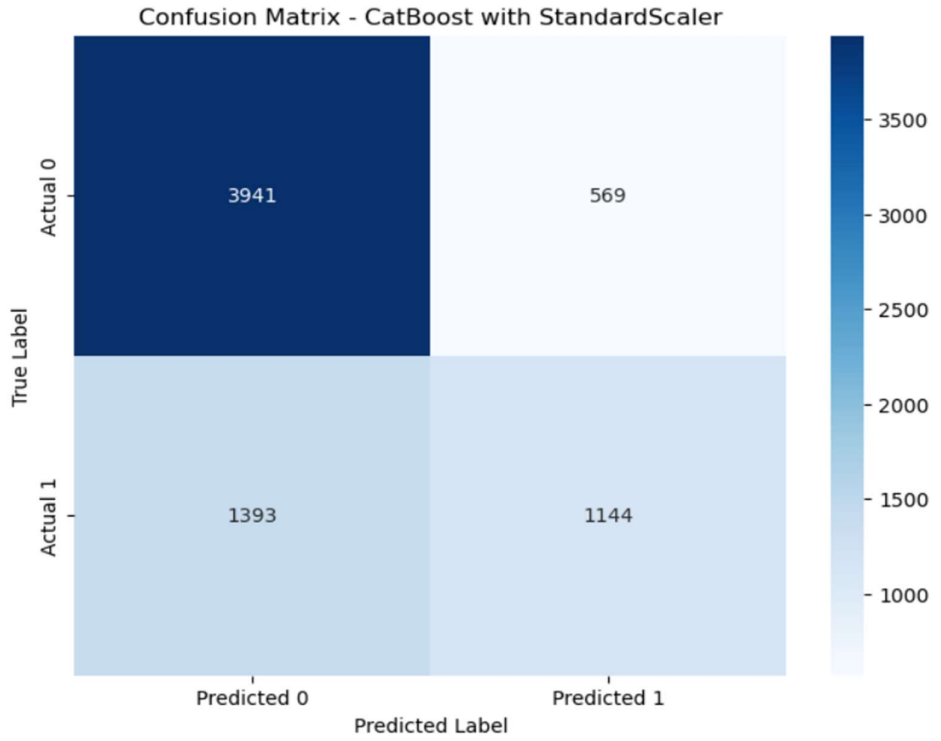

Confusion Matrix - CatBoost with RobustScaler
Actual 0
Actual 1
True Label
3940
570
1400
1137
Predicted 0
Predicted 1
Predicted Label
3500
3000
2500
2000
1500
1000

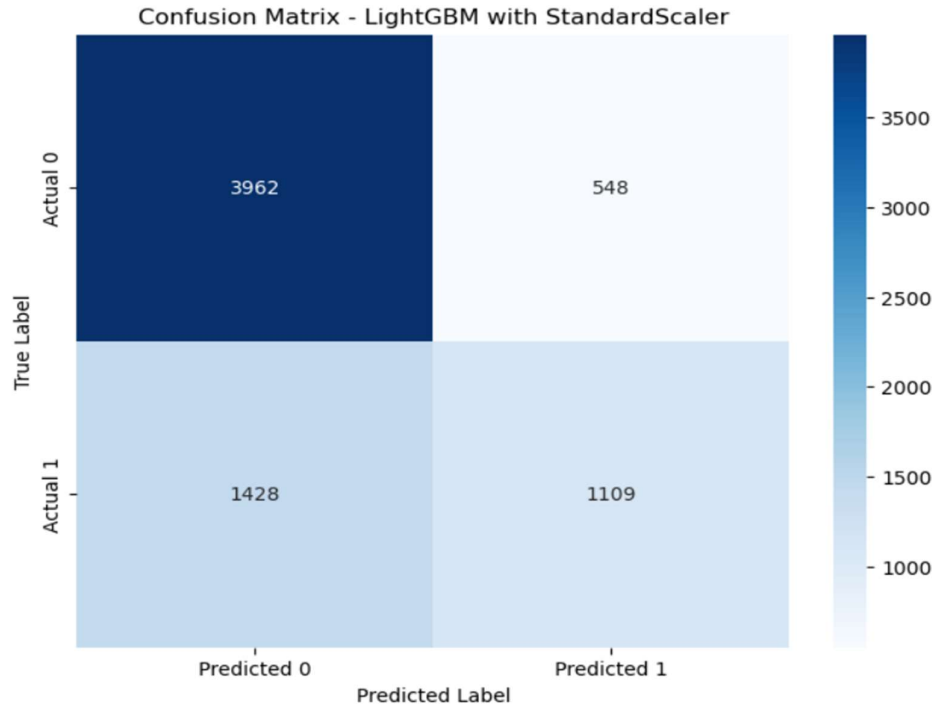

Confusion Matrix - LightGBM with RobustScaler
Actual 0
Actual 1
True Label
3956
554
1426
1111
Predicted 0
Predicted 1
Predicted Label
3500
3000
2500
2000
1500
1000

## 2. Standard Scaler and XGBoost/CATBoost/LightGBM

```
Scaling Function:  StandardScaler
Model Name:  XGBoost
Accuracy:  0.6509152830991911
Scaling Function:  StandardScaler
Model Name:  CatBoost
Accuracy:  0.7215836526181354
Scaling Function:  StandardScaler
Model Name:  LightGBM
Accuracy:  0.719596991627643
```

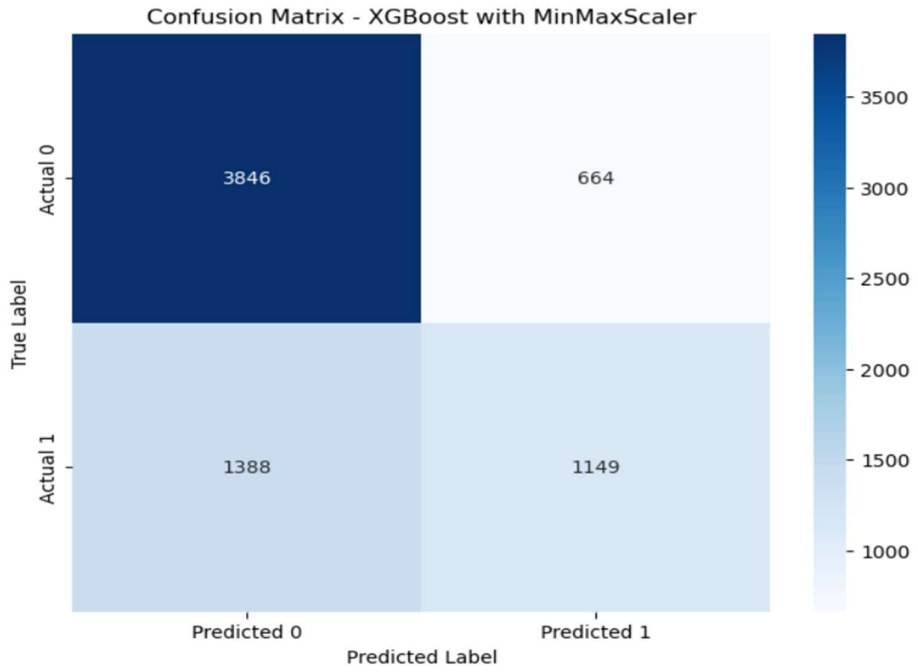


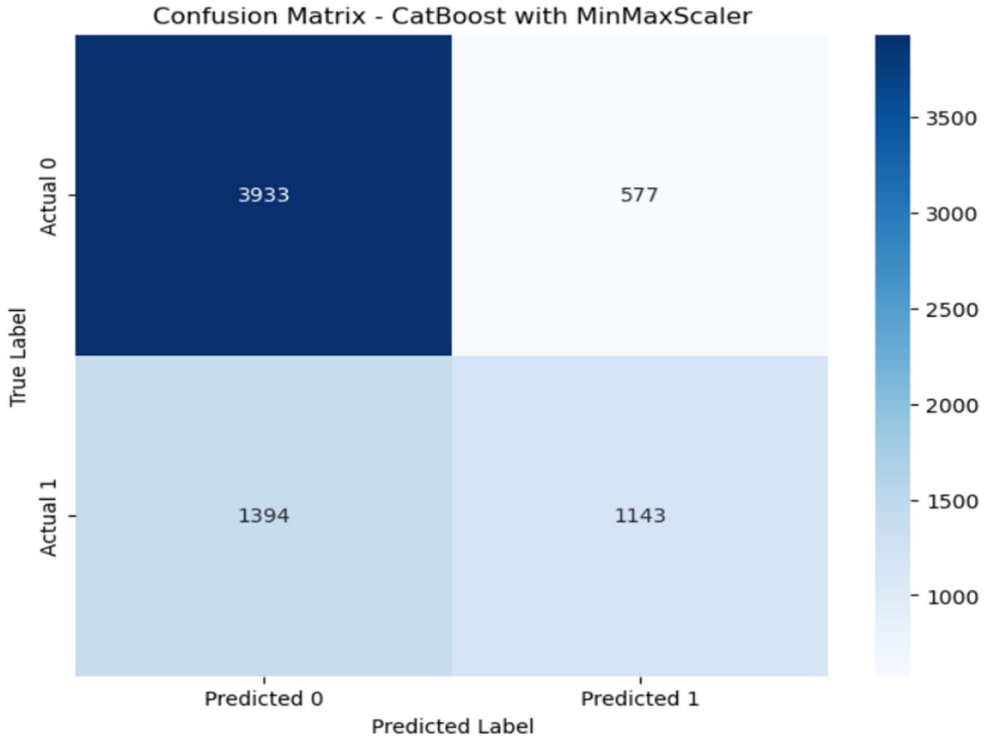

Confusion Matrix - CatBoost with StandardScaler
3941
569
1393
1144
Actual 0
Actual 1
True Label
Predicted 0
Predicted 1
Predicted Label
3500
3000
2500
2000
1500
1000

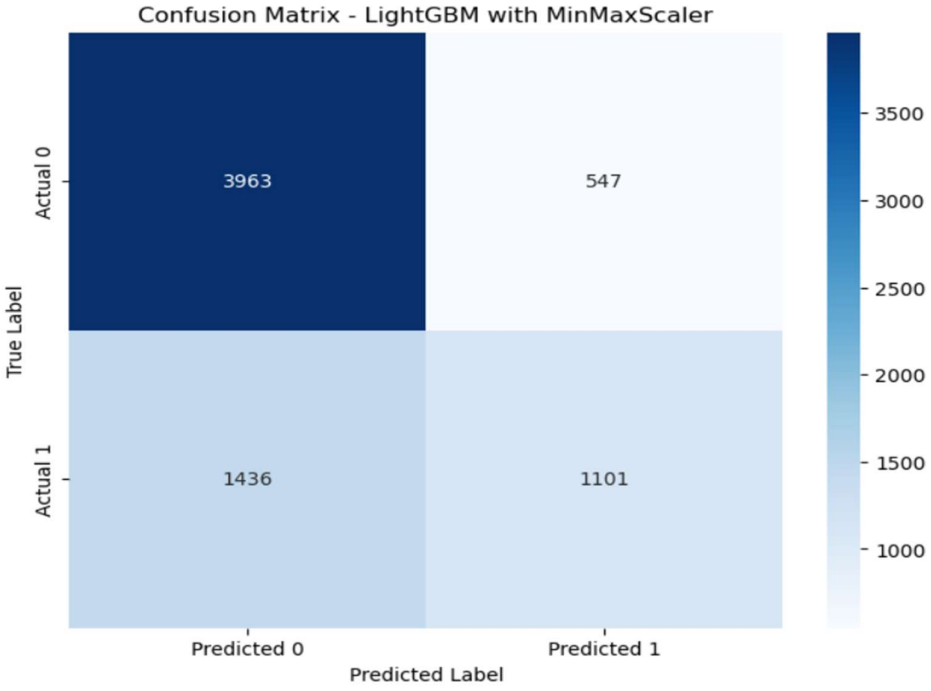

Confusion Matrix - LightGBM with StandardScaler
3962
548
1428
1109
Actual 0
Actual 1
True Label
Predicted 0
Predicted 1
Predicted Label
3500
3000
2500
2000
1500
1000

## 3. MinMax Scaler and XGBoost/CATBoost/LightGBM

```
Scaling Function:  MinMaxScaler
Model Name:  XGBoost
Accuracy:  0.7088122605363985
Scaling Function:  MinMaxScaler
Model Name:  CatBoost
Accuracy:  0.7203065134099617
Scaling Function:  MinMaxScaler
Model Name:  LightGBM
Accuracy:  0.7186036611323967
```

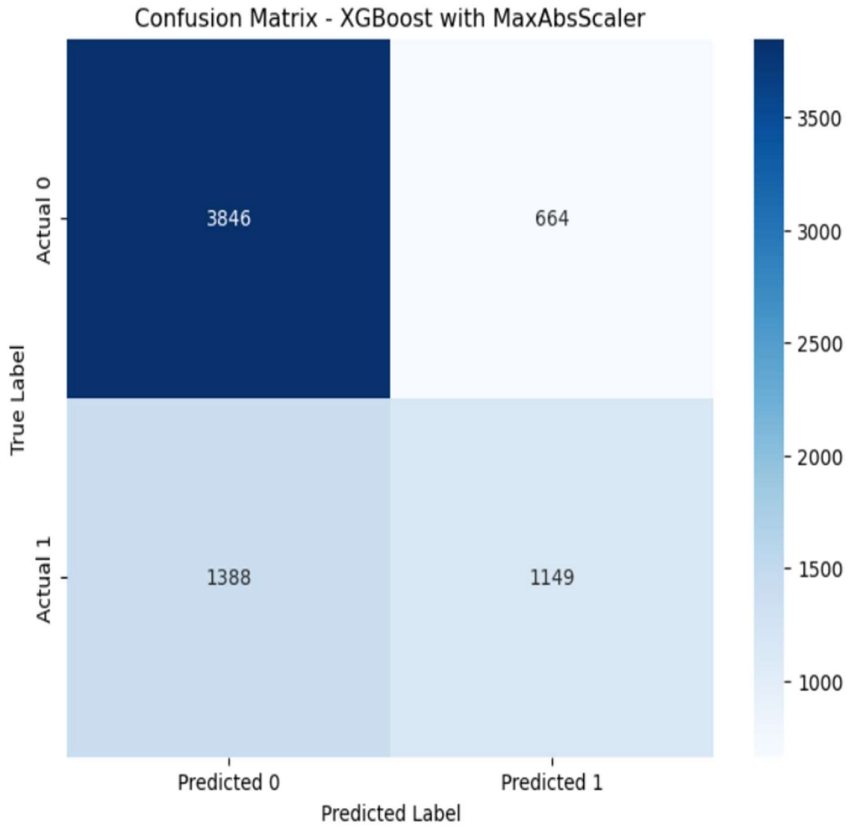


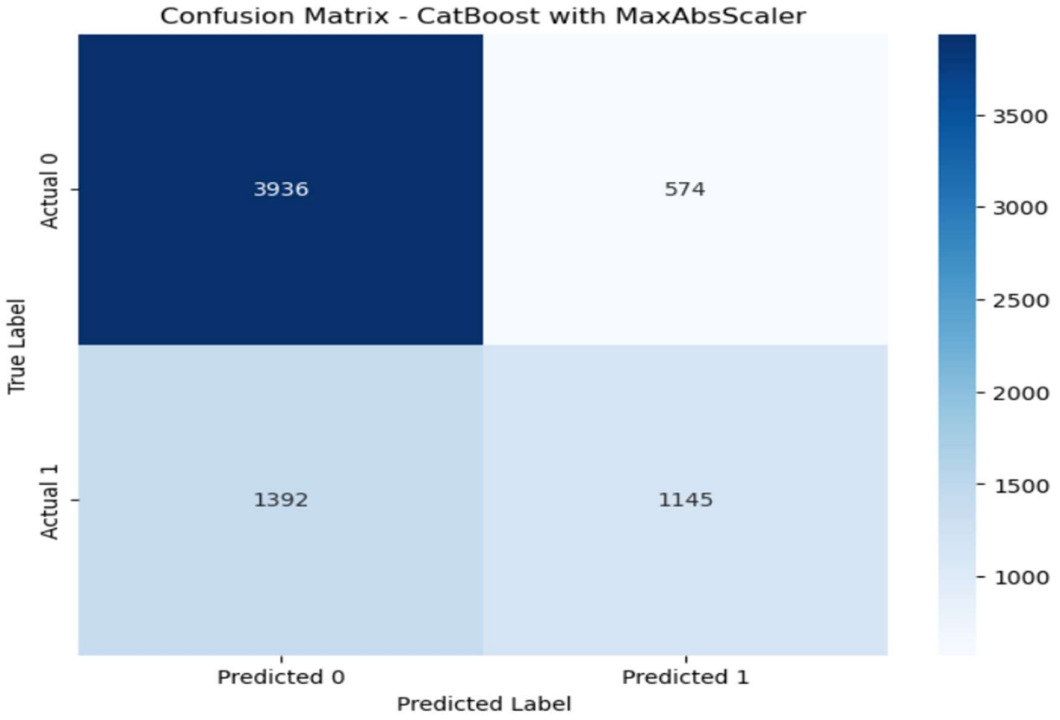

Confusion Matrix - CatBoost with MinMaxScaler
3933
577
1394
1143
Actual 0
Actual 1
True Label
Predicted 0
Predicted 1
Predicted Label
3500
3000
2500
2000
1500
1000

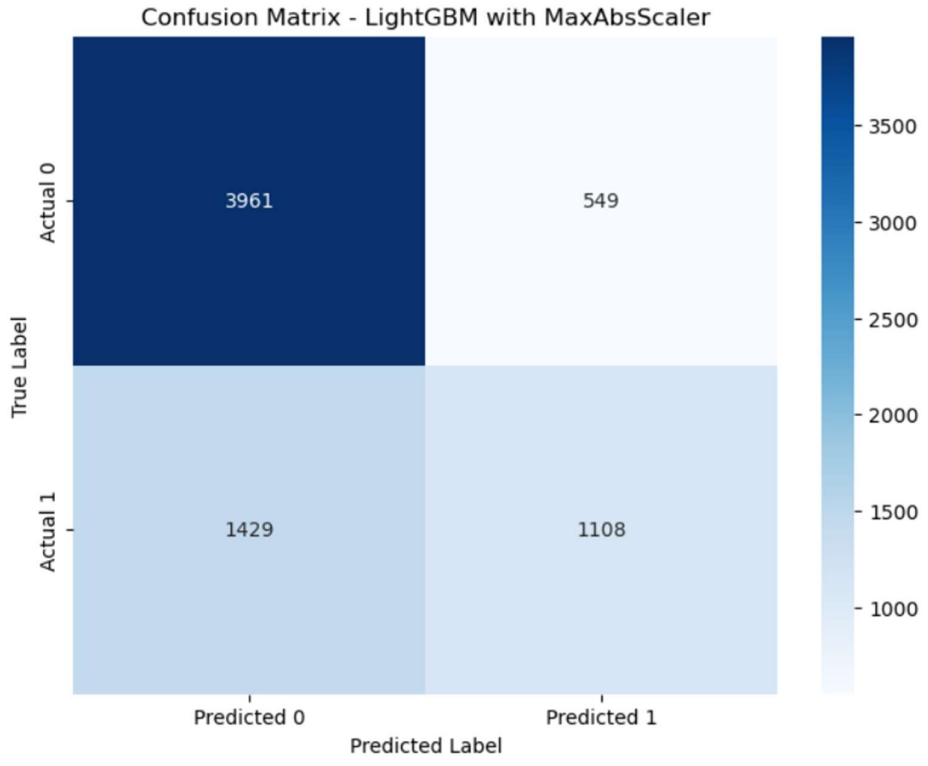

Confusion Matrix - LightGBM with MinMaxScaler
3963
547
1436
1101
Actual 0
Actual 1
True Label
Predicted 0
Predicted 1
Predicted Label
3500
3000
2500
2000
1500
1000

## 4. MaxAbs Scaler and XGBoost/CATBoost/LightGBM

```
Scaling Function:  MaxAbsScaler
Model Name:  XGBoost
Accuracy:  0.7088122605363985
Scaling Function:  MaxAbsScaler
Model Name:  CatBoost
Accuracy:  0.7210160351922804
Scaling Function:  MaxAbsScaler
Model Name:  LightGBM
Accuracy:  0.7193131829147155
```




Confusion Matrix - CatBoost with MaxAbsScaler
Actual 0
Actual 1
True Label
3936
574
1392
1145
Predicted 0
Predicted 1
Predicted Label
3500
3000
2500
2000
1500
1000


Confusion Matrix - LightGBM with MaxAbsScaler
Actual 0
Actual 1
True Label
3961
549
1429
1108
Predicted 0
Predicted 1
Predicted Label
3500
3000
2500
2000
1500
1000

# 3. Requirements Specification

## 4.1 Hardware Requirement:

- 500 GB hard drive (Minimum requirement)
- 8 GB RAM (Minimum requirement)
- PC x64-bit CPU

## 4.2 Software Requirement:

- Windows/Mac/Linux
- Python-3.9.1
- VS Code/Anaconda/Spyder
- Python Extension for VS Code
- Libraries:
  - Numpy 1.18.2
  - Pandas 1.2.1
  - Matplotlib 3.3.3
  - Scikit-learn 0.24.1
  - Flask 1.1.2
- Any Modern Web Browser like Google Chrome
  - To access the web application written in Flask

## 4. Conclusion:

- This project has proposed a quick and accurate prediction of employees' performance. This research work uses the data from a software development company having various branches in familiar cities.
- The project has showcased the technical capabilities of ML in the HR domain but also highlighted the significance of data-driven insights in making informed decisions related to workforce management.
- The findings of this report can be leveraged by organizations to enhance their recruitment, training, and performance evaluation processes, ultimately leading to improved productivity and employee satisfaction.
- Furthermore, the project has laid the foundation for future research and development in this area, with potential extensions including the integration of real-time data, additional features, and the exploration of advanced ML techniques.
- Overall, the outcomes of this project underscore the potential for ML to revolutionize traditional HR practices and contribute to the strategic management of human capital within organizations.

## 5. References


- Predicting the Quality of Questions on Stackoverflow, Antoaneta Baltadzhieva Grzegorz Chrupała (2015) https://www.aclweb.org/anthology/R15-1005.pdf
- Stackoverflow Question Quality Dataset - https://www.kaggle.com/imoore/60k-stack-overflow-questions-with-quality-rate