PROJECT REPORT
A Study of Member Attrition prediction

Submitted towards the partial fulfillment of the criteria for award of Post Graduate In Data Analytics by Imarticus

Submitted By:

SURYA KUMAR.R (IL035802)

Course and Batch: PGA 25

Abstract

Website customer churn refers to the rate at which visitors to a website discontinue their engagement with the site, such as leaving the site without taking any further action or failing to return to the site. High website churn rates can indicate problems with the user experience, such as slow loading times, confusing navigation, or irrelevant content. To reduce website churn, businesses can analyze website data, gather feedback from users, and implement changes to improve the user experience. Strategies for reducing website churn may include optimizing website performance, personalizing content, simplifying navigation, and providing clear calls-to-action. By improving website engagement and reducing churn, businesses can increase website traffic, conversions, and customer satisfaction.

Acknowledgements

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of this group project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

Further, I am fortunate to have **Surya Kumar** as our mentor. He has readily shared his immense knowledge in data analytics and guide us in a manner that the outcome resulted in enhancing our data skills.

I wish to thank, all the faculties, as this project utilized knowledge gained from every course that formed the PGA program.

I certify that the work done by me for conceptualizing and completing this project is original and authentic.

Date: Feb 28, 2023 Surya Kumar R

Place:Chennai

Certificate of Completion

I hereby certify that the project titled "**A Study of Member Attrition prediction "**
was undertaken and completed under my supervision by Surya Kumar R from the batch of PGA -25

Mentor: VijayaKumar

Date: Feb 28,2023

Place – Chennai

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Title & Objective of the study

- **A Study of Member Attrition prediction**

Churn rate is a marketing metric that describes the number of customers who leave a business over a specific time period. . Every user is assigned a prediction value that estimates their state of churn at any given time. This value is based on:

- User demographic information
- Browsing behavior
- Historical purchase data among other information

It factors in our unique and proprietary predictions of how long a user will remain a customer. This score is updated every day for all users who have a minimum of one conversion. The values assigned are between 1 and 5.

1.2 Need of the Study

Customer churn analysis helps businesses understand why customers don't return for repeat business. Churn rate tells you what portion of your customers leave over a period of time. It's often useful to look at churn by product, region or other granular factors.

Data Sources

- Kaggle

1.3 Tools & Techniques

Tools:

• Python • Matplotlib • Statsmodel

• Numpy • Seaborn • Scipy.stats

• Pandas • Sklearn • XGBoost

Techniques:

To evaluate the performance of five classification algorithms (Logistic Regression, Decision Tree, Random Forest, XGBoost ,Ada Boosting ,Naïve Byesand KNN), we use accuracy, precision, recall, and F1 score as evaluation metrics.

CHAPTER 2: DATA PREPARATION AND UNDERSTANDING

2.1 Phase I – Data Extraction and Cleaning:

- **Reading the dataset using Pandas**
- **Check for Missing Values**

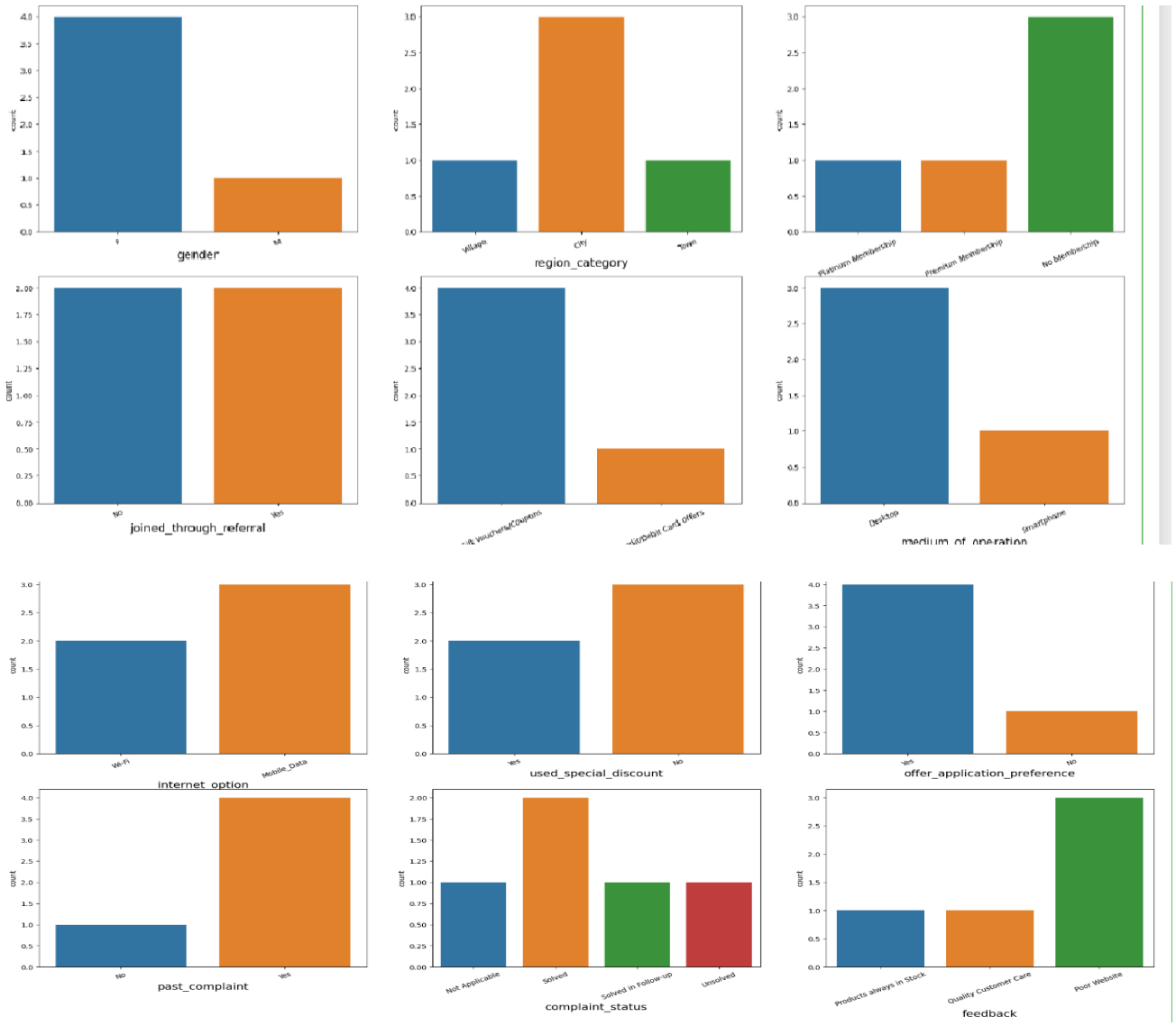
2.2 Phase II - Feature Engineering

- **Drop the unnecessary columns**

2.3 Exploratory Data Analysis

- **Finding the shape of the dataset**
  - 36992 – No of rows
  - 24 - No of columns

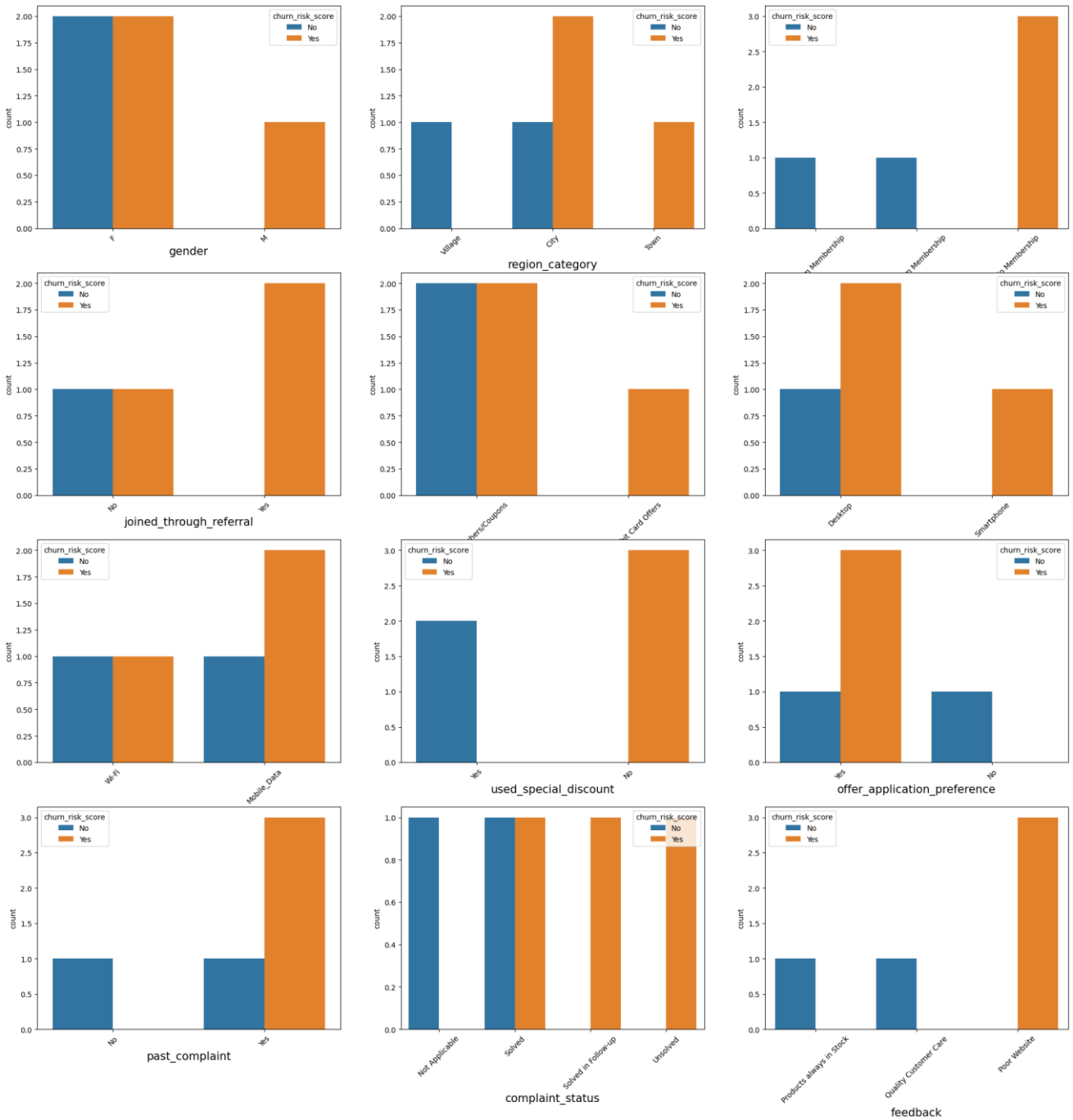
- **Univariate analysis**



<u>Observations of Univariate Analysis:</u>

- we can see both genders, Male and Female are equally distributed
- As Most of the customers are from town region while least number of customer belongs to village which is = 4600.
- No & Basic category are leading one in membership category while premium & platinum are least subscribed which is around ~4300+
- Around 15K customers have joined through referral program
- Most of the customers either use Desktop or Smartphone to access website
- Most of the customers has given negative feedback about the service such as poor website, poor customer service etc.

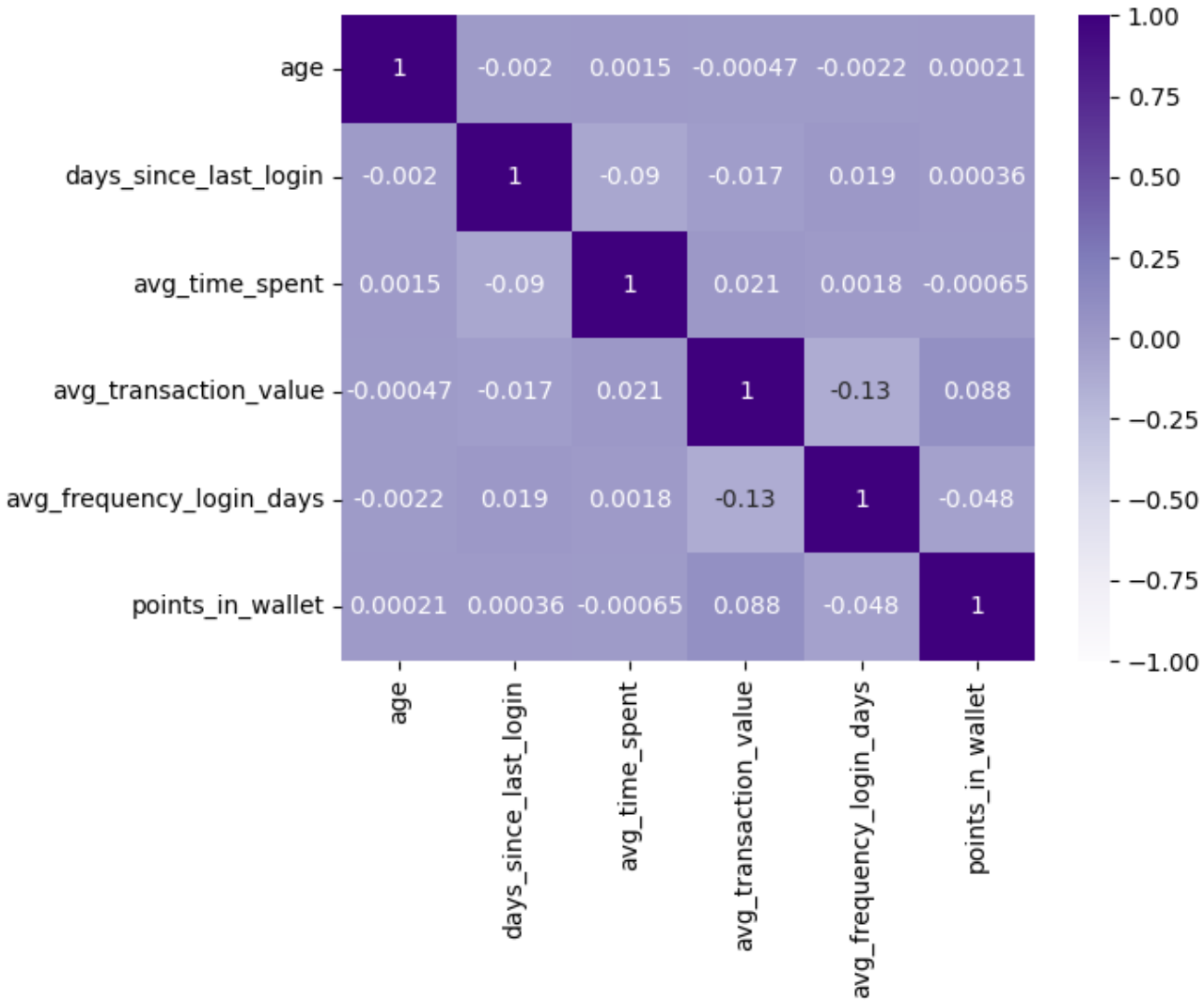
- **Bivariate analysis :-**



Observations of bivariate Analysis:

- Female are more likely to leave comparatively to men.
- People with No or Basic membership are more likely to leave the service.
- People who didn't have a good experience and gave negative feedbacks about the service are more likely to leave.

- **Multi collinearity Check**



CHAPTER 3: FITTING MODELS TO DATA

3.1 Train Test Split

The "train-test split" is a technique used to evaluate the performance of a machine learning model. It involves splitting a dataset into two separate sets: one for training the model, and another for testing its performance. The training set is used to train the machine learning model by feeding it with input data and the corresponding target output. The model learns to recognize patterns in the data and adjust its parameters to minimize the error between its predicted output and the target output. The test set, on the other hand, is used to evaluate the performance of the trained model. The model is presented with input data from the test set, and its predicted output is compared to the actual output to determine its accuracy.

1. Logistic Regression

- Logistic regression is a statistical model used to analyze the relationship between a binary dependent variable (a variable with two possible outcomes, often represented as 0 and 1) and one or more independent variables.

```
print(log_report)
```

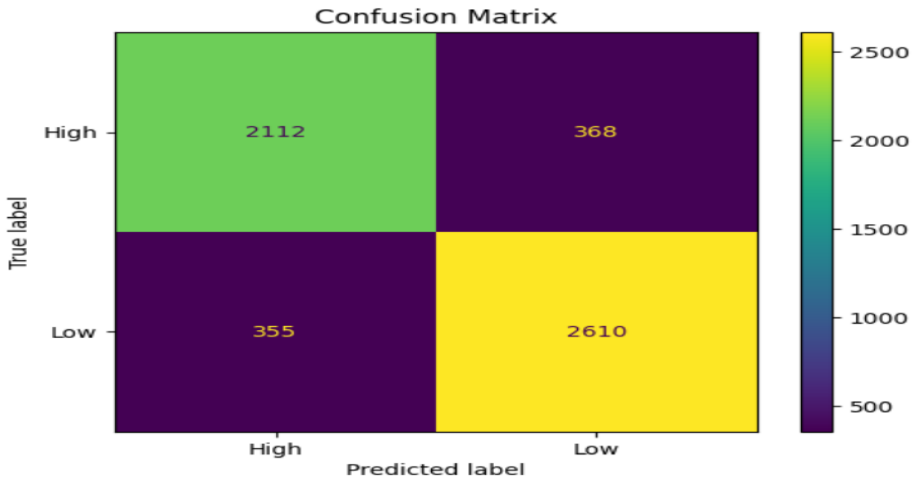
```
                precision    recall  f1-score   support

           0       0.86      0.85      0.85      2480
           1       0.88      0.88      0.88      2965

    accuracy                           0.87      5445
   macro avg       0.87      0.87      0.87      5445
weighted avg       0.87      0.87      0.87      5445
```

```
conf_matrix = confusion_matrix(log_pred,y_test1)

cf = ConfusionMatrixDisplay(conf_matrix, display_labels=["High","Low"])
cf.plot()
plt.title("Confusion Matrix")
```



2. Decision Tree Classifier

- Decision tree algorithm is a machine learning algorithm used for both regression and classification tasks. It is a predictive modeling tool that works by recursively partitioning the data into smaller subsets based on the features of dataset

```
print(tree_report)
```

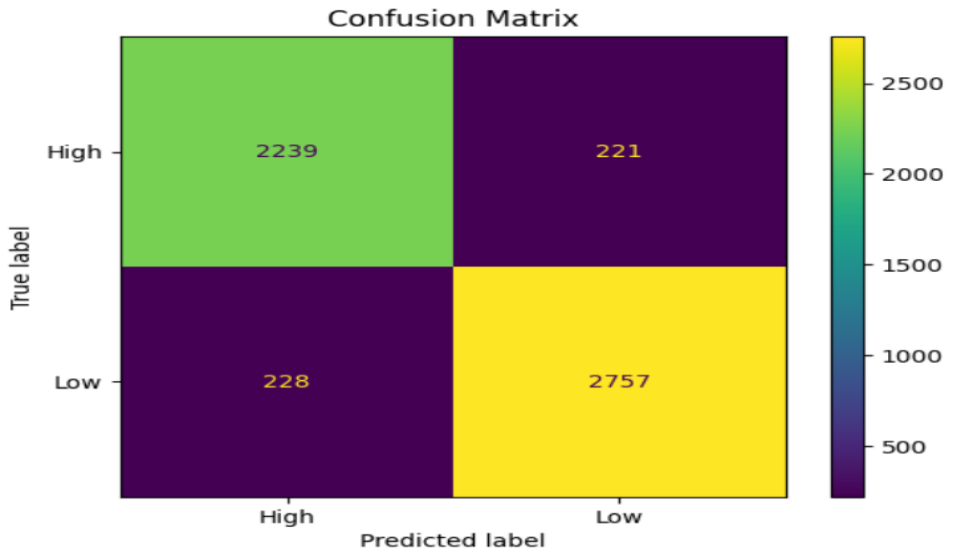
```
              precision    recall  f1-score   support

           0       0.91      0.91      0.91      2467
           1       0.92      0.93      0.92      2978

    accuracy                           0.92      5445
   macro avg       0.92      0.92      0.92      5445
weighted avg       0.92      0.92      0.92      5445
```

```
conf_matrix = confusion_matrix(tree_pred,y_test1)

cf = ConfusionMatrixDisplay(conf_matrix, display_labels=["High","Low"])
cf.plot()
plt.title("Confusion Matrix")
plt.show()                                                    # model confust
```



3. Random Forest Classifier

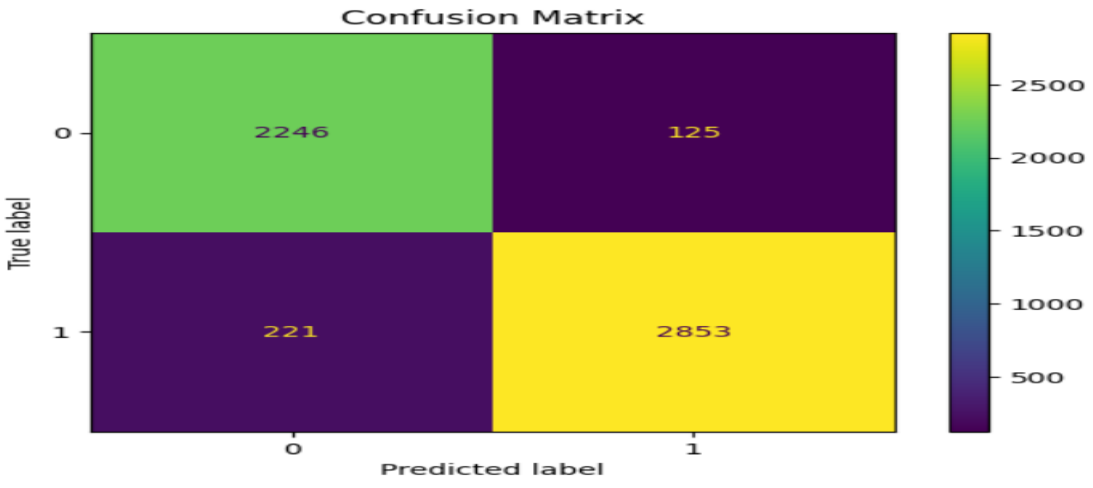
- Random Forest is an ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

```
print(rf_report)
```

	precision	recall	f1-score	support
0	0.95	0.91	0.93	2467
1	0.93	0.96	0.94	2978
accuracy			0.94	5445
macro avg	0.94	0.93	0.94	5445
weighted avg	0.94	0.94	0.94	5445

```
conf_matrix = confusion_matrix(rf_pred,y_test1)

cf = ConfusionMatrixDisplay(conf_matrix, display_labels=None)
cf.plot()
plt.title("Confusion Matrix")
plt.show()
```



4. Ada Boosting classifier:

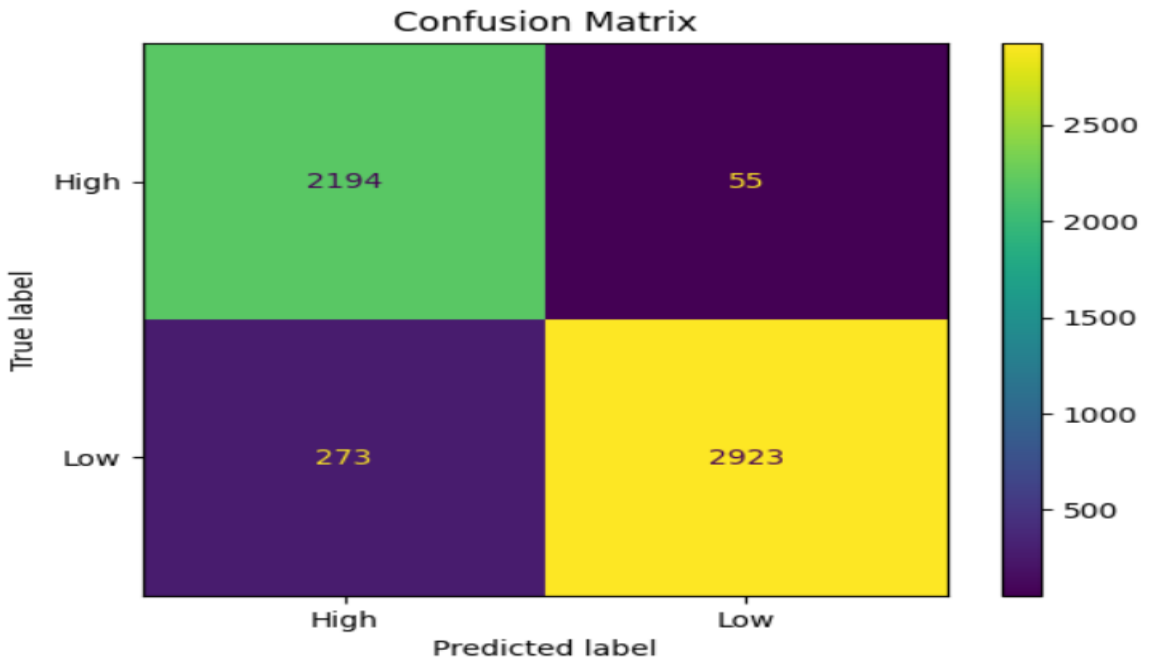
Ada Boost (Adaptive Boosting) is a machine learning algorithm that is used for classification and regression problems. It is a type of ensemble learning technique that combines several weak learners (i.e., models that perform slightly better than random guessing) to create a stronger model.

```
print(ada_report)`
```

	precision	recall	f1-score	support
0	0.98	0.89	0.93	2467
1	0.91	0.98	0.95	2978
accuracy			0.94	5445
macro avg	0.95	0.94	0.94	5445
weighted avg	0.94	0.94	0.94	5445

```
conf_matrix = confusion_matrix(ada_pred,y_test1)

cf = ConfusionMatrixDisplay(conf_matrix, display_labels=["High","Low"])
cf.plot()
plt.title("Confusion Matrix")
plt.show()
```



5. XG Boost Classifier

- **XG Boost (Extreme Gradient Boosting) is a popular and efficient machine learning algorithm used for classification and regression tasks. It is an**

ensemble algorithm that combines the predictions of multiple decision trees to produce a final prediction.

```
print(xg_report)
```

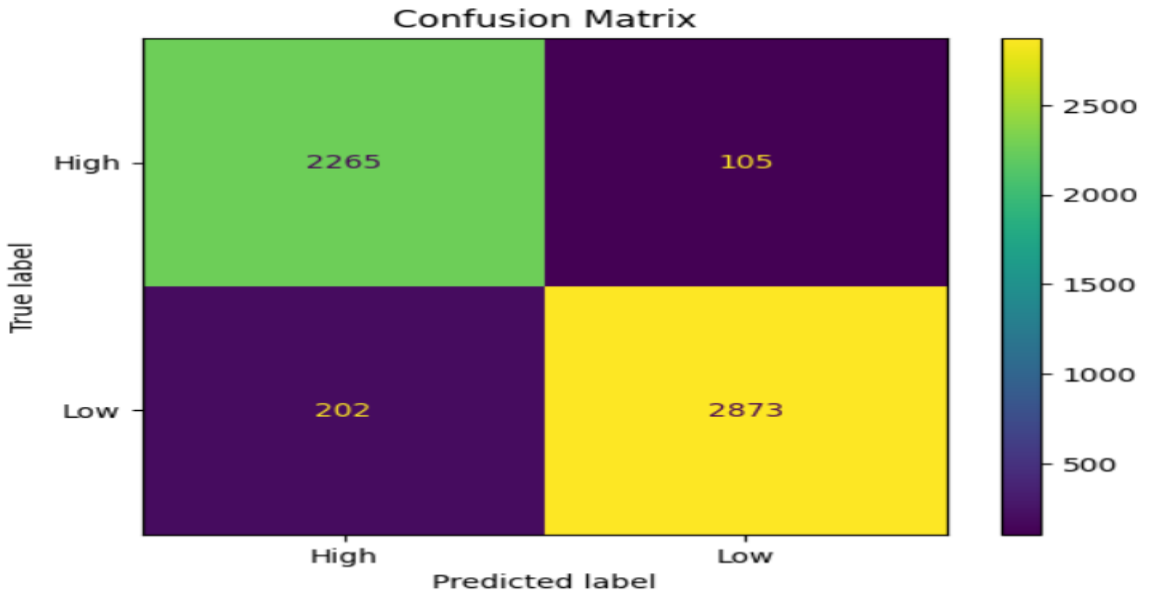
```
              precision    recall  f1-score   support

           0       0.96      0.92      0.94      2467
           1       0.93      0.96      0.95      2978

    accuracy                           0.94      5445
   macro avg       0.95      0.94      0.94      5445
weighted avg       0.94      0.94      0.94      5445
```

```
conf_matrix = confusion_matrix(xg_pred,y_test1)

cf = ConfusionMatrixDisplay(conf_matrix, display_labels=["High","Low"])
cf.plot()
plt.title("Confusion Matrix")
plt.show()
```



6. K-nearest Neighbor:

- The k-nearest neighbor (k-NN) algorithm is a type of supervised machine learning algorithm the "k" represents the number of nearest neighbors to consider when making a prediction for a new data point. To make a prediction for a new data point, the algorithm looks at the "k" closest data points in the training set and assigns the new data point the most common class (for classification) or the average value (for regression) of those "k" nearest neighbors.

```
print(knn_report)
```

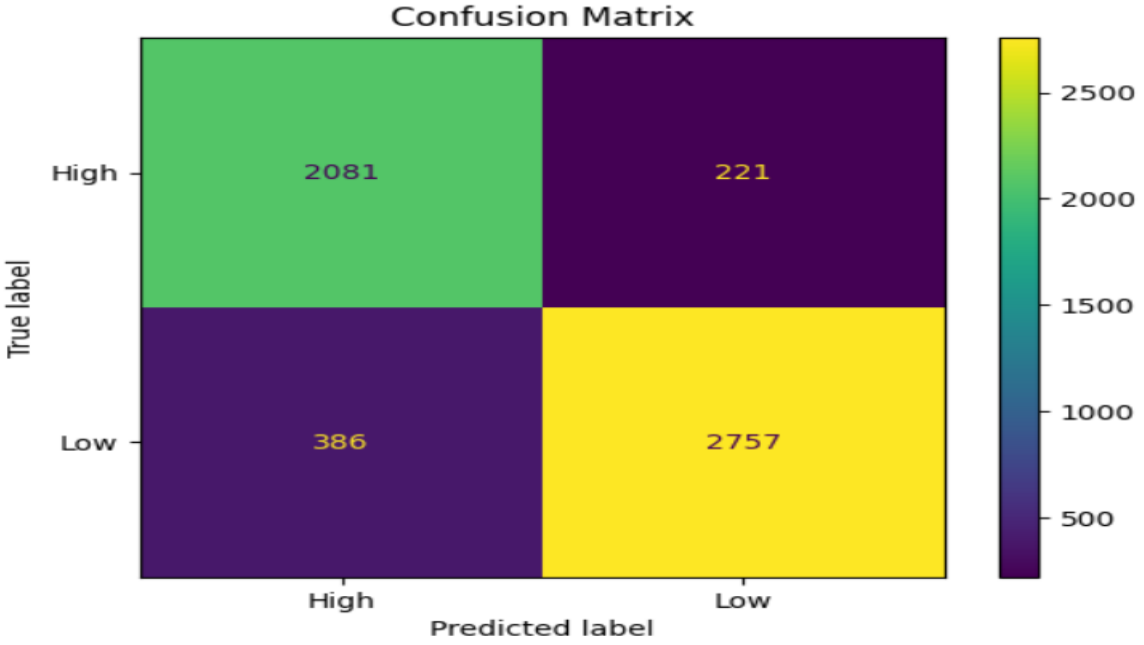
```
              precision    recall  f1-score   support

           0       0.90      0.84      0.87      2467
           1       0.88      0.93      0.90      2978

    accuracy                           0.89      5445
   macro avg       0.89      0.88      0.89      5445
weighted avg       0.89      0.89      0.89      5445
```

```
conf_matrix = confusion_matrix(knn_pred,y_test1)

cf = ConfusionMatrixDisplay(conf_matrix, display_labels=["High","Low"])
cf.plot()
plt.title("Confusion Matrix")
plt.show()
```



7. Naïve Bayes:

Naive Bayes is a classification algorithm that is based on Bayes' theorem, which describes the probability of an event occurring based on prior knowledge of conditions that might be related to the event. In the context of machine learning, Naive Bayes is used to classify data into one of several categories based on the p

```
print(naive_report)
```

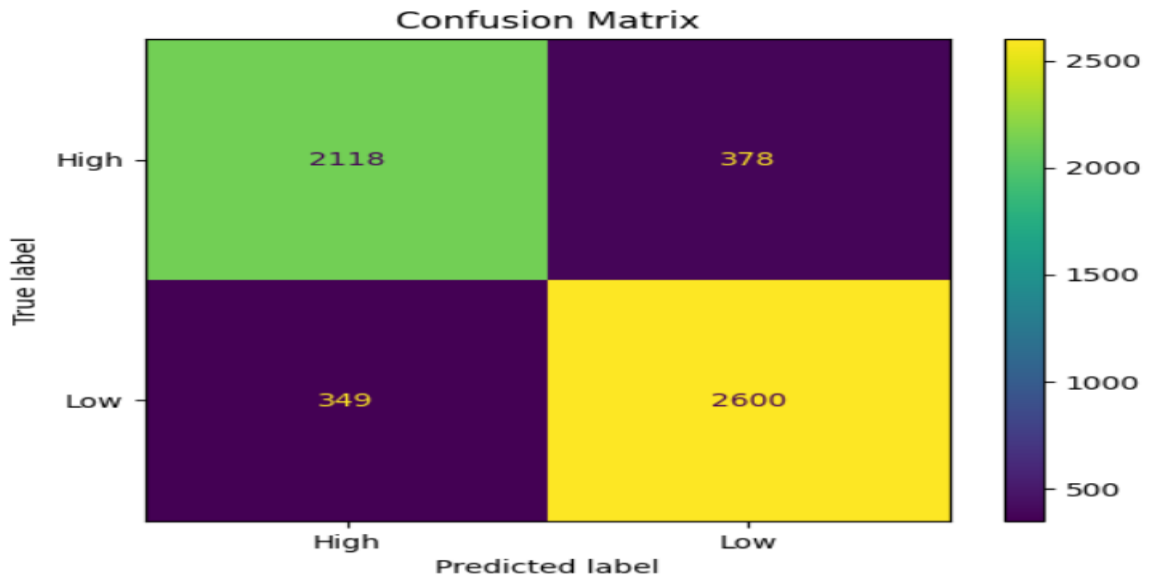
```
              precision    recall  f1-score   support

           0       0.85      0.86      0.85      2467
           1       0.88      0.87      0.88      2978

    accuracy                           0.87      5445
   macro avg       0.87      0.87      0.87      5445
weighted avg       0.87      0.87      0.87      5445
```

```
conf_matrix = confusion_matrix(naive_pred,y_test1)

cf = ConfusionMatrixDisplay(conf_matrix, display_labels=["High","Low"])
cf.plot()
plt.title("Confusion Matrix")
plt.show()
```



CHAPTER 4: KEY FINDINGS:

	Model Name	Train_Accuracy	Test_Accuracy
0	Logistic_Regression	0.85	0.87
1	Decision_Tree	1.	0.92
2	Random_Forest	1.	0.94
3	XGBoost	0.98	0.94
4	naive_bayes	0.85	0.87
5	ada_Boosting	0.93	0.94
6	knn	0.92	0.89

CHAPTER 5: RECOMMENDATIONS AND CONCLUSION:

After finding the Accuracy of all the models it is recommended that our Data is best with **Ada Boosting**. We are concluding that the Random forest classifier with the **Accuracy of 94% and F1 Score 93%** is the best efficient model for the " **A Study of Member Attrition prediction**