National College of Ireland

Configuration Manual

MSc Data Analytics
Research Project

Rahul Prakash
X23101237

School of Computing
National College of Ireland

Supervisor: Eamon Nolan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Rahul Prakash
Student ID:	X23101237
Programme:	MSc Data Analytics **Year:** 2024/2025
Module:	Research Project
Supervisor:	Eamon Nolan
Submission Due Date:	12/12/2024
Project Title:	**CUSTOMER CHURN PREDICTION IN RETAIL BANKING USING PREDICTIVE ANALYTICS**
Word Count:	**761** **Page Count:** 15

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Rahul Prakash

Date: 11/12/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	

Date:	
Penalty Applied (if applicable):	

Configuration Manual

Rahul Prakash
X23101237

Introduction:

This research investigates the effectiveness of several machine learning models, Logistic Regression, Decision Trees, Random Forest, and also XGBoost, in forecasting customer churn within the banking field.

1 System Requirements:

To guarantee efficient model processing and to minimize the duration required, it's crucial to be equipped with the necessary hardware and software resources.

1.1. Hardware Requirements:

The implementation is performed on an HP Pavilion; the configuration of the device is as follows.

1. Processor: 11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz 2.50 GHz
2. RAM: 8.00 GB (7.75 GB usable)
3. Hard Disk: 476.94 GB SSD
4. OS: Windows 11

1.2. Software Requirements:

Before beginning the model construction phase, the below mentioned software, libraries, and tools were set up and installed on the system.

Software/Tools	Version	Information
Python		To develop machine learning models, to use it as libraries and functions to develop predictive models and analyse patterns from a dataset

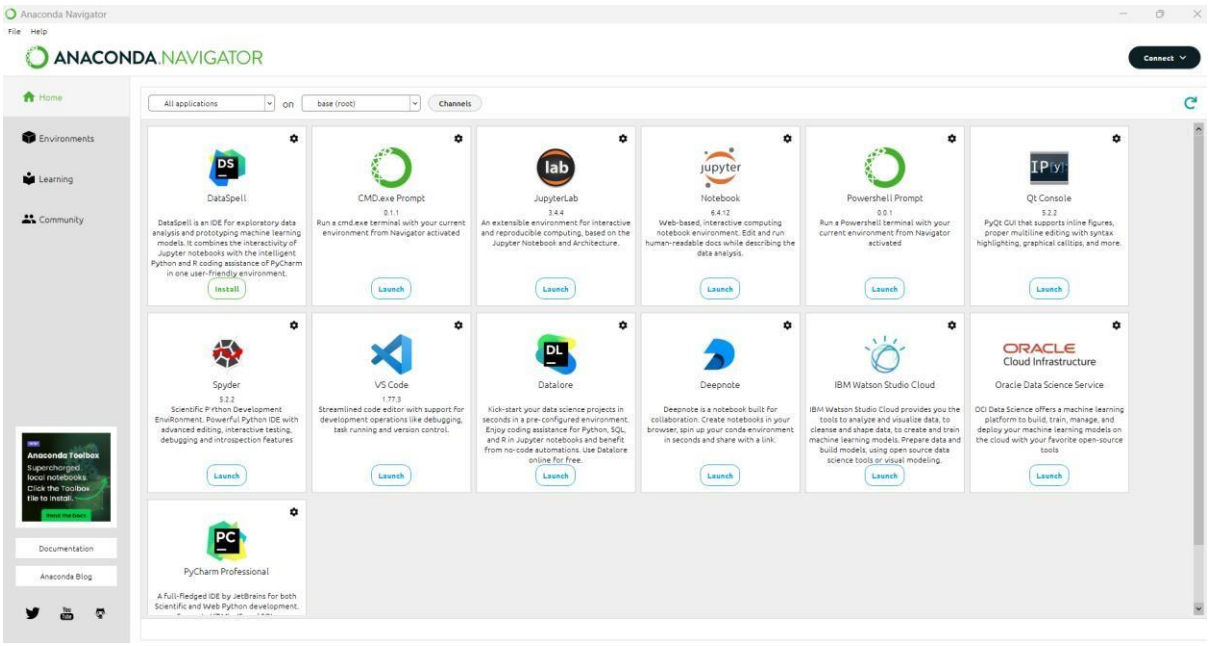
Anaconda		A popular open-source distribution for Python and R, designed for data science, machine learning, and scientific computing. It simplifies package management and environment creation using tools like Conda. It comes pre-installed with numerous libraries like Pandas, NumPy, and Sci-kit Learn.
Pandas		A Python library for data manipulation and analysis.It provides data structures like DataFrames and Series, enabling tasks such as data cleaning, transformation, and exploration. It's highly efficient for handling structured data
Matplotlib		Matplotlib is a widely used Python library for data visualization. It provides tools to create static, animated, and interactive plots, making it ideal for visualizing data in 2D and even basic 3D.
Sci-kit Learn		A robust Python library for machine learning. It offers tools for supervised and unsupervised learning, model evaluation, and preprocessing. It supports algorithms like linear regression, k-means clustering, and Random Forest.

2. Implementation:

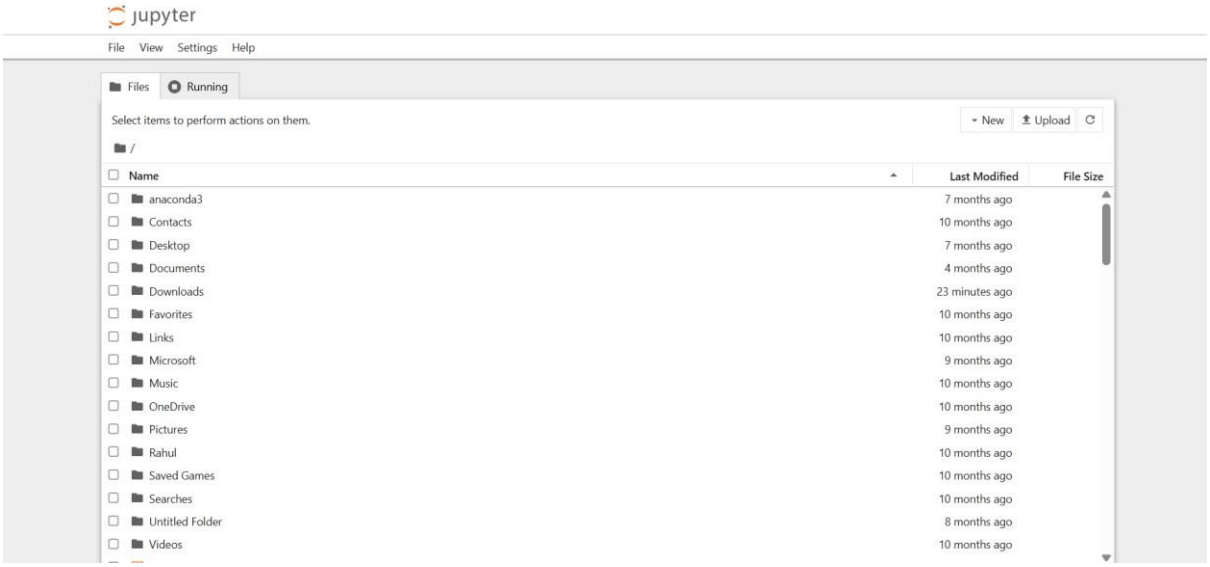
In this section there is a complete guide to run the project in any windows system.

1. Download and Install Anaconda Software in the windows system.

   (https://www.anaconda.com/products/individual)



2. Open the Jupyter Notebook from Anaconda.



3. After opening jupyter notebook click on the new notebook (python 3).
4. In notebook, Import all the required libraries.

Importing Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score, roc_curve
!pip install xgboost

import xgboost as xgb
```

5. Import the Provided Dataset.

```
customer_churn_data = pd.read_csv("C:/Users/Rahul Prakash/Downloads/Customer-Churn-Records.csv")
```

6. Next Step will be Pre Processing Step will be performed using following Code.

```
customer_churn_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 18 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   RowNumber          10000 non-null  int64
 1   CustomerId         10000 non-null  int64
 2   Surname            10000 non-null  object
 3   CreditScore        10000 non-null  int64
 4   Geography          10000 non-null  object
 5   Gender             10000 non-null  object
 6   Age                10000 non-null  int64
 7   Tenure             10000 non-null  int64
 8   Balance            10000 non-null  float64
 9   NumOfProducts      10000 non-null  int64
 10  HasCrCard          10000 non-null  int64
 11  IsActiveMember     10000 non-null  int64
 12  EstimatedSalary    10000 non-null  float64
 13  Exited             10000 non-null  int64
 14  Complain           10000 non-null  int64
 15  Satisfaction Score 10000 non-null  int64
 16  Card Type          10000 non-null  object
 17  Point Earned       10000 non-null  int64
dtypes: float64(2), int64(12), object(4)
memory usage: 1.4+ MB
```

Performing Data Cleaning Operations

```
# Check Missing Values
customer_churn_data.isnull().sum()
```

```
RowNumber            0
CustomerId           0
Surname              0
CreditScore          0
Geography            0
Gender               0
Age                  0
Tenure               0
Balance              0
NumOfProducts        0
HasCrCard            0
IsActiveMember       0
EstimatedSalary      0
Exited               0
Complain             0
Satisfaction Score   0
Card Type            0
Point Earned         0
dtype: int64
```

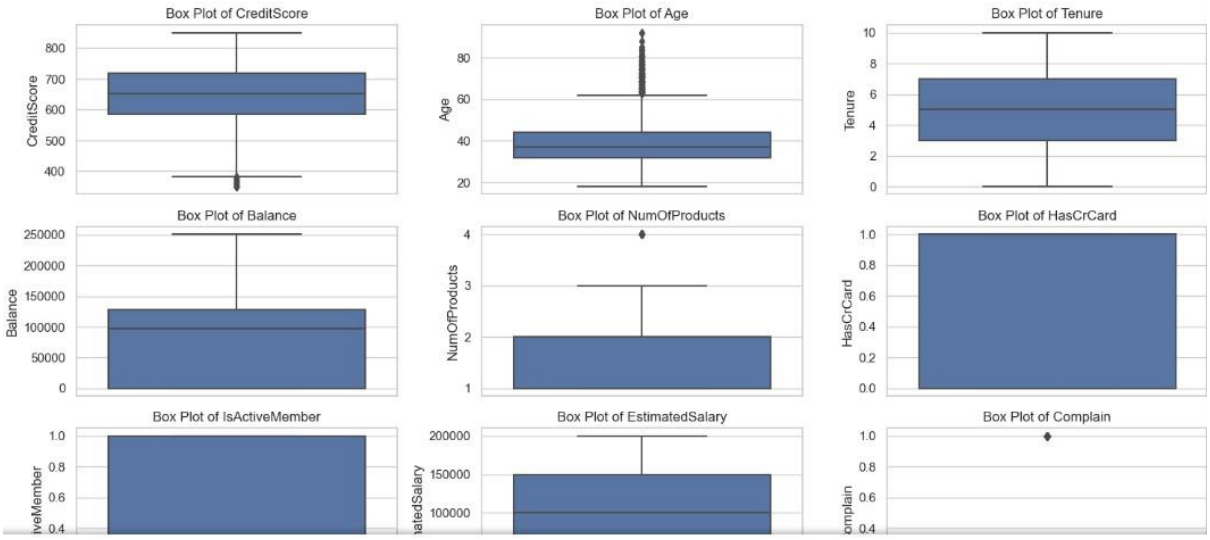
```
# Check Duplicate Values
customer_churn_data.duplicated().sum()
```

```
0
```

7. Exploratory Data Analysis has been Performed and Visualisation has been done using following Code

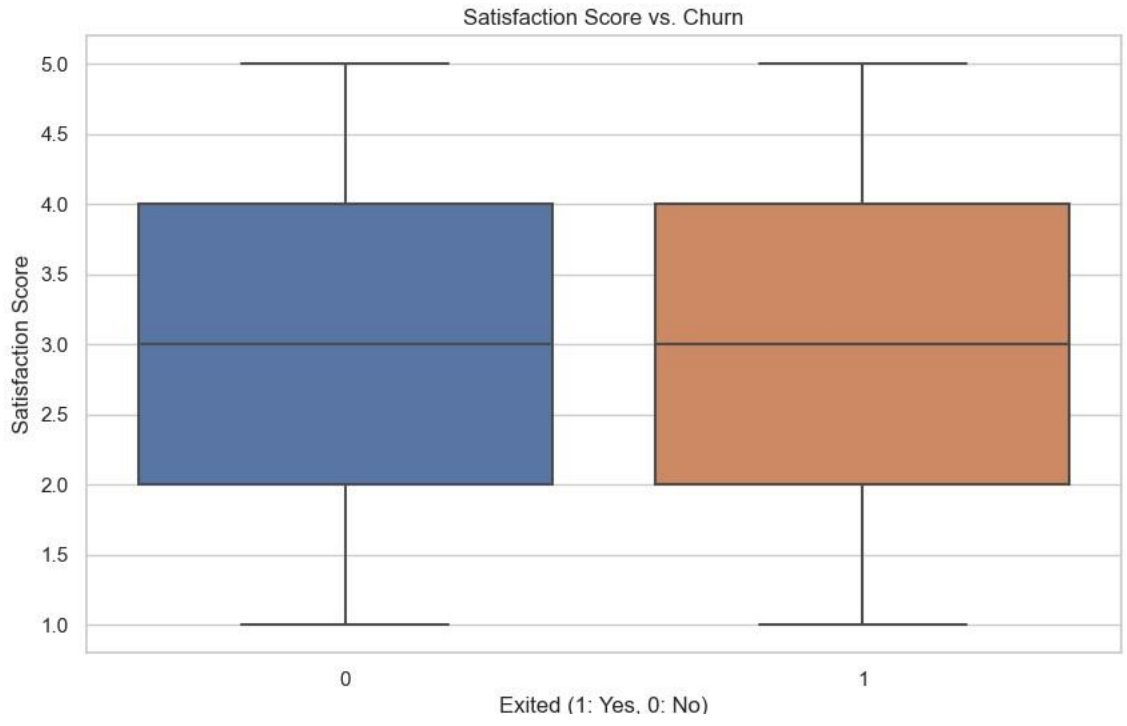
```
# Create box plots for each numerical column
for i, col in enumerate(numerical_cols, 1):
    plt.subplot(4, 3, i)  # Adjust the grid size as needed
    sns.boxplot(y=col, data=customer_churn_data)
    plt.title(f'Box Plot of {col}')

plt.tight_layout()  # Adjusts the layout to prevent overlap
plt.show()
```

```
<Figure size 1500x1000 with 0 Axes>
```



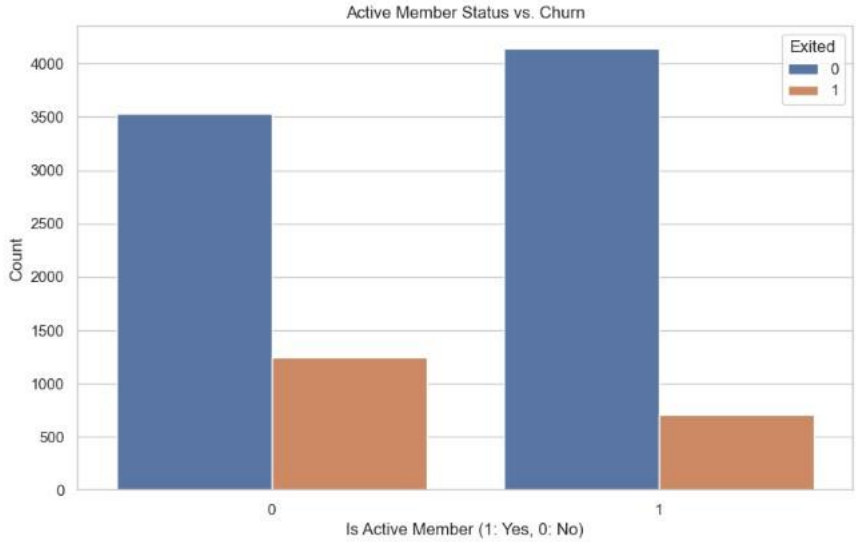
Poor Customer Service

```
# Correlation analysis between Satisfaction Score and Exited
plt.figure(figsize=(10, 6))
sns.boxplot(x='Exited', y='Satisfaction Score', data=customer_churn_data)
plt.title('Satisfaction Score vs. Churn')
plt.xlabel('Exited (1: Yes, 0: No)')
plt.ylabel('Satisfaction Score')
plt.show()
```



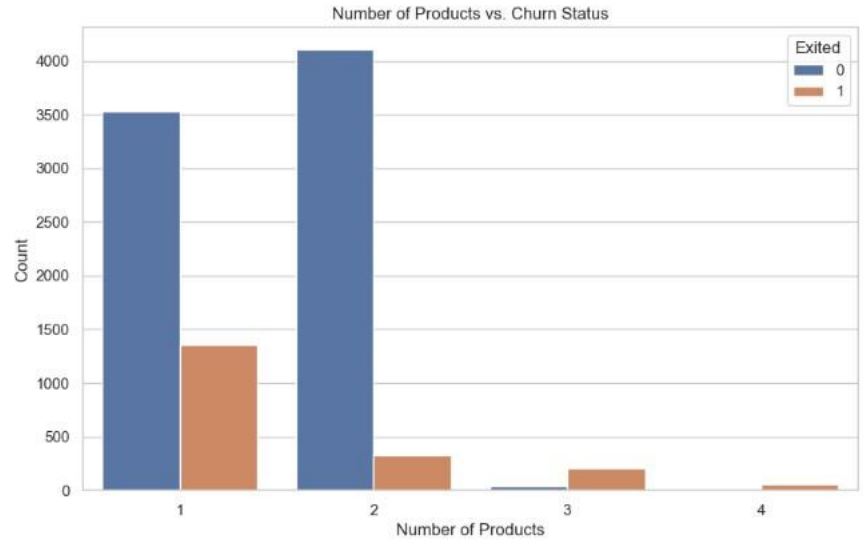
Technological Limitations

```
# Analyze IsActiveMember and churn
plt.figure(figsize=(10, 6))
sns.countplot(x='IsActiveMember', hue='Exited', data=customer_churn_data)
plt.title('Active Member Status vs. Churn')
plt.xlabel('Is Active Member (1: Yes, 0: No)')
plt.ylabel('Count')
plt.show()
```



Lack of Personalization

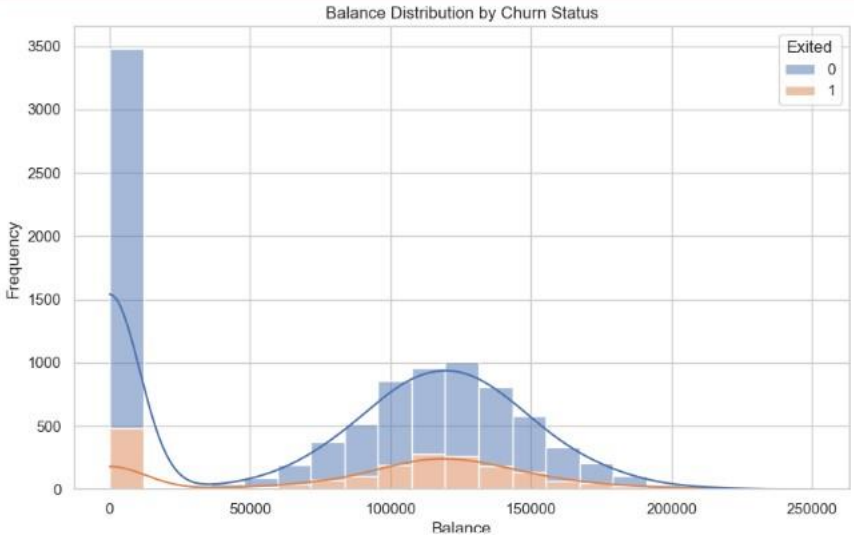
```
# Analyze NumOfProducts and churn status
plt.figure(figsize=(10, 6))
sns.countplot(x='NumOfProducts', hue='Exited', data=customer_churn_data)
plt.title('Number of Products vs. Churn Status')
plt.xlabel('Number of Products')
plt.ylabel('Count')
plt.show()
```



High Fees and Charges

```
# Compare Balance of churned vs non-churned customers
plt.figure(figsize=(10, 6))
sns.histplot(customer_churn_data, x='Balance', hue='Exited', multiple='stack', kde=True)
plt.title('Balance Distribution by Churn Status')
plt.xlabel('Balance')
plt.ylabel('Frequency')
plt.show()
```

```
C:\Users\Rahul Prakash\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



8. After Data Pre Processing the Data Splitting is Performed before Building a Model

Model Development

```
# Separate features and target variable
X = customer_churn_data.drop(columns=['Exited'])
y = customer_churn_data['Exited']
```

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)
```

9. Models Implementation has been Performed with the following Code

```
# Logistic Regression Model
logistic_model = Pipeline(steps=[('classifier', LogisticRegression(max_iter=1000))])
logistic_model.fit(X_train, y_train)
```

Pipeline
LogisticRegression

```
# Decision Tree Model
decision_tree_model = Pipeline(steps=[('classifier', DecisionTreeClassifier(random_state=42))])
decision_tree_model.fit(X_train, y_train)
```

Pipeline
DecisionTreeClassifier

```
# Random Forest Model
random_forest_model = Pipeline(steps=[('classifier', RandomForestClassifier(random_state=42))])
random_forest_model.fit(X_train, y_train)
```

Pipeline
RandomForestClassifier

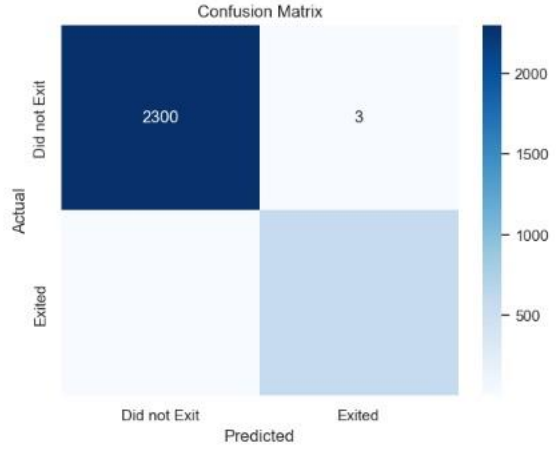
```
# XGBoost Model
xgb_model = Pipeline(steps=[('classifier', xgb.XGBClassifier(use_label_encoder=False, eval_metric='logloss'))])
xgb_model.fit(X_train, y_train)
```

```
C:\Users\Rahul Prakash\anaconda3\Lib\site-packages\xgboost\core.py:158: UserWarning: [09:44:43] WARNING: C:\buildkite-agent\builds\buildkite-windows-cp
u-autoscaling-group-i-0ed59c031377d09b8-1\xgboost\xgboost-ci-windows\src\learner.cc:740:
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

Pipeline
XGBClassifier

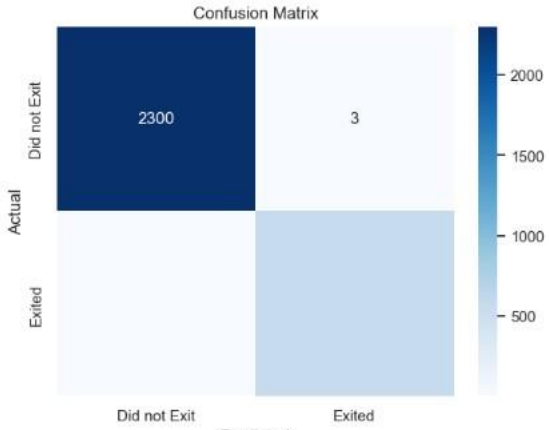
10. The Accuracy is considered as evaluation factor after Model Implementation

```
Logistic Regression Model Evaluation:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2303
           1       0.99      1.00      1.00       585

    accuracy                           1.00      2888
   macro avg       1.00      1.00      1.00      2888
weighted avg       1.00      1.00      1.00      2888
```



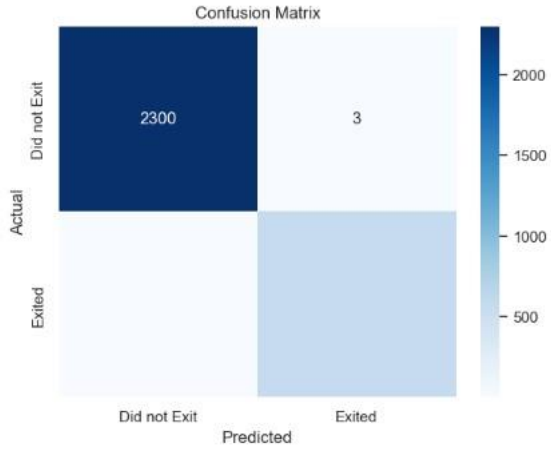
Decision Tree Model Evaluation:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2303
1	0.99	0.99	0.99	585
accuracy			1.00	2888
macro avg	1.00	0.99	0.99	2888
weighted avg	1.00	1.00	1.00	2888



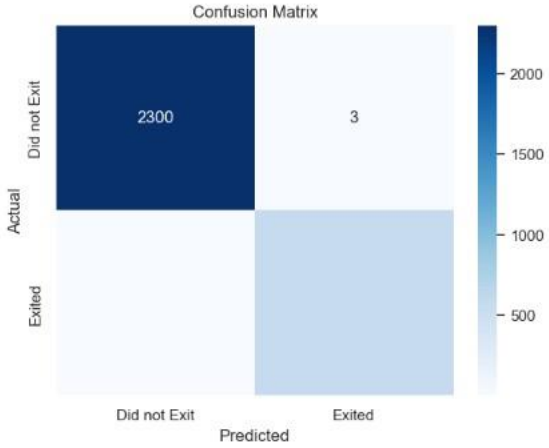
Random Forest Model Evaluation:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2303
1	0.99	1.00	1.00	585
accuracy			1.00	2888
macro avg	1.00	1.00	1.00	2888
weighted avg	1.00	1.00	1.00	2888



XGBoost Model Evaluation:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2303
1	0.99	1.00	1.00	585
accuracy			1.00	2888
macro avg	1.00	1.00	1.00	2888
weighted avg	1.00	1.00	1.00	2888



The concluding code files are include the ipynb file and the csv dataset.

11. Appendix

Pseudocode for Customer Churn Prediction

Step 1: Import Libraries

1. Import libraries for:
    - Data manipulation (pandas)
    - Visualization (matplotlib, seaborn)
    - Machine Learning models and utilities (sklearn, xgboost)
    - Evaluation metrics (sklearn.metrics)

Step 2: Load and Explore the Dataset

1. Load the CSV file containing customer churn data into a DataFrame.
2. Display the first few rows and check the dataset's structure using .info().

Step 3: Data Cleaning

1. Check for missing and duplicate values.
2. Drop unnecessary columns: RowNumber, CustomerId, and Surname.

3. Visualize distributions and check for outliers in numerical columns using box plots.

4. Define a function to remove outliers based on the Interquartile Range (IQR) and apply it to relevant columns (CreditScore, Age).

Step 4: Perform Exploratory Data Analysis (EDA)

1. Investigate correlations and distributions for factors contributing to churn:

    - Plot relationships between Satisfaction Score and Exited. ○ Plot the balance distribution for churned vs. non-churned customers.
    - Visualize relationships between NumOfProducts, IsActiveMember, and Exited.

Step 5: Normalize and Encode Data

1. Apply one-hot encoding for categorical columns: Geography, Gender, and Card Type.

2. Scale numerical columns using StandardScaler.

Step 6: Split Data into Features and Target

1. Separate the target variable (Exited) from features.

2. Split the dataset into training and testing sets using an 70-30 split and stratify by the target variable.

Step 7: Train Machine Learning Models

1. Initialize and train the following models using a Pipeline:

    - XGBoost Classifier ○

Step 8: Evaluate Models Logistic Regression ○ Decision Tree Classifier ○ Random Forest Classifier ○ Predict on test data. ○ Generate and display the classification report and confusion matrix. ○ Calculate AUC-ROC and plot the ROC curve.

Evaluate each model (Logistic Regression, Decision Tree, Random

2. Define a function to evaluate models:

3. Forest, XGBoost) using the evaluation function.

4. Store and compare the AUC scores for all models.

Step 9: Output Results

1. Print classification performance metrics and visualization for each model.

2. Display and compare AUC scores to determine the best-performing model.