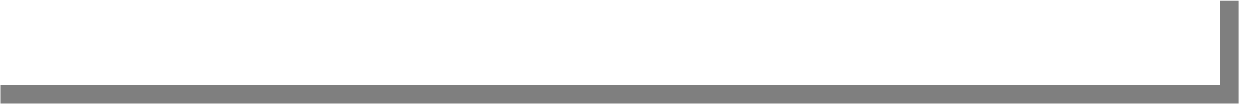
**1.**

A Project Report On



**Bank Customer Churn Prediction**

Submitted in partial fulfillment of the requirement for the award of the degree

MASTER OF COMPUTR APPLICATION

from

Marwadi University

Academic Year 2024 – 25

## Rutik Parmar (92400584150)

**Shyam Tanna (92400584172)**

**Internal Guide**

Miss. Gehna Sachdeva



Rajkot-Morbi Road, At & PO : Gauridad, Rajkot 360 003. Gujarat. India.



**Faculty of Computer Applications (FoCA)**

****

**This is to certify that the project work entitled Heart Failure Prediction Using Logistic Regression submitted in partial fulfillment of the requirement for**

**the award of the degree of**

Master of Science (Data Science)

**of the**

Marwadi University

**is a result of the bonafide work carried out by**

## Rutik Parmar (92400584150)

**Shyam Tanna (92400584172)**

**during the academic year 2024 – 2025**

|  |  |  |
| --- | --- | --- |
| *Miss. Gehna Sachdeva* | *Dr. Sunil Bajeja* | *Dr. R. Sridaran* |
| **Faculty Guide** | **HOD** | **Dean** |

# DECLARATION

We hereby declare that this project work entitled **Bank Customer Churn Prediction Using Classification** is a record done by us.

We also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place: Rajkot Date:

Rutik Parmar (92400584150) Signature: Shyam Tanna (92400584172) Signature:

# ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us. No serious and lasting achievement or success one can ever achieve without the help of friendly guidance and co-operation of so many people involved in the work.

We are very thankful to our guide **Miss. Gehna Sachdeva,** the person who makes us to follow the right steps during our project work. We express our deep sense of gratitude to for his guidance, suggestions and expertise at every stage. A part from that his valuable and expertise suggestion during documentation of our report indeed help us a lot.

Thanks to our friend and colleague who have been a source of inspiration and motivation that helped to us during our project work.

We are heartily thankful to the Dean of our department **Dr. R. Sridaran** for giving us an opportunity to work over this project and for their end-less and great support to all other people who directly or indirectly supported and help us to fulfil our task.

Rutik Parmar (92400584150) Signature: Shyam Tanna (92400584172) Signature:

## CONTENTS

|  |  |  |
| --- | --- | --- |
| **Chapters** | **Particulars** | **Page No.** |
| 1 | **Introduction**   * 1. Objective of the New System   2. Problem Definition   3. Core Components   4. Project Profile   5. Assumptions and Constraints   6. Advantages and Limitations of the Proposed System | 6 |
| 2 | **Requirement Determination & Analysis**   * 1. Requirement Determination   2. Targeted Users   3. Tool details (Python / PowerBI/ Tableau)   4. Library description (Details on various libraries / packages used) | 7 |
| 3 | **System Design**   * 1. Flowchart / Algorithm with steps   2. Dataset Design   3. Details on preprocessing steps applied | 8 |
| 4 | **Development**  4.1 Script details / Source code  4.2. Screen Shots / UI Design of simulation (if applicable)  4.3. Test reports | 10 |
| 5 | **Proposed Enhancements** | 20 |
| 6 | **Conclusion** | 21 |
| 7 | **Bibliography** | 21 |

1. **Introduction:**

### Objective of the New System:

* To preprocess the dataset (ml.csv) by handling missing values, encoding categorical features, and scaling numerical features.
* To train and evaluate multiple **classification algorithms** (Logistic Regression, KNN, Decision Tree, Random Forest, SVM, Naïve Bayes) on the dataset.
* To compare the models based on **accuracy, confusion matrix, and classification report**.
* To visualize important insights using **heatmaps, histograms, scatter plots, and boxplots**.
* To build a robust model that can predict whether a customer will **exit (1) or stay (0)** with high accuracy

### Problem Definition:

* In the competitive business environment, retaining customers is a major challenge. Companies face significant losses when customers discontinue their services (churn/exit). Predicting whether a customer will **exit or remain loyal** based on their historical and behavioral data can help businesses take preventive actions.
* This project focuses on developing a **machine learning classification model** to analyze customer data and predict the likelihood of customer exit (Exited column). By applying different classification algorithms and comparing their performance, the project aims to identify the most effective model for churn prediction

### Core Components:

* + - **Dataset:** Bank Customer Churn dataset
    - **Preprocessing:** Handling null values, encoding, scaling
    - **Algorithms:**  Logistic Regression, KNN, Decision Tree Classifier, Random Forest Classifier, SVM, NB
    - **Evaluation:** Accuracy, Confusion Matrix, Graphs

### Project Profile:

* + - **Project Title:** Bank Customer Churn Prediction Using Classification
    - **Domain:** Bank
    - **Technology:** Python (sklearn, pandas, seaborn, matplotlib)

### Assumptions and Constraints:

* + - **Assumptions:**
      * The dataset is clean after preprocessing.
      * Models are evaluated using accuracy and classification reports.

### Constraints:

* + - * Dataset may not represent real-world noise.
      * Imbalanced data may affect results.

### Advantages and Limitations of the Proposed System:

* + - **Advantages:**
      * Early Churn Prediction
      * Multiple Algorithms Used

### Data-Driven Decision Making

## Requirement Determination & Analysis:

### Requirement Determination:

To build a predictive system for heart failure, the following requirements were identified:

* + - Data preprocessing pipeline
    - Machine learning algorithms
    - Model evaluation techniques
    - Visualization tools for understanding the data

### Targeted Users:

Bank customers who are at risk of leaving

### Tool details (Python / PowerBI/ Tableau):

This project is developed using Python due to its wide range of libraries and ease of implementation for data analysis and machine learning.

### Library description (Details on various libraries / packages used):

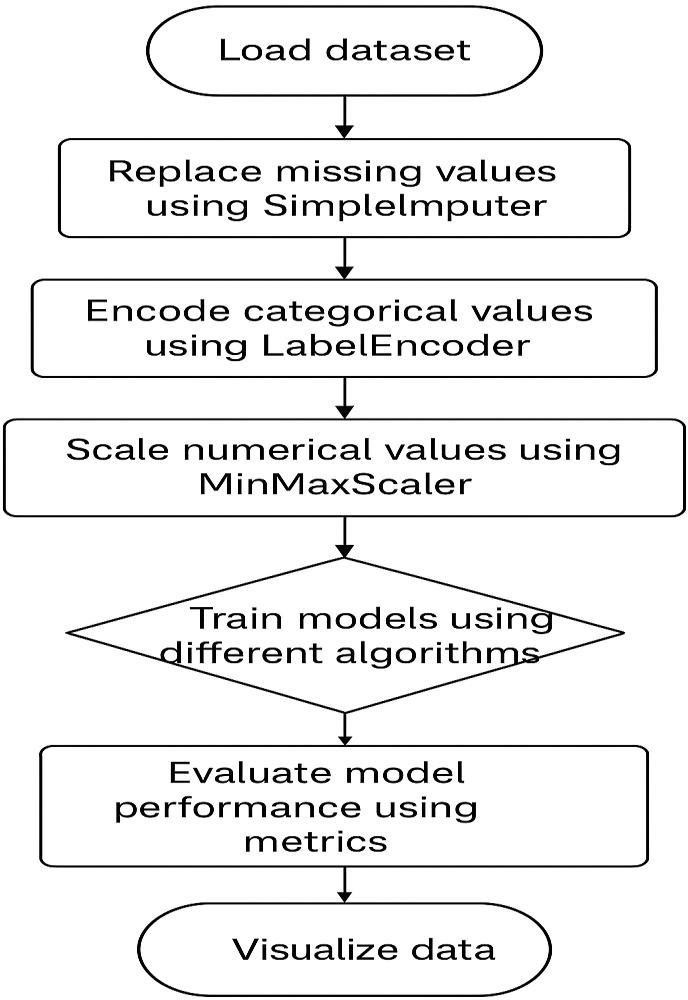
* + - **pandas:** Data manipulation and analysis
    - **numpy:** Numerical operations
    - **sklearn:** Machine learning algorithms and preprocessing tools
    - **seaborn:** Statistical data visualization
    - **matplotlib:** Plotting graphs

## System Design:

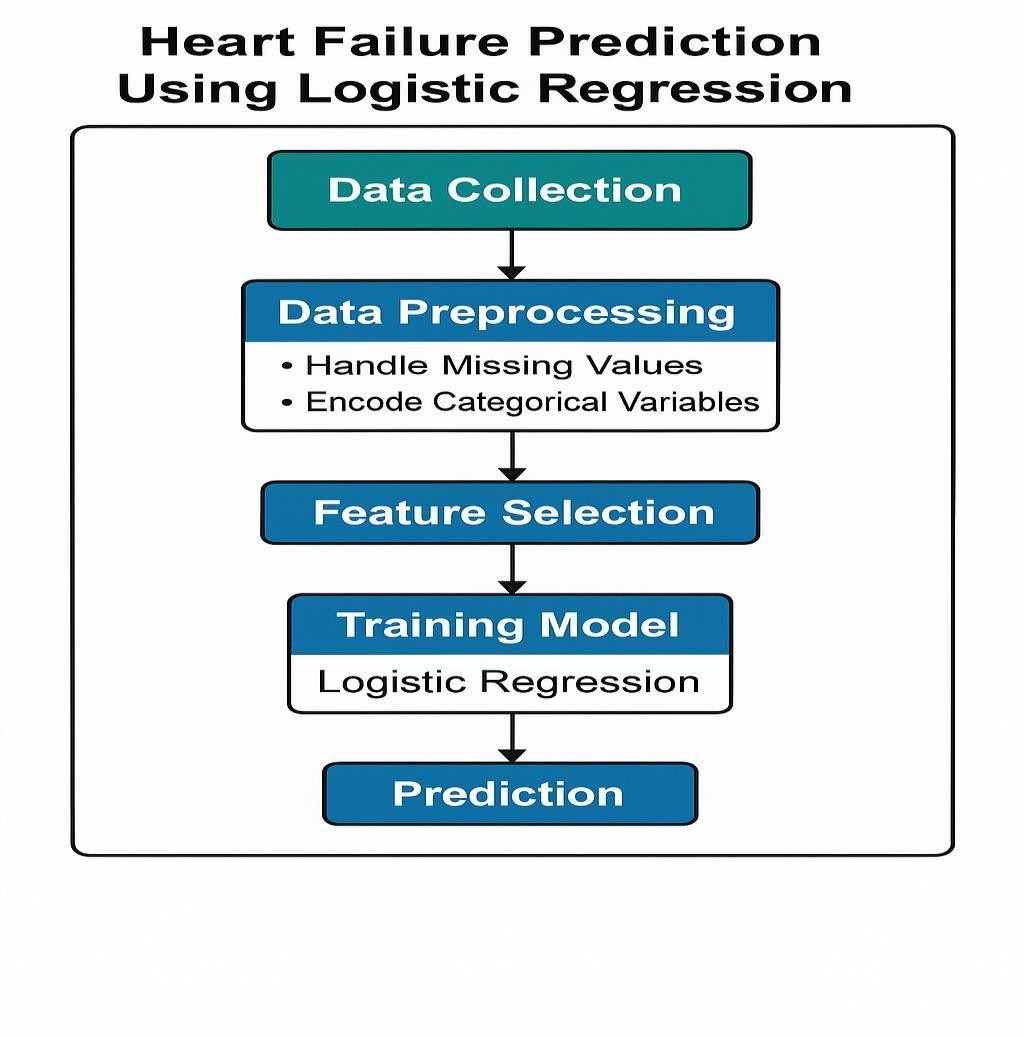
### Flowchart / Algorithm with steps:

* **Algorithm:**
* Step 1: Load dataset
* Step 2: Replace missing values using SimpleImputer
* Step 3: Encode categorical values using LabelEncoder
* Step 4: Scale numerical values using MinMaxScaler
* Step 5: Split data into training and test sets
* Step 6: Train models using different algorithms
* Step 7: Evaluate model performance using metrics
* Step 8: Visualize data using heatmap, scatter, boxplot, and histogram

### Flowchart:

****

* 1. **Dataset Design:**



### Details on preprocessing steps applied:

* + - **Missing Values:** Filled using SimpleImputer (mean for numerical, most frequent for categorical)
    - **Categorical Encoding:** LabelEncoder applied to all object type columns
    - **Feature Scaling:** MinMaxScaler used on numerical columns to normalize them between 0 and 1.

## Development:

### Script Details / Source Code:

The project is divided into two main Python scripts:

1. **preprocessing.py:** Responsible for cleaning the dataset using SimpleImputer, encoding with LabelEncoder, and scaling using MinMaxScaler.

### Code:

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder, MinMaxScaler

from sklearn.impute import SimpleImputer

# Models

from sklearn.linear\_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

from sklearn.naive\_bayes import GaussianNB

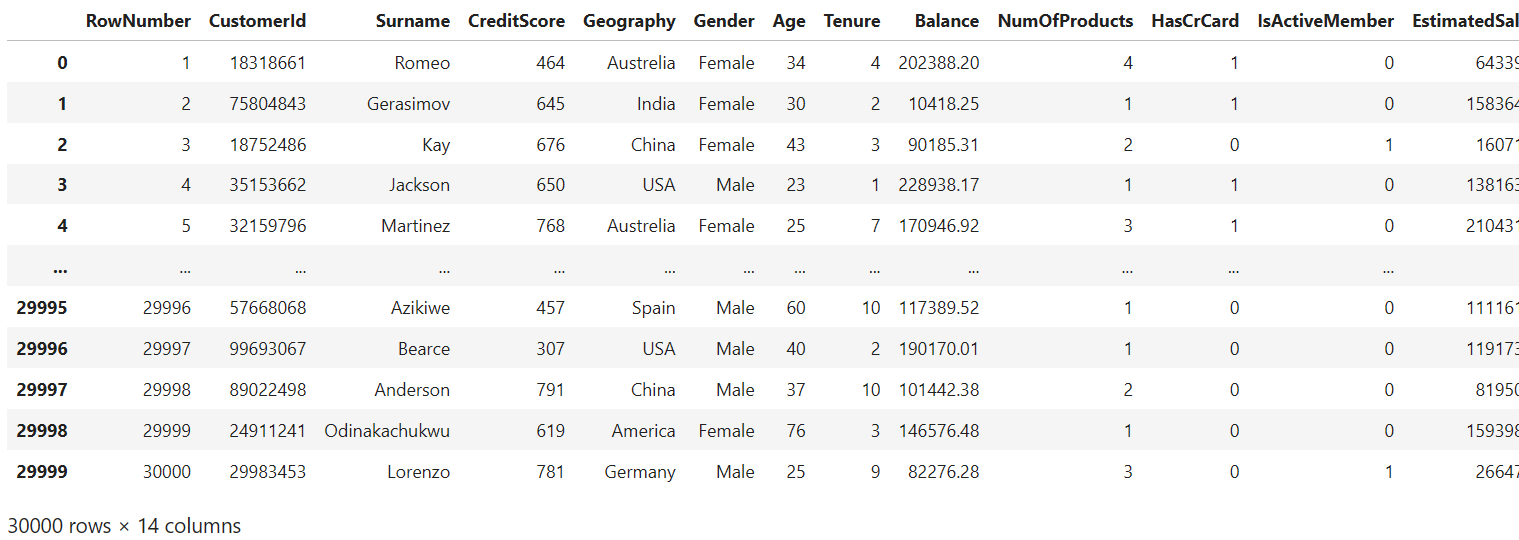
# Evaluation

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# 1. Load Dataset

df = pd.read\_csv("D:/ML\_Project/ml.csv")

df



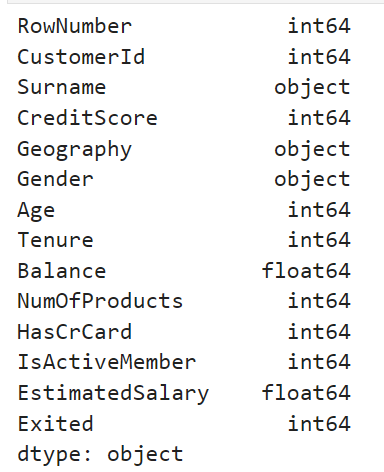
# 2. Handle Missing Values

print("Dataset preview: ")

print(df.head())

### Output:

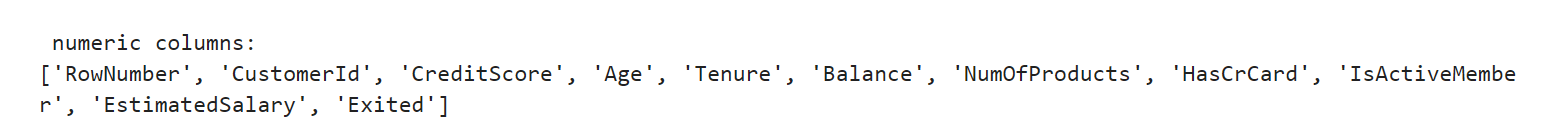
print(df.dtypes)



num\_cols = df.select\_dtypes(include=['int64','float64']).columns.tolist()

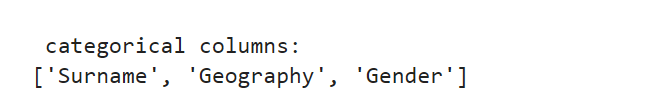
cat\_cols = df.select\_dtypes(include=['object']).columns.tolist()

print("\n numeric columns: ")

print(num\_cols)

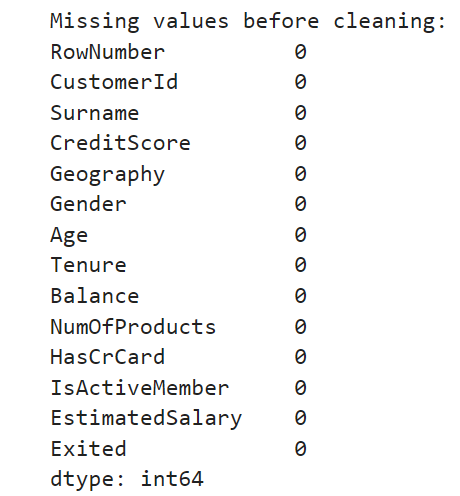
print("\n categorical columns: ")

print(cat\_cols)



print("Missing values before cleaning:")

print(df.isnull().sum())



num\_imputer = SimpleImputer(strategy="mean")

df[num\_cols] = num\_imputer.fit\_transform(df[num\_cols])

cat\_imputer = SimpleImputer(strategy="most\_frequent")

df[cat\_cols] = cat\_imputer.fit\_transform(df[cat\_cols])

# 3. Encode Categorical

label\_encoders = {}

for col in cat\_cols:

le = LabelEncoder()

df[col] = le.fit\_transform(df[col])

label\_encoders[col] = le

# 4. Choose Target Column

if "Exited" in df.columns:

target = "Exited"

else:

# fallback: last column ko target maan lo

target = df.columns[-1]

print("🎯 Target Selected:", target)

🎯 Target Selected: Exited

X = df.drop(columns=[target])

y = df[target]

# 5. Feature Scaling

from sklearn.preprocessing import MinMaxScaler

# Identify numeric columns again

num\_cols = df.select\_dtypes(include=['int64','float64']).columns.tolist()

# Remove target column if present

if "Exited" in num\_cols:

num\_cols.remove("Exited")

# Apply scaling

scaler = MinMaxScaler()

X[num\_cols] = scaler.fit\_transform(X[num\_cols])

# 6. Train-Test Split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size=0.3, random\_state=42, stratify=y

)

#Train Models

models = {

"Logistic Regression": LogisticRegression(max\_iter=500),

"KNN": KNeighborsClassifier(n\_neighbors=5),

"Decision Tree": DecisionTreeClassifier(random\_state=42),

"Random Forest": RandomForestClassifier(n\_estimators=100, random\_state=42),

"SVM (RBF)": SVC(kernel="rbf", probability=True),

"Naive Bayes": GaussianNB()

}

results = {}

for name, model in models.items():

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

acc = round(accuracy\_score(y\_test, y\_pred)\*100, 2)

results[name] = acc

print("\n===============================")

print(f"🔹 {name}")

print("Accuracy:", acc, "%")

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

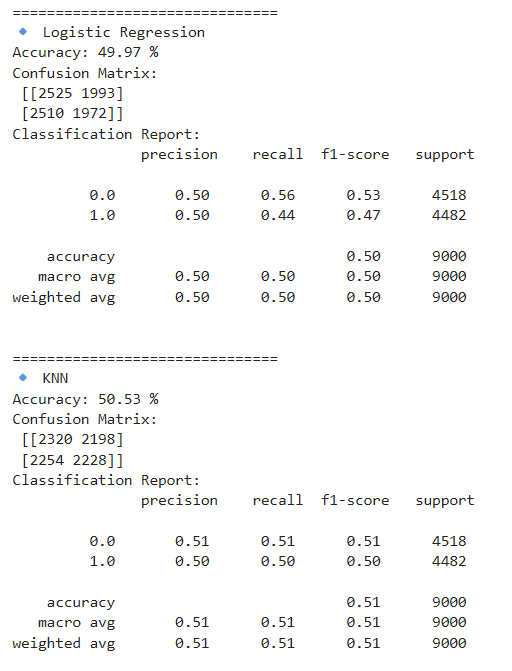
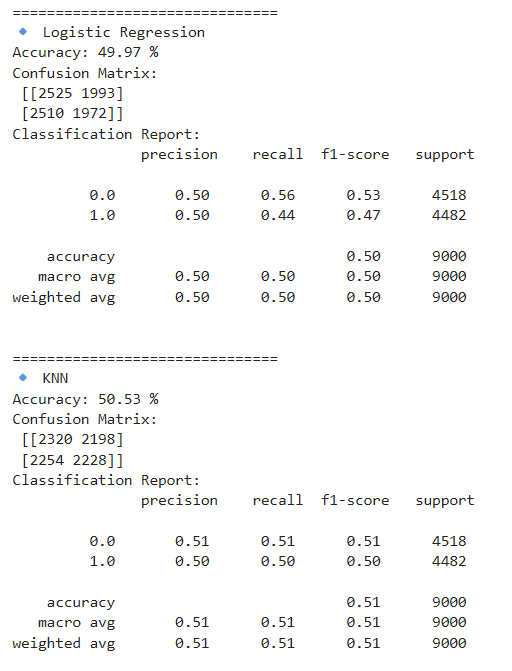
print("\n===============================")

print(f"🔹 {name}")

print("Accuracy:", acc, "%")

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(y\_test, y\_pred))



#**Accuracy Comparison Plot**

plt.figure(figsize=(8,5))

sns.barplot(x=list(results.keys()), y=list(results.values()), palette="viridis")

plt.title("Model Accuracy Comparison")

plt.ylabel("Accuracy %")

plt.xticks(rotation=30)

plt.show()

#**Feature Corelation Heatmap**

plt.figure(figsize=(12, 8))

sns.heatmap(df.corr(), annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)

plt.title("Feature Correlation Heatmap")

plt.show(block=False)

**# Histogram for all numerical columns dynamically**

fig, axes = plt.subplots(len(num\_cols), 1, figsize=(10, 4 \* len(num\_cols)))

for i, col in enumerate(num\_cols):

sns.histplot(df[col], bins=30, kde=True, ax=axes[i])

axes[i].set\_title(f"Histogram of {col}")

plt.tight\_layout()

plt.show()

**#Box Plot**

fig, axes = plt.subplots(1, len(num\_cols), figsize=(15, 5))

for i, col in enumerate(num\_cols):

sns.boxplot(x=y, y=df[col], ax=axes[i])

axes[i].set\_title(f"{col} vs {target}")

plt.tight\_layout()

plt.show()

**#Scatter Graph**

if "Age" in df.columns and "Balance" in df.columns:

plt.figure(figsize=(10, 6))

sns.scatterplot(x=df["Age"], y=df["Balance"], hue=df["Exited"], palette="Set1", alpha=0.7)

plt.title("Scatter Plot: Age vs Balance (Colored by Exited)", fontsize=14)

plt.show()

**#Prediction**

new\_data = pd.DataFrame([{

"CreditScore": 650,

"Geography": "France",

"Gender": "Male",

"Age": 40,

"Tenure": 5,

"Balance": 60000.0,

"NumOfProducts": 2,

"HasCrCard": 1,

"IsActiveMember": 1,

"EstimatedSalary": 50000.0

}])

pred\_class = model.predict(new\_data)[0]

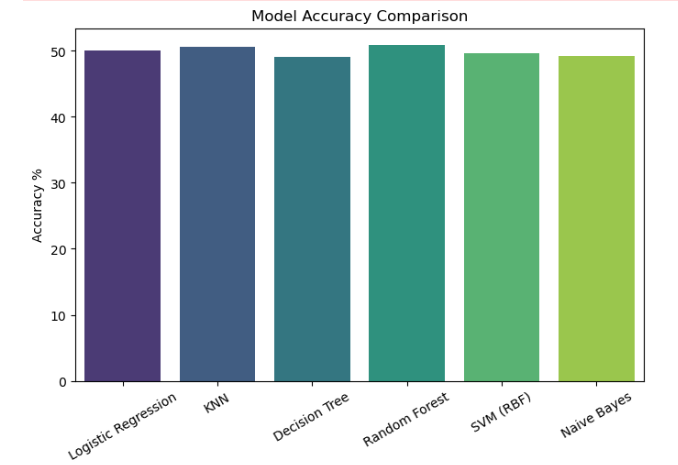
print("Here is prediction value from our sample data ")

print("Predicted Exited:", pred\_class)

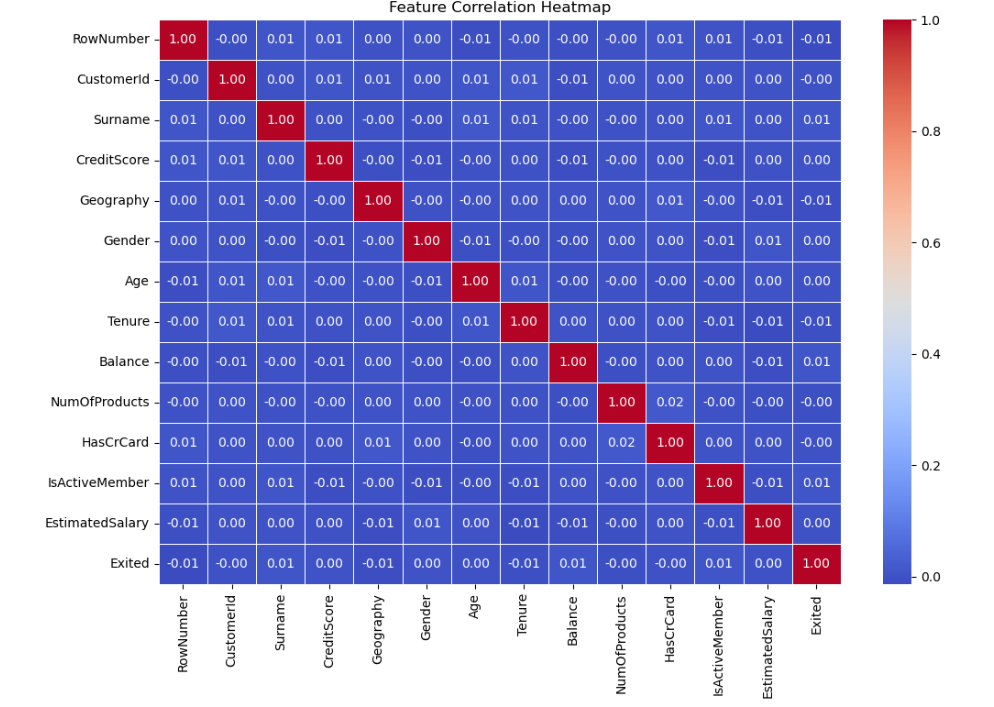
Graphical Visualization (heatmap, scatter plot, histogram, box plot):

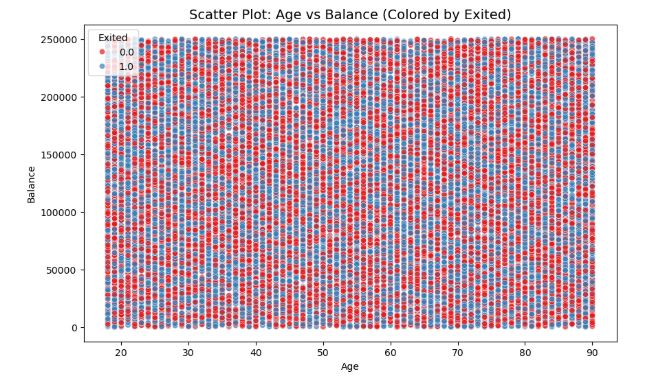
The following visualizations were used:

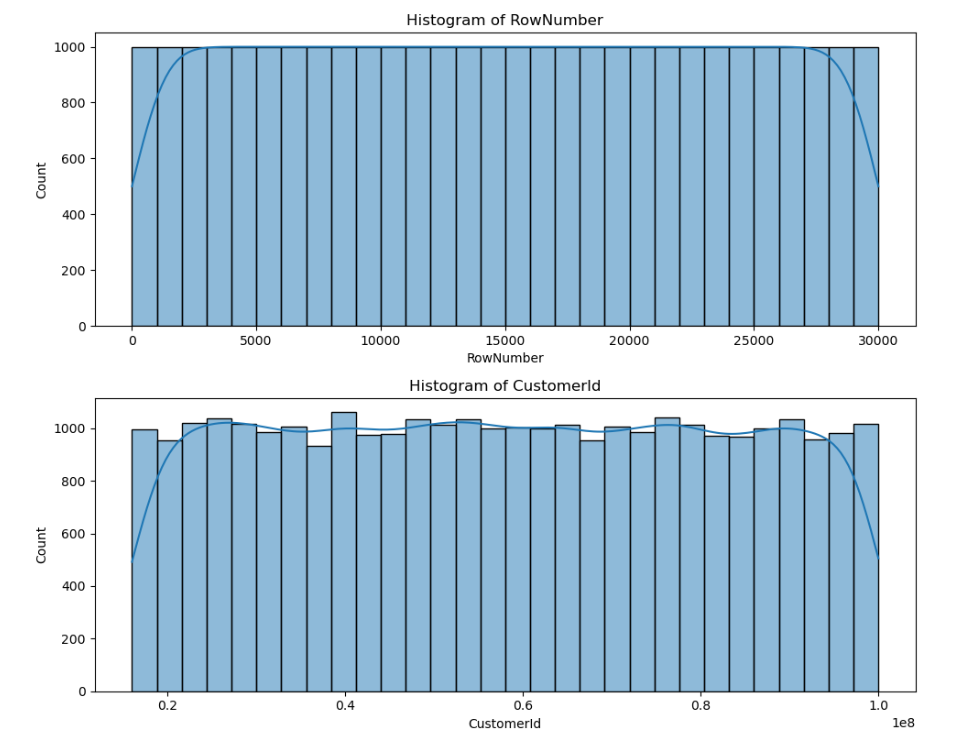
* **Accuracy Comparision Plot:**

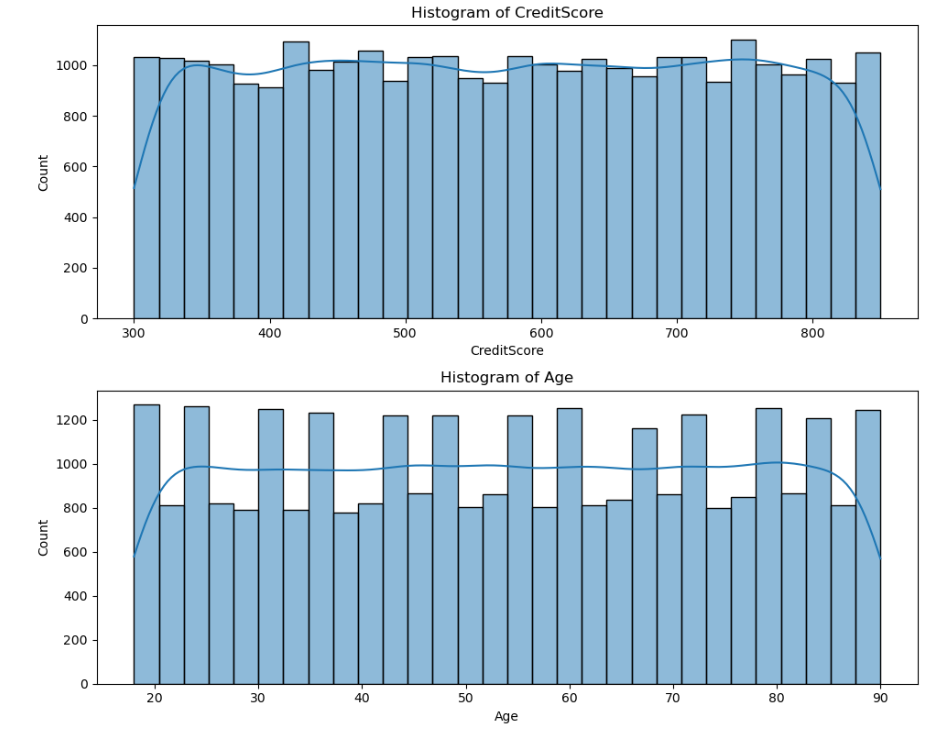
****

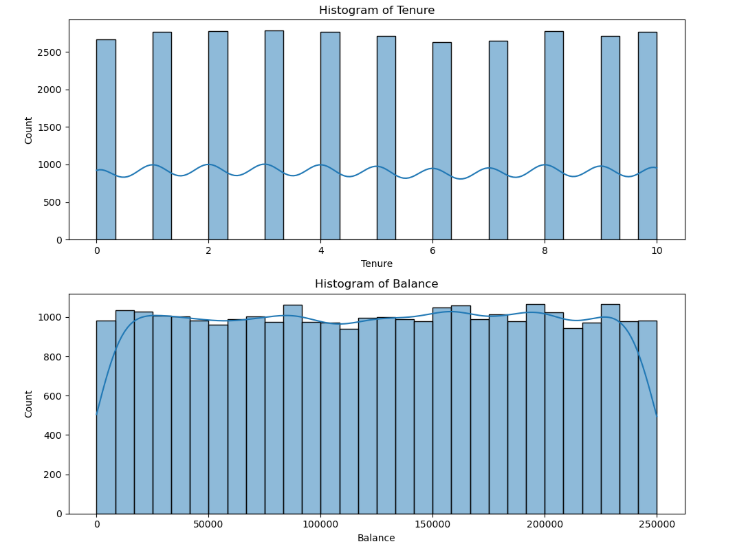
* + - **Heatmap:** Shows correlation between all features to identify the most influential variables.

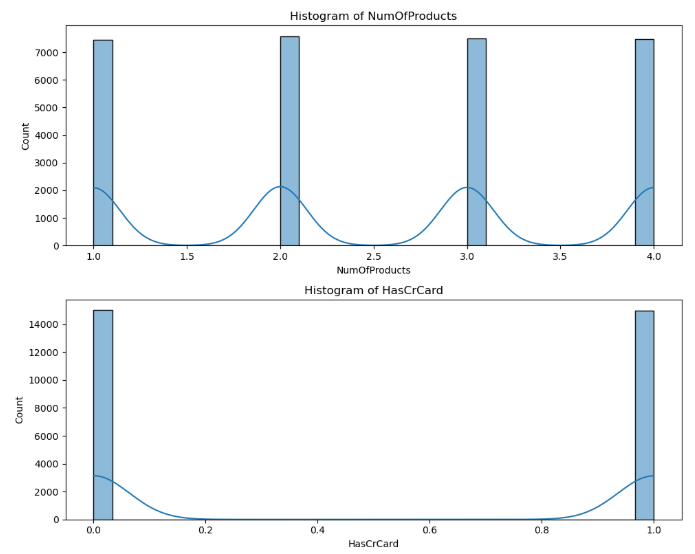


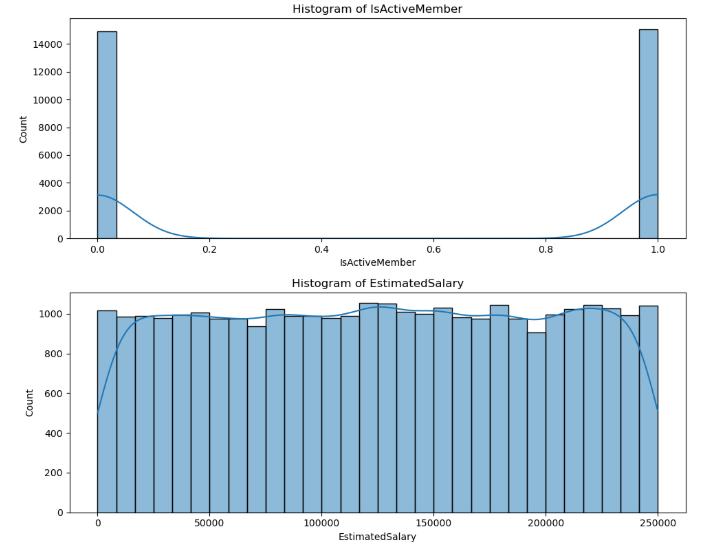
* + - **Scatter Plot:** Visualizes pairwise relationships between numerical features.
    - **Histogram:** Displays the distribution of individual numerical features.











* **Box Plot :**

****

### Test Reports (Accuracy, Classification Reports):

Each model was evaluated using Confusion Matrix, Accuracy Score, and Classification Report.

Results include precision, recall, F1-score, and overall accuracy. Below are some sample findings:

* + - **Logistic Regression:** Accuracy – 49.97 %
    - **KNN :** Accuracy – 50.53%
    - **Dicision Tree :** Accuracy – 49.06%
    - **Random Forest :** Accuracy – 50.88%

## Proposed Enhancements:

* The current project can be enhanced by integrating deep learning models such as Artificial Neural Networks for improved accuracy.  
  Additional customer behavior data like transaction history, feedback, and social interactions can be included to strengthen predictions.  
  Real-time churn prediction systems can be developed using streaming data for instant decision-making.  
  Feature selection and hyperparameter tuning can be applied to further optimize model performance.  
  Finally, deploying the model as a web or mobile application will make it easily accessible for business users.

## Conclusion:

This project successfully applies machine learning algorithms to predict customer churn (Exited).  
By identifying at-risk customers in advance, businesses can take proactive steps to improve retention.  
Overall, the system helps reduce revenue loss and enhances customer satisfaction through data-driven insights.

## Bibliography:

1. scikit-learn: Machine Learning in Python – <https://scikit-learn.org/>
2. pandas Documentation – <https://pandas.pydata.org/>
3. seaborn Documentation – <https://seaborn.pydata.org/>
4. matplotlib Documentation – <https://matplotlib.org/>
5. Heart Disease Dataset (original inspiration) – <https://www.kaggle.com/>