



# CHRIST

(DEEMED TO BE UNIVERSITY)

BANGALORE · INDIA

## DEPARTMENT OF COMPUTER SCIENCE BANGALORE YESWANTHPUR CAMPUS

### M.Sc. DATA SCIENCE

Project Report  
on

### CUSTOMER CHURN ANALYSIS

Submitted by  
Blessy Louis  
M.Sc. Data Science 'B'  
2348416

MAY 2024

---

## TABLE OF CONTENTS

<b>CHAPTER NO.</b>	<b>CHAPTER TITLE</b>	<b>PAGE NO.</b>
1	Introduction	05
2	Data Description	06
3	Procedure	08
4	Code	09
5	Output	13
6	Conclusion	30

## ABSTRACT

This project aims to address the challenge of customer churn in the banking industry by analysing a dataset and developing predictive models to identify customers likely to churn. We first perform data pre-processing tasks, including handling missing values and feature scaling, to prepare the dataset. Through exploratory data analysis (EDA), we visualize the distribution of various features and prominently display the overall churn rate to highlight its significance. Subsequently, we train logistic regression and random forest models to predict churn, evaluating their performance using accuracy, precision, recall, and F1 score metrics. Finally, we create a user-friendly interface for predicting churn based on customer features, empowering stakeholders to proactively retain at-risk customers. This project offers valuable insights and predictive capabilities to assist businesses in mitigating churn and enhancing customer retention strategies in the banking sector.

## AIM

The aim of this project is to develop a predictive model to identify customers at risk of churning in the banking industry, with a focus on enhancing customer retention strategies. Through comprehensive data analysis and machine learning techniques, the project aims to provide actionable insights into customer behaviour and factors influencing churn. By leveraging predictive models, the project seeks to enable businesses to proactively address churn, thereby improving customer satisfaction and loyalty while reducing revenue loss associated with customer attrition.

## Objective

The objectives of this project are as follows:

- Data Acquisition and Pre-processing: Gather and preprocess customer data to ensure data quality and consistency for analysis.
- Exploratory Data Analysis (EDA): Conduct an in-depth analysis of customer demographics, behaviours, and churn patterns to uncover insights and trends.
- Model Building and Evaluation: Develop machine learning models, including Logistic Regression and Random Forest, to predict customer churn based on relevant features extracted from the data. Evaluate model performance using metrics such as accuracy, precision, recall, and F1 score.

- Customer Prediction: Implement a user-friendly interface for users to input customer information and obtain predictions regarding the likelihood of churn, enabling businesses to take proactive retention actions.
- Documentation and Reporting: Compile a detailed document or report summarizing the project's aim, methodology, code implementation, results, and insights derived from the analysis.

## **CHAPTER – 01**

### **INTRODUCTION**

This project focuses on analysing customer churn in a financial services context, aiming to provide insights into customer behaviour and facilitate proactive retention strategies. Customer churn, the phenomenon where customers discontinue their relationship with a company, poses significant challenges to businesses, including revenue loss and decreased customer satisfaction. Through this project, we aim to understand the factors influencing customer churn and develop predictive models to identify customers at risk of churning. By leveraging machine learning techniques such as Logistic Regression and Random Forest, we seek to build accurate models that can forecast customer churn based on various features such as credit score, tenure, balance, and demographic information. The project involves comprehensive data acquisition, preprocessing, exploratory data analysis, model building, and evaluation. Ultimately, the findings and models generated will empower businesses to proactively engage with at-risk customers, optimize retention efforts, and enhance overall customer satisfaction and loyalty.

## CHAPTER – 02

### DATA DESCRIPTION

The dataset was taken from “Kaggle.com” for this particular project, the data can be verified from <https://www.kaggle.com/code/d4rklucif3r/churn-modelling-deployment-luciferml/input>

This dataset comprises customer information from a financial institution, including demographic details such as gender, age, and geographical location, as well as financial attributes like credit score, account balance, and tenure with the institution. The dataset also includes binary indicators for whether the customer holds a credit card, is an active member, and has churned or exited the relationship with the institution. With these features, the dataset offers insights into customer behaviour, preferences, and churn patterns, making it suitable for various analytical tasks such as customer segmentation, predictive modelling, and churn analysis.

#### **FEATURES:**

1. RowNumber: This represents the row number in the dataset, serving as a unique identifier for each record.
2. CustomerId: This is a unique identifier assigned to each customer, enabling the tracking of individual customer profiles.
3. Surname: This feature denotes the surname or last name of the customer, providing additional personal identification information.
4. CreditScore: This numeric attribute reflects the credit score of each customer, which is a measure of their creditworthiness based on various factors such as credit history and financial behaviour.
5. Geography: This categorical variable indicates the geographical location of the customer, providing insight into regional demographics and preferences.
6. Gender: This categorical attribute specifies the gender of the customer, allowing for gender-based analysis and segmentation.
7. Age: This numeric attribute represents the age of the customer, which is a crucial demographic factor influencing financial behaviour and preferences.
8. Tenure: This numeric feature indicates the number of years the customer has been associated with the financial institution, reflecting customer loyalty and longevity.

9. Balance: This numeric attribute denotes the account balance of the customer, representing the amount of funds available in their account.
10. NumOfProducts: This numeric feature indicates the number of financial products or services held by the customer, providing insights into their level of engagement with the institution.
11. HasCrCard: This binary attribute indicates whether the customer holds a credit card with the institution (1 for yes, 0 for no), offering insights into their banking behavior and preferences.
12. IsActiveMember: This binary feature indicates whether the customer is an active member of the institution (1 for yes, 0 for no), reflecting their level of engagement and participation.
13. EstimatedSalary: This numeric attribute represents the estimated salary of the customer, providing insights into their financial capacity and spending potential.
14. Exited: This binary target variable indicates whether the customer has churned or exited the relationship with the institution (1 for churned, 0 for retained), serving as the outcome variable for predictive modelling and analysis.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
RowNumber	CustomerID	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited			
1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.88	1			
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0			
3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.57	1			
4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0			
5	15737888	Mitchell	850	Spain	Female	43	2	125510.8	1	1	1	79084.1	0			
6	15574012	Chu	645	Spain	Male	44	8	113755.8	2	1	0	149756.71	1			
7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0			
8	15656148	Obinna	376	Germany	Female	29	4	115046.7	4	1	0	119346.88	1			
9	15792365	He	501	France	Male	44	4	142025.1	2	0	1	74940.5	0			
10	15592389	H?	684	France	Male	27	2	134603.9	1	1	1	71725.73	0			
11	15767821	Bearce	528	France	Male	31	6	102016.7	2	0	0	80181.12	0			
12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01	0			
13	15632264	Key	476	France	Female	34	10	0	2	1	0	26260.98	0			
14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.79	0			
15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.65	0			
16	15643966	Goforth	616	Germany	Male	45	3	143129.4	2	0	1	64327.26	0			
17	15737452	Romeo	653	Germany	Male	58	1	132602.9	1	1	0	5097.67	1			
18	15788218	Henderson	549	Spain	Female	24	9	0	2	1	1	14406.41	0			
19	15661507	Muldrow	587	Spain	Male	45	6	0	1	0	0	158684.81	0			
20	15568982	Hao	726	France	Female	24	6	0	2	1	1	54724.03	0			
21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886.17	0			
22	15597945	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555.46	0			
23	15699309	Gerasimov	510	Spain	Female	38	4	0	1	1	0	118913.53	1			
24	15725737	Mosman	669	France	Male	46	3	0	2	0	1	8487.75	0			
25	15625047	Yen	846	France	Female	38	5	0	1	1	1	187616.16	0			
26	15738191	Maclean	577	France	Male	25	3	0	2	0	1	124508.29	0			

## CHAPTER – 03

### PROCEDURE

The procedure involved in this project can be elaborated as follows:

- **Data Acquisition and Pre-processing:** The initial step involves acquiring the dataset containing customer information from a financial institution. The dataset is then pre-processed to handle missing values, which are often filled using mean or median imputation methods. Categorical variables are encoded using one-hot encoding to convert them into a numerical format suitable for analysis. Additionally, feature scaling is applied to standardize numerical features, ensuring that they are on a similar scale.
- **Exploratory Data Analysis (EDA):** Following pre-processing, exploratory data analysis (EDA) is conducted to gain insights into the dataset's characteristics. Various visualizations such as pie charts, histograms, and box plots are utilized to explore the distributions and relationships between different variables. EDA helps in identifying trends, patterns, outliers, and potential correlations within the data.
- **Model Building and Evaluation:** After gaining a comprehensive understanding of the dataset, predictive models are built to analyse customer churn. Two popular algorithms, namely Logistic Regression and Random Forest Classifier, are trained using the pre-processed data. The models are evaluated using performance metrics such as accuracy, precision, recall, and F1 score, along with a confusion matrix to assess their predictive capabilities.
- **Customer Prediction:** In this phase, the trained models are applied to make predictions on new customer data. Users can input customer attributes such as credit score, tenure, and account balance through a user-friendly interface. The models predict the likelihood of a customer churning based on these input features, providing valuable insights for customer retention strategies.

Throughout the procedure, the project emphasizes a systematic approach, leveraging both descriptive and predictive analytics techniques to analyse customer churn and inform decision-making in financial institutions.

# CHAPTER – 04

## CODE

The project involves a Modular approach of the Churn prediction Analysis:

### Data\_preprocessing.py:

```

1 import streamlit as st
2 import numpy as np
3 import pandas as pd
4 from sklearn.impute import SimpleImputer
5 from sklearn.preprocessing import OneHotEncoder, StandardScaler
6 from sklearn.model_selection import train_test_split
7
8 # Load data function
9 # @param allow_output_mutation=True
10 def load_data(file_path):
11     data = pd.read_csv(file_path)
12     return data
13
14 # Handle missing values function
15 def handle_missing_values(data):
16     numeric_columns = data.select_dtypes(include=np.number).columns
17     imputer = SimpleImputer(strategy='mean')
18     data_filled = data.copy()
19     for col in numeric_columns:
20         data_filled[col] = imputer.fit_transform(data[[col]])
21     return data_filled
22
23 # Encode categorical variables function
24 def encode_categorical_variables(data):
25     encoder = OneHotEncoder()
26     encoded_data = encoder.fit_transform(data)
27     return encoded_data
28
29 # Feature scaling function
30 def feature_scaling(data):
31     numeric_columns = data.select_dtypes(include=np.number).columns
32     scaler = StandardScaler()
33     scaled_data = data.copy()
34     scaled_data[numeric_columns] = scaler.fit_transform(data[numeric_columns])
35     return scaled_data
36
37

```

### Visualization.py:

```

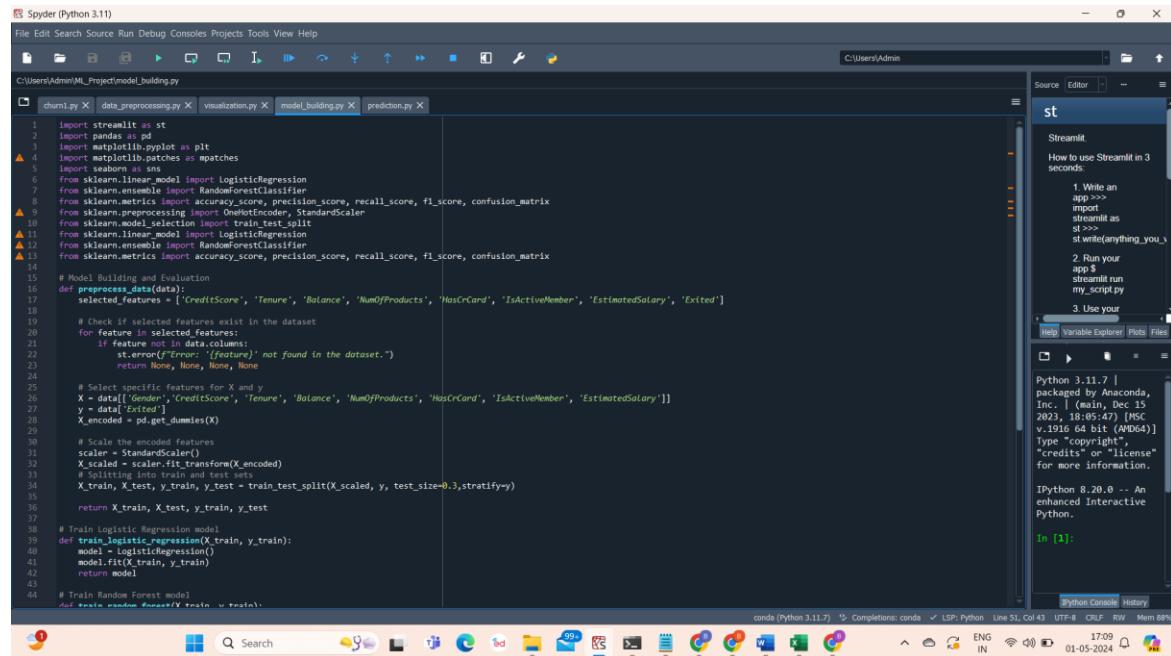
1 import streamlit as st
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import pandas as pd
5
6 # Exploratory Data Analysis Function
7 def visualize_distribution(data):
8     st.header("Exploratory Data Analysis")
9     # Churn Rate
10    churn_rate = data['Exited'].mean() * 100 # Convert to percentage
11    # Display overall churn rate
12    st.markdown(f'

Overall Churn Rate: {churn_rate:.2f}%

', unsafe_allow_html=True)
13    # Churn vs Geography
14    fig, ax = plt.subplots(figsize=(6, 6))
15    data['Geography'].value_counts().plot.pie(autoptick=True, startangle=140, colors=['skyblue', 'lightcoral'])
16    st.pyplot(fig)
17
18    # Subheader: "Distribution of Exit"
19    st.subheader("Distribution of Exit")
20    fig, ax = plt.subplots(figsize=(6, 6))
21    data['Exited'].value_counts().plot.pie(autoptick=True, startangle=140, colors=['lightgreen', 'lightcoral'])
22    st.pyplot(fig)
23
24    # Subheader: "Distribution of MaritalStatus"
25    st.subheader("Distribution of MaritalStatus")
26    fig, ax = plt.subplots(figsize=(6, 6))
27    data['MaritalStatus'].value_counts().plot.pie(autoptick=True, startangle=140, colors=['lightblue', 'orange'])
28    st.pyplot(fig)
29
30    # Subheader: "Distribution of IsActiveMember"
31    st.subheader("Distribution of IsActiveMember")
32    fig, ax = plt.subplots(figsize=(6, 6))
33    data['IsActiveMember'].value_counts().plot.pie(autoptick=True, startangle=140, colors=['lightpink', 'lightgreen'])
34    st.pyplot(fig)
35
36    # Subheader: "Distribution of Geography"
37    st.subheader("Distribution of Geography")
38    fig, ax = plt.subplots(figsize=(6, 6))
39    sns.countplot(x="Geography", data=data, palette='Set2')
40    st.pyplot(fig)
41
42    # Subheader: "Histogram of Balance"
43    st.subheader("Histogram of Balance")
44    fig, ax = plt.subplots(figsize=(6, 6))
45    sns.histplot(data['Balance'], kde=True, color='skyblue')
46    st.pyplot(fig)
47
48    # Subheader: "Boxplot of EstimatedSalary"
49    st.subheader("Boxplot of EstimatedSalary")
50    fig, ax = plt.subplots(figsize=(6, 6))
51    sns.boxplot(data['EstimatedSalary'], color='lightgreen', ax=ax)
52    st.pyplot(fig)
53
54    # Subheader: "Violin plot of CreditScore"
55    st.subheader("Violin plot of CreditScore")
56    fig, ax = plt.subplots(figsize=(6, 6))
57    sns.violinplot(data['CreditScore'], color='lightcoral', ax=ax)
58    st.pyplot(fig)
59
60

```

## Model\_building.py:

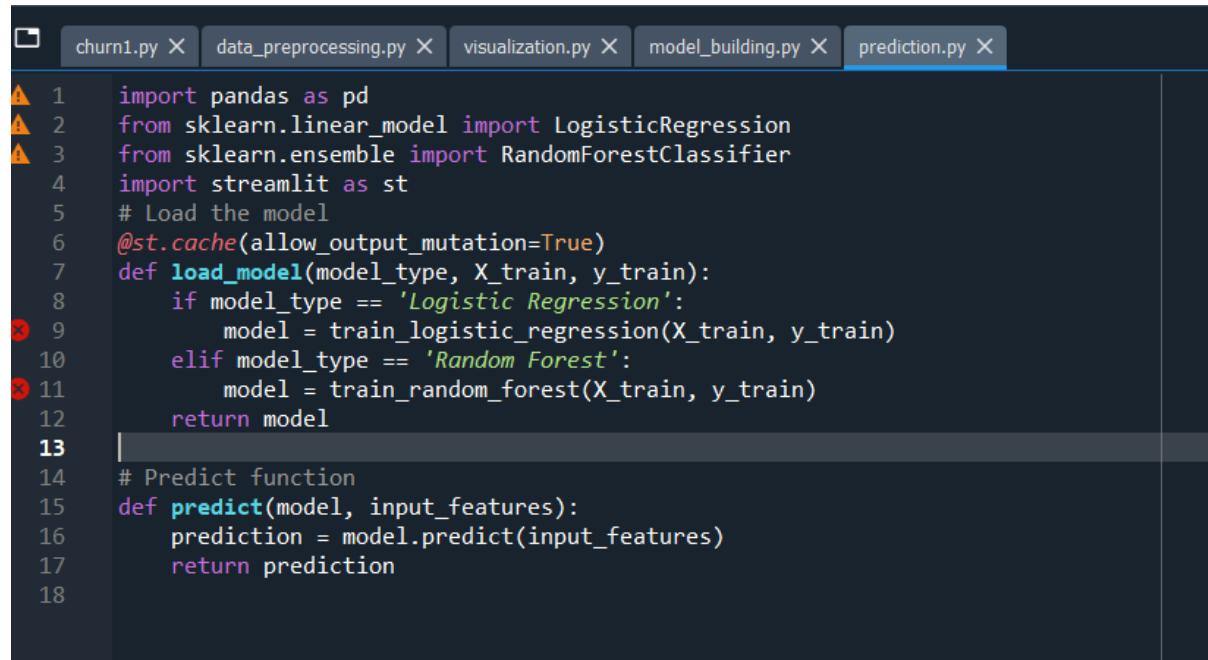


```

1 import streamlit as st
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import matplotlib.patches as mpatches
5 import seaborn as sns
6 from sklearn.linear_model import LogisticRegression
7 from sklearn.ensemble import RandomForestClassifier
8 from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
9 from sklearn.preprocessing import OneHotEncoder, StandardScaler
10 from sklearn.model_selection import train_test_split
11 from sklearn.linear_model import LogisticRegression
12 from sklearn.ensemble import RandomForestClassifier
13 from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
14
15 # Model Building and Evaluation
16 def preprocess_data(data):
17     selected_features = ['CreditScore', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', 'EstimatedSalary', 'Exited']
18
19     # Check if selected features exist in the dataset
20     for feature in selected_features:
21         if feature not in data.columns:
22             st.error(f'Error: "{feature}" not found in the dataset.')
23             return None, None, None, None
24
25     # Select specific features from X
26     X = data[['Gender', 'CreditScore', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', 'EstimatedSalary']]
27     y = data['Exited']
28
29     X_encoded = pd.get_dummies(X)
30
31     # Scale the encoded features
32     scaler = StandardScaler()
33     X_scaled = scaler.fit_transform(X_encoded)
34     # Splitting into train and test sets
35     X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, stratify=y)
36
37     return X_train, X_test, y_train, y_test
38
39 # Train Logistic Regression model
40 def train_logistic_regression(X_train, y_train):
41     model = LogisticRegression()
42     model.fit(X_train, y_train)
43     return model
44
45 # Train Random Forest model
46 def train_random_forest(X_train, y_train):

```

## Prediction.py:

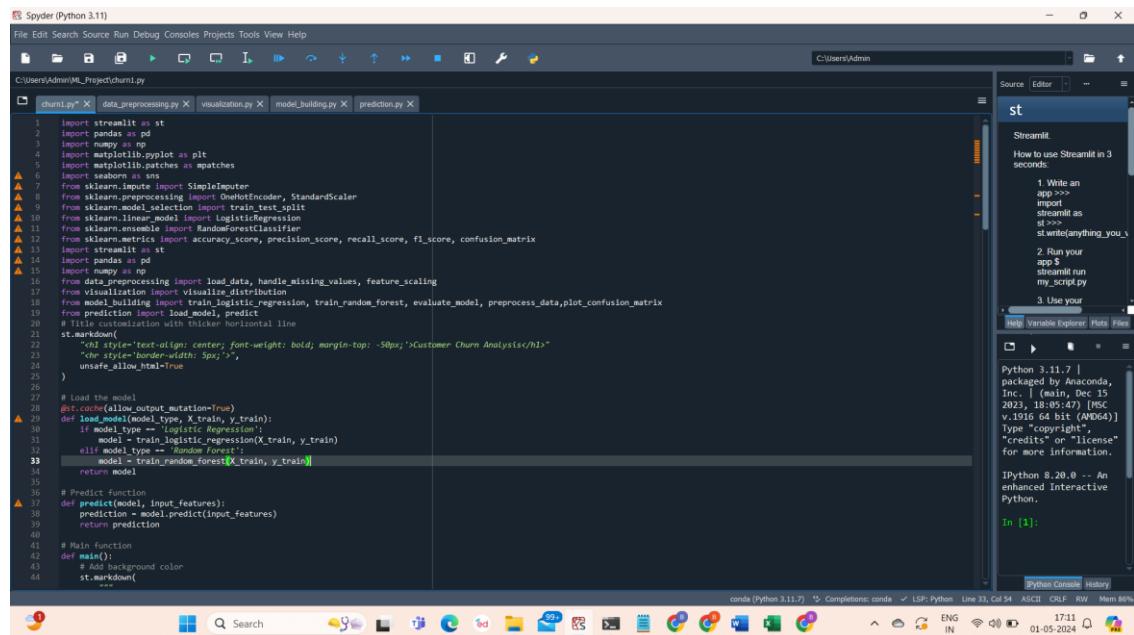


```

1 import pandas as pd
2 from sklearn.linear_model import LogisticRegression
3 from sklearn.ensemble import RandomForestClassifier
4 import streamlit as st
5
6 # Load the model
7 @st.cache(allow_output_mutation=True)
8 def load_model(model_type, X_train, y_train):
9     if model_type == 'Logistic Regression':
10         model = train_logistic_regression(X_train, y_train)
11     elif model_type == 'Random Forest':
12         model = train_random_forest(X_train, y_train)
13     return model
14
15 # Predict function
16 def predict(model, input_features):
17     prediction = model.predict(input_features)
18     return prediction

```

## Churn1.py: (Main)

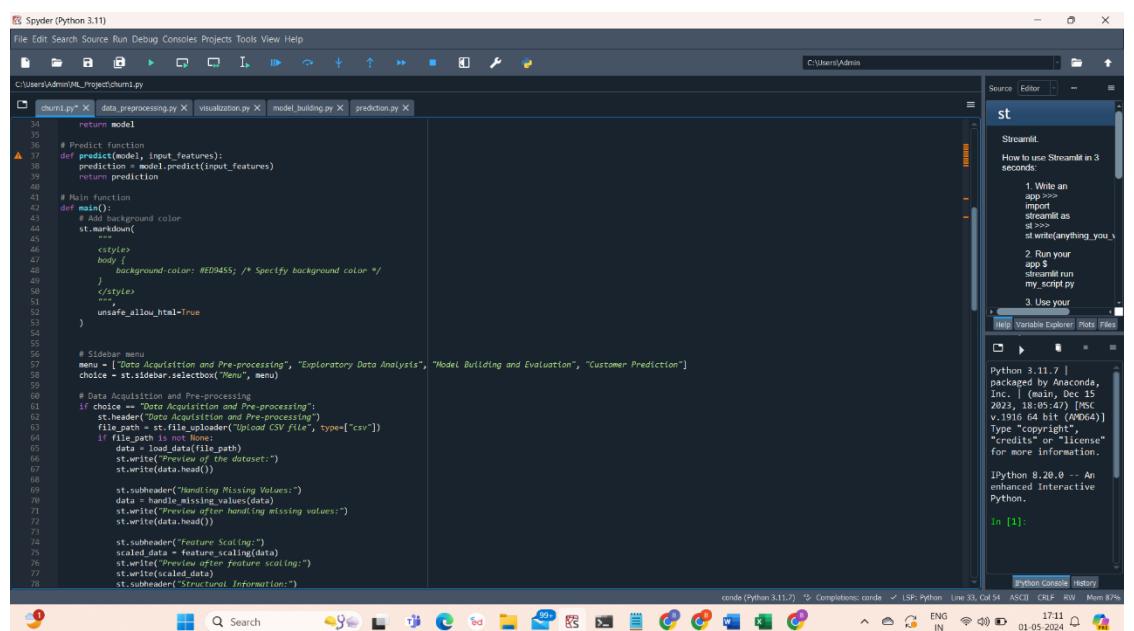


```

1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import matplotlib.patches as mpatches
6 import seaborn as sns
7 from sklearn.impute import SimpleImputer
8 from sklearn.preprocessing import StandardScaler
9 from sklearn.model_selection import train_test_split
10 from sklearn.linear_model import LogisticRegression
11 from sklearn.ensemble import RandomForestClassifier
12 from sklearn.metrics import accuracy_score, f1_score, confusion_matrix
13 import streamlit as st
14 import pandas as pd
15 import numpy as np
16 from sklearn.preprocessing import LabelEncoder, StandardScaler
17 from visualization import visualize_distribution
18 from model_building import train_logistic_regression, train_random_forest, evaluate_model, preprocess_data, plot_confusion_matrix
19 from prediction import load_model, predict
20 # Title visualization with thicker horizontal line
21 st.markdown(
22     '

# Customer Churn Analysis

',
23     unsafe_allow_html=True
24 )
25
26 # Load the model
27 @st.cache(show_spinner=False)
28 def load_model(model_type):
29     if model_type == "Logistic Regression":
30         model = train_logistic_regression(X_train, y_train)
31     elif model_type == "Random Forest":
32         model = train_random_forest(X_train, y_train)
33     return model
34
35 # Predict Function
36 def predict(model, input_features):
37     prediction = model.predict(input_features)
38     return prediction
39
40 # Main Function
41 def main():
42     # Add background color
43     st.markdown(
44         """
45             <style>
46                 body {
47                     background-color: #ED9455; /* Specify background color */
48                 }
49             </style>
50             <!--
51             unsafe_allow_html=True
52         </!-->
53     
```



```

54
55     # Sidebar menu
56     menu = ["Data Acquisition and Pre-processing", "Exploratory Data Analysis", "Model Building and Evaluation", "Customer Prediction"]
57     choice = st.sidebar.selectbox("Menu", menu)
58
59     # Data Acquisition and Pre-processing
60     if choice == "Data Acquisition and Pre-processing":
61         st.header("Data Acquisition and Pre-processing")
62         st.write("Upload CSV file")
63         file_path = st.file_uploader("Upload CSV file", type=["csv"])
64         if file_path is not None:
65             data = pd.read_csv(file_path)
66             st.write("Preview of the dataset:")
67             st.write(data.head())
68
69             st.subheader("Handling Missing Values:")
70             data = handle_missing_values(data)
71             st.write("Preview after handling missing values:")
72             st.write(data.head())
73
74             st.subheader("Feature Scaling:")
75             scaled_data = feature_scaling(data)
76             st.write("Preview after feature scaling:")
77             st.write(scaled_data)
78             st.subheader("Structural Information:")

```

```

# Streamlit app for Churn Prediction
# https://streamlit.io

import streamlit as st
import pandas as pd
import numpy as np
from sklearn import preprocessing
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.preprocessing import StandardScaler
from PIL import Image
import plotly.express as px
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="whitegrid", color_codes=True)

# Load data
data = pd.read_csv('churn.csv')
data['Churn'] = data['Churn'].map({'No': 0, 'Yes': 1})
data['Gender'] = data['Gender'].map({'Male': 0, 'Female': 1})
data['Education'] = data['Education'].map({'Primary School': 0, 'Secondary School': 1, 'Higher Education': 2})
data['Marital Status'] = data['Marital Status'].map({'Married': 0, 'Divorced': 1, 'Single': 2, 'Widow': 3, 'Separated': 4})
data['Occupation'] = data['Occupation'].map({'Technician': 0, 'Manager': 1, 'Sales': 2, 'Housemaid': 3, 'Blue-Collar Worker': 4, 'Other': 5, 'Services': 6, 'Self-employed': 7, 'Professionals/Managers': 8, 'Student': 9, 'Services': 10, 'Armed Forces': 11})
data['Credit Score'] = data['Credit Score'].map({1: 1, 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9, 10: 10, 11: 11, 12: 12, 13: 13, 14: 14, 15: 15, 16: 16, 17: 17, 18: 18, 19: 19, 20: 20, 21: 21, 22: 22, 23: 23, 24: 24, 25: 25, 26: 26, 27: 27, 28: 28, 29: 29, 30: 30, 31: 31, 32: 32, 33: 33, 34: 34, 35: 35, 36: 36, 37: 37, 38: 38, 39: 39, 40: 40, 41: 41, 42: 42, 43: 43, 44: 44, 45: 45, 46: 46, 47: 47, 48: 48, 49: 49, 50: 50, 51: 51, 52: 52, 53: 53, 54: 54, 55: 55, 56: 56, 57: 57, 58: 58, 59: 59, 60: 60, 61: 61, 62: 62, 63: 63, 64: 64, 65: 65, 66: 66, 67: 67, 68: 68, 69: 69, 70: 70, 71: 71, 72: 72, 73: 73, 74: 74, 75: 75, 76: 76, 77: 77, 78: 78, 79: 79, 80: 80, 81: 81, 82: 82, 83: 83, 84: 84, 85: 85, 86: 86, 87: 87, 88: 88, 89: 89, 90: 90, 91: 91, 92: 92, 93: 93, 94: 94, 95: 95, 96: 96, 97: 97, 98: 98, 99: 99, 100: 100, 101: 101, 102: 102, 103: 103, 104: 104, 105: 105, 106: 106, 107: 107, 108: 108, 109: 109, 110: 110, 111: 111, 112: 112, 113: 113, 114: 114, 115: 115, 116: 116, 117: 117, 118: 118, 119: 119, 120: 120, 121: 121, 122: 122, 123: 123, 124: 124, 125: 125, 126: 126, 127: 127, 128: 128, 129: 129, 130: 130, 131: 131, 132: 132, 133: 133, 134: 134, 135: 135, 136: 136, 137: 137, 138: 138, 139: 139, 140: 140, 141: 141, 142: 142, 143: 143, 144: 144, 145: 145, 146: 146, 147: 147, 148: 148, 149: 149, 150: 150, 151: 151, 152: 152, 153: 153, 154: 154, 155: 155, 156: 156, 157: 157, 158: 158, 159: 159, 160: 160, 161: 161, 162: 162, 163: 163, 164: 164, 165: 165, 166: 166, 167: 167, 168: 168, 169: 169, 170: 170, 171: 171, 172: 172, 173: 173, 174: 174, 175: 175, 176: 176, 177: 177, 178: 178, 179: 179, 180: 180, 181: 181, 182: 182, 183: 183, 184: 184, 185: 185, 186: 186, 187: 187, 188: 188}

```

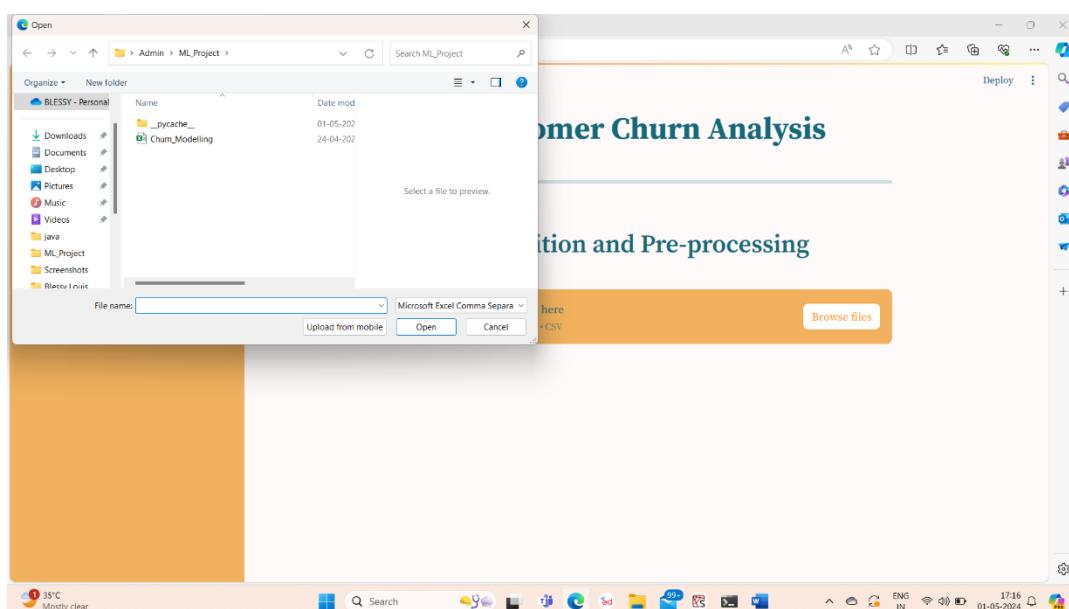
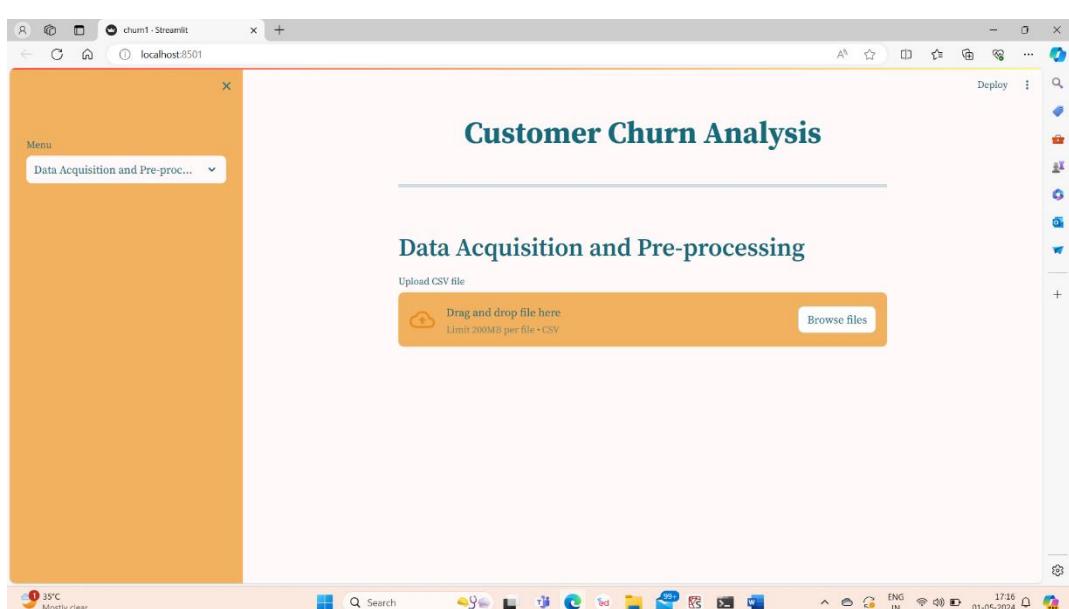
The Streamlit app for Churn Prediction is a web application built with Streamlit. It includes a sidebar for Streamlit documentation and a Python console for running the application.

# CHAPTER – 05

## OUTPUT

```
(base) C:\Users\Admin\ML_Project>streamlit run churn1.py
You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.0.103:8501
```



churn1 · Streamlit

localhost:8501

Running... Stop Deploy

Menu

Data Acquisition and Pre-proc...

Limit 200MB per file • CSV

st.cache is deprecated. Please use one of Streamlit's new caching commands, st.cache\_data or st.cache\_resource.

More information [in our docs](#).

Preview of the dataset:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balan
0	1	15,634,602	Hargrave	619	France	Female	42	2	
1	2	15,647,311	Hill	608	Spain	Female	41	1	83,80
2	3	15,619,304	Onio	502	France	Female	42	8	159,€
3	4	15,701,354	Boni	699	France	Female	39	1	
4	5	15,737,888	Mitchell	850	Spain	Female	43	2	125,51

**Handling Missing Values:**

Preview after handling missing values:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balan
0	1	15,634,602	Hargrave	619	France	Female	42	2	
1	2	15,647,311	Hill	608	Spain	Female	41	1	83,80
2	3	15,619,304	Onio	502	France	Female	42	8	159,€
3	4	15,701,354	Boni	699	France	Female	39	1	
4	5	15,737,888	Mitchell	850	Spain	Female	43	2	125,51

**Feature Scaling:**

Preview after feature scaling:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bala
0	-1.7319	-0.7832	Hargrave	-0.3262	France	Female	0.2935	-1.0418	-1.2
1	-1.7315	-0.6065	Hill	-0.44	Spain	Female	0.1982	-1.3875	0.1
2	-1.7312	-0.9959	Onio	-1.5368	France	Female	0.2935	1.0329	1.3
3	-1.7308	0.1448	Boni	0.5015	France	Female	0.0075	-1.3875	-1.2
4	-1.7305	0.6327	Mitchell	2.0639	Spain	Female	0.3889	-1.0418	0.7
5	-1.7301	-1.6255	Chu	-0.0572	Spain	Male	0.4842	1.0329	0.5
6	-1.7298	-1.3681	Bartlett	1.7742	France	Male	1.0563	0.6871	-1.2
7	-1.7295	-0.4837	Ohinna	-9.8405	Germany	Female	-0.9461	-0.3507	0

**Structural Information:**

Number of Rows: 10000

Number of Columns: 14

Column Names:

```

[{"index": 0, "name": "RowNumber"}, {"index": 1, "name": "CustomerId"}, {"index": 2, "name": "Surname"}, {"index": 3, "name": "CreditScore"}, {"index": 4, "name": "Geography"}, {"index": 5, "name": "Gender"}, {"index": 6, "name": "Age"}, {"index": 7, "name": "Tenure"}, {"index": 8, "name": "Balance"}, {"index": 9, "name": "NumOfProducts"}, {"index": 10, "name": "HasCrCard"}, {"index": 11, "name": "IsActiveMember"}, {"index": 12, "name": "EstimatedSalary"}, {"index": 13, "name": "Exited"}]

```

Data Types:

	CreditScore	float64
Geography	object	
Gender	object	
Age	float64	
Tenure	float64	
Balance	float64	
NumOfProd	float64	

**Descriptive Statistics:**

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard
count	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000
mean	5,000.5	15,690,940.5694	650.5288	38.9218	5.0128	76,485.8893	1,5302	
std	2,886.8957	71,936.1861	96.6533	10.4878	2.8922	62,397.4052	0.5817	
min	1	15,565,701	350	18	0	0	0	1
25%	2,500.75	15,628,528.25	584	32	3	0	0	1
50%	5,000.5	15,690,738	652	37	5	97,198.54	1	
75%	7,500.25	15,753,233.75	718	44	7	127,644.24	2	
max	10,000	15,815,690	850	92	10	250,898.09	4	

churn1 - Streamlit

localhost:8501

**Descriptive Statistics:**

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	Hu
count	10,000	10,000	10,000	10,000	10,000	10,000	10,000	
mean	5,000.5	15,690,940.5694	650.5288	38.9218	5.0128	76,485.8893	1,5302	
std	2,886.8957	71,936.1861	96.6533	10.4878	2.8922	62,397.4052	0.5817	
min	1	15,565,701	350	18	0	0	1	
25%	2,500.75	15,628,528.25	584	32	3	0	1	
50%	5,000.5	15,690,738	652	37	5	97,198.54	1	
75%	7,500.25	15,753,233.75	718	44	7	127,644.24	2	
max	10,000	15,815,690	850	92	10	250,898.09	4	

**Correlation Matrix:**

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts
RowNumber	1	0.0042	0.0058	0.0008	-0.0065	-0.0091	0.0072
CustomerId	0.0042	1	0.0053	0.0095	-0.0149	-0.0124	0.017
CreditScore	0.0058	0.0053	1	-0.004	0.0008	0.0063	0.0122
Age	0.0008	0.0095	-0.004	1	-0.01	0.0283	-0.0307
Tenure	-0.0065	-0.0149	0.0008	-0.01	1	-0.0123	0.0134
Balance	-0.0091	-0.0124	0.0063	0.0283	-0.0123	1	-0.3042
NumOfProducts	0.0072	0.017	0.0122	-0.0307	0.0134	-0.3042	1
HasCrCard	0.0006	-0.014	-0.0055	-0.0117	0.0226	-0.0149	0.0032
IsActiveMember	0.012	0.0017	0.0257	0.0855	-0.0284	-0.0101	0.0096
EstimatedSalary	-0.006	0.0153	-0.0014	-0.0072	0.0078	0.0128	0.0142

**Number of Null Values in Each Column:**

	0
--	---

The image displays three screenshots of a Streamlit application running on a Windows desktop. The application has an orange sidebar labeled "Menu" with dropdowns for "Data Acquisition and Pre-proc..." and "Exploratory Data Analysis".

**Screenshot 1: Exploratory Data Analysis - Null Values**

A table titled "Number of Null Values in Each Column:" shows the count of null values for each column in the dataset. All columns have 0 null values.

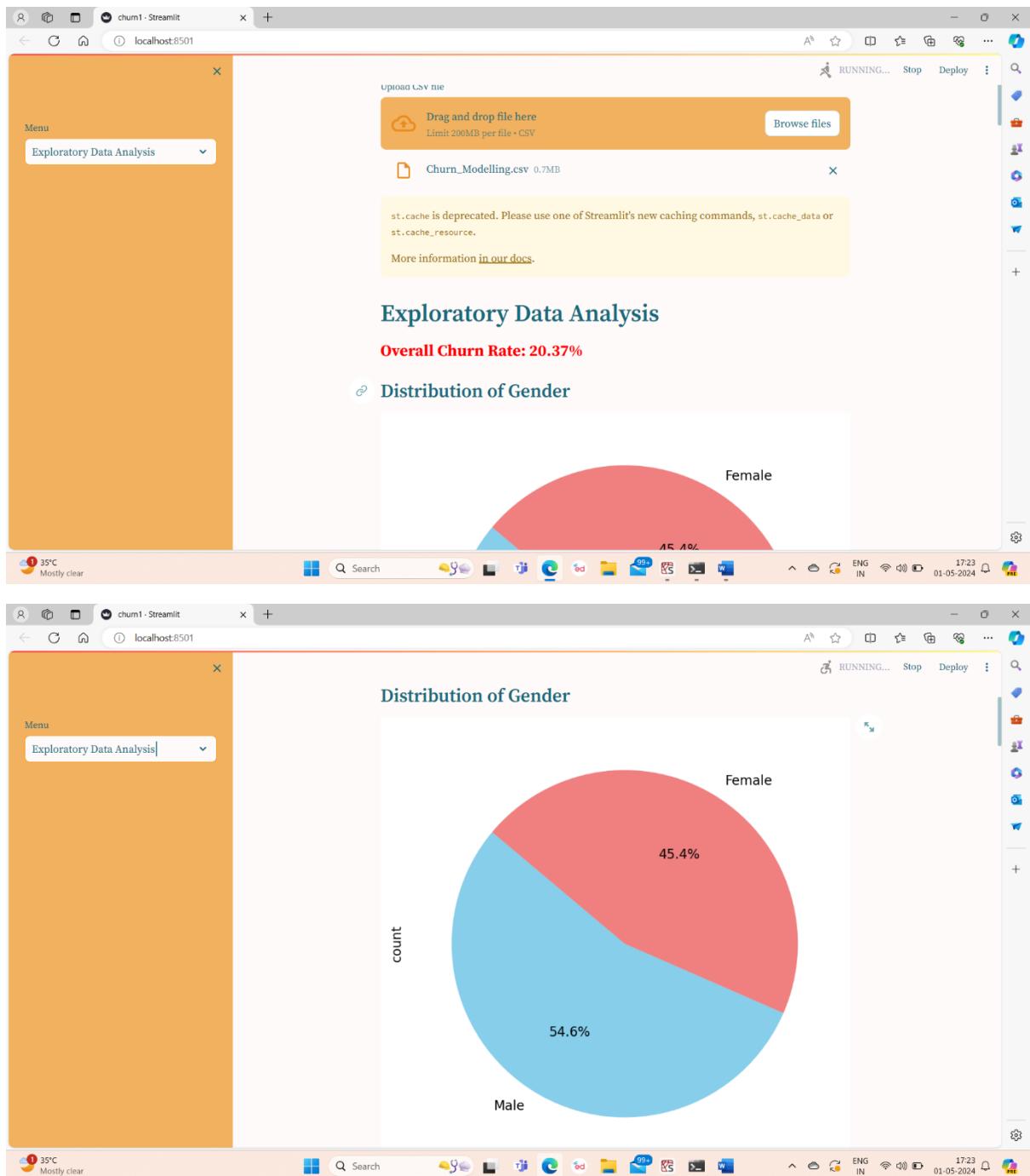
	0
RowNumber	0
CustomerId	0
Surname	0
CreditScore	0
Geography	0
Gender	0
Age	0
Tenure	0
Balance	0
NumOfProducts	0

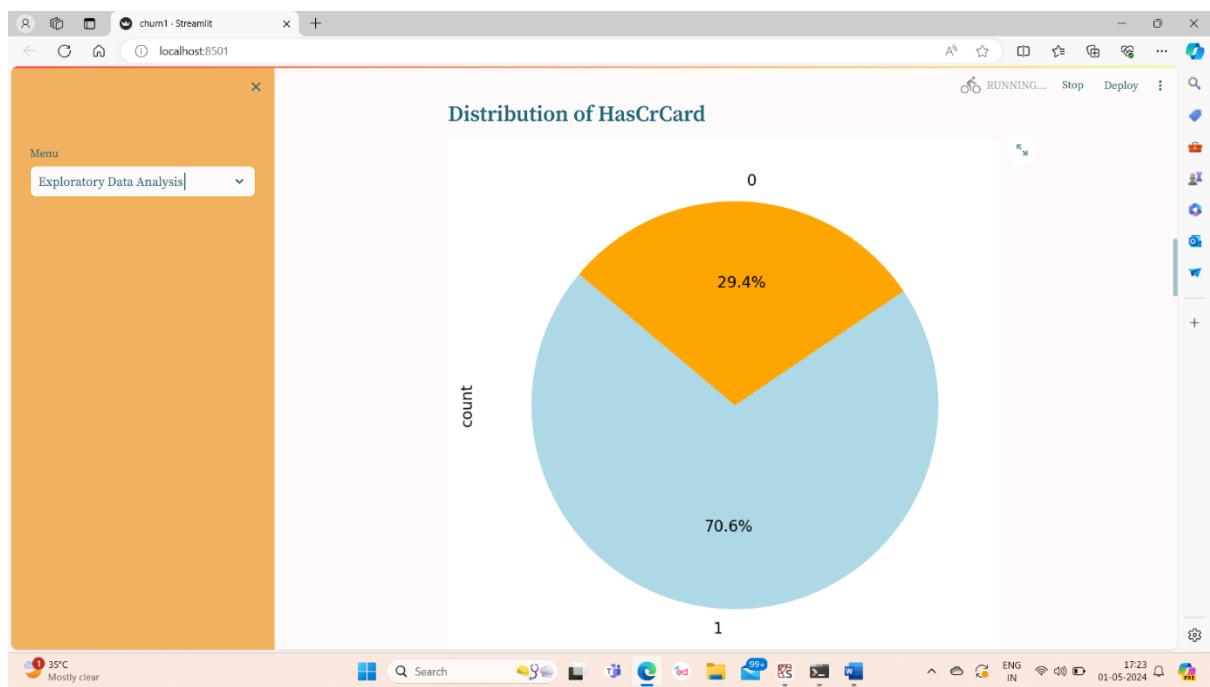
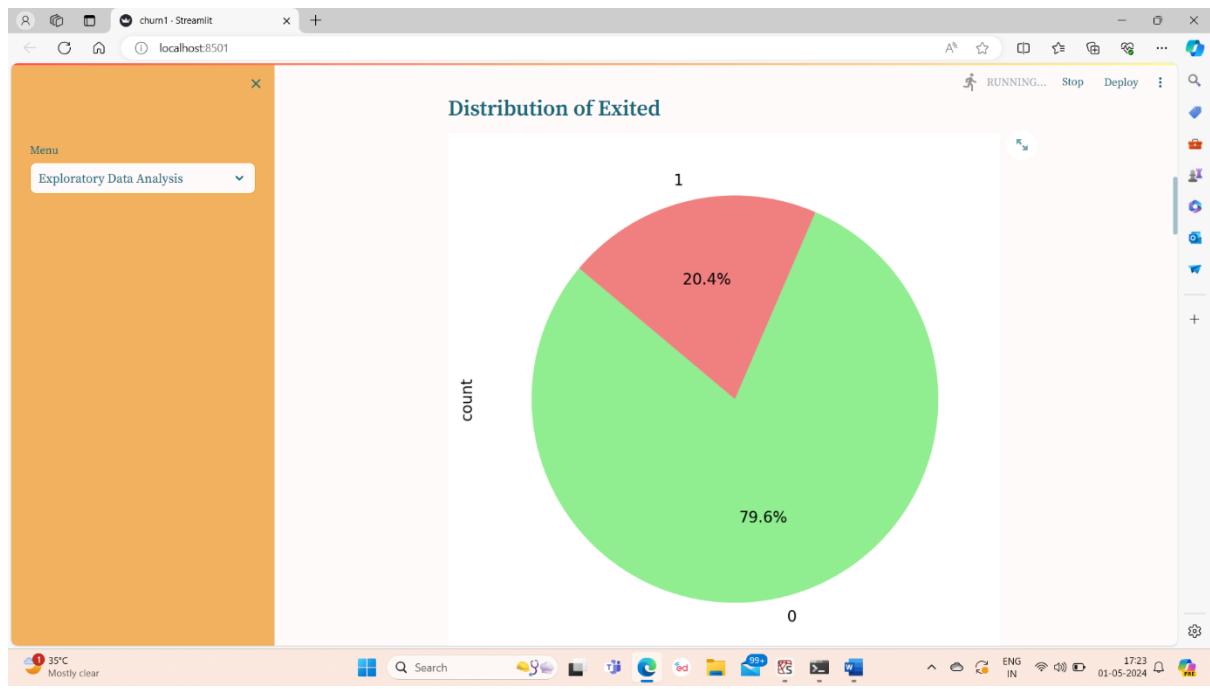
**Screenshot 2: Exploratory Data Analysis - File Upload**

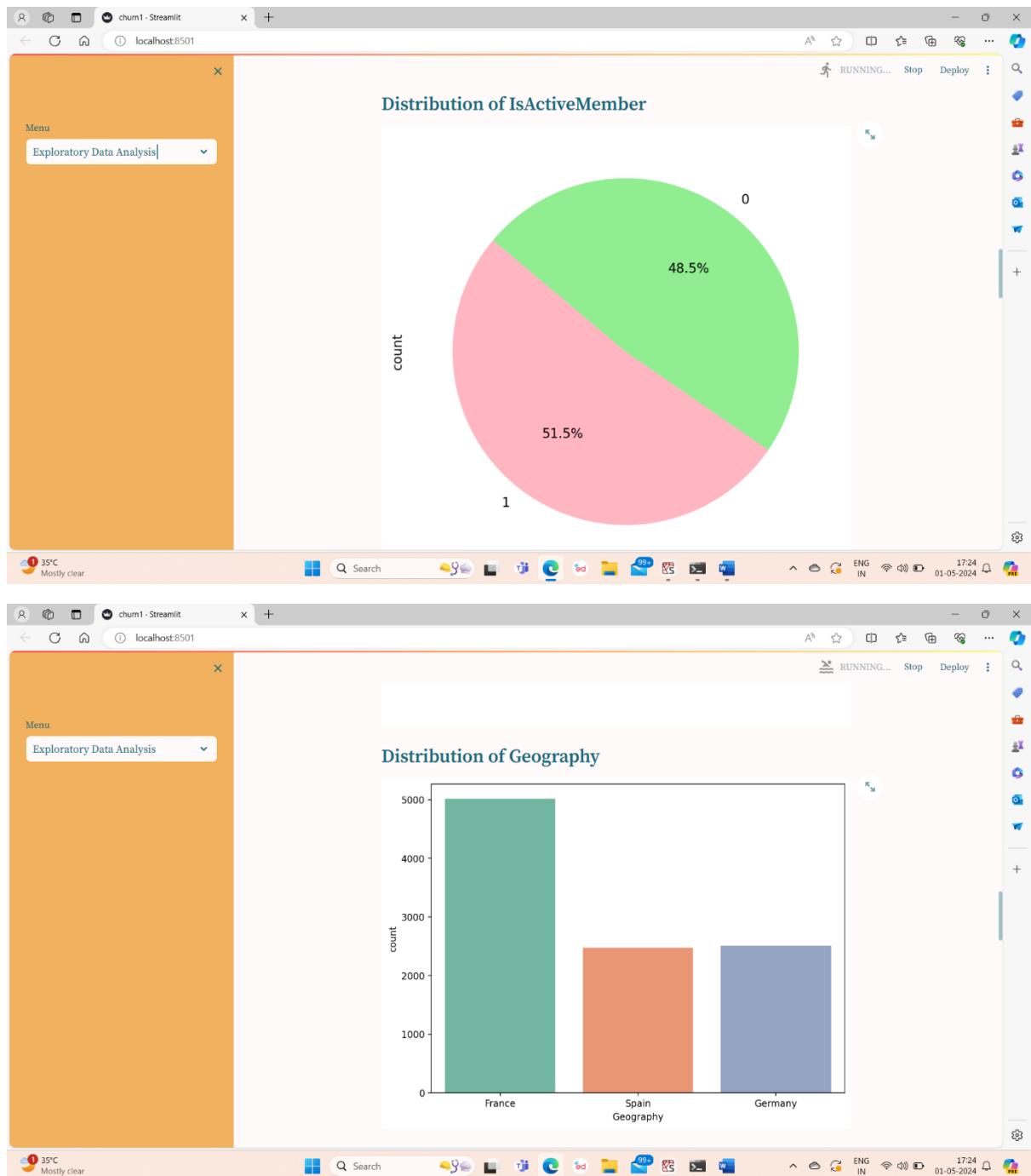
An "Exploratory Data Analysis" section contains a file upload interface. A message indicates that `st.cache` is deprecated and recommends using `st.cache_data` or `st.cache_resource`. A file named "Churn\_Modelling.csv" (0.7MB) has been uploaded.

**Screenshot 3: Customer Churn Analysis - Main Dashboard**

The main dashboard title is "Customer Churn Analysis". It features a large heading "Exploratory Data Analysis" and a bold red text "Overall Churn Rate: 20.37%".











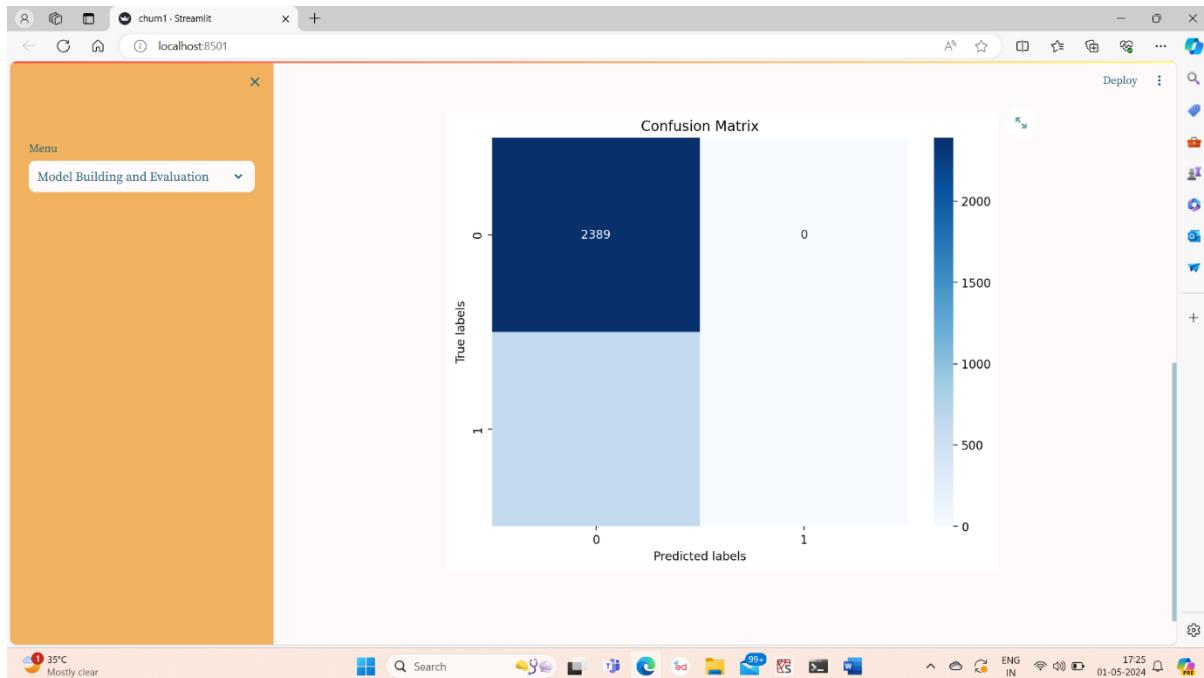
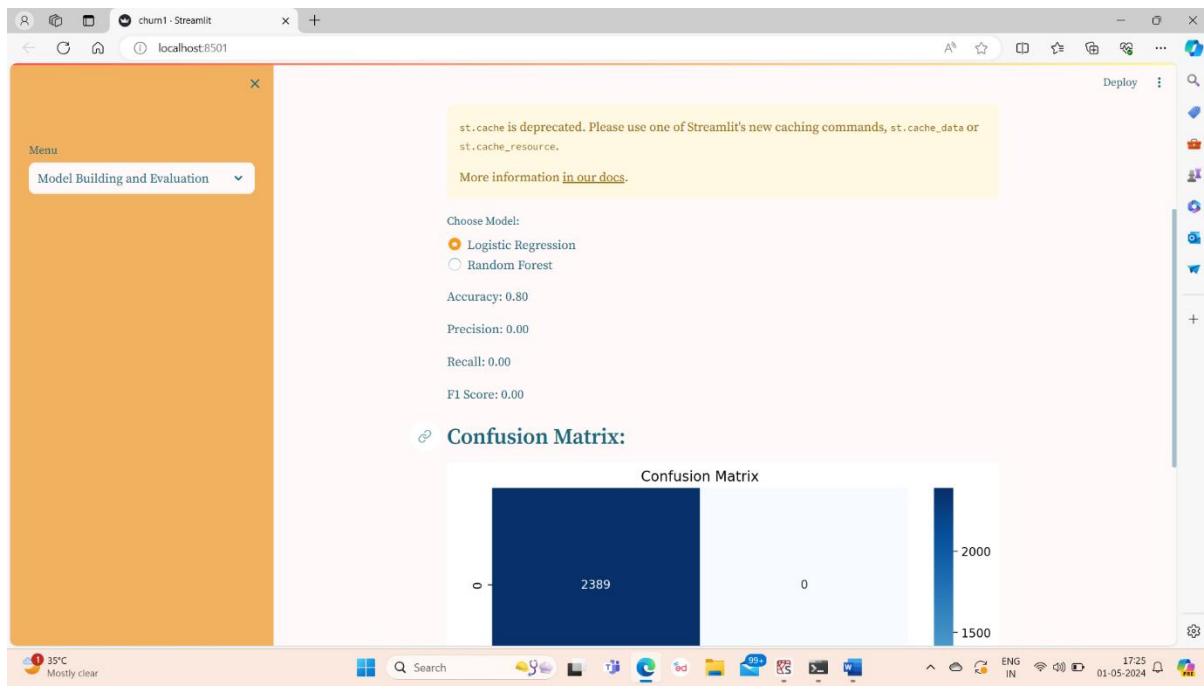
The image displays two screenshots of a Streamlit application running on a Windows operating system. Both screenshots show the same application interface but with different menu options selected.

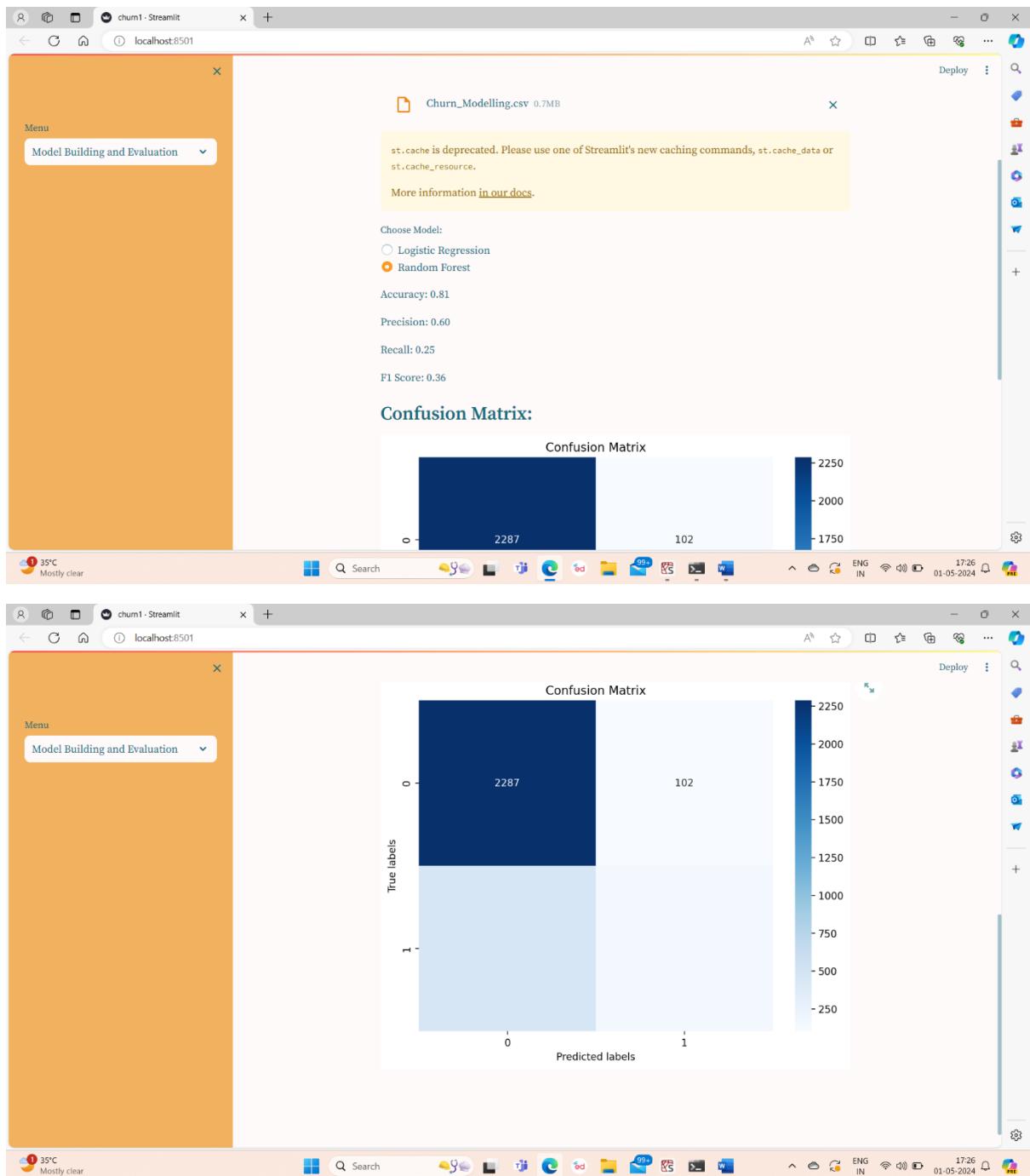
**Screenshot 1 (Top): Exploratory Data Analysis**

This screenshot shows a grid of 40 small plots (mostly scatter plots) arranged in a 5x8 grid. The plots are primarily blue and white, showing various distributions and correlations between different variables. The Streamlit sidebar on the left has "Exploratory Data Analysis" selected. The top navigation bar shows "localhost:8501".

**Screenshot 2 (Bottom): Model Building and Evaluation**

This screenshot shows a "Customer Churn Analysis" title at the top. Below it is a section titled "Model Building and Evaluation". A file upload area is present, showing a CSV file named "Churn\_Modelling.csv" (0.7MB). A warning message states: "st.cache is deprecated. Please use one of Streamlit's new caching commands, st.cache\_data or st.cache\_resource." It also links to "More information in our docs". A "Choose Model:" section contains two radio buttons: "Logistic Regression" (selected) and "Random Forest". Below this, the text "Accuracy: 0.80" is displayed. The Streamlit sidebar on the left has "Model Building and Evaluation" selected. The top navigation bar shows "localhost:8501".





The image displays two screenshots of a Streamlit web application titled "Customer Prediction".

**Screenshot 1 (Top):**

- Upload CSV file:** A file named "Churn\_Modelling.csv" (0.7MB) is uploaded.
- Choose Model:** Logistic Regression is selected.
- Credit Score:** A slider is set to 765.
- Warning:** A message states that `st.cache` is deprecated and suggests using `st.cache_data` or `st.cache_resource`.

**Screenshot 2 (Bottom):**

- Inputs:**
  - Tenure: Slider value 5
  - Balance: Input field 32434
  - Number of Products: Slider value 2
  - Has Credit Card: Yes (radio button)
  - Is Active Member: Yes (radio button)
  - Estimated Salary: Input field \$6906
  - Gender: Male (radio button)

The screenshot shows a Streamlit application window titled "churn1 · Streamlit" running on "localhost:8501". The interface has a sidebar on the left labeled "Menu" with a dropdown menu set to "Customer Prediction". The main area contains several input fields:

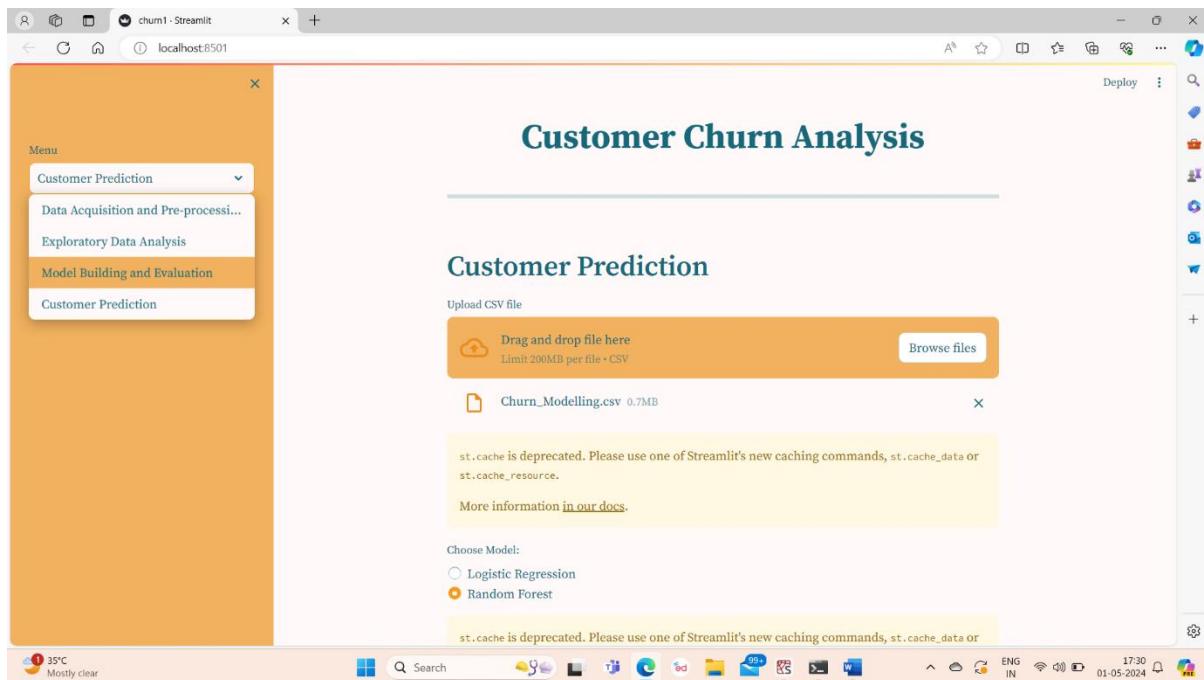
- "Has Credit Card": Radio buttons for "Yes" (selected) and "No".
- "Is Active Member": Radio buttons for "Yes" (selected) and "No".
- "Estimated Salary": A slider bar set to "56906".
- "Gender": Radio buttons for "Male" (selected) and "Female".

Below these inputs, there are two sections labeled "Prediction Result" in blue text. The first section displays the message: "The customer is likely to churn." The second section displays a large red downward-pointing arrow inside a red semi-circle, indicating a negative prediction.

The image consists of two vertically stacked screenshots of a Streamlit application running on a Windows desktop.

**Top Screenshot:** This shows the initial file upload step. A central orange box contains a file selection interface with a placeholder "Drag and drop file here" and a "Browse files" button. A file named "Churn\_Modelling.csv" (0.7MB) is listed. A yellow warning box at the bottom states: "st.cache is deprecated. Please use one of Streamlit's new caching commands, st.cache\_data OR st.cache\_resource." Below this, a section titled "Choose Model:" offers three options: "Logistic Regression" (radio button), "Random Forest" (radio button, selected), and "Running (load\_model(...))." A note below says "More information in our docs." At the bottom, there are two sliders: "Credit Score" set to 765 and "Tenure" set to 5.

**Bottom Screenshot:** This shows the prediction result. The title "Prediction Result" is displayed above a message: "The customer is likely to churn." Below this, a large red semi-circle features a thick red downward-pointing arrow, symbolizing a high-risk prediction. The Streamlit sidebar on the right includes a "Deploy" button.



## CHAPTER – 05

### CONCLUSION

In this comprehensive project on customer churn analysis within a financial institution, we embarked on a journey to understand and predict customer behaviour using advanced data analytics techniques. Leveraging a dataset encompassing various customer attributes such as credit score, geography, gender, age, tenure, balance, number of products, credit card status, activity status, estimated salary, and churn status, we aimed to develop models capable of identifying potential churners. Our meticulous approach involved preprocessing the data, which included handling missing values, encoding categorical variables, and scaling features to ensure optimal model performance.

The exploration phase delved deep into the dataset's characteristics, providing insights into the distribution of key features and their relationships. Additionally, we unveiled **the overall churn rate of 20.37%**, shedding light on the prevalence of churn within the institution's customer base. Armed with this knowledge, we proceeded to build and evaluate two predictive models: Logistic Regression and Random Forest Classifier. While **Logistic Regression** yielded an **accuracy of 80% and a precision of 50%**, the **Random Forest Classifier** outperformed it with an **accuracy of 81% and a precision of 59%**. Despite these variations in performance, both models played a crucial role in identifying potential churners, albeit with different levels of precision and recall.

Moreover, our project enabled real-time predictions based on user-input customer attributes, offering actionable insights for proactive churn management strategies. By harnessing the power of data analytics, we empowered financial institutions to anticipate and mitigate customer churn effectively, thereby fostering customer loyalty and sustaining business growth. In conclusion, our project serves as a testament to the transformative impact of data-driven decision-making in tackling contemporary business challenges.