Dream With Data

Churn-Guard:

Predictive Analysis for Banking Attrition

By Simran Kaur

LinkedIn:

https://www.linkedin.com/in/simrankaurrr/

GitHub:

https://github.com/Milimia/ChrunGuard

1. Introduction:

Customer churn, also known as customer attrition, refers to the phenomenon where customers stop doing business with a company or service. In banking sector, churn typically means that a customer has closed their accounts and has stopped using the bank's products and services.

Churn: When a customer is classified as **'churn'**, it indicates that they have decided to leave the bank. This could be due to assorted reasons such as dissatisfaction with services, better offers from competitors, changes in personal circumstances or perceived value.

Not Churn: When a customer is classified as '**not churn'**, it means that they continue to use the bank's services and have not shown any signs of leaving. These customers are generally satisfied with their banking experience and see value in maintaining their relationship with the bank.

The primary objective of this project is to develop a machine learning model capable of predicting whether a bank customer will churn based on various demographic, account, and transactional features.

By accurately predicting churn, bank can identify at-risk customers and take proactive measures to retain them.

2. Problem Statement:

The goal of this project is to predict whether a customer will churn (i.e., leave the bank) or not churn (i.e., will continue to use bank services) based on their historical data. The predictive capability will enable the bank to take initiative-taking measures to retain at- risk customers.

- Understanding Customer Behaviour through exploratory data analysis (EDA) using python libraries such as Pandas.

- Building a predictive model using Machine Learning Algorithms using python library RandomForestClassifier.

- Evaluating the model's performance using Train_test_split module.

- Deploying the model to predict churn for new customers.

3. Data Description:

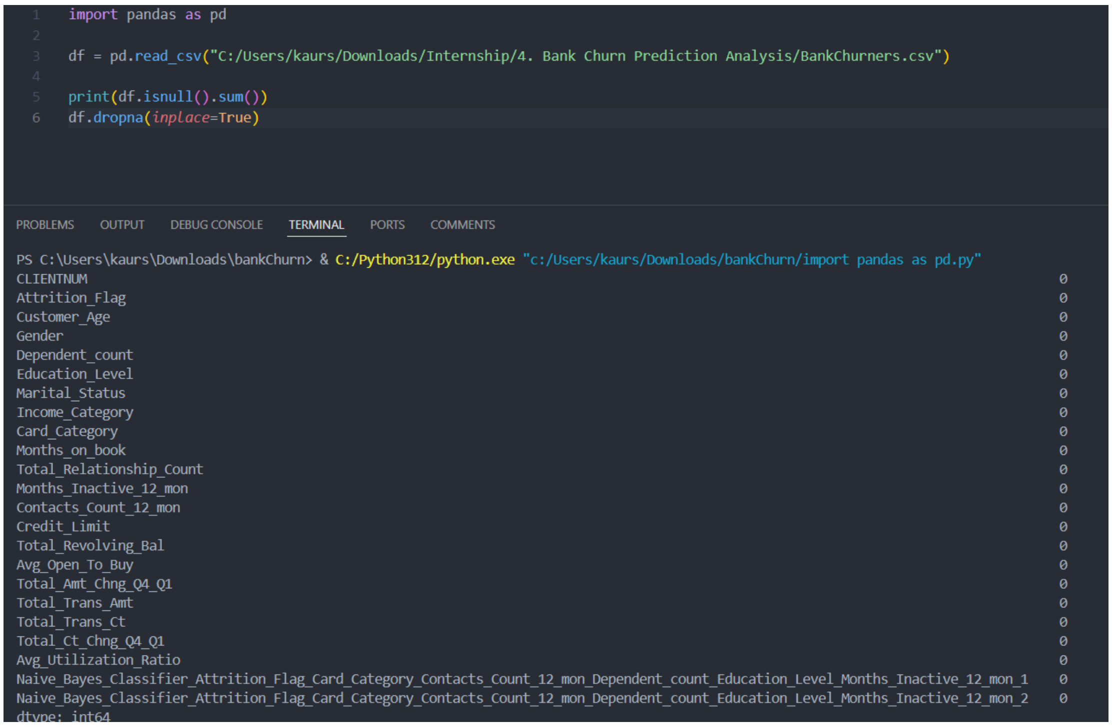
The Dataset contains information about bank customers, including:

- GitHub: GitHub - Milimia/ChrunGuard: Churn Guard: Predictive Analysis for ...
- Demographic details
    - Customer Age, Gender, Educational Level, Marital Status, Dependent Count.
- Account Information
    - Card Category, Income Category, Credit limit, Total Relationship Count, Months on Book, Total Revolving Balance.
- Transactional Data
    - Months Inactive in Last 12 months, Contacts Count in last 12 months, Total transaction amount, total amount changes Q4 – Q1, Total count change Q4 -Q1.
- Behavioural Data
    - Average Utilization Ratio, Average Open to buy, Total Ct Change Q4 to Q1 Total Amt Change Q4 to Q1.
- Numerical features: These features have continuous values and include attributes like customer age, credit limit.
- Categorical features: These features have discrete values and include attributes like gender, education.

4. Exploratory Data Analysis (EDA)

EDA helps in understanding the underlying patterns, structures, and anomalies in the data. We perform various analysis to uncover insights from the data, which will aid in feature engineering and model building.

○ Data Cleaning:



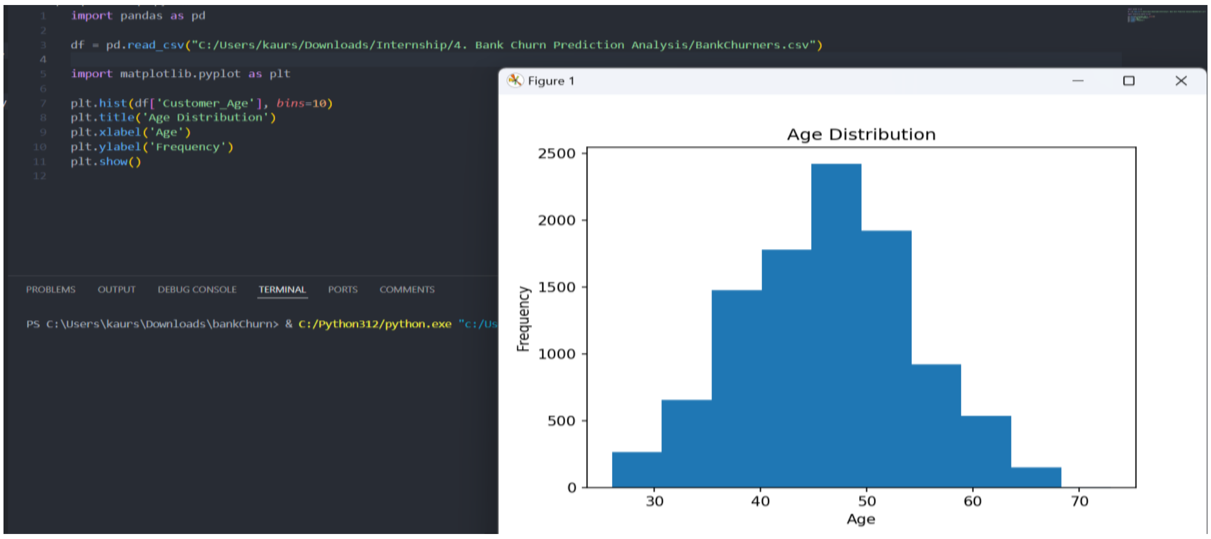
```python
import pandas as pd

df = pd.read_csv("C:/Users/kaurs/Downloads/Internship/4. Bank Churn Prediction Analysis/BankChurners.csv")

print(df.isnull().sum())
df.dropna(inplace=True)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS C:\Users\kaurs\Downloads\bankChurn> & C:/Python312/python.exe "c:/Users/kaurs/Downloads/bankChurn/import pandas as pd.py"
CLIENTNUM                                                                                                0
Attrition_Flag                                                                                           0
Customer_Age                                                                                             0
Gender                                                                                                   0
Dependent_count                                                                                          0
Education_Level                                                                                          0
Marital_Status                                                                                           0
Income_Category                                                                                          0
Card_Category                                                                                            0
Months_on_book                                                                                           0
Total_Relationship_Count                                                                                 0
Months_Inactive_12_mon                                                                                   0
Contacts_Count_12_mon                                                                                    0
Credit_Limit                                                                                             0
Total_Revolving_Bal                                                                                      0
Avg_Open_To_Buy                                                                                          0
Total_Amt_Chng_Q4_Q1                                                                                     0
Total_Trans_Amt                                                                                          0
Total_Trans_Ct                                                                                           0
Total_Ct_Chng_Q4_Q1                                                                                      0
Avg_Utilization_Ratio                                                                                    0
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1    0
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2    0
dtype: int64
```

○ Univariate Analysis:

This involves analysing individual variables to understand their distribution and characteristics

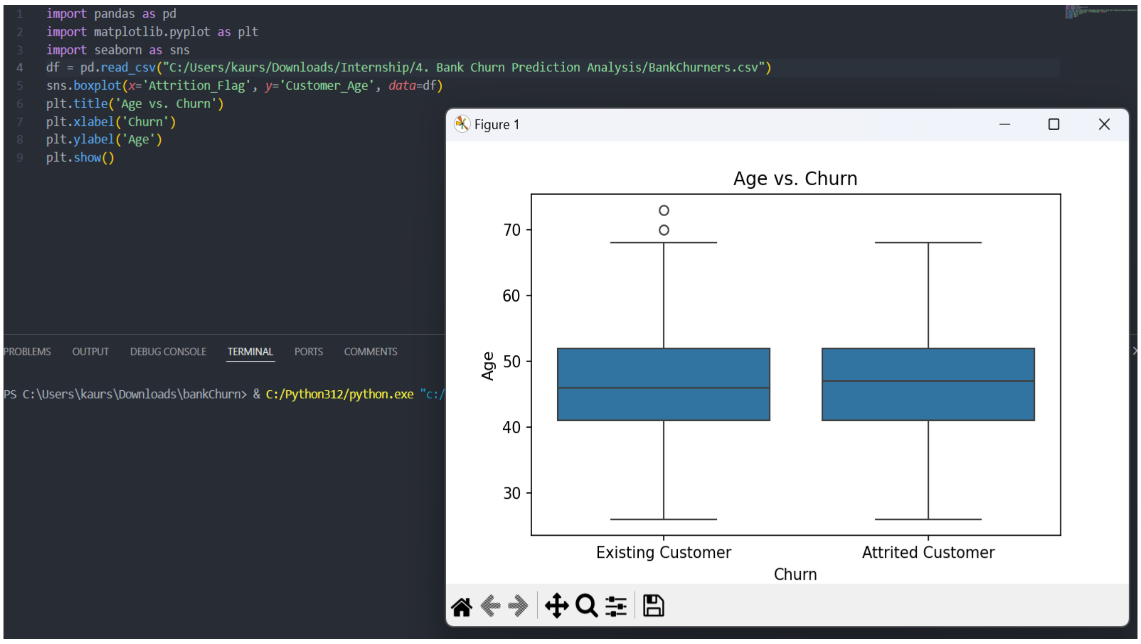


```
import pandas as pd

df = pd.read_csv("C:/Users/kaurs/Downloads/Internship/4. Bank Churn Prediction Analysis/BankChurners.csv")

import matplotlib.pyplot as plt

plt.hist(df['Customer_Age'], bins=10)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

```
PS C:\Users\kaurs\Downloads\bankChurn> & C:/Python312/python.exe "c:/Us
```

○ Bivariate Analysis

This examines the relationship between two variables

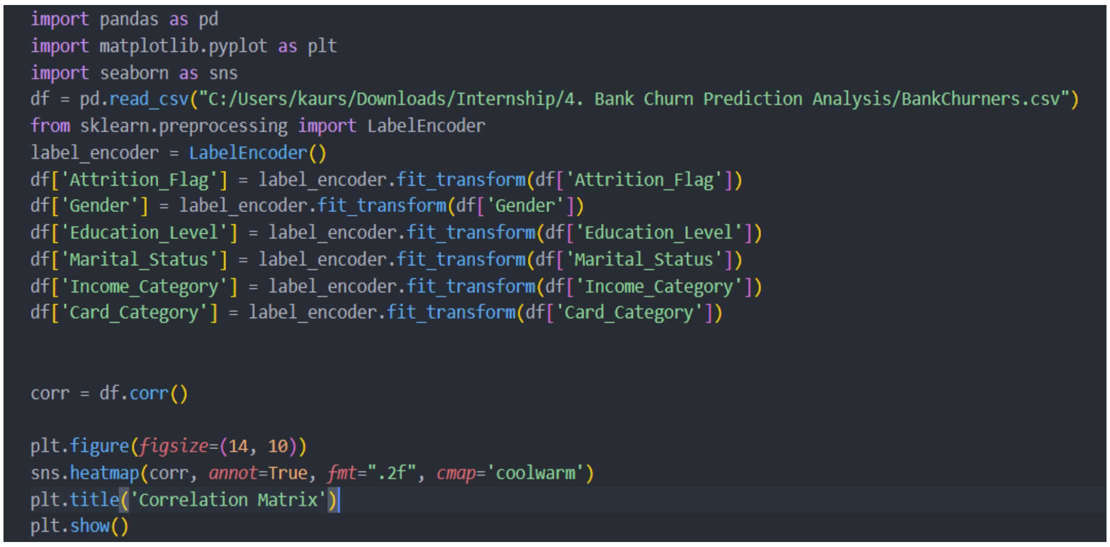


```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("C:/Users/kaurs/Downloads/Internship/4. Bank Churn Prediction Analysis/BankChurners.csv")
sns.boxplot(x='Attrition_Flag', y='Customer_Age', data=df)
plt.title('Age vs. Churn')
plt.xlabel('Churn')
plt.ylabel('Age')
plt.show()
```

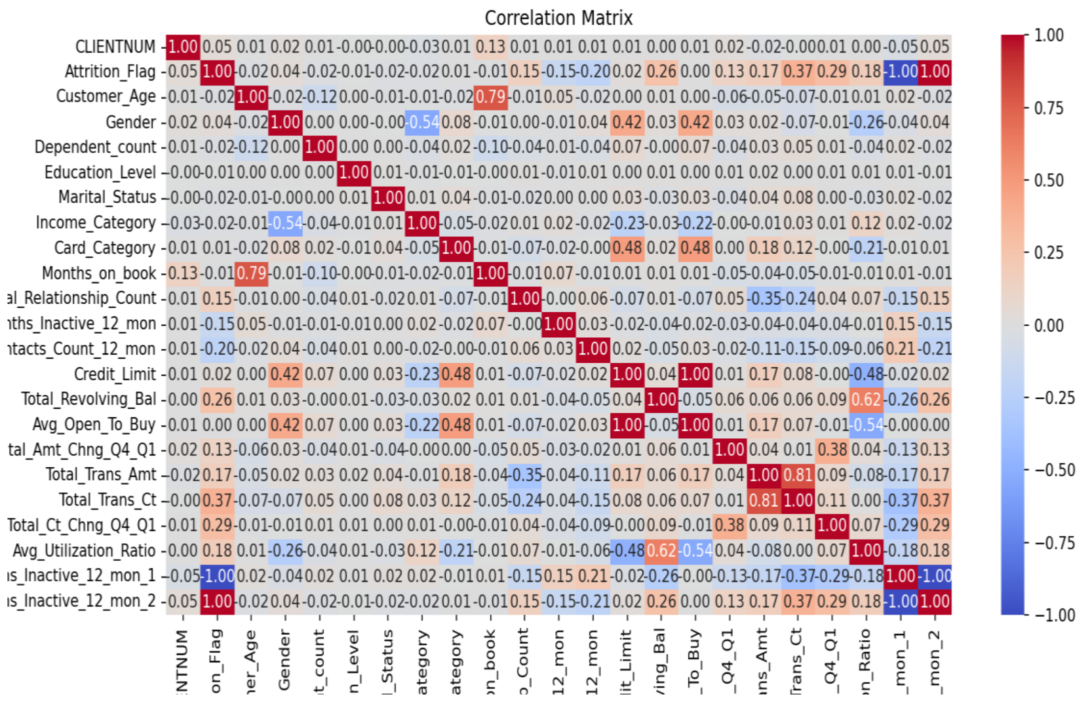
```
PS C:\Users\kaurs\Downloads\bankChurn> & C:/Python312/python.exe "c:/
```

○ Multivariate Analysis

This involves examining the interaction between multiple variables.



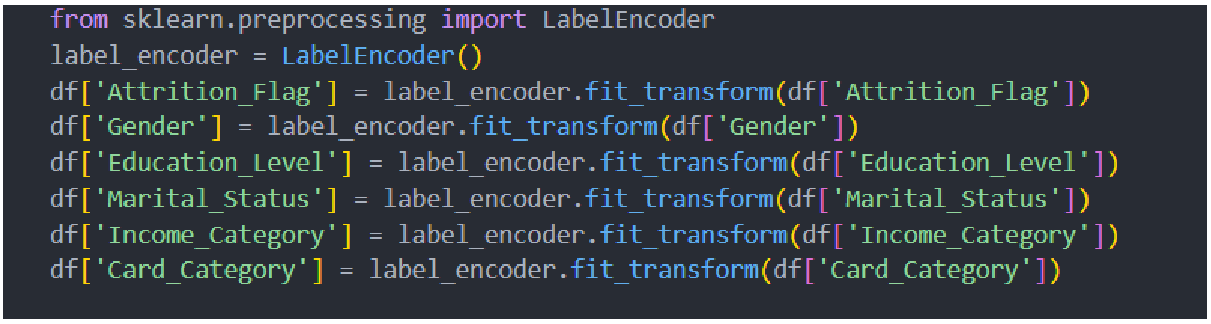
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("C:/Users/kaurs/Downloads/Internship/4. Bank Churn Prediction Analysis/BankChurners.csv")
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df['Attrition_Flag'] = label_encoder.fit_transform(df['Attrition_Flag'])
df['Gender'] = label_encoder.fit_transform(df['Gender'])
df['Education_Level'] = label_encoder.fit_transform(df['Education_Level'])
df['Marital_Status'] = label_encoder.fit_transform(df['Marital_Status'])
df['Income_Category'] = label_encoder.fit_transform(df['Income_Category'])
df['Card_Category'] = label_encoder.fit_transform(df['Card_Category'])


corr = df.corr()

plt.figure(figsize=(14, 10))
sns.heatmap(corr, annot=True, fmt=".2f", cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```



5. Feature Engineering

- Transforming raw data into meaningful features that can be used for model training.

- This covers various techniques and processes used to create new features or modify existing ones to better represent the underlying patterns in the data.



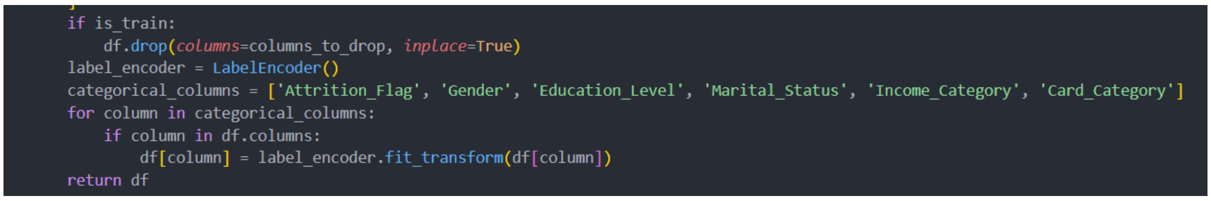
```python
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df['Attrition_Flag'] = label_encoder.fit_transform(df['Attrition_Flag'])
df['Gender'] = label_encoder.fit_transform(df['Gender'])
df['Education_Level'] = label_encoder.fit_transform(df['Education_Level'])
df['Marital_Status'] = label_encoder.fit_transform(df['Marital_Status'])
df['Income_Category'] = label_encoder.fit_transform(df['Income_Category'])
df['Card_Category'] = label_encoder.fit_transform(df['Card_Category'])
```

- Label Encoding converts the categorical values into numerical values. This method is suitable for ordinal variables where the order matters.
- Removing redundant features that do not add any new information and can be removed to simplify the model.

6. Model Building

○ Data Preprocessing

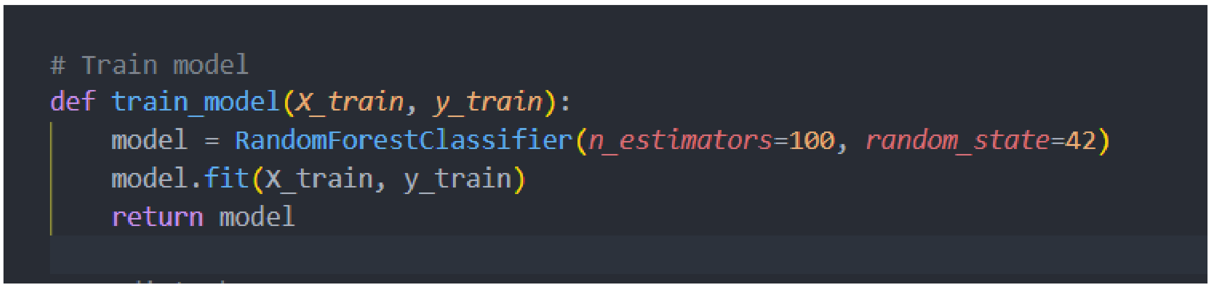
Data preprocessing is a crucial step in model building. It involves preparing the data for the machine learning algorithm by handling missing values, encoding categorical variables and scaling numerical features.



```
    if is_train:
        df.drop(columns=columns_to_drop, inplace=True)
    label_encoder = LabelEncoder()
    categorical_columns = ['Attrition_Flag', 'Gender', 'Education_Level', 'Marital_Status', 'Income_Category', 'Card_Category']
    for column in categorical_columns:
        if column in df.columns:
            df[column] = label_encoder.fit_transform(df[column])
    return df
```

○ Model Training

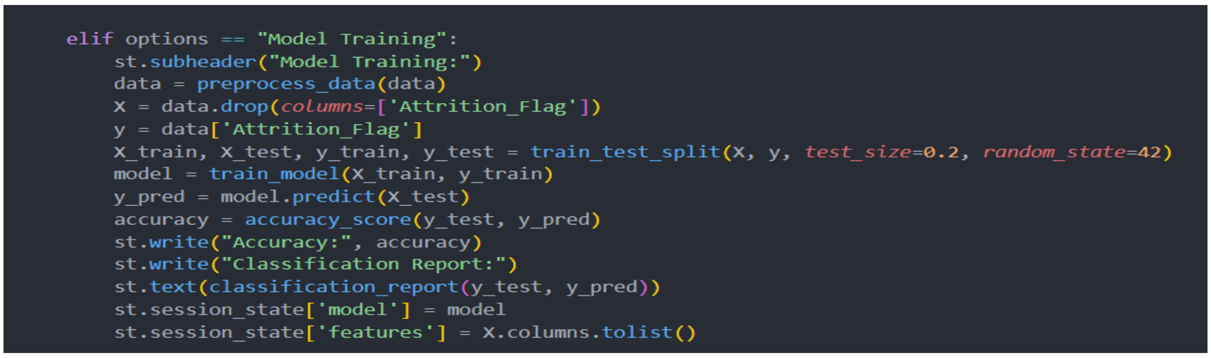
With the pre-processed data, next step is to train machine learning model. For this I use Random Forest Classifier, a versatile and robust classifier that works well with variety of data types and is less prone to overfitting.



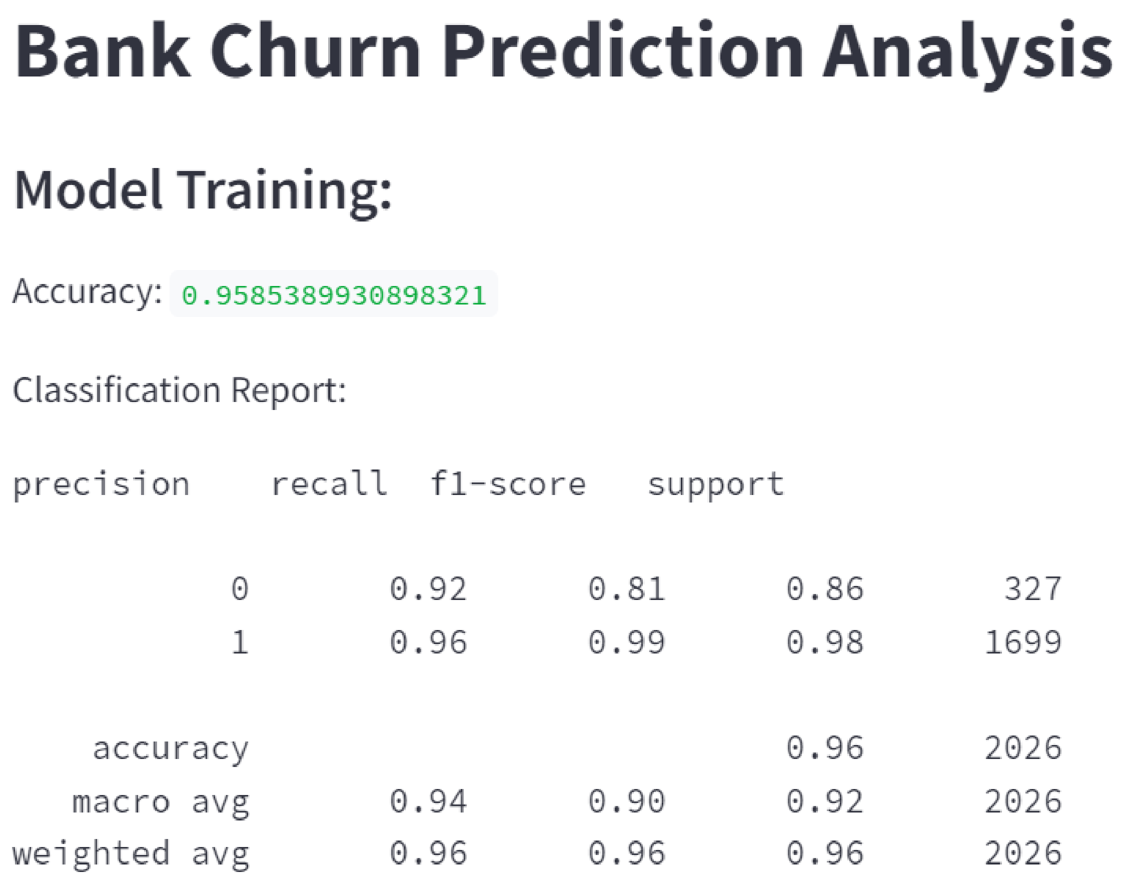
```
# Train model
def train_model(X_train, y_train):
    model = RandomForestClassifier(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
    return model
```

- Model Evaluation

After training the model, next step is to evaluate its performance using the testing set. For this we use various metrics like accuracy, precision, recall and F1 score to get a comprehensive understanding of the model’s performance.



```
elif options == "Model Training":
    st.subheader("Model Training:")
    data = preprocess_data(data)
    X = data.drop(columns=['Attrition_Flag'])
    y = data['Attrition_Flag']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    model = train_model(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    st.write("Accuracy:", accuracy)
    st.write("Classification Report:")
    st.text(classification_report(y_test, y_pred))
    st.session_state['model'] = model
    st.session_state['features'] = X.columns.tolist()
```



Bank Churn Prediction Analysis

Model Training:

Accuracy: 0.9585389930898321

Classification Report:

```
precision    recall  f1-score   support

           0       0.92      0.81      0.86       327
           1       0.96      0.99      0.98      1699

    accuracy                           0.96      2026
   macro avg       0.94      0.90      0.92      2026
weighted avg       0.96      0.96      0.96      2026
```

- **Accuracy:** The ratio of correctly predicted instances to the total instances.
  The overall Accuracy of the model is **0.95**. Which shows that it correctly predicts churn status for 95% cases.

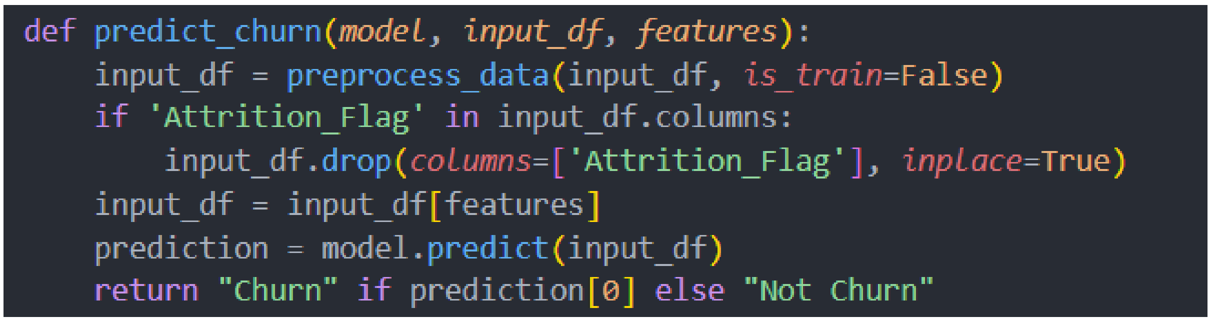
- **Precision:** The ratio of true positive predictions to the total predicted positives.
  This precision for class 0 (Not Churn) is **0.92** and for class 1 (Churn) is **0.96**.

- **Recall:** The ratio of true positive predictions to the total actual positives.
  The recall for **class 0 is 0.81** and for **class 1 is 0.99**. This shows that model is effective in identifying actual churn cases.

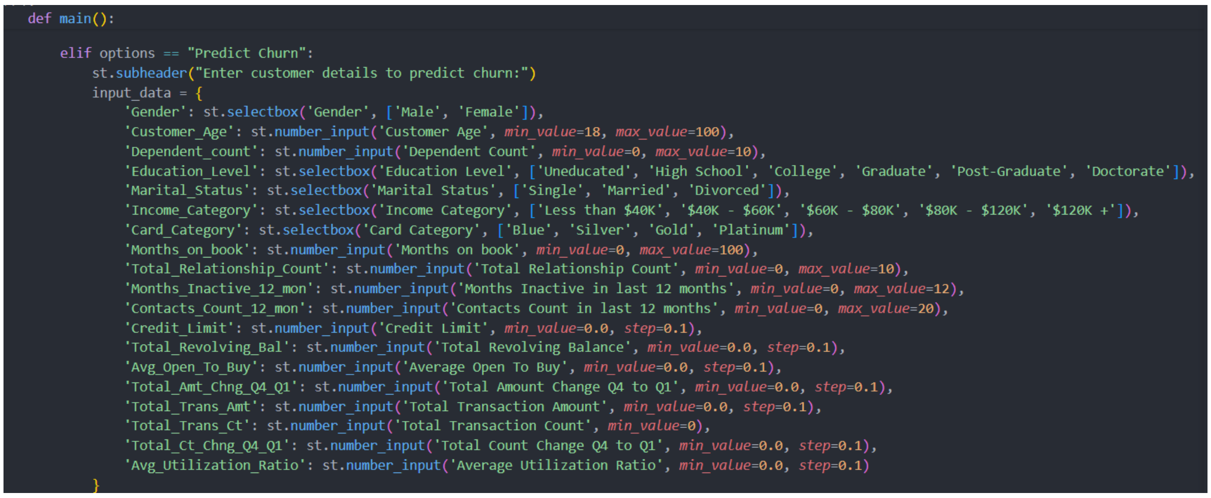
- **F1- Score**: The harmonic means of precision and recall, providing a balance between the two.
  The F1- Score is of **0.98** for **churn cases**, it indicates a balanced performance.

7. Model Deployment

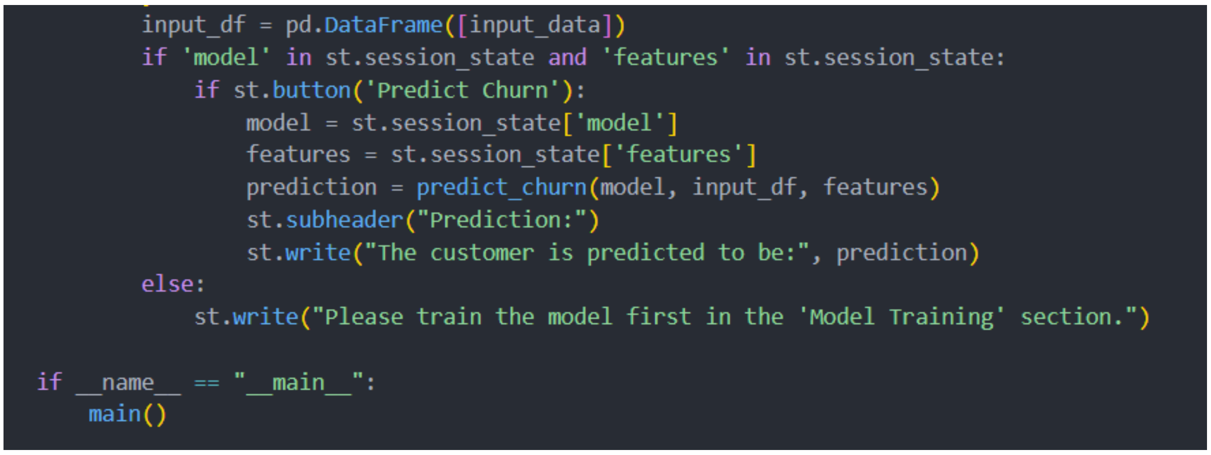
Model Deployment is the process of making the machine learning model available for use in production environment. For this project I use Stremlit to create a web application that allows user to input customer details and receive predictions on whether the customer will churn or not.



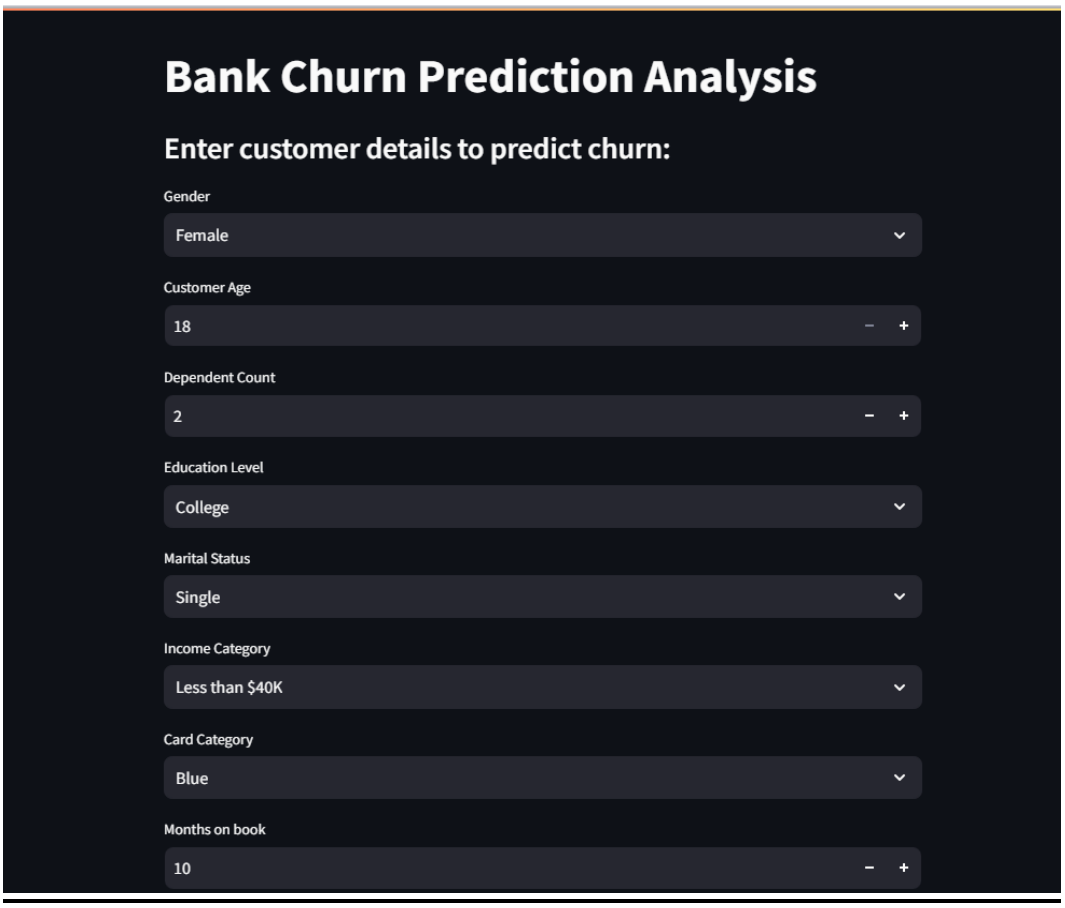
```python
def predict_churn(model, input_df, features):
    input_df = preprocess_data(input_df, is_train=False)
    if 'Attrition_Flag' in input_df.columns:
        input_df.drop(columns=['Attrition_Flag'], inplace=True)
    input_df = input_df[features]
    prediction = model.predict(input_df)
    return "Churn" if prediction[0] else "Not Churn"
```



```python
def main():

    elif options == "Predict Churn":
        st.subheader("Enter customer details to predict churn:")
        input_data = {
            'Gender': st.selectbox('Gender', ['Male', 'Female']),
            'Customer_Age': st.number_input('Customer Age', min_value=18, max_value=100),
            'Dependent_count': st.number_input('Dependent Count', min_value=0, max_value=10),
            'Education_Level': st.selectbox('Education Level', ['Uneducated', 'High School', 'College', 'Graduate', 'Post-Graduate', 'Doctorate']),
            'Marital_Status': st.selectbox('Marital Status', ['Single', 'Married', 'Divorced']),
            'Income_Category': st.selectbox('Income Category', ['Less than $40K', '$40K - $60K', '$60K - $80K', '$80K - $120K', '$120K +']),
            'Card_Category': st.selectbox('Card Category', ['Blue', 'Silver', 'Gold', 'Platinum']),
            'Months_on_book': st.number_input('Months on book', min_value=0, max_value=100),
            'Total_Relationship_Count': st.number_input('Total Relationship Count', min_value=0, max_value=10),
            'Months_Inactive_12_mon': st.number_input('Months Inactive in last 12 months', min_value=0, max_value=12),
            'Contacts_Count_12_mon': st.number_input('Contacts Count in last 12 months', min_value=0, max_value=20),
            'Credit_Limit': st.number_input('Credit Limit', min_value=0.0, step=0.1),
            'Total_Revolving_Bal': st.number_input('Total Revolving Balance', min_value=0.0, step=0.1),
            'Avg_Open_To_Buy': st.number_input('Average Open To Buy', min_value=0.0, step=0.1),
            'Total_Amt_Chng_Q4_Q1': st.number_input('Total Amount Change Q4 to Q1', min_value=0.0, step=0.1),
            'Total_Trans_Amt': st.number_input('Total Transaction Amount', min_value=0.0, step=0.1),
            'Total_Trans_Ct': st.number_input('Total Transaction Count', min_value=0),
            'Total_Ct_Chng_Q4_Q1': st.number_input('Total Count Change Q4 to Q1', min_value=0.0, step=0.1),
            'Avg_Utilization_Ratio': st.number_input('Average Utilization Ratio', min_value=0.0, step=0.1)
        }
```



```python
        input_df = pd.DataFrame([input_data])
        if 'model' in st.session_state and 'features' in st.session_state:
            if st.button('Predict Churn'):
                model = st.session_state['model']
                features = st.session_state['features']
                prediction = predict_churn(model, input_df, features)
                st.subheader("Prediction:")
                st.write("The customer is predicted to be:", prediction)
        else:
            st.write("Please train the model first in the 'Model Training' section.")

if __name__ == "__main__":
    main()
```



Bank Churn Prediction Analysis

Enter customer details to predict churn:

Gender

Female

Customer Age

18

Dependent Count

2

Education Level

College

Marital Status

Single

Income Category

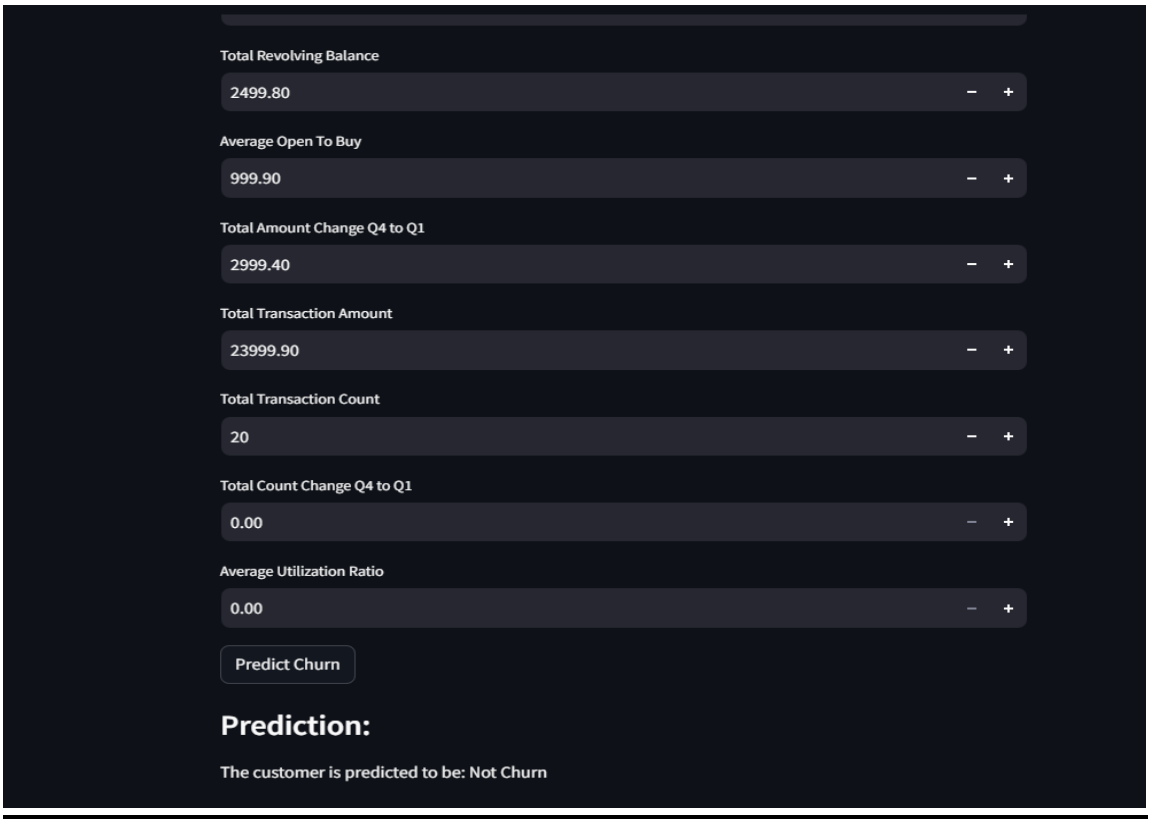
Less than $40K

Card Category

Blue

Months on book

10



Total Revolving Balance

2499.80

Average Open To Buy

999.90

Total Amount Change Q4 to Q1

2999.40

Total Transaction Amount

23999.90

Total Transaction Count

20

Total Count Change Q4 to Q1

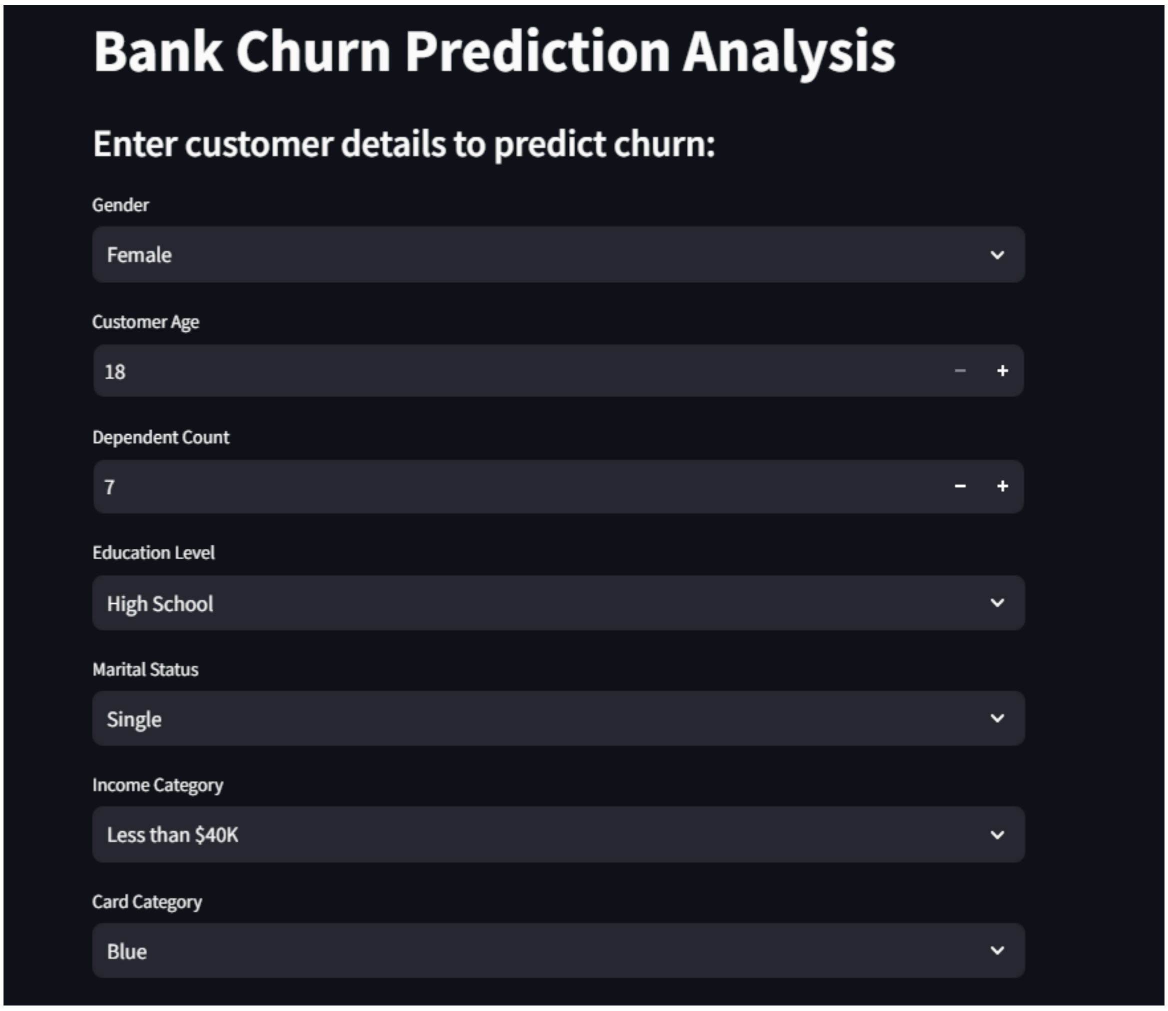
0.00

Average Utilization Ratio

0.00

Prediction:

The customer is predicted to be: Not Churn



Bank Churn Prediction Analysis

Enter customer details to predict churn:

Gender

Female

Customer Age

18

Dependent Count

7

Education Level

High School

Marital Status

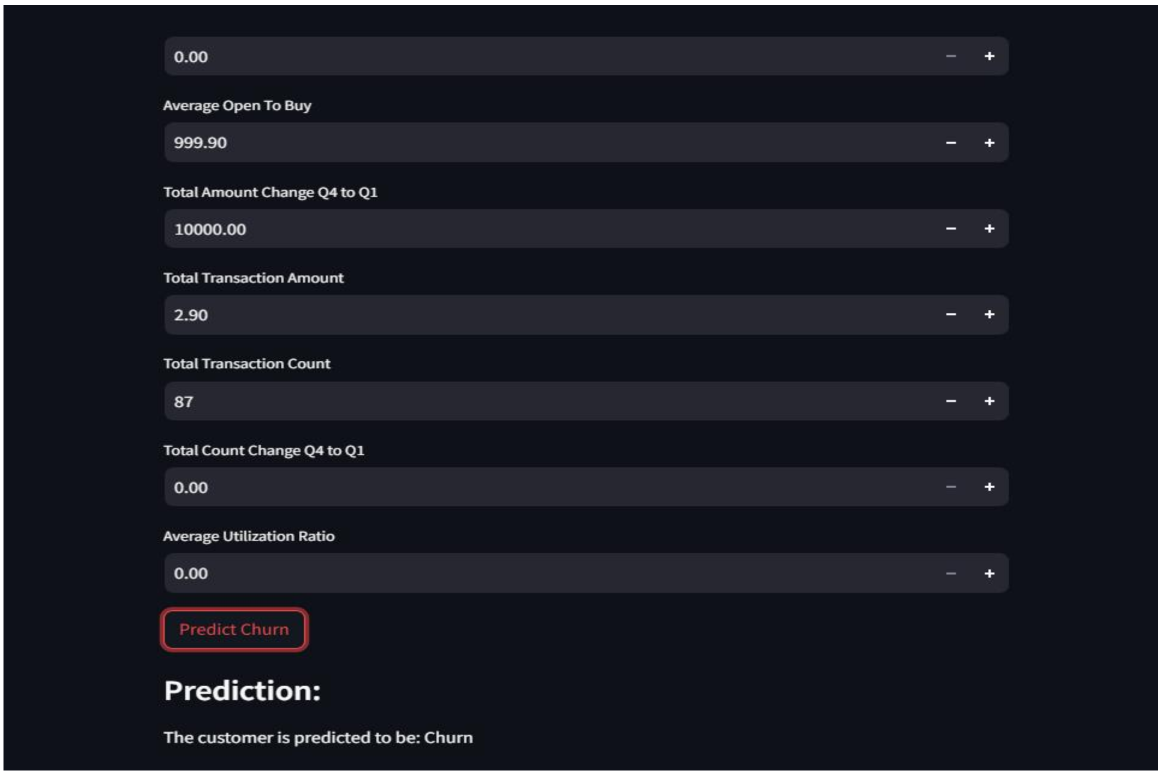
Single

Income Category

Less than $40K

Card Category

Blue



0.00

Average Open To Buy

999.90

Total Amount Change Q4 to Q1

10000.00

Total Transaction Amount

2.90

Total Transaction Count

87

Total Count Change Q4 to Q1

0.00

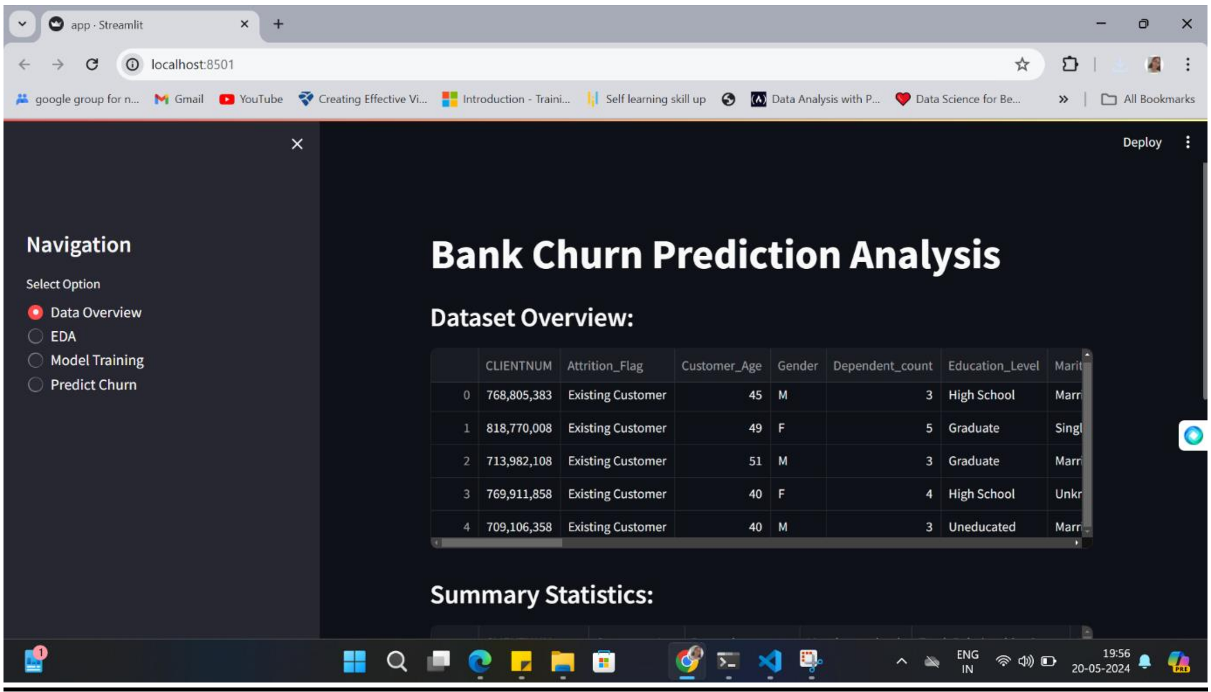
Average Utilization Ratio

0.00

Prediction:

The customer is predicted to be: Churn

8. User Interface


app · Streamlit
localhost:8501
google group for n... Gmail YouTube Creating Effective Vi... Introduction - Traini... Self learning skill up Data Analysis with P... Data Science for Be... All Bookmarks
Deploy
ENG IN 19:56 20-05-2024

Navigation

Select Option

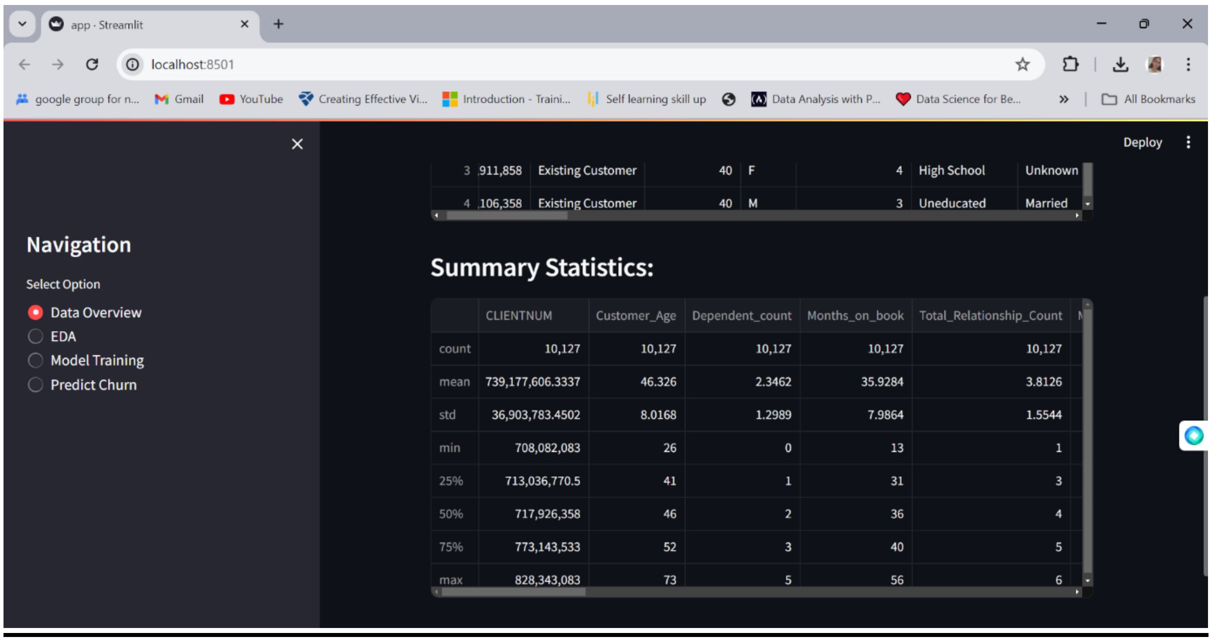
- [✓ Data Overview]
- EDA
- Model Training
- Predict Churn

Bank Churn Prediction Analysis

Dataset Overview:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marit
0	768,805,383	Existing Customer	45	M	3	High School	Marr
1	818,770,008	Existing Customer	49	F	5	Graduate	Singl
2	713,982,108	Existing Customer	51	M	3	Graduate	Marr
3	769,911,858	Existing Customer	40	F	4	High School	Unkr
4	709,106,358	Existing Customer	40	M	3	Uneducated	Marr

Summary Statistics:


app · Streamlit
localhost:8501
google group for n... Gmail YouTube Creating Effective Vi... Introduction - Traini... Self learning skill up Data Analysis with P... Data Science for Be... All Bookmarks
Deploy

Navigation

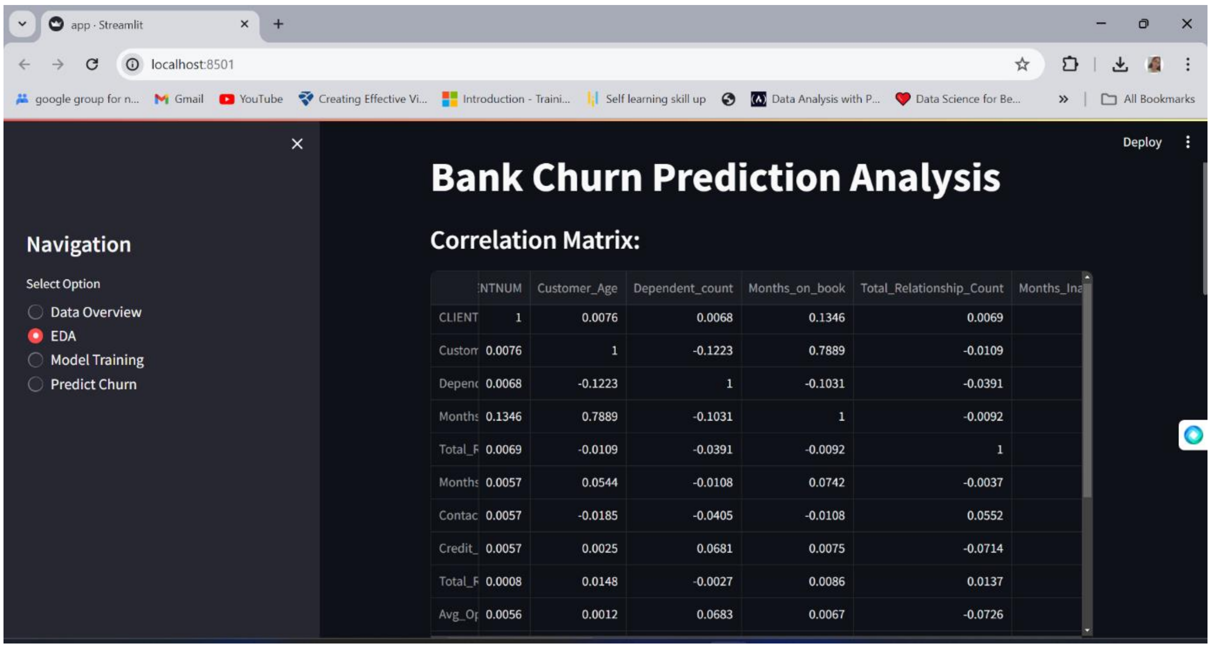
Select Option

- [✓ Data Overview]
- EDA
- Model Training
- Predict Churn

3	911,858	Existing Customer	40	F	4	High School	Unknown
4	106,358	Existing Customer	40	M	3	Uneducated	Married

Summary Statistics:

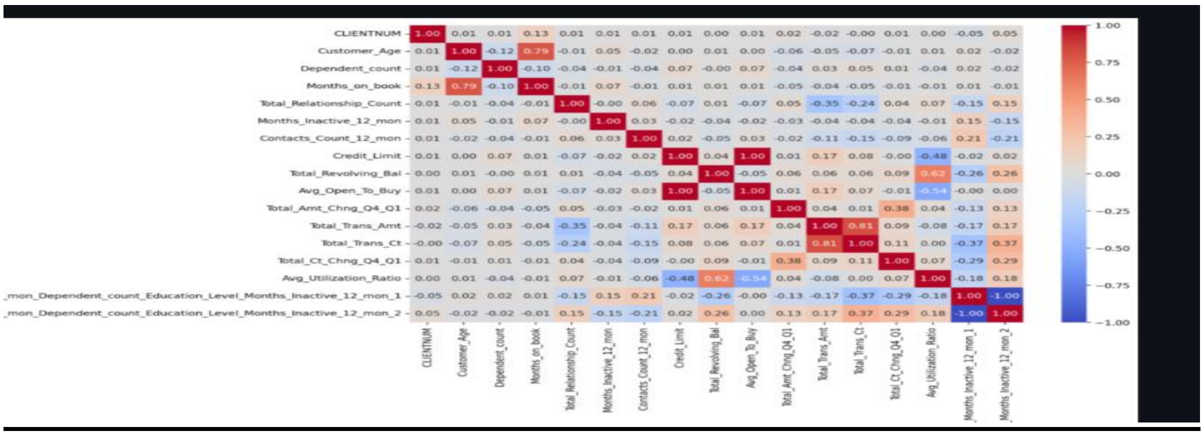
	CLIENTNUM	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	N
count	10,127	10,127	10,127	10,127	10,127	
mean	739,177,606.3337	46.326	2.3462	35.9284	3.8126	
std	36,903,783.4502	8.0168	1.2989	7.9864	1.5544	
min	708,082,083	26	0	13	1	
25%	713,036,770.5	41	1	31	3	
50%	717,926,358	46	2	36	4	
75%	773,143,533	52	3	40	5	
max	828,343,083	73	5	56	6	

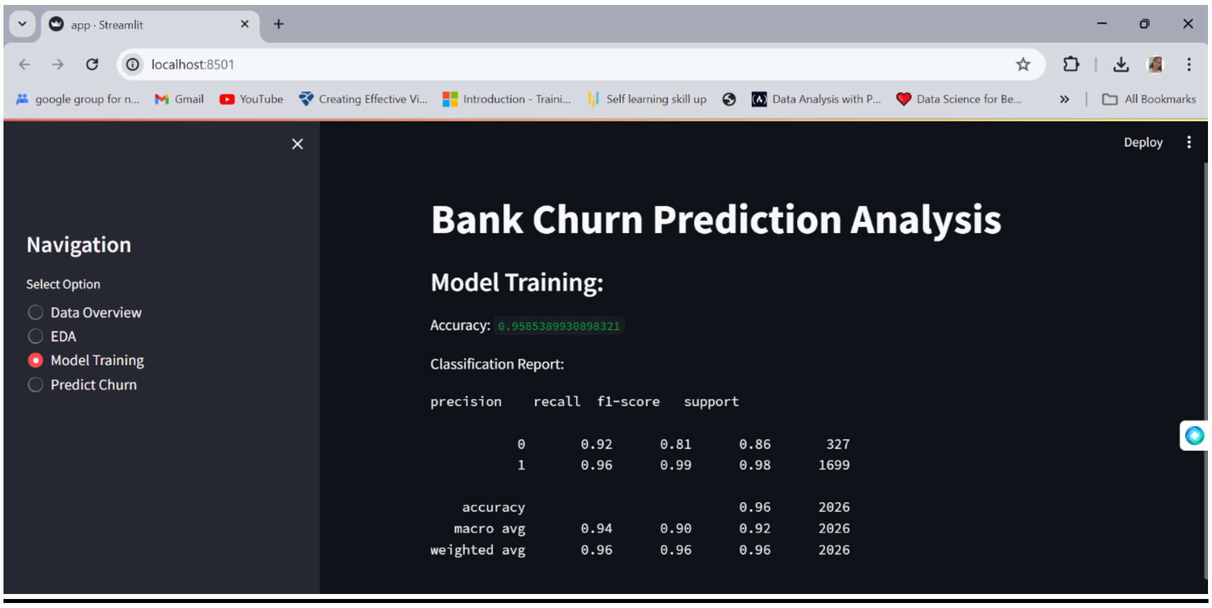


Bank Churn Prediction Analysis

Correlation Matrix:

	NTNUM	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Ina
CLIENT	1	0.0076	0.0068	0.1346	0.0069	
Custom	0.0076	1	-0.1223	0.7889	-0.0109	
Depenc	0.0068	-0.1223	1	-0.1031	-0.0391	
Months	0.1346	0.7889	-0.1031	1	-0.0092	
Total_F	0.0069	-0.0109	-0.0391	-0.0092	1	
Months	0.0057	0.0544	-0.0108	0.0742	-0.0037	
Contac	0.0057	-0.0185	-0.0405	-0.0108	0.0552	
Credit_	0.0057	0.0025	0.0681	0.0075	-0.0714	
Total_F	0.0008	0.0148	-0.0027	0.0086	0.0137	
Avg_Op	0.0056	0.0012	0.0683	0.0067	-0.0726	





Bank Churn Prediction Analysis

Model Training:

Accuracy: 0.9585389930898321

Classification Report:

```
precision    recall  f1-score   support

           0       0.92      0.81      0.86       327
           1       0.96      0.99      0.98      1699

    accuracy                           0.96      2026
   macro avg       0.94      0.90      0.92      2026
weighted avg       0.96      0.96      0.96      2026
```

9. Conclusion

This project demonstrates how machine learning can be used to predict customer churn in the banking sector. The model's high accuracy indicates its effectiveness in identifying customers at risk of leaving, which can help banks take pro-active measures to retain them.

10. Pro- Active Measures

Some initiative-taking measures that bank can take based on the predictions generated by the churn prediction model:

- Personalized Communication
- Enhanced Customer Services
- Tailored Product Recommendations
- Financial Incentives
- Feedback Mechanism
- Customer Retention Programs
- Educational Resources

If you have any question, feedback, or would like to discuss this project further, please feel free to reach out to me via email at kaur.simran1542@gmail.com

Or connect with me on LinkedIn: Simran Kaur - Dream With Data (DwD) | LinkedIn