customer-1

July 19, 2025

0.1 Load the data

0.1.1 Subtask:

Load the provided "Churn_Modelling.csv" file into a pandas DataFrame.

Reasoning: Import pandas and load the dataset into a DataFrame, then display the first 5 rows.

0.2 Explore the data

0.2.1 Subtask:

Perform some initial exploration to understand the dataset, including checking for missing values and examining the data types.

[ ]:
```
import pandas as pd

df = pd.read_csv("Churn_Modelling.csv")
display(df.head())
```

```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender   Age  \
0          1    15634602  Hargrave          619    France  Female  42.0
1          2    15647311      Hill          608     Spain  Female  41.0
2          3    15619304      Onio          502    France  Female  42.0
3          4    15701354      Boni          699    France  Female  39.0
4          5    15737888  Mitchell          850     Spain  Female  43.0

   Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0       2       0.00              1        1.0             1.0
1       1   83807.86              1        0.0             1.0
2       8  159660.80              3        1.0             0.0
3       1       0.00              2        0.0             0.0
4       2  125510.82              1        NaN             1.0

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
3         93826.63       0
4         79084.10       0
```

```
Error: Runtime no longer has a reference to this dataframe, please re-run this
cell and try again.
```

Reasoning: Display the data types and check for missing values to understand the dataset structure and identify potential data quality issues.

[ ]:
```
display(df.info())
display(df.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10002 entries, 0 to 10001
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   CreditScore        10002 non-null  int64
 1   Age                10002 non-null  float64
 2   Tenure             10002 non-null  int64
 3   Balance            10002 non-null  float64
 4   NumOfProducts      10002 non-null  int64
 5   HasCrCard          10002 non-null  float64
 6   IsActiveMember     10002 non-null  float64
 7   EstimatedSalary    10002 non-null  float64
 8   Exited             10002 non-null  int64
 9   Geography_Germany  10002 non-null  bool
 10  Geography_Spain    10002 non-null  bool
 11  Gender_Male        10002 non-null  bool
dtypes: bool(3), float64(5), int64(4)
memory usage: 732.7 KB

None

CreditScore          0
Age                  0
Tenure               0
Balance              0
NumOfProducts        0
HasCrCard            0
IsActiveMember       0
EstimatedSalary      0
Exited               0
Geography_Germany    0
Geography_Spain      0
Gender_Male          0
dtype: int64
```

0.3 Preprocess the data

0.3.1 Subtask:

Prepare the data for machine learning, which may include handling categorical variables, scaling numerical features, and splitting the data into training and testing sets.

Reasoning: I need to prepare the data for machine learning by dropping irrelevant columns, handling missing values, encoding categorical variables, and splitting the data. I can perform steps 1, 2, and 3 in a single code block.

[ ]:
```
df = df.drop(['RowNumber', 'CustomerId', 'Surname'], axis=1)

for col in ['Geography', 'Age', 'HasCrCard', 'IsActiveMember']:
    mode_val = df[col].mode()[0]
    df[col] = df[col].fillna(mode_val)

df = pd.get_dummies(df, columns=['Geography', 'Gender'], drop_first=True)

display(df.head())
```

```
   CreditScore   Age  Tenure    Balance  NumOfProducts  HasCrCard  \
0          619  42.0       2       0.00              1        1.0
1          608  41.0       1   83807.86              1        0.0
2          502  42.0       8  159660.80              3        1.0
3          699  39.0       1       0.00              2        0.0
4          850  43.0       2  125510.82              1        1.0

   IsActiveMember  EstimatedSalary  Exited  Geography_Germany  \
0             1.0        101348.88       1              False
1             1.0        112542.58       0              False
2             0.0        113931.57       1              False
3             0.0         93826.63       0              False
4             1.0         79084.10       0              False

   Geography_Spain  Gender_Male
0            False        False
1             True        False
2            False        False
3            False        False
4             True        False
```

Reasoning: Now that the data is cleaned and encoded, I need to separate the features (X) and the target variable (y) and then split the data into training and testing sets as per steps 4 and 5 of the instructions.

[ ]:
```
from sklearn.model_selection import train_test_split

X = df.drop('Exited', axis=1)
y = df['Exited']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)

display(X_train.head())
display(y_train.head())
```

```
      CreditScore   Age  Tenure    Balance  NumOfProducts  HasCrCard  \
1561          632  42.0       4  119624.60              2        1.0
1670          559  24.0       3  114739.92              1        1.0
6088          661  30.0       7       0.00              2        1.0
9575          483  30.0       9       0.00              2        0.0
8304          675  69.0       1       0.00              2        1.0

      IsActiveMember  EstimatedSalary  Geography_Germany  Geography_Spain  \
1561             1.0        195978.86               True            False
1670             0.0         85891.02              False             True
6088             0.0         72196.57              False            False
9575             0.0        136356.97              False            False
8304             0.0        157097.09              False            False

      Gender_Male
1561         True
1670         True
6088        False
9575        False
8304         True

1561    0
1670    1
6088    0
9575    0
8304    0
Name: Exited, dtype: int64
```

0.4 Build and train a model

0.4.1 Subtask:

Choose a suitable machine learning model and train it on the prepared data.

Reasoning: Import and train a RandomForestClassifier model on the training data.

[ ]:
```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

[ ]:
```
RandomForestClassifier(random_state=42)
```

0.5 Evaluate the model

0.5.1 Subtask:

Assess the performance of the trained model using appropriate metrics.

Reasoning: Use the trained model to make predictions on the test set and then calculate and print the evaluation metrics.

[ ]:
```
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
↪f1_score

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1-score: {f1:.4f}")
```

```
Accuracy: 0.8701
Precision: 0.7591
Recall: 0.5174
F1-score: 0.6154
```