# BankCustomerChurn

February 20, 2025

# 1 Factors Affecting Bank Customer Churn

## 1.1 Purpose and Questions

In today's financial landscape, customer retention has become a critical challenge for banks and financial institutions. With the rise of digital banking and increased competition, understanding why customers leave—also known as churn—is more important than ever. Many factors may contribute to a customer's decision to close their account, including financial stability, customer service quality, account fees, and available banking features. Some argue that demographics, income levels, or transaction behaviors play a significant role, while others believe that customer loyalty programs and personalized banking experiences can reduce churn rates. Despite extensive research, the debate continues. Through this analysis, I aim to uncover key factors influencing bank customer churn, explore potential predictors, and determine whether correlation implies causation or if there are deeper insights driving customer behavior. ### Questions to Answer

- Does financial stability influence a bank customer's likelihood to churn?
- Do premium banking services, often accessible to higher-income customers, reduce churn rates?
- How do customer engagement and personalized banking experiences impact retention?

I will be using a dataset from Kaggle. You can access the dataset with this link https://www.kaggle.com/datasets/saurabhbadole/bank-customer-churn-prediction-dataset.

[4]:
```
#first, let's load the dataset, we'll need pandas
import pandas as pd

df = pd.read_csv('Churn_Modelling.csv')
```

[5]:
```
df.shape #rows x columns
print(df.head())
```

```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age  \
0          1    15634602  Hargrave          619    France  Female   42
1          2    15647311      Hill          608     Spain  Female   41
2          3    15619304      Onio          502    France  Female   42
3          4    15701354      Boni          699    France  Female   39
4          5    15737888  Mitchell          850     Spain  Female   43

   Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0       2       0.00              1          1               1
```

```
1       1   83807.86              1          0               1
2       8  159660.80              3          1               0
3       1       0.00              2          0               0
4       2  125510.82              1          1               1

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

## 1.2 Preprocessing

Before we start visualizing data, we should evaluate the dataset and determine whether the dataset needs any cleaning done.

### 1.2.1 Check Nulls

[8]:
```python
print(df.isnull().sum())   #check for nulls in every column
print(df.isnull().sum().sum())
print(df.dtypes)
```

```
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
dtype: int64
0
RowNumber            int64
CustomerId           int64
Surname             object
CreditScore          int64
Geography           object
Gender              object
Age                  int64
Tenure               int64
Balance            float64
NumOfProducts        int64
```

```
HasCrCard             int64
IsActiveMember        int64
EstimatedSalary     float64
Exited                int64
dtype: object
```

#### 1.2.2 Check Duplicates

[10]:
```python
print(df.duplicated().sum())  # Count duplicate rows
```

```
0
```

### Encode Categorical Variables

[12]:
```python
import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Label Encoding for Gender (Binary)
df['Gender'] = LabelEncoder().fit_transform(df['Gender'])  # Male: 1, Female: 0

# One-Hot Encoding for Geography (Multiclass)
df = pd.get_dummies(df, columns=['Geography'], drop_first=True)

print(df.head())
```

```
   RowNumber  CustomerId   Surname  CreditScore  Gender  Age  Tenure  \
0          1    15634602  Hargrave          619       0   42       2
1          2    15647311      Hill          608       0   41       1
2          3    15619304      Onio          502       0   42       8
3          4    15701354      Boni          699       0   39       1
4          5    15737888  Mitchell          850       0   43       2

     Balance  NumOfProducts  HasCrCard  IsActiveMember  EstimatedSalary  \
0       0.00              1          1               1        101348.88
1   83807.86              1          0               1        112542.58
2  159660.80              3          1               0        113931.57
3       0.00              2          0               0         93826.63
4  125510.82              1          1               1         79084.10

   Exited  Geography_Germany  Geography_Spain
0       1              False            False
1       0              False             True
2       1              False            False
3       0              False            False
4       0              False             True
```

So far, we have checked for duplicates, encoded categorical variables, and ensured that the dataset is clean and well-structured. Since there were no missing values, no imputation was necessary, allowing us to retain the original integrity of the data. I also converted categorical variables into numerical representations to make them compatible with machine learning models. These preprocessing steps ensure that the dataset is properly formatted and ready for analysis. With a clean and structured dataset, we can now proceed confidently with further exploration and modeling.

## 1.3 Method

I decided to use Random Forest for this churn prediction problem because it balances accuracy, interpretability, and robustness. Since churn prediction involves a mix of numerical and categorical features, Random Forest handles both effectively without needing heavy preprocessing. Unlike a single decision tree, which can easily overfit, Random Forest builds multiple trees and averages their predictions, making it more stable and reliable. Another big reason I chose it is that it provides feature importance rankings, so I can see which factors—like Credit Score, Balance, or IsActiveMember—actually influence churn the most. Plus, it works well even with imbalanced data, which is common in churn problems. Overall, it gives me a solid mix of performance and explainability, making it a great choice for this dataset.

[ ]:

### 1.3.1

[ ]:

### 1.3.2

[ ]:

### 1.3.3

[ ]:

### 1.3.4

[ ]:

[ ]:

[ ]:

### 1.3.5

[ ]:

[ ]:

### 1.3.6

[ ]:

[ ]:

```

```

## 1.4 Summary

### 1.4.1 Bias and Limitations