Customer Churn Analytics for European Bank

Internship Project Report

(Jan 2026 - March 2026)

Submitted By: Rutuja Kamble (Data Analyst Intern)

Tools & Technologies Used: Excel, MySQL, Python, Power BI, DAX, Pandas, NumPy, Matplotlib, Seaborn

GitHub Link: https://github.com/rutuja-2003/Customer-Segmentation-Churn-Analysis-of-Bank

Organization: Unified Mentor Pvt. Ltd.

Abstract

Customer churn is one of the most critical challenges faced by the banking sector, directly impacting revenue, customer lifetime value, and acquisition costs. This project aims to analyse churn patterns in a European bank using a complete end-to-end data analytics pipeline, including Excel, SQL, Python, and Power BI.

The study focuses on identifying key factors contributing to customer attrition such as age, credit score, geography, balance, and product usage. Data was cleaned, transformed, and validated across multiple tools to ensure reliability. Exploratory Data Analysis (EDA) in Python further strengthened insights, while Power BI dashboards visualized churn behaviours clearly and interactively.

The final outcome includes actionable business recommendations that will help the bank reduce churn, improve customer retention, and strengthen high-value customer relationships.

Introduction

Customer churn represents a major challenge in the financial services industry. For retail banks, losing customers often means losing long-term revenue, cross-selling opportunities, and brand credibility. Understanding churn behaviour and identifying at-risk customer segments is crucial for improving retention and reducing overall acquisition costs.

This project provides a comprehensive churn analysis for European Bank as part of an internship under Unified Mentor Pvt. Ltd. The project follows an end-to-end analytics workflow starting from raw data cleaning to advanced insights extraction and business recommendations. Multiple tools such as Excel, Power BI, SQL, and Python were used to ensure accurate, multi-angle analysis.

The insights derived from this study will help the bank identify high-risk customers, uncover the factors influencing churn, and design strategic actions to improve customer loyalty.

Problem Statement

The European Bank has been experiencing a growing number of customer exits (churn). Losing customers affects profitability and increases operational costs, making churn prediction and prevention a top priority.

The central questions addressed in this project include:

1. Which customer segments have the highest churn rates?
2. What demographic and behavioural factors influence churn the most?
3. How do credit score, age, products, geography, and balance relate to churn?
4. Which customers are at the highest risk of leaving the bank?
5. What strategies can help reduce churn and retain valuable customers?

This report delivers a structured analytical solution using dashboards, EDA, data cleaning, modelling (basic), segmentation, and actionable recommendations.

Project Objectives

The main objectives of this project are:

1. Clean and Prepare the Dataset

- Handle missing values, inconsistencies, incorrect data types, and outliers.
- Standardize fields such as Age, Credit Score, Balance, Products, etc.

2. Perform Exploratory Data Analysis (EDA)

- Understand distributions, correlations, and data patterns.
- Identify demographic and behavioral attributes affecting churn.

3. Build Customer Segmentation Logic

- Divide customers by demographics (age groups, geography).
- Group them by behavioral factors (products, balance, credit score).

4. Create Interactive Power BI Dashboards

- Dashboard 1: Visual storytelling of churn patterns.
- Dashboard 2: Insights & Recommendations page with KPIs.
- One button slicer = Customer Segment for user-driven navigation.

5. Extract Insights & Business Recommendations

- Based on EDA findings and dashboard analysis.

- Identify strengths and risks.

6. Prepare Final Documentation (This Report + PPT)

- Summarize methodology, insights, recommendations, and conclusion.

Tools & Technologies Used

Tool	Purpose
Excel	Initial data understanding & minor cleaning
MySQL	Data validation & SQL-based analysis
Python (Google Colab)	EDA, data visualization, correlation analysis
Power BI	Main dashboard creation, insights page, DAX calculations
DAX	KPI Measures & calculated columns
Power Query (Transform Data)	Main dataset cleaning & transformation

---

Dataset Description

The project uses a structured banking dataset containing customer demographic, financial, and behavioral information. The data was provided as part of the internship assignment by Unified Mentor Pvt. Ltd.

Dataset Overview

- Total Records: ~10,000 customers
- File Name: european_bank.csv
- Source: Internship project dataset provided by Unified Mentor
- Target Variable: Exited
    - 1 = Customer Churned
    - 0 = Customer Retained

Key Attributes

Feature	**Description**
CustomerId	Unique identifier for each customer
Surname	Customer last name
Geography	Country (France, Spain, Germany)
Gender	Male/Female
Age	Age of customer
CreditScore	Score between 350–850
Balance	Bank account balance
NumOfProducts	Number of bank products used
IsActiveMember	Whether customer interacts actively
EstimatedSalary	Annual salary
Tenure	Number of years with the bank
Exited	Churn flag (1 = left, 0 = stayed)

Additional Calculated Features (Created During Project)

New Feature	Purpose
Age_Group	Categorizing age into bins for better segmentation
CreditScore_Band	Grouping credit scores into ranges
Customer_Segment	Low/Medium/High-value segmentation
Risk_Score (from Insights page)	Helps categorize high-risk churn customers

These engineered features improved segmentation and visual interpretation.

Data Cleaning & Preparation

Data cleaning was done using Excel, Power BI (Transform Data), SQL, and Python. Each tool served a purpose to ensure maximum accuracy and professional workflow.

Data Cleaning in Excel

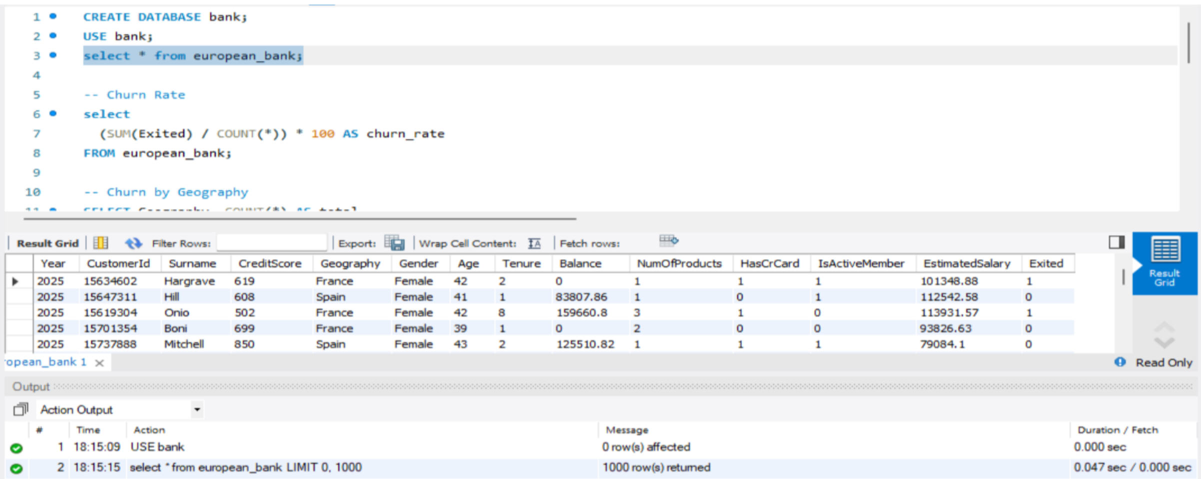
- Removed duplicates based on CustomerId
- Fixed inconsistent categorical values (e.g., "France"/"france")
- Cleaned nulls in:
  - Credit Score
  - Age
  - Balance
- Verified logical constraints:
  - Age > 18
  - Products between 1–4
  - Credit Score between 350–850

8.2 Data Validation in MySQL

SQL was used to reconfirm critical data integrity:

- Count of unique customers
- Distribution of churn vs non-churn
- Gender distribution
- Geography correctness
- Outliers in Credit Score
- Average age & balance
- Churn patterns across Products, Active Status, Credit Score

Using SQL gave confidence in dataset consistency before visualization.



Limit to 1000 rows

```sql
CREATE DATABASE bank;
USE bank;
select * from european_bank;

-- Churn Rate
select
  (SUM(Exited) / COUNT(*)) * 100 AS churn_rate
FROM european_bank;

-- Churn by Geography
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

churn_rate
20.3700

Result 2 × — Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	18:15:15	select * from european_bank LIMIT 0, 1000	1000 row(s) returned	0.047 sec / 0.000 sec
3	18:16:17	select (SUM(Exited) / COUNT(*)) * 100 AS churn_rate FROM european_bank LI...	1 row(s) returned	0.031 sec / 0.000 sec

Limit to 1000 rows

```sql
select
  (SUM(Exited) / COUNT(*)) * 100 AS churn_rate
FROM european_bank;

-- Churn by Geography
SELECT Geography, COUNT(*) AS total,
       SUM(Exited) AS churned,
       (SUM(Exited)/COUNT(*))*100 AS churn_rate
FROM european_bank
GROUP BY Geography;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Geography	total	churned	churn_rate
France	5014	810	16.1548
Germany	2509	814	32.4432
Spain	2477	413	16.6734

Result 3 × — Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
3	18:16:17	select (SUM(Exited) / COUNT(*)) * 100 AS churn_rate FROM european_bank LI...	1 row(s) returned	0.031 sec / 0.000 sec
4	18:17:01	SELECT Geography, COUNT(*) AS total, SUM(Exited) AS churned, (SUM(...	3 row(s) returned	0.046 sec / 0.000 sec

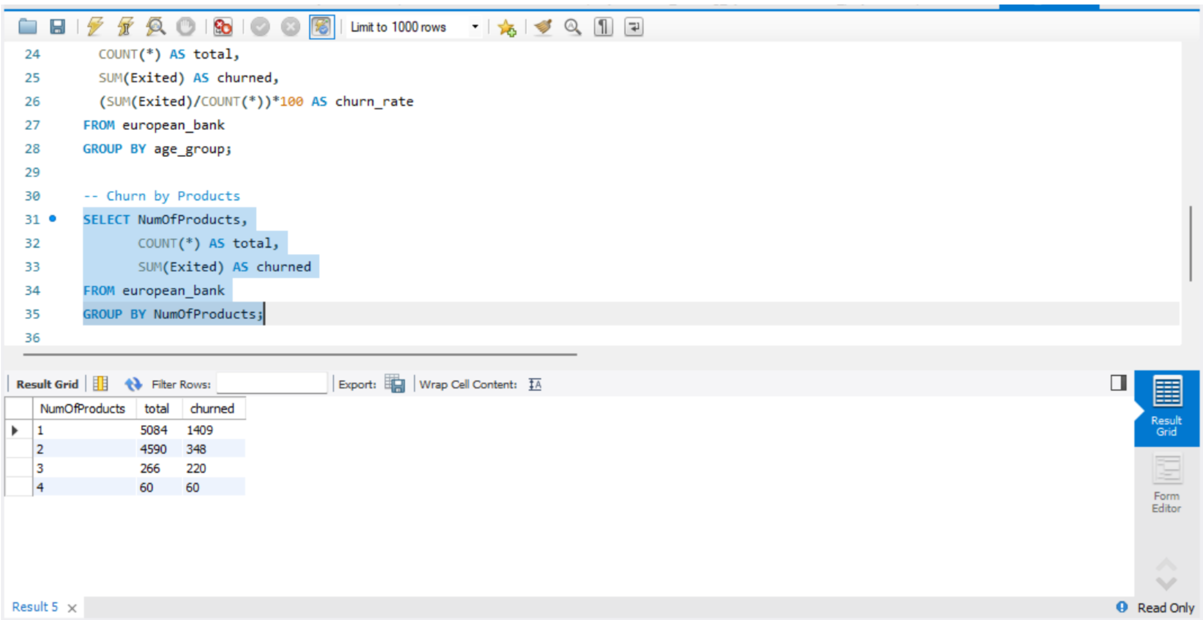
```sql
-- Churn by Age Group
SELECT
  CASE
    WHEN Age < 30 THEN 'Young'
    WHEN Age BETWEEN 30 AND 50 THEN 'Middle-Age'
    ELSE 'Senior'
  END AS age_group,
  COUNT(*) AS total,
  SUM(Exited) AS churned,
  (SUM(Exited)/COUNT(*))*100 AS churn_rate
FROM european_bank
GROUP BY age_group;
```

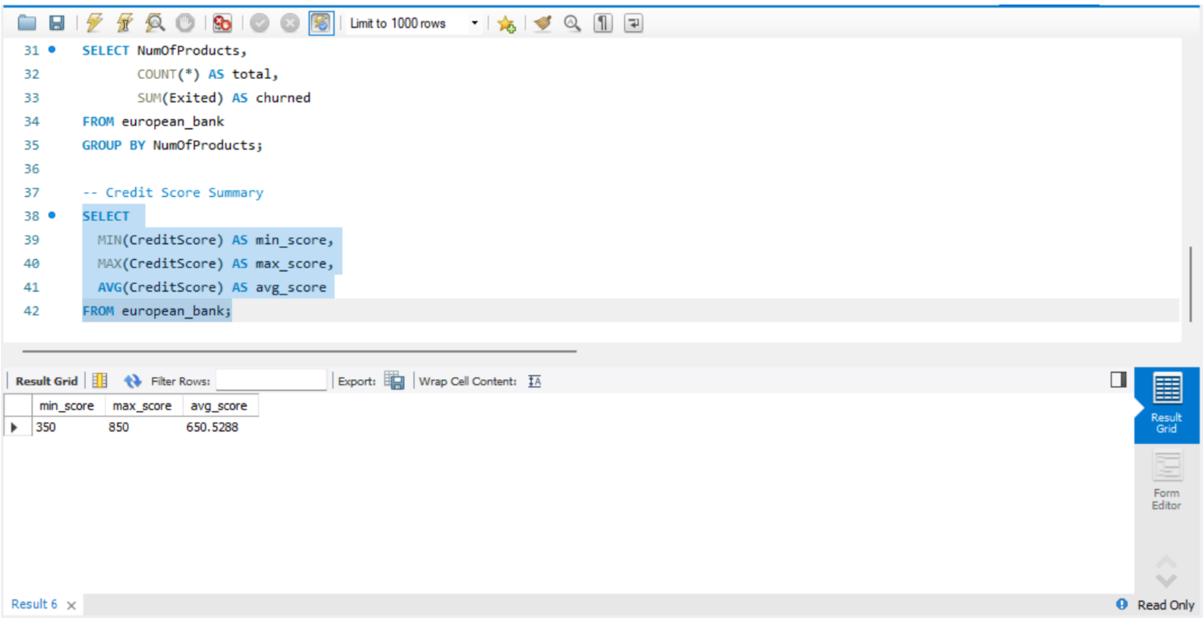
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

age_group	total	churned	churn_rate
Middle-Age	7098	1350	19.0194
Senior	1261	563	44.6471
Young	1641	124	124 5564

Result 4 × — Read Only

By: Rutuja Kamble (Data Analyst Intern)





8.3 Data Cleaning & EDA in Python (Colab)

Performed EDA using:

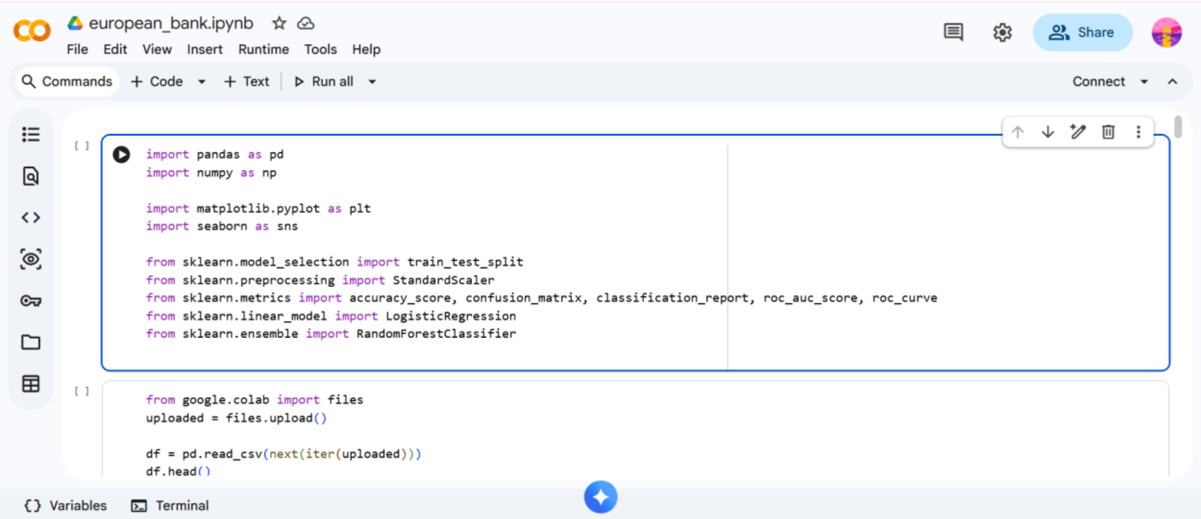
- Pandas
- Matplotlib
- Seaborn

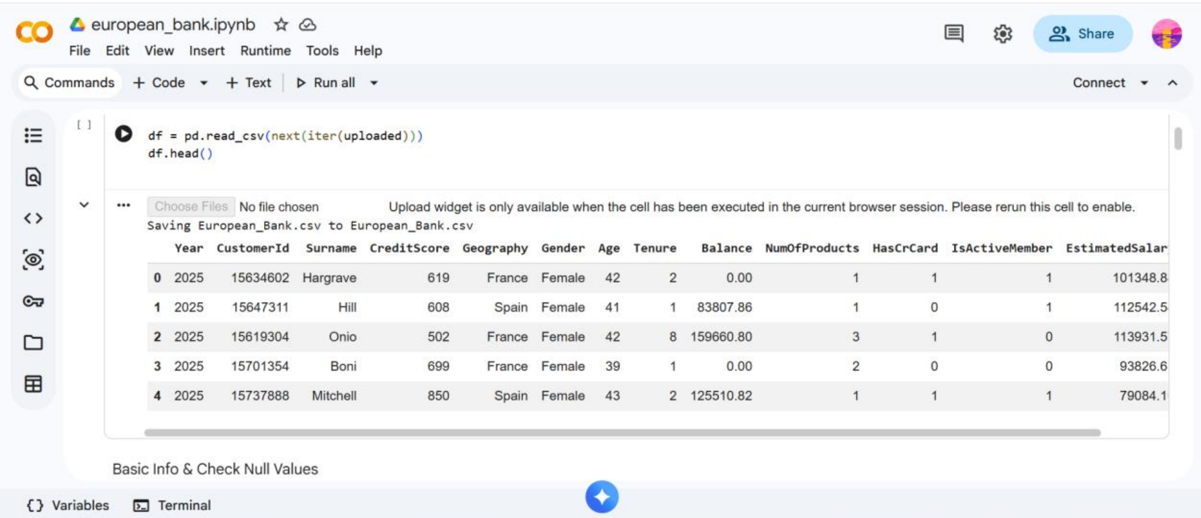
Python tasks included:

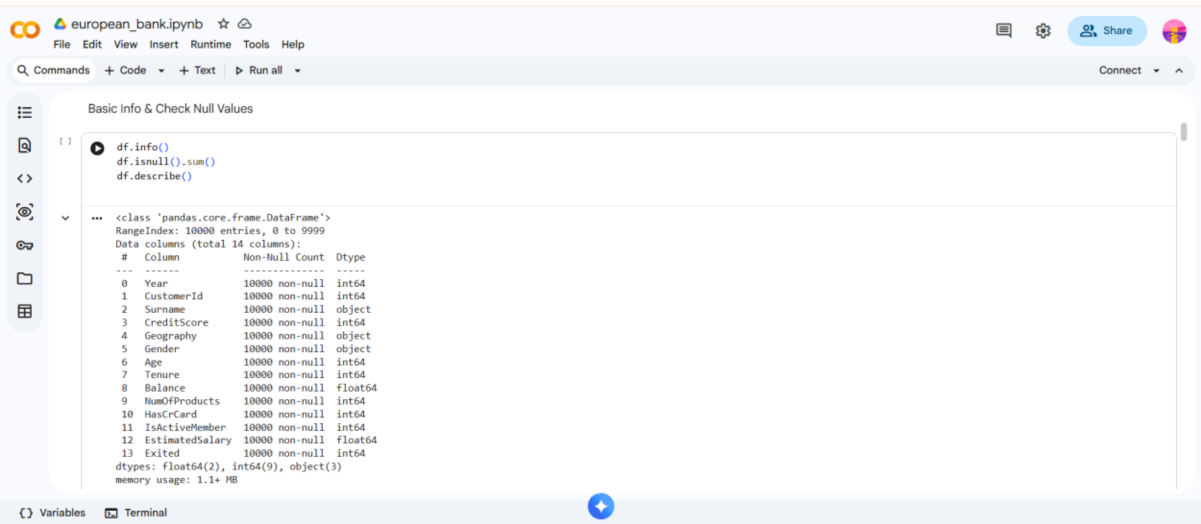
- Checking nulls & duplicates
- Summary statistics

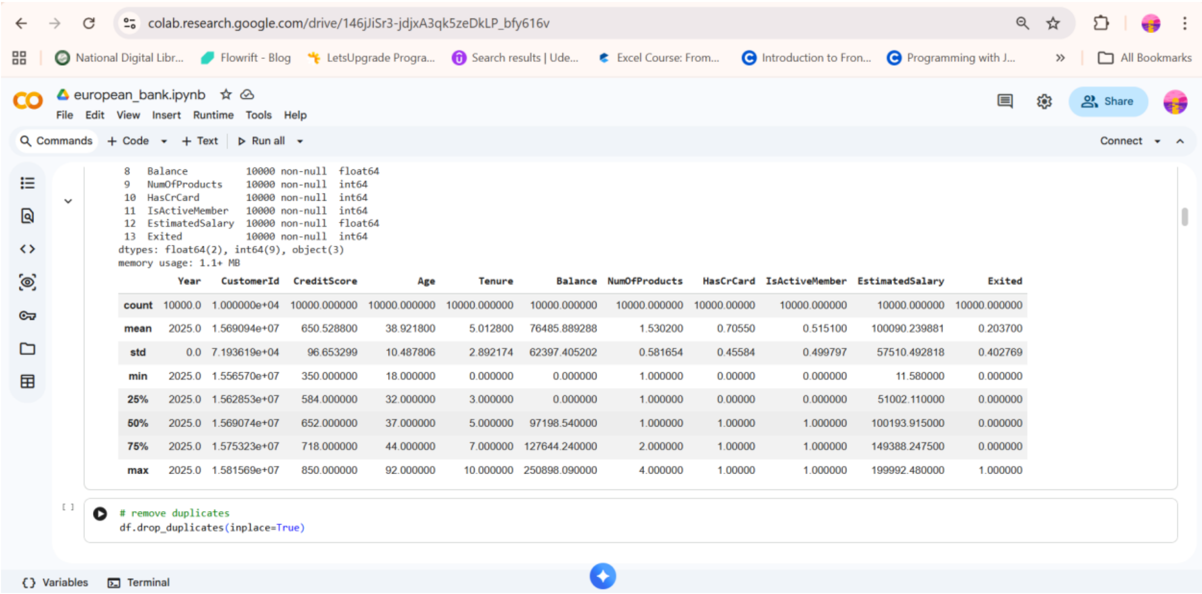
- Visualizations (histograms, pairplots, heatmaps)
- Correlation heatmap
- Exporting cleaned dataset → cleaned_european_bank.csv

This cleaned file was imported into Power BI for dashboard creation.









```
8   Balance          10000 non-null  float64
9   NumOfProducts    10000 non-null  int64
10  HasCrCard        10000 non-null  int64
11  IsActiveMember   10000 non-null  int64
12  EstimatedSalary  10000 non-null  float64
13  Exited           10000 non-null  int64
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

	Year	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.0	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.000000
mean	2025.0	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	0.203700
std	0.0	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402769
min	2025.0	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	0.000000
25%	2025.0	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	0.000000
50%	2025.0	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	0.000000
75%	2025.0	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	0.000000
max	2025.0	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	1.000000

```
# remove duplicates
df.drop_duplicates(inplace=True)
```

8.4 Power BI Transform Data Cleaning

Power Query cleaning included:

- Removing additional nulls after merging
- Changing data types (Integer, Text, Decimal)
- Creating calculated columns:
    - Age_Group
    - Customer_Segment
    - CreditScore_Band
- Final shaping for visualization-ready dataset

This became the master dataset for dashboard creation.

---

9. Exploratory Data Analysis (EDA)

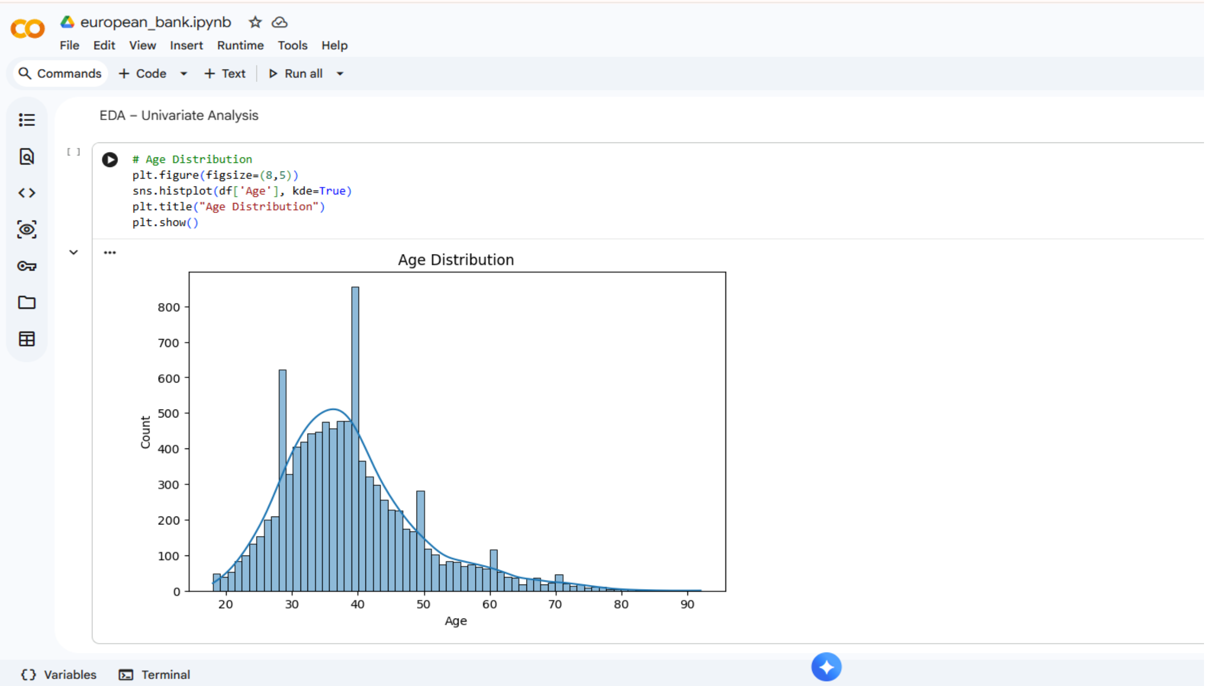
The EDA was performed using Power BI, Python, and SQL, covering:

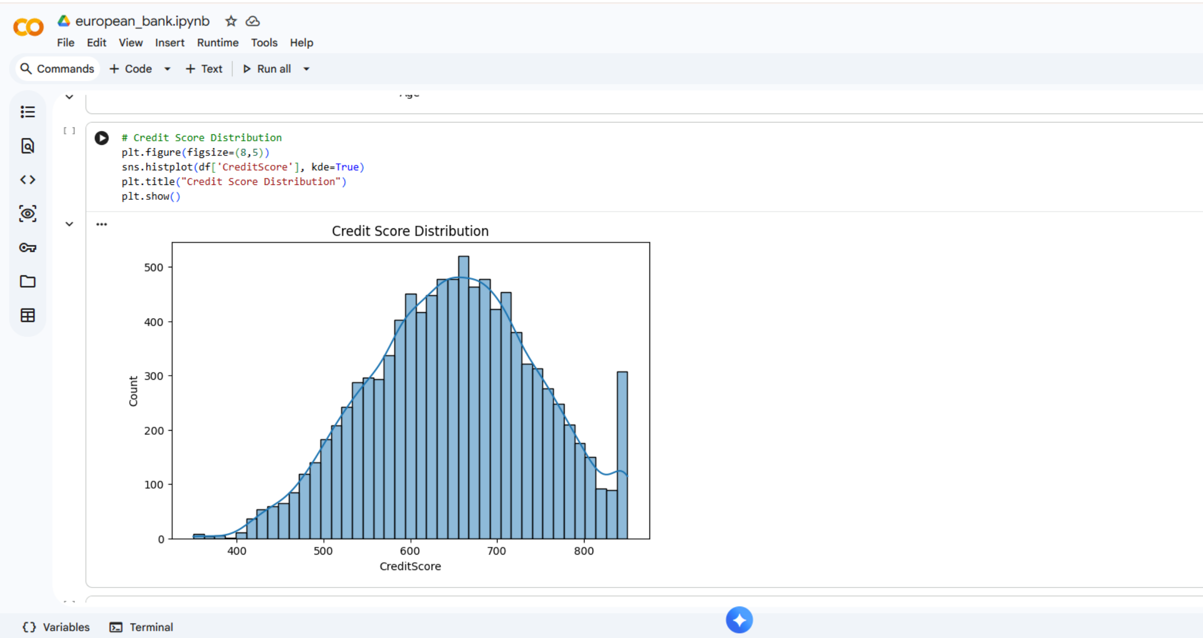
Customer Demographics

- Gender distribution
- Age groups
- Geography split

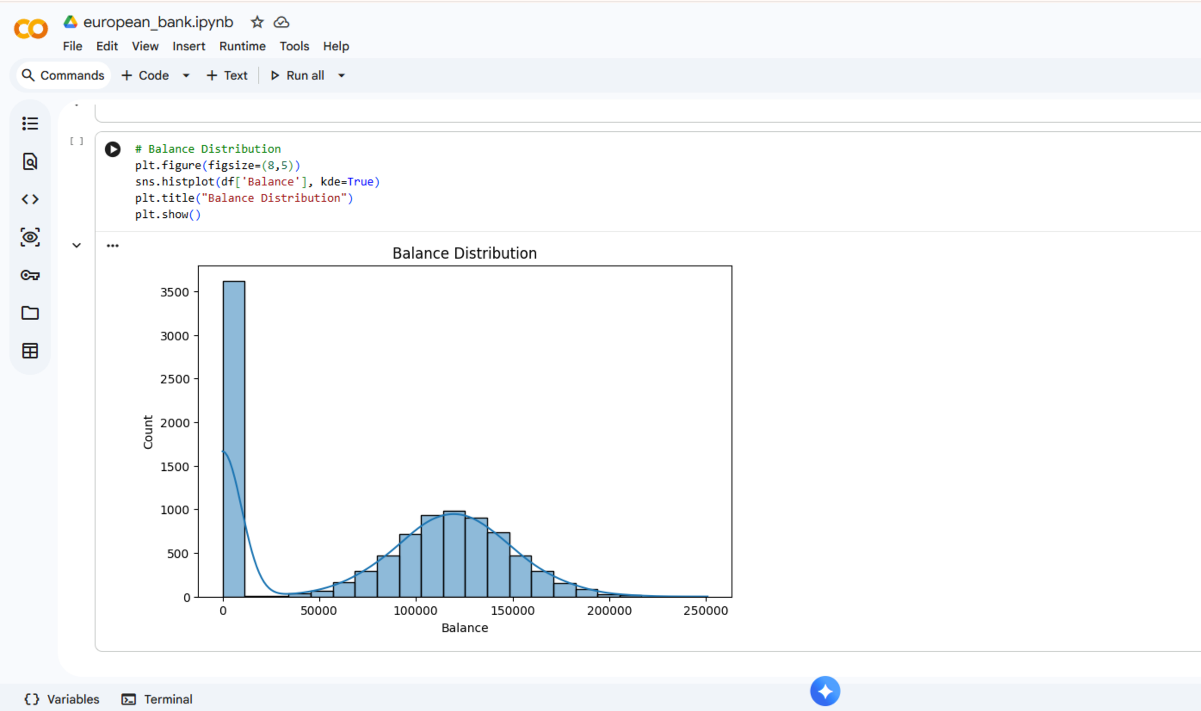
Churn Behaviour

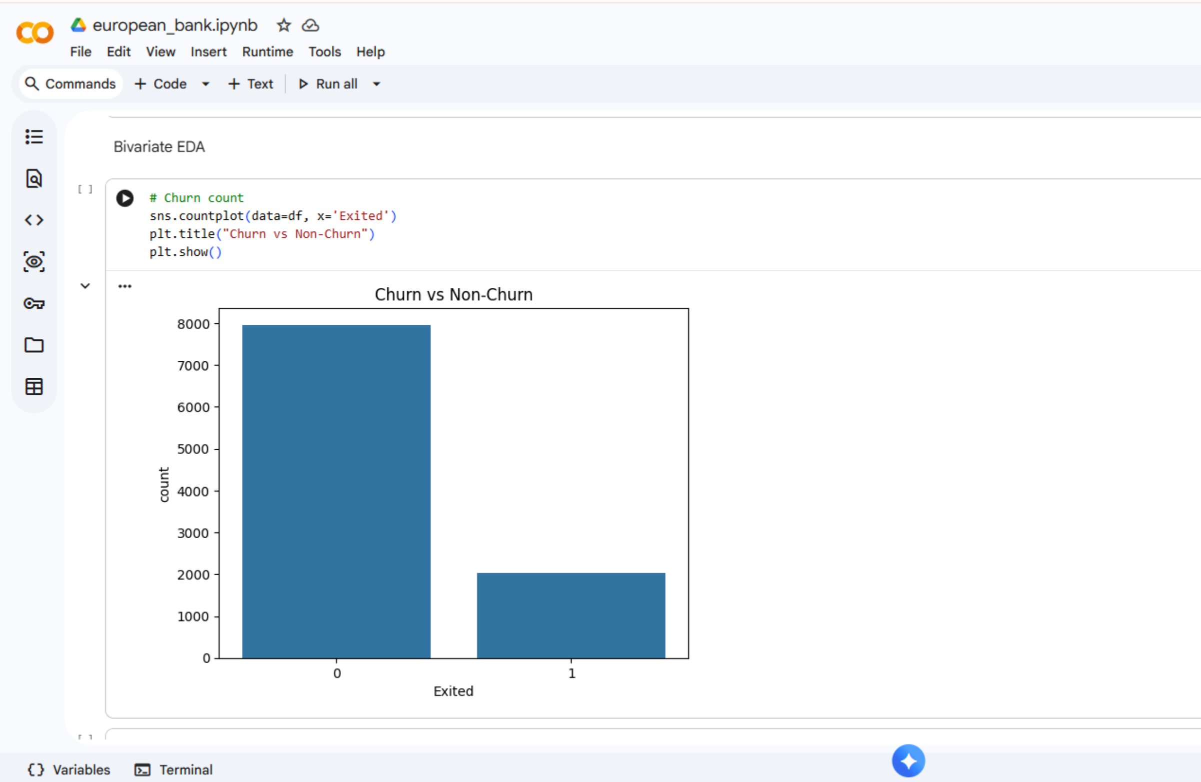
- Churn by age
- Churn by gender
- Churn by geography
- Churn by credit score
- Churn by products
- Churn by balance vs age
- Churn by activity level





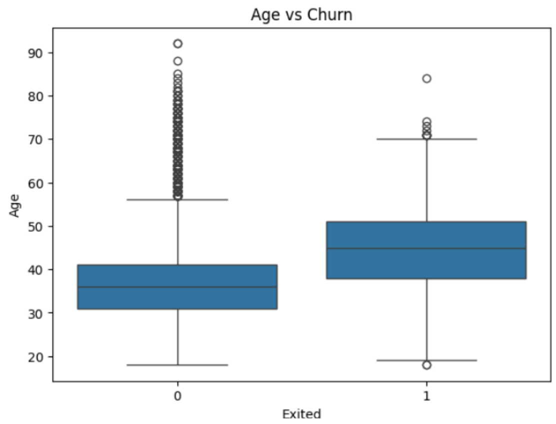
By: Rutuja Kamble (Data Analyst Intern)



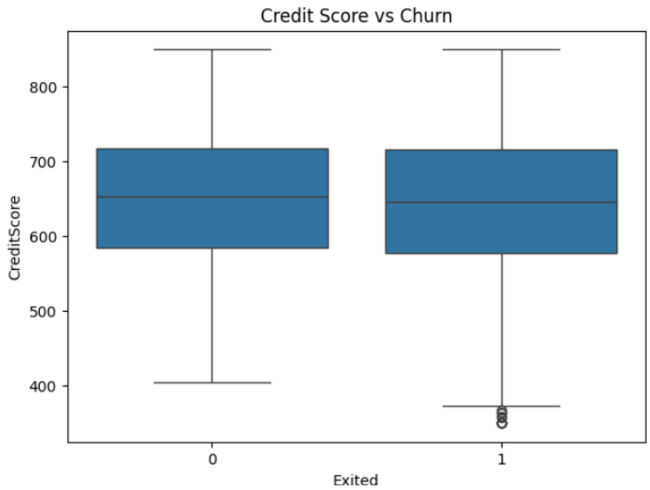


By: Rutuja Kamble (Data Analyst Intern)

```python
# Age vs Churn
plt.figure(figsize=(7,5))
sns.boxplot(data=df, x='Exited', y='Age')
plt.title("Age vs Churn")
plt.show()
```




```python
# Credit Score vs Churn
plt.figure(figsize=(7,5))
sns.boxplot(data=df, x='Exited', y='CreditScore')
plt.title("Credit Score vs Churn")
plt.show()
```



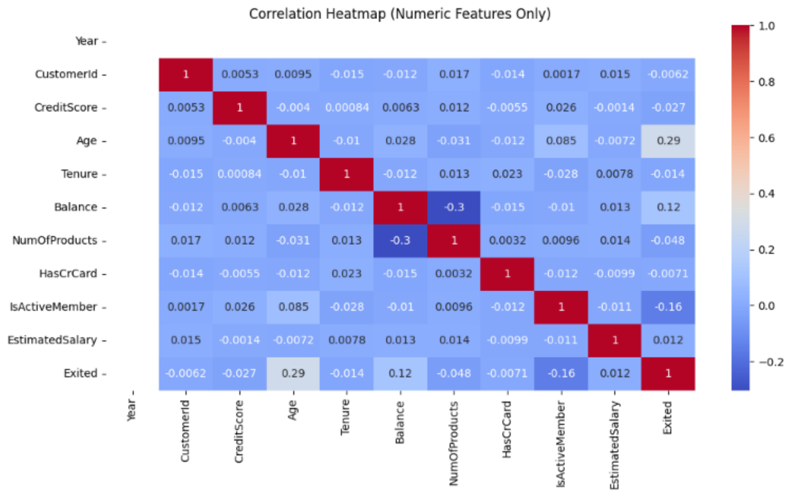
By: Rutuja Kamble (Data Analyst Intern)

european_bank.ipynb

File Edit View Insert Runtime Tools Help


```python
# Select numeric columns only
numeric_df = df.select_dtypes(include=['int64', 'float64'])

# Correlation Heatmap
plt.figure(figsize=(12,6))
sns.heatmap(numeric_df.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap (Numeric Features Only)")
plt.show()
```



european_bank.ipynb

File Edit View Insert Runtime Tools Help

Feature Engineering

```python
# Age Groups
df['Age_Group'] = pd.cut(df['Age'],
                         bins=[0, 30, 50, 100],
                         labels=['Young', 'Middle-Aged', 'Senior'])
```

```python
# Credit Score Band
df['CreditScore_Band'] = pd.cut(df['CreditScore'],
                                bins=[300, 580, 670, 740, 850],
                                labels=['Poor', 'Fair', 'Good', 'Excellent'])
```

```python
#Customer Segment (Example Logic)
df['Customer_Segment'] = np.where(df['Balance'] > 100000, 'High Value',
                         np.where(df['Balance'] > 50000, 'Mid Value', 'Low Value'))
```

```python
df.head()
```

	Year	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Age_Group	CreditScore_Band	Customer_Segment
0	2025	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1	Middle-Aged	Fair	Low Value
1	2025	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0	Middle-Aged	Fair	Mid Value
2	2025	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1	Middle-Aged	Poor	High Value
3	2025	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0	Middle-Aged	Good	Low Value
4	2025	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0	Middle-Aged	Excellent	High Value

european_bank.ipynb

```
#Encode Categorical Variables
df_encoded = pd.get_dummies(df, drop_first=True)
df_encoded.head()
```

	Year	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	...	Geography_Germany	Geography_Spain	Gender_Male	Age_Group_Middle-Aged	Age_Group_Senior	CreditScore_Band_Fa
0	2025	15634602	619	42	2	0.00	1	1	1	101348.88	...	False	False	False	True	False	T
1	2025	15647311	608	41	1	83807.86	1	0	1	112542.58	...	False	True	False	True	False	T
2	2025	15619304	502	42	8	159660.80	3	1	0	113931.57	...	False	False	False	True	False	Fa
3	2025	15701354	699	39	1	0.00	2	0	0	93826.63	...	False	False	False	True	False	Fa
4	2025	15737888	850	43	2	125510.82	1	1	1	79084.10	...	False	True	False	True	False	Fa

5 rows × 2952 columns

```
# Split Data (X, y)
X = df_encoded.drop('Exited', axis=1)
y = df_encoded['Exited']
```

```
#Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
#Feature Scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

european_bank.ipynb

```
#Logistic Regression Model
log_model = LogisticRegression(max_iter=500)
log_model.fit(X_train, y_train)

y_pred_log = log_model.predict(X_test)
```

```
print("Logistic Regression Accuracy:", accuracy_score(y_test, y_pred_log))
print(classification_report(y_test, y_pred_log))
```

```
Logistic Regression Accuracy: 0.7785
              precision    recall  f1-score   support

           0       0.83      0.91      0.87      1607
           1       0.39      0.23      0.29       393

    accuracy                           0.78      2000
   macro avg       0.61      0.57      0.58      2000
weighted avg       0.74      0.78      0.75      2000
```

```
# Random Forest Classifier
rf = RandomForestClassifier(n_estimators=200, random_state=42)
rf.fit(X_train, y_train)

y_pred_rf = rf.predict(X_test)

#Evaluation
print("Random Forest Accuracy:", accuracy_score(y_test, y_pred_rf))
print(classification_report(y_test, y_pred_rf))
```

```
Random Forest Accuracy: 0.858
              precision    recall  f1-score   support
```

european_bank.ipynb

```
# Random Forest Classifier
rf = RandomForestClassifier(n_estimators=200, random_state=42)
rf.fit(X_train, y_train)

y_pred_rf = rf.predict(X_test)

#Evaluation
print("Random Forest Accuracy:", accuracy_score(y_test, y_pred_rf))
print(classification_report(y_test, y_pred_rf))
```

```
Random Forest Accuracy: 0.858
              precision    recall  f1-score   support

           0       0.86      0.98      0.92      1607
           1       0.82      0.35      0.49       393

    accuracy                           0.86      2000
   macro avg       0.84      0.67      0.71      2000
weighted avg       0.85      0.86      0.83      2000
```

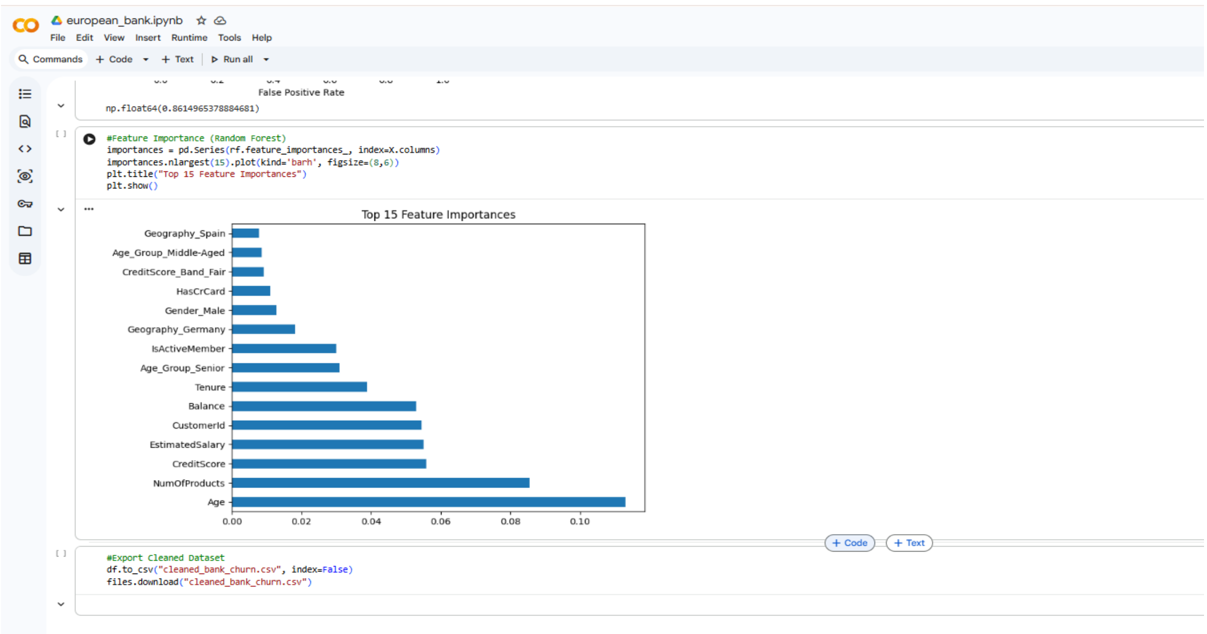
```
#ROC-AUC Curve
y_probs = rf.predict_proba(X_test)[:,1]
fpr, tpr, thresh = roc_curve(y_test, y_probs)

plt.figure(figsize=(6,5))
plt.plot(fpr, tpr)
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve (Random Forest)")
plt.show()

roc_auc_score(y_test, y_probs)
```

ROC Curve (Random Forest)

By: Rutuja Kamble (Data Analyst Intern)



Important Findings

(These appear later in Insights page — but repeating here for report completeness)

- Customers aged 50+ churn the most
- Churn is highest in Germany
- Customers with high balance but low activity churn more
- Customers with 2 products are most loyal

By: Rutuja Kamble (Data Analyst Intern)

- Credit score has moderate impact on churn
- Active customers churn less

---

10. Dashboard Design & Development

The dashboard was created in **Power BI**, consisting of **two pages** designed for stakeholders, team leads, and bank management. This dashboard enables quick understanding of customer behaviour, churn patterns, and actionable business metrics.

10.1 Page 1 – Customer Churn Analytics Dashboard

Objective:

Provide a clear overview of churn distribution across different customer attributes.

Components Included

✓ Three KPI Cards

1. **Total Customers**
2. **Total Churned Customers**
3. **Overall Churn Rate**

Six Visuals

1. **Churn % by Age Group & Gender**
   → Analyzes how age and gender influence customer churn.
2. **Churn % by Age Group (Age_Group)**
   → Identifies the high-risk age segments.
3. **Churn % by CreditScore Band**
   → Helps understand the stability of customers based on financial credibility.
4. **Balance vs Age by Churn**
   → Shows if financial value impacts churn.
5. **Churn % by Number of Products**
   → Reveals loyalty trends depending on product usage.
6. **Churn Count & Customer Count by Geography**
   → Highlights high-churn countries.

One Button Slicer

- **Customer Segment** (Low, Medium, High Value Customers)
  → Helps filter all visuals based on customer worth.

10.2 Page 2 – Insights & Recommendations Dashboard

This page converts visuals into actionable insights.

Components Included

Five KPI Cards

1. Total Customers
2. Churn Rate
3. High-Risk Customers
4. Average Balance
5. Active Customer Ratio

Risk Indicator

- Badge showing **High Risk Regions / Segments**

Strength Indicator

- Badge showing the bank's stability areas

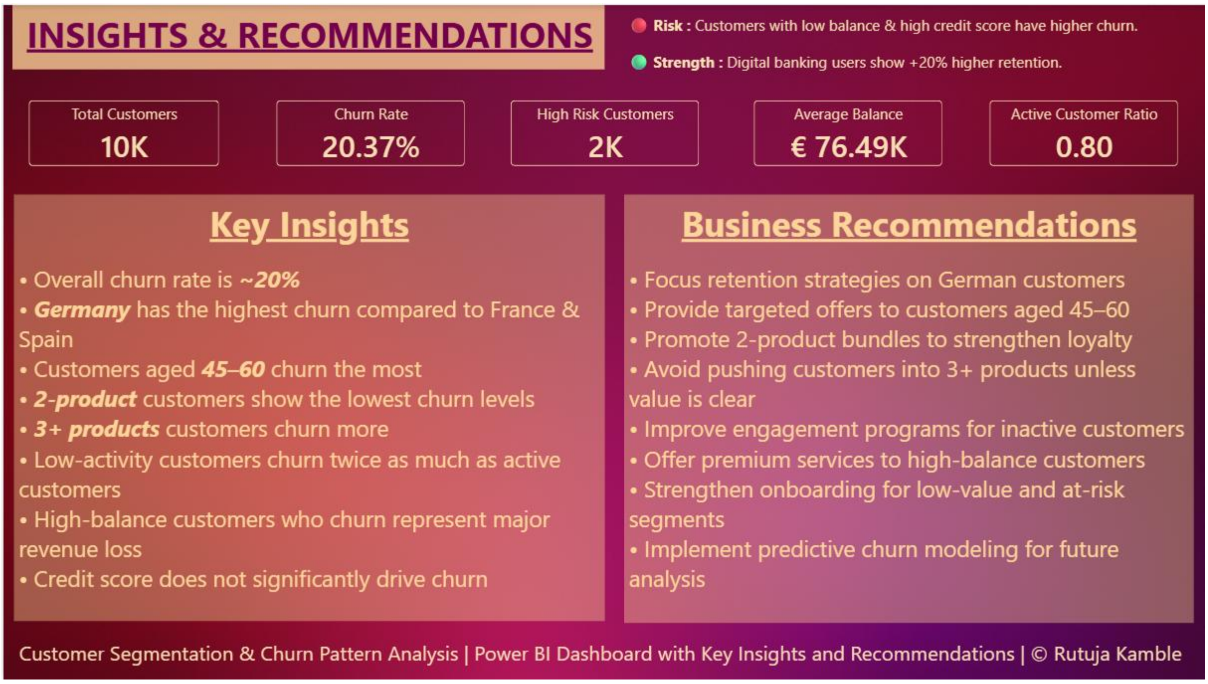
Insights Section

Beautifully structured written insights based on the visuals.

Recommendations Section

Actionable strategies for customer retention.

This page is the **storytelling layer** of the project—helpful for presentations and reporting.



---

11. Key Insights (As Displayed in Power BI Insights Page)

1. Age & Churn

- Customers above **50 years** show significantly higher churn.
- Younger customers (18–35) are more stable.

2. Geography Impact

- **Germany** has the highest churn among all regions.
- France & Spain customers are more loyal.

3. Products & Loyalty

- Customers with **two products** show the lowest churn.
- Customers with **3+ products** tend to churn more due to dissatisfaction or product complexity.

4. Activity Level

- **Inactive customers churn twice as much** as active members.

5. Balance & Churn

- High-balance customers leaving the bank indicates a business threat.

6. Credit Score

- Customers with **low credits** churn more, but credit score is not the strongest predictor of churn.

---

12. Business Recommendations

1. Customer Retention Program for 50+ Age Group

- Personalized financial advisory
- Senior-friendly service plans
- Loyalty rewards

2. Reduce Churn in Germany

- Conduct customer interviews
- Improve support handling
- Launch market-specific offers

3. Strengthen Engagement for Inactive Customers

- Monthly activity reminders
- Free consultation services
- App-based financial goals

4. Promote Two-product Bundling

- Two-product customers are most loyal
- Offer discounted product combos
- Encourage cross-selling

5. Balance-based Alerts

- Create high-balance customer monitoring list
- Provide premium customer care services

6. Improve Customer Experience

- Predictive churn modelling (future model)
- Automate follow-up reminders
- Feedback-driven service improvements

---

13. Conclusion

This project provides a comprehensive understanding of customer churn behaviour for European Bank. By integrating Excel, SQL, Python, and Power BI, a complete analytical pipeline was created. The findings highlight churn drivers such as age, geography, product usage, and customer activity.

The reports and dashboards help bank leadership adopt proactive strategies to reduce churn, retain high-value customers, and improve customer satisfaction. These insights can significantly strengthen long-term business performance and enhance customer loyalty.

By: Rutuja Kamble (Data Analyst Intern)