Customer Churn Prediction Using Machine Learning

Prepared by:
Mahima Advilkar

Role: Data Analyst
Domain: Predictive Modeling | Banking Analytics
Tools Used: Python, scikit-learn, pandas, seaborn, matplotlib

1. Project Objective

The main objective of this project is to build a machine learning model that can accurately predict customer churn in a banking context. By analyzing historical customer data, the model helps identify individuals who are at high risk of leaving the bank.

This enables the business to take proactive steps—such as targeted engagement or retention campaigns—ultimately aiming to reduce customer loss, improve satisfaction, and support long-term revenue growth. The project also provides insight into the key factors that drive churn, supporting more data-driven decision-making across teams.

2. Data Description

The dataset used in this project is a **synthetic customer churn dataset** from a European bank. It contains information on 10,000 customers, capturing both demographic and behavioral factors that may influence whether a customer decides to leave the bank.

📂 **Dataset Overview:**

- **Source:** Synthetic data simulating a European bank's customer base
- **Total Records:** 10,000 customer entries
- **Target Variable:** Exited (1 = churned, 0 = retained)

📊 **Feature Breakdown:**

- **Demographics:**
  - Geography – Country of residence
  - Gender – Male/Female
  - Age – Age of the customer
- **Account Behavior:**
  - CreditScore – Creditworthiness
  - Tenure – Number of years as a customer
  - Balance – Current account balance
  - NumOfProducts – Number of bank products used
  - HasCrCard – Whether the customer has a credit card
  - IsActiveMember – Engagement indicator
- **Financials:**
  - EstimatedSalary – Annual estimated income

This structured dataset provided a strong foundation for building predictive models and drawing actionable business insights.

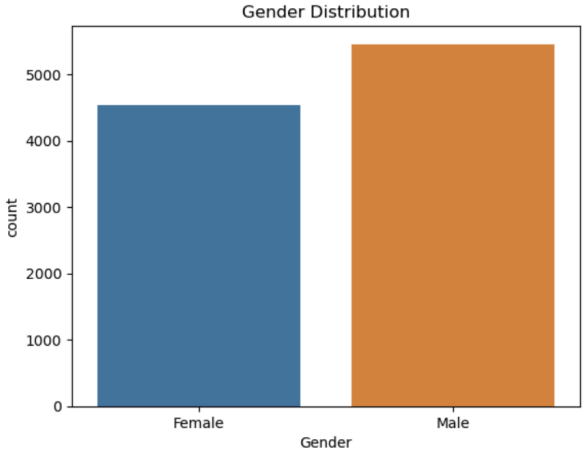
3. Exploratory Data Analysis (EDA)

Exploratory Data Analysis was performed to better understand the structure of the dataset, identify trends, and surface patterns relevant to customer churn.

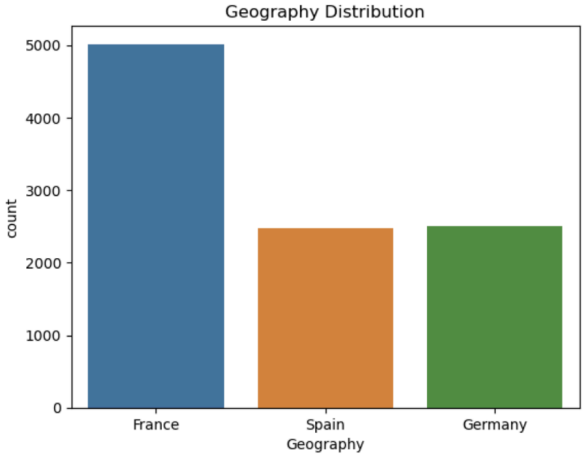
🔍 Key Insights:

- **Churn Rate:** Roughly **20%** of customers have churned, showing a moderate class imbalance.
- **Age & Activity:** Older and less active customers are significantly more likely to churn.
- **Balance Patterns:** Customers with **zero balance** tend to stay, while those with a **high balance-to-salary ratio** show a higher risk of leaving.
- **Demographics:** The customer base is predominantly **Male** and mostly from **France**

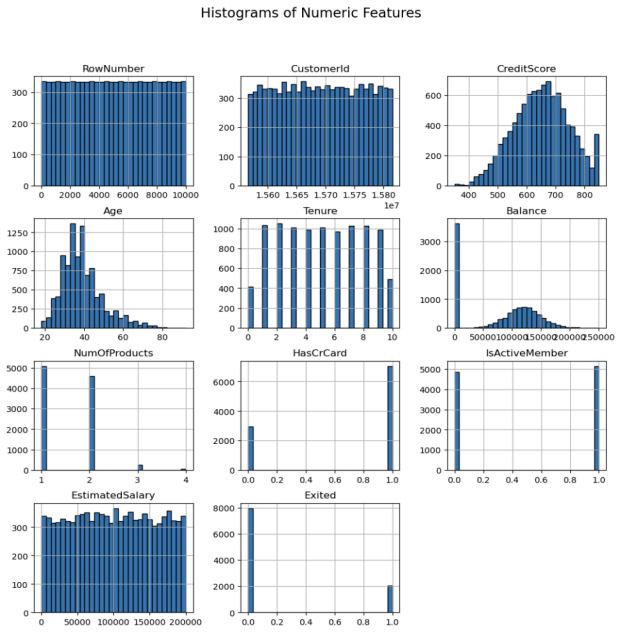
🔹 Gender Distribution



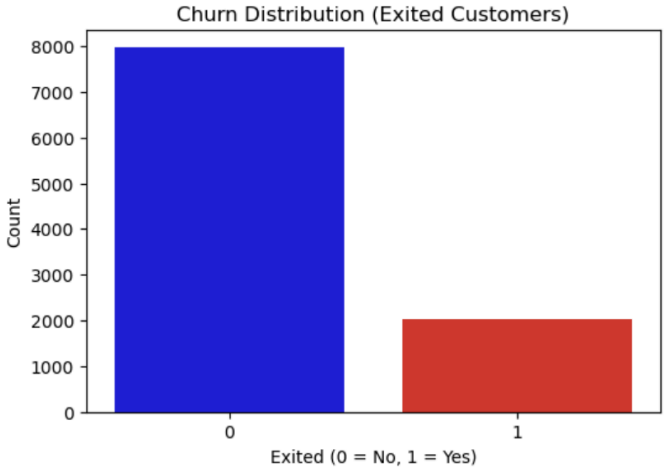
🔹 Geography Distribution



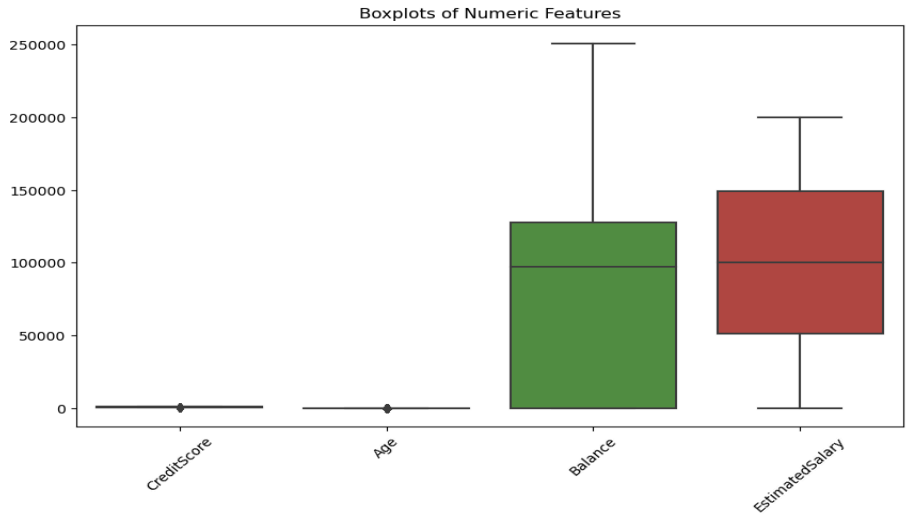
🔹 Histograms of Numeric Features



🔹 Histograms of Numeric Features



🔹 **Outlier Detection (Boxplot) : Revealed outliers in Balance and EstimatedSalary**



4. Data Preprocessing & Feature Engineering

Before building the models, the data was prepared and enhanced to improve performance and interpretability.

🔹 Encoding Categorical Variables

- **Label Encoding** was applied to binary variables like `Gender`.
- **One-Hot Encoding** was used for categorical features like `Geography`, `AgeGroup`, and `TenureGroup` to avoid giving them any ordinal meaning.

🛠️ Engineered Features:

Feature	Description
`BalanceZero`	Binary flag for zero balance
`BalanceToSalaryRatio`	Normalized balance feature
`ProductUsage`	NumOfProducts × IsActiveMember
`Male_Germany`, `Male_Spain`	Interaction terms
`AgeGroup`, `TenureGroup`	Binned categories for segmentation

Feature Scaling

- StandardScaler was applied to normalize numerical columns (e.g., CreditScore, Balance, EstimatedSalary) to ensure models like KNN and SVM perform optimally.

This step ensured that the data was clean, structured, and ready for efficient model training

5. Model Development

Model	Accuracy	Notable Strengths
Gradient Boosting	86.45%	Strong predictive power, handles complexity well
Random Forest	86.25%	Fast, interpretable, robust to overfitting
Logistic Regression	80.45%	High interpretability and fast execution
K-Nearest Neighbors	79.85%	Simple, effective for small datasets
Support Vector Machine (Linear)	79.60%	Works well with high-dimensional data. Poor on recall for churn

Multiple classification algorithms were trained to compare performance:
Evaluation Metrics:

- Accuracy – Overall correctness of predictions
- Precision – Accuracy of churn predictions
- Recall – Ability to catch actual churners
- F1-Score – Balance between precision and recall
- Confusion Matrix – Visual breakdown of correct vs incorrect predictions

📈 **Gradient Boosting** performed the best with **86.45% accuracy**, followed closely by **Random Forest** at **86.25%**.

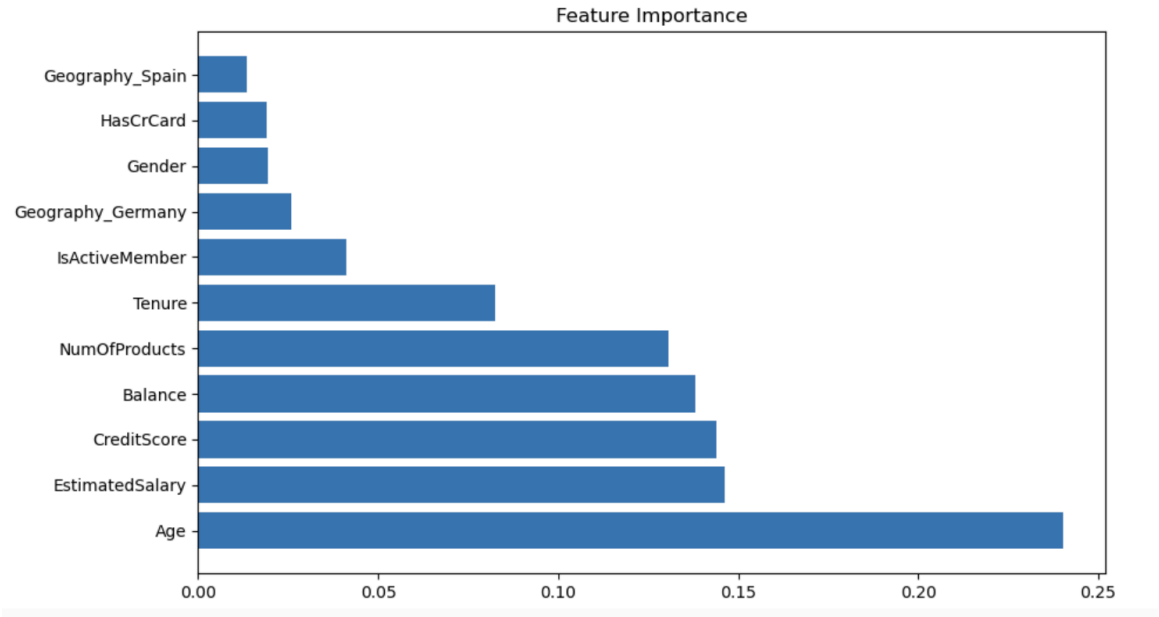
The confusion matrix from the Random Forest model highlights the prediction performance across churn and non-churn classes:

This evaluation helped select the most effective and business-ready model for deployment.

6. Feature Importance Analysis

Using Random Forest:

```
In [28]: # 1st Model I will be using is Random Forest

In [29]: model= RandomForestClassifier(n_estimators=100, random_state=42)
         model.fit(X_train, Y_train)

Out[29]: RandomForestClassifier
         RandomForestClassifier(random_state=42)

In [30]: Y_PRED = model.predict(X_test)

In [31]: conf_matrix= confusion_matrix(Y_test, Y_PRED)
         class_report= classification_report(Y_test, Y_PRED)
         accuracy= accuracy_score(Y_test, Y_PRED)

In [32]: print(conf_matrix)
         print(class_report)
         print(accuracy)

[[1550   57]
 [ 208  185]]
              precision    recall  f1-score   support

           0       0.88      0.96      0.92      1607
           1       0.76      0.47      0.58       393

    accuracy                           0.87      2000
   macro avg       0.82      0.72      0.75      2000
weighted avg       0.86      0.87      0.85      2000

0.8675
```



To understand which features had the biggest impact on churn prediction, I used the feature importance scores from the Random Forest model.

Here's a visual showing how much each variable contributed to the model's decisions:

🔝 Top Influential Features:

- **Age:** The most important factor. Older customers are more likely to leave, especially if not engaged regularly.
- **EstimatedSalary:** Played a strong role in interaction with other features like balance and credit score.
- **CreditScore:** Lower credit scores were slightly more likely to be linked with churn, although it wasn't as strong as Age.
- **Balance:** Higher balances didn't always mean loyalty—especially if the customer wasn't active.
- **NumOfProducts:** Customers with fewer products were more likely to churn. Offering more services could help.
- **Tenure:** Interestingly, customers in the mid-tenure range had a higher churn tendency than those very new or very old.
- **IsActiveMember:** A customer's engagement level made a clear difference in whether they stayed or not.

7. Interpretation of Results

The model results gave us some clear patterns to focus on:

- **Age stood out as the strongest predictor of churn.** Older customers had a much higher likelihood of leaving, especially when other risk factors—like inactivity—were present.

- **Inactivity played a major role.** Customers who weren't actively using their accounts were more likely to churn, even if they had a good credit score or decent balance.

- **Geography had an impact too.** Churn rates were noticeably higher among customers from Germany, which might reflect differences in service expectations or competitive banking options in that market.

- **A high account balance doesn't guarantee loyalty.** In fact, one of the riskiest combinations was customers who had a large balance but showed very low activity—likely indicating disengagement.

8. Business Recommendations

🔹 **Customer Segmentation**

Focus on older and inactive customers—these were the most likely to churn according to the model. We can segment the customer base into risk categories (high, medium, low) based on the churn probability scores and prioritize the high-risk group for retention efforts.

🔹 **Proactive Engagement**

Reach out to high-risk customers through personalized messages—email, phone calls, or even in-app notifications. These messages could include check-ins, financial advice, or offers tailored to their needs. Early engagement could help keep them from leaving.

🔹 **Product Adoption Programs**

Encourage customers to use more than one product. Churn was noticeably lower for customers who were using 2 or more products. We could offer them incentives like bundled services, fee waivers, or loyalty points for adopting additional products like credit cards or savings accounts.

🔹 **Activity-Based Retention**

Keep an eye on customer activity levels. For example, if a user hasn't logged in for a while or stopped using their account actively, we can trigger automated reminders, exclusive offers, or rewards to bring them back.Simple nudges can make a big difference, especially if done before the customer decides to leave.

9. Deployment Plan

- The final Gradient Boosting model can be integrated into the CRM system so that every customer has a churn risk score attached to their profile. This score can update regularly based on the most recent activity and financial data.
- A simple internal dashboard can be built to show weekly churn predictions. Teams like marketing or customer success can use this to quickly identify which customers need attention.
- We can also set up automatic alerts for users who hit a high churn risk threshold (say, 80% or more). These alerts can trigger emails, special offers, or even assign a rep to reach out directly.
- The entire setup doesn't need to be complex—it can start as a batch process, refreshing once a week, and evolve over time into something more real-time if needed.
- Integrate the trained Gradient Boosting model into CRM systems.
- Create a churn scoring dashboard updated weekly.
- Build alert systems for high-probability churn customers.

10. Next Steps

To ensure this churn prediction model remains reliable and production-ready, I've proposed the following steps for future improvement and deployment:

🔹 **Hyperparameter Optimization with GridSearchCV**

I plan to use GridSearchCV to fine-tune the model's parameters and improve performance beyond the current baseline. This will help identify the best combination of model settings, such as the number of trees, depth, and split criteria, and optimize accuracy and recall.

🔹 **Addressing Class Imbalance with SMOTE or Class Weights**

Given that churn cases represent only around 20% of the dataset, I will address this imbalance using two approaches:

- **SMOTE** to oversample the minority class, or
- **Class weights** within the model to penalize misclassified churn predictions.

This step is important to improve the model's sensitivity toward actual churners and reduce false negatives.

🔹 **Deploying a Real-Time Churn Prediction API**

As a next step, I plan to wrap the final trained model into a simple web API using Flask or FastAPI. This will allow real-time churn predictions to be integrated directly into customer-facing systems like CRMs or dashboards—enabling immediate action on high-risk accounts.

🔹 **Monitoring for Model Drift and Scheduled Retraining**

To keep the model relevant over time, I propose a quarterly retraining schedule. Customer behavior and engagement patterns can shift, and regular monitoring of performance metrics (like recall and F1-score) will help identify when retraining is necessary. This ensures the model remains accurate and actionable in the long term.

11. Conclusion

This project successfully built a machine learning model to predict customer churn using real-world banking data. By engineering new features, testing multiple classification algorithms, and carefully interpreting the results, we were able to identify key drivers of churn like age, inactivity, and limited product usage.

The Gradient Boosting model gave the best performance, reaching **86.45% accuracy**, and proved reliable for identifying at-risk customers. More importantly, the insights gained from the analysis can now guide business decisions—helping the bank take action before valuable customers leave.

Overall, this project shows how data and machine learning can be used not just for prediction, but to shape strategy in a meaningful, measurable way.