Decoding Customer Churn: A Data Science Approach

Nikhil Kotha

February 17, 2025

1 Exploratory Data Analysis

Customer churn refers to the percentage of customers who stop using a company's services over a given period. In the banking sector, tracking and understanding churn is essential for maintaining a stable customer base and ensuring long-term profitability.

1.1 Customer Churn Distribution

The pie chart below illustrates the proportion of customers who have churned versus those who have been retained.

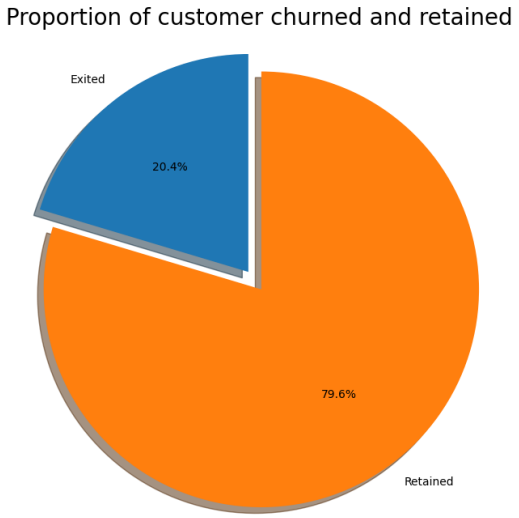

Figure 1: Proportion of customer churned and retained.

Definition of Churn Rate:

$$\text{Churn Rate} = \left(\frac{\text{Number of Customers Who Left}}{\text{Total Number of Customers}} \right) \times 100$$

Importance of Churn Analysis:

- Helps businesses proactively identify customers at risk of leaving.
- Guides the development of retention strategies and personalized interventions.
- Reduces acquisition costs since retaining existing customers is generally cheaper than acquiring new ones.

Observations from the Pie Chart:

- Approximately 20.4% of customers have churned, while 79.6% remain with the bank.
- The class imbalance suggests a need for resampling techniques like SMOTE or undersampling to improve model training.

1.2 Categorical Feature Distribution

Categorical variables can significantly influence the likelihood of customer churn.

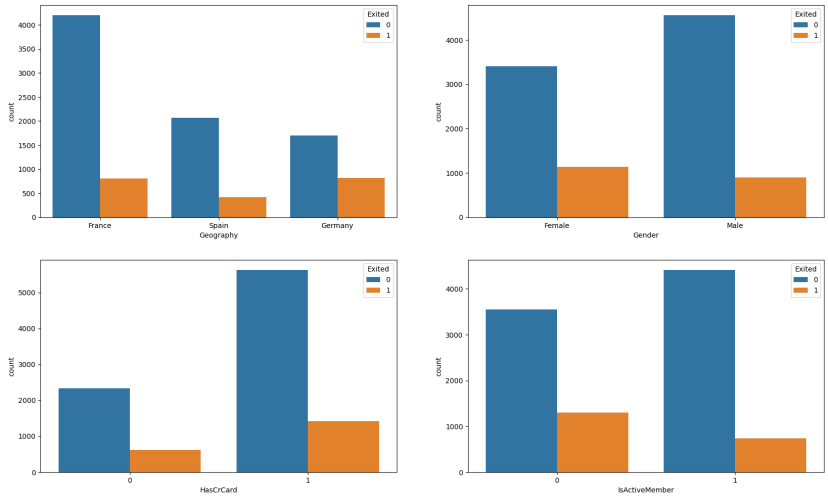

Figure 2: Distribution of categorical features in churned and retained customers.

Definitions:

- **Geography:** Customer's country (e.g., France, Spain, Germany). Economic and cultural factors can impact satisfaction.

- **Gender:** Male or Female. Financial product usage and churn behavior may differ slightly by gender.
- **HasCrCard:** Indicates credit card ownership. A lack of a credit card may suggest lower product engagement.
- **IsActiveMember:** Signifies active engagement with the bank. An inactive member often has minimal product usage.

Observations from Bar Plots:

- Customers in Germany exhibit a higher churn rate than those in France or Spain, possibly due to local competition or service preferences.
- Female customers show a slightly higher churn rate than male customers, indicating possible differences in product fit or service experience.
- Customers without a credit card are more likely to churn, implying that additional banking products may increase loyalty.
- Inactive members tend to leave the bank at higher rates, suggesting a correlation between low engagement and churn.

1.3 Numerical Feature Distributions

Numerical attributes (e.g., Credit Score, Age, Tenure, Balance, Number of Products, Estimated Salary) often play a key role in predicting churn.

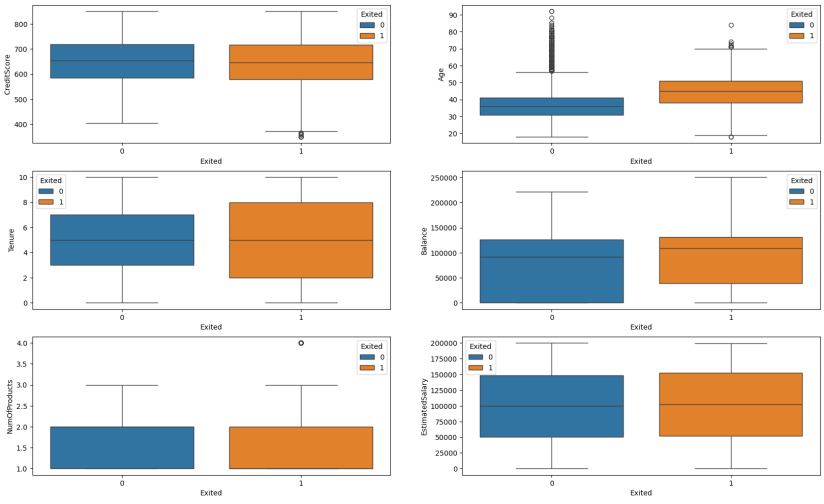

Figure 3: Box plots of numerical features based on customer churn status.

Definitions:

- **Credit Score:** A numeric representation of customer creditworthiness.
- **Age:** Could correlate with banking needs and loyalty patterns.
- **Tenure:** The number of years a customer has been with the bank. Higher tenure often indicates stronger loyalty.
- **Balance:** Total money in the account. Extremely high or low balances might correlate with churn.
- **NumOfProducts:** The number of products a customer uses (e.g., loans, credit cards). Multi-product usage often increases retention.
- **Estimated Salary:** Higher income customers might have different financial needs and churn tendencies.

Observations from Box Plots:

- Churned customers generally have higher balances than retained ones, possibly indicating idle funds or underutilized relationships.
- Older customers tend to churn more frequently, potentially due to changing financial priorities or retirement planning.
- Customers with fewer products (1 or 2) have a higher likelihood of leaving, suggesting limited engagement.

2 Feature Engineering

Feature engineering is the process of creating additional or transformed features from raw data to improve model performance.

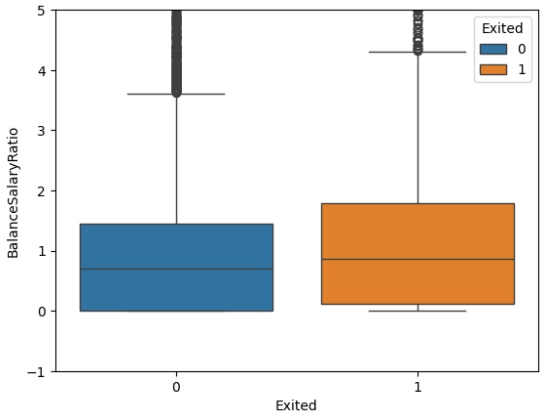

Figure 4: Balance-to-Salary Ratio for churned vs. retained customers.

Newly Engineered Features:

- **Balance-to-Salary Ratio**:

$$\text{Balance-Salary Ratio} = \frac{\text{Balance}}{\text{Estimated Salary}}$$

This metric helps compare customer savings relative to their income.

  - A high ratio might indicate idle balances or dissatisfaction with products.
  - Customers with a high ratio in the churned group highlight potential missed cross-sell opportunities.

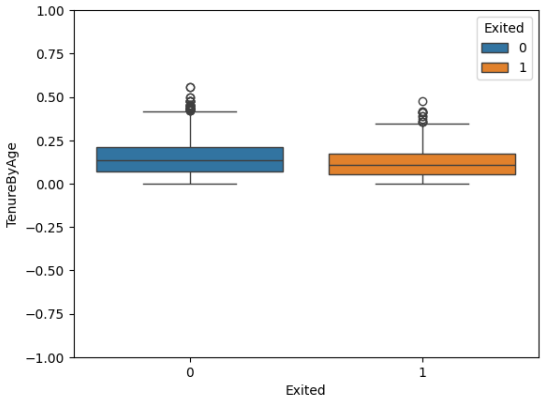

Figure 5: Tenure-to-Age Ratio for churned vs. retained customers.

- **Tenure-to-Age Ratio**:

$$\text{Tenure-to-Age Ratio} = \frac{\text{Tenure}}{\text{Age}}$$

Standardizing tenure by age provides insight into customers' relative longevity with the bank.

  - A lower ratio may correlate with weaker long-term engagement.
  - Customers below a certain threshold could be flagged for targeted retention campaigns.

Observations from Feature Engineering Plots:

- Customers with a higher **Balance-to-Salary Ratio** appear more likely to churn.
- Customers with a **shorter tenure relative to their age** also exhibit a higher propensity to leave.

3 Model Evaluation and Performance

3.1 ROC-AUC Curve

The Receiver Operating Characteristic (ROC) curve assesses the trade-off between True Positive Rate (Sensitivity) and False Positive Rate. The Area Under the Curve (AUC) provides an aggregate measure of performance.

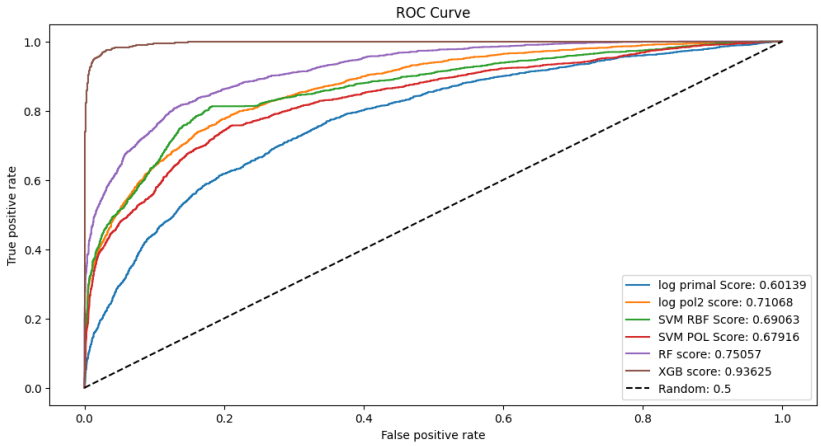

Figure 6: ROC Curve comparison for different machine learning models.

Observations:

- **XGBoost** has the highest ROC-AUC, indicating its strong predictive capabilities.
- **Random Forest** and **Logistic Regression (with polynomial features)** also show competitive performance.
- **SVM** models exhibit comparatively lower ROC-AUC, which could be due to feature scaling or parameter settings.

Metrics to Note:

- **Accuracy:** Overall rate of correct predictions.
- **Precision & Recall:** Important for focusing on actual churners (recall) and avoiding false alarms (precision).
- **F1-Score:** Harmonic mean of precision and recall.

4 Conclusion

- **Key Findings:**

  - Customers in Germany, inactive members, and those without a credit card exhibit higher churn.
  - High **balance-to-salary ratio** and low **tenure-to-age ratio** correlate with churn risk.
  - **XGBoost** outperforms other models in predictive power, offering valuable insights into churn drivers.

- **Recommended Actions:**
  - Implement targeted strategies (e.g., personalized offers) for high-risk customers.
  - Encourage multiple product usage to deepen engagement and reduce churn.
  - Continuously monitor model performance and retrain to keep predictions accurate.

Appendix: Code Implementation

Data Preprocessing and Feature Engineering

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Load dataset
df = pd.read_csv('Churn_Modelling.csv')

# Drop unnecessary columns
df.drop(['RowNumber', 'CustomerId', 'Surname'], axis=1, inplace=
    True)

# Convert categorical variables
df['Geography'] = df['Geography'].astype('category').cat.codes
df['Gender'] = df['Gender'].map({'Male': 1, 'Female': 0})

# Feature engineering
df['BalanceSalaryRatio'] = df['Balance'] / (df['EstimatedSalary'] +
     1e-9)
df['TenureToAgeRatio'] = df['Tenure'] / (df['Age'] + 1e-9)

# Separate features and target
X = df.drop('Exited', axis=1)
y = df['Exited']

# Split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Scaling numerical features
scaler = StandardScaler()
num_cols = ['CreditScore', 'Age', 'Balance',
            'EstimatedSalary', 'BalanceSalaryRatio', '
                TenureToAgeRatio']

X_train[num_cols] = scaler.fit_transform(X_train[num_cols])
X_test[num_cols] = scaler.transform(X_test[num_cols])
```

Model Training and Evaluation

```
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report,
    roc_auc_score

# Random Forest
rf_model = RandomForestClassifier(n_estimators=100, random_state
    =42)
```

```
rf_model.fit(X_train, y_train)
rf_preds = rf_model.predict(X_test)
rf_probs = rf_model.predict_proba(X_test)[:, 1]

# Evaluate Random Forest
rf_acc = accuracy_score(y_test, rf_preds)
rf_auc = roc_auc_score(y_test, rf_probs)
print("Random Forest Accuracy:", rf_acc)
print("Random Forest AUC:", rf_auc)
print(classification_report(y_test, rf_preds))

# XGBoost
xgb_model = XGBClassifier(use_label_encoder=False, eval_metric='
    logloss')
xgb_model.fit(X_train, y_train)
xgb_preds = xgb_model.predict(X_test)
xgb_probs = xgb_model.predict_proba(X_test)[:, 1]

# Evaluate XGBoost
xgb_acc = accuracy_score(y_test, xgb_preds)
xgb_auc = roc_auc_score(y_test, xgb_probs)
print("XGBoost Accuracy:", xgb_acc)
print("XGBoost AUC:", xgb_auc)
print(classification_report(y_test, xgb_preds))

# Logistic Regression
log_model = LogisticRegression(max_iter=300)
log_model.fit(X_train, y_train)
log_preds = log_model.predict(X_test)
log_probs = log_model.predict_proba(X_test)[:, 1]

# Evaluate Logistic Regression
log_acc = accuracy_score(y_test, log_preds)
log_auc = roc_auc_score(y_test, log_probs)
print("Logistic Regression Accuracy:", log_acc)
print("Logistic Regression AUC:", log_auc)
print(classification_report(y_test, log_preds))
```

Notes:

- **Random Forest** provides a robust baseline with relatively high accuracy.
- **XGBoost** typically yields higher AUC scores, showcasing strong performance for imbalanced datasets.
- **Logistic Regression** is straightforward and interpretable, useful for baseline comparisons.