Credit Card Customer Churn Prediction

1. Business Situation

A manager at a bank is disturbed with more and more customers leaving their credit card services. He is looking for a data-driven solution to proactively identify customers likely to churn, so they can take preventive actions and retain them.

2. Key Problems and Objective

The key problem is the increasing number of customers leaving the credit card services. The objective of this project is to build a predictive model that can accurately classify whether a customer is likely to churn or not, enabling targeted retention strategies.

3. Dataset Overview

- Source: Kaggle - Credit Card customers (www.kaggle.com/datasets/sakshigoyal7/credit-card-customers)
- Records: 10,127 customers
- Features: 21 features including demographics, credit usage, and customer activity
- Target Variable: Attrition_Flag (Existing vs. Attrited Customer)
- Churn Rate: 16.07%

4. Tools and Techniques Used

- Programming: Python
- Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, xgboost, shap, flask, joblib
- Visualization: seaborn, matplotlib, SHAP
- Model Deployment: Flask Web API with HTML forms
- Model Persistence: joblib
- Environment: Jupyter Lab, Anaconda

5. Data Preprocessing

- Dropped irrelevant column: CLIENTNUM
- Handled ordinal features with category ordering (Education_Level, Income_Category)
- Applied one-hot encoding to categorical features
- Performed feature selection based on correlation analysis
- Standardized numerical features using StandardScaler

6. Exploratory Data Analysis

- Distribution analysis of churn vs. non-churn
- Count plots of categorical variables against churn
- Box plots of numerical variables against churn
- Correlation heatmaps for feature relationships
- Target-wise percentage breakdown per category for insights

7. Model Building

Trained and evaluated four classification models:

- Logistic Regression
- Naive Bayes Classifier
- Random Forest Classifier
- XGBoost Classifier (final model)

All models trained on a stratified 80/20 train-test split.

8. Evaluation Metrics

- Accuracy
- Precision, Recall, F1-Score
- ROC-AUC Score

- XGBoost outperformed all models with the best F1 and ROC-AUC scores

- Feature importance and SHAP values used for interpretability

9. Key Takeaways

- Churn is low rate (16%) (classification report will be needed for evaluation)

- Transaction-related features (Total_Trans_Ct, Amt_Change, Utilization) are key churn indicators

- XGBoost provides the best balance of accuracy and interpretability

- Flask app allows easy real-time predictions via user input

10. Resources

GitHub Repo

Kaggle Notebook

Flask API Demo