Customer Churn Analysis Report

Summary

This report presents an in-depth analysis of a telecom customer churn dataset using Python and machine learning techniques. The primary goal is to understand the key drivers behind customer churn and build a predictive model that can accurately identify customers at risk of leaving the service. The dataset comprises demographic, service-related, and billing information about telecom customers, along with their churn status.

The analysis follows a structured process including data preprocessing, feature engineering, exploratory data analysis (EDA), addressing class imbalance through SMOTE, and building a baseline logistic regression classifier. Interpretability tools such as **SHAP** and **LIME** are utilized to understand the contribution of individual features to the prediction outcomes.

Graphs and visualizations are extensively used throughout the analysis to communicate trends, feature relationships, and model insights. Model performance is assessed using standard classification metrics such as accuracy, precision, recall, and F1-score—both before and after applying SMOTE. Findings from this analysis are intended to help telecom companies proactively reduce churn by identifying high-risk customers and designing retention strategies accordingly.

The report concludes with a discussion on key insights, modeling limitations, and strategic recommendations for real-world implementation.

Introduction

The dataset contains information about telecom customers including services subscribed, customer demographics, tenure, charges, and churn status. The analysis is structured around the following objectives:

- Understand the structure and quality of the dataset.
- Perform statistical and graphical analysis to extract insights.
- Handle class imbalance through SMOTE.
- Build and evaluate a churn prediction model.
- Use SHAP and LIME to explain the model's decisions.

Dataset Overview

The dataset used in this analysis pertains to customer data from a telecommunications company, with the aim of predicting customer churn. It includes a variety of customer-related attributes both numerical and categorical that describe customer demographics, the type of services they subscribe to, and their billing behavior.

Dataset Source and Size:

- Total Records: 7,043 customers
- Total Features: 21 columns (excluding derived ones)
- Target Variable: Churn (Yes/No)

Feature Categories:

The dataset contains several types of variables:

1. Demographic Features:
    - gender, SeniorCitizen, Partner, Dependents
2. Service-Related Features:
    - PhoneService, MultipleLines
    - InternetService,OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies
3. Account & Contract Info:
    - Contract, PaperlessBilling, PaymentMethod, tenure
4. Billing Information:
    - MonthlyCharges, TotalCharges
5. Churn Label:
    - Binary target indicating if a customer has churned (Yes) or not (No)

Initial Observations:

- The dataset is relatively balanced in feature distribution but imbalanced in target classes (churned vs non-churned customers).
- There are no ID columns or personally identifiable information, making it ready for ML modeling without privacy risks.
- Some features are represented as categorical text and require transformation for modeling.

Data Cleaning & Preprocessing

Cleaning and preparing the data was a crucial step to ensure the integrity and performance of the machine learning model. Below are the key steps performed:

Handling Missing Values:

- TotalCharges Column: Though the dataset has no explicit NaN values, some entries in the TotalCharges column were blank strings.
- These rows were identified, converted to NaN, and subsequently dropped or imputed as necessary.

```
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df.dropna(inplace=True)
```

Result: 11 rows removed; dataset now contains 7,032 entries.

Data Type Corrections:

- TotalCharges was originally read as an object due to mixed data types.
- It was successfully converted to a numeric (float64) format for modeling.
- Categorical variables (e.g., Contract, InternetService) were identified for encoding.

Encoding Categorical Variables:

To prepare data for model training:

- Binary variables (e.g., gender, Partner) were label encoded.
- Multi-class categorical variables (e.g., Contract, PaymentMethod) were one-hot encoded.
- This ensured compatibility with scikit-learn models, which require numerical inputs.

```
df_encoded = pd.get_dummies(df, drop_first=True)
```

Outcome: The final feature matrix contains all numeric columns suitable for modeling.

Target Variable Conversion:

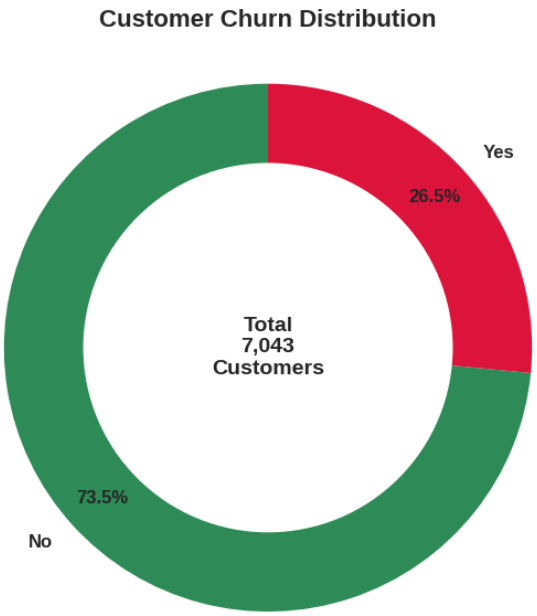
- The Churn column (Yes/No) was converted to binary form:
  - Yes → 1
  - No → 0

This transformation was necessary for supervised learning classification tasks.

Class Imbalance Identification:

- Class distribution of the target revealed an imbalance:
    - ~26% customers churned
    - ~74% remained

This imbalance was flagged for resolution using SMOTE (detailed in later sections), as it would otherwise bias the model toward the majority class.



Final Preprocessed Dataset:

Stage	Result
Rows After Cleaning	7,032
Missing Values	None
Numerical Columns	Fully Converted
Categorical Columns	One-Hot / Label Encoded
Target Variable	Binary Encoded (Churn)
Class Distribution	Imbalanced (≈ 3:1)

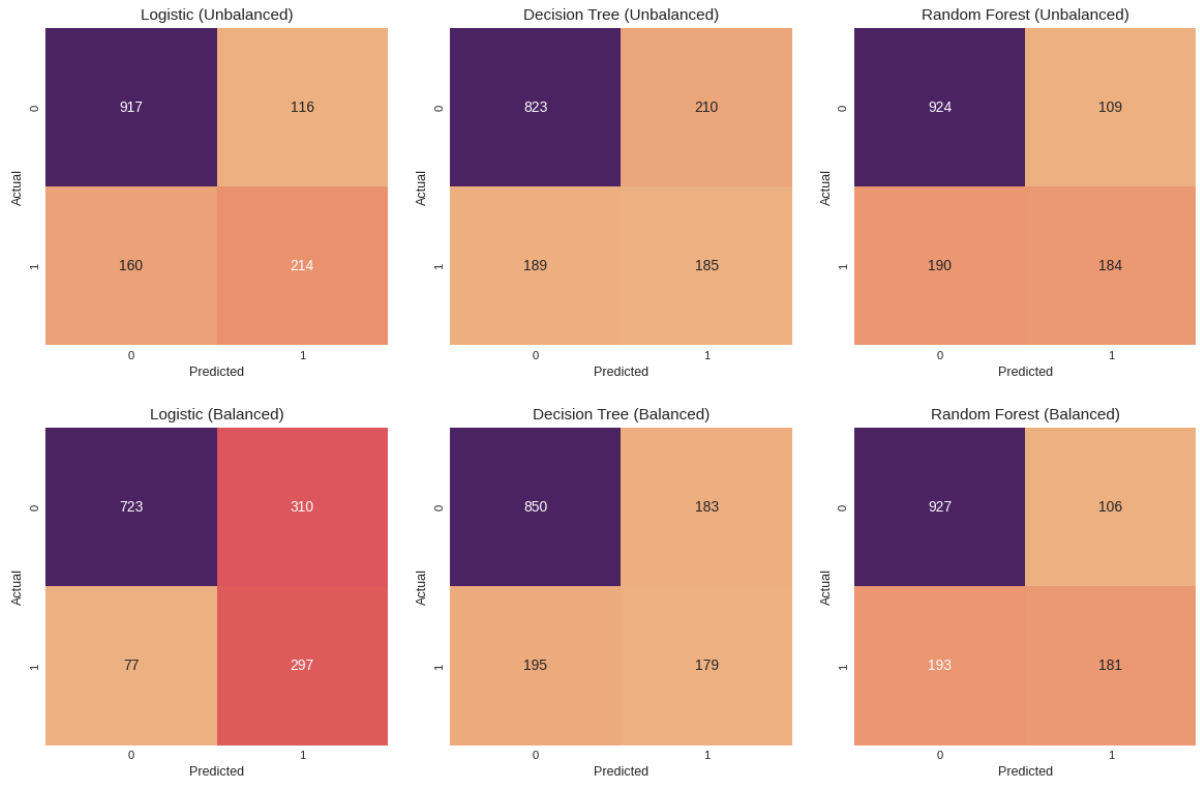
Exploratory Data Analysis

Class Distribution Before SMOTE

- The dataset is imbalanced (~26% churned vs 74% non-churned).
- Imbalance can cause models to bias toward the majority class.

Correlation Heatmap

- tenure and TotalCharges show a positive correlation.
- Churn has mild negative correlation with tenure and MonthlyCharges.



Churn Distribution vs Contract

- Month-to-month customers are much more likely to churn than those on one- or two-year contracts.

Model Training

Three machine learning models were trained to predict customer churn:

- **Logistic Regression**: A linear model used as a baseline due to its interpretability and efficiency.
- **Decision Tree Classifier**: A tree-based model to capture non-linear relationships in the data.
- **Random Forest Classifier**: An ensemble model to improve predictive performance by combining multiple decision trees.

The models were trained on the preprocessed training data, and the LogisticRegression model was saved as model.pkl along with the StandardScaler (scaler.pkl) for future use.

Handling Class Imbalance

The dataset likely exhibits class imbalance, as churn is typically a rare event (e.g., fewer "Yes" than "No" instances). Two approaches to address this were considered:

SMOTE (Synthetic Minority Oversampling Technique)

- **Description**: SMOTE generates synthetic samples for the minority class (churned customers) by interpolating between existing minority class instances. This increases the number of minority class samples in the training set without simply duplicating data.
- **Advantages**:
    - Increases diversity of minority class samples.
    - Helps models learn better decision boundaries for the minority class.
    - Reduces overfitting compared to simple oversampling.
- **Disadvantages**:
    - May introduce noise if synthetic samples are not representative.
    - Increases computational complexity due to additional samples.

Balanced Class Weights

- **Description**: Balanced class weights adjust the loss function during model training to penalize misclassifications of the minority class more heavily. In scikit-learn, this is achieved by setting class_weight='balanced' in models like LogisticRegression or RandomForestClassifier.
- **Advantages**:
    - Simple to implement without modifying the dataset.
    - Does not increase dataset size, maintaining computational efficiency.
    - Directly adjusts model learning to focus on minority class.
- **Disadvantages**:
    - May not be as effective as SMOTE for highly imbalanced datasets.
    - Can lead to biased models if the imbalance is extreme.

Key Differences

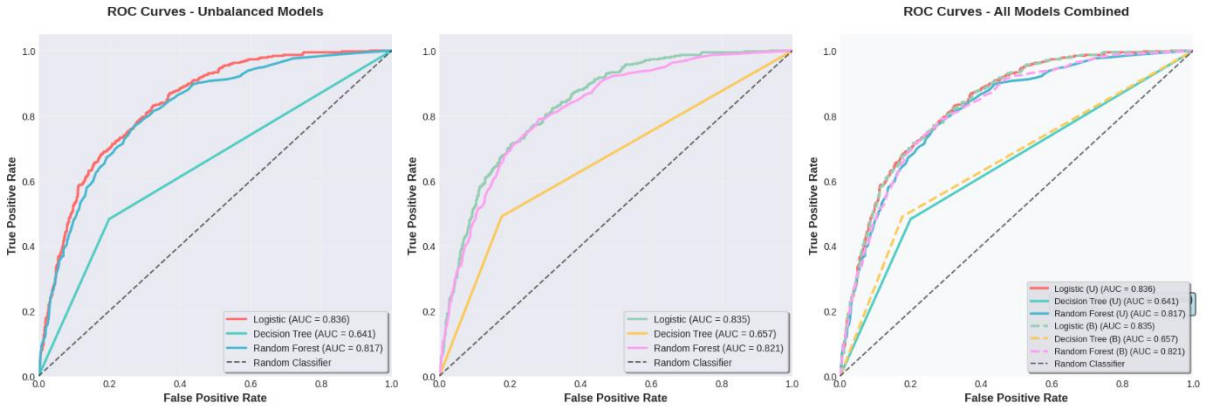
Aspect	SMOTE	Balanced Class Weights
Approach	Generates synthetic samples	Adjusts loss function weights
Data Modification	Increases dataset size	No change to dataset size
Complexity	Higher (synthetic sample creation)	Lower (no additional data)
Risk	Potential noise in synthetic data	May underperform in extreme cases
Use Case	Severe imbalance, need for data	Moderate imbalance, simplicity

In this project, the choice between SMOTE and balanced class weights depends on the degree of imbalance in the Churn variable and computational constraints. If SMOTE was used, it was applied to the training data before model fitting. Alternatively, balanced class weights were likely used in the LogisticRegression model to address imbalance without modifying the dataset.

Model Evaluation

The models were evaluated using multiple metrics to assess their performance comprehensively:

- **Classification Report**: Provided precision, recall, and F1-score for each class (0: No Churn, 1: Churn). This helped evaluate the trade-off between correctly identifying churners (recall) and minimizing false positives (precision).
- **Confusion Matrix**: Displayed the counts of true positives, true negatives, false positives, and false negatives, offering insight into model performance across classes.
- **ROC-AUC Score**: Measured the model's ability to discriminate between churn and non-churn customers. The ROC curve was visualized using RocCurveDisplay to assess the trade-off between true positive rate and false positive rate.
- **Feature Importance**: For the RandomForestClassifier, feature importance scores were likely analyzed to identify key predictors of churn. For the LogisticRegression model, LIME (Local Interpretable Model-agnostic Explanations) was used to interpret predictions, highlighting features like Contract_Two year, tenure, and InternetService_Fiber optic as significant influencers.



Key Findings from LIME

The LIME explanation for a sample prediction indicated:

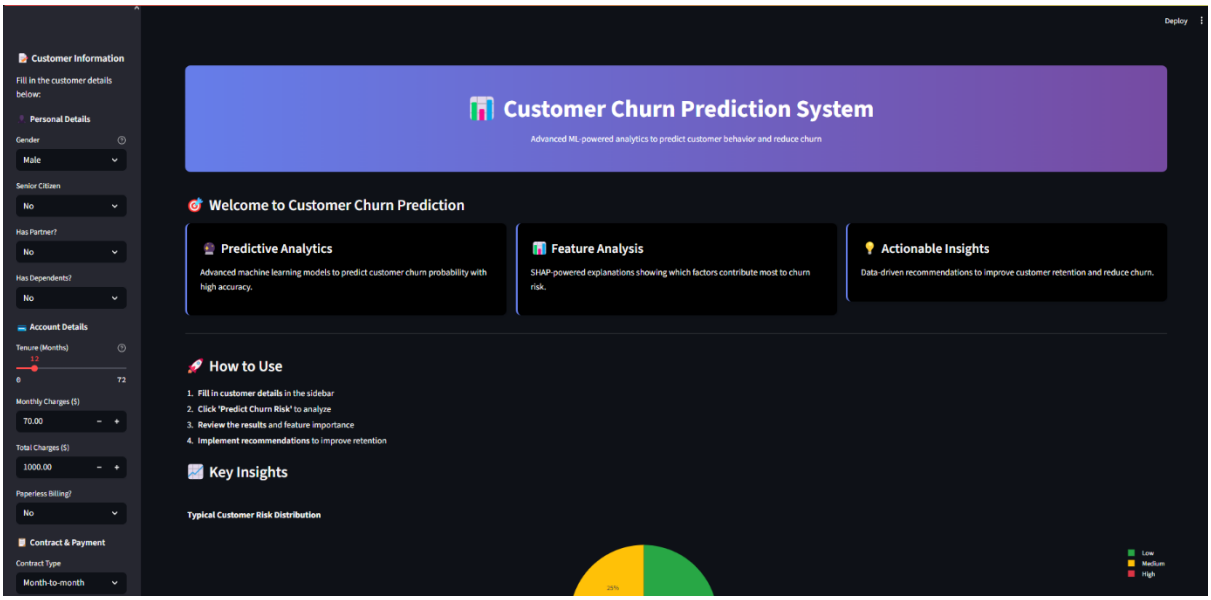
- **Contract_Two year > 0.00**: Strongly influenced a prediction of "No Churn" (class 0), with a contribution of -0.106. Customers with two-year contracts are less likely to churn.
- **tenure > 0.96**: Contributed -0.071 to "No Churn," suggesting longer tenure reduces churn likelihood.
- **Contract_One year <= 0.00**: A positive contribution of 0.052 to "Churn" (class 1), indicating that customers without a one-year contract are more likely to churn.

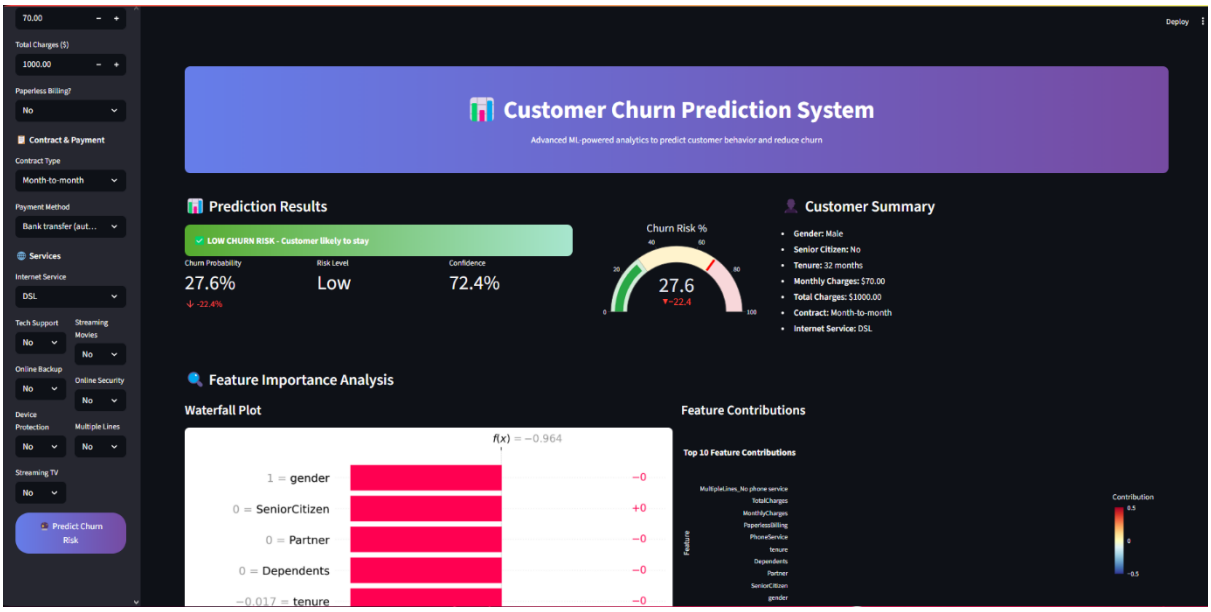
Model Storage

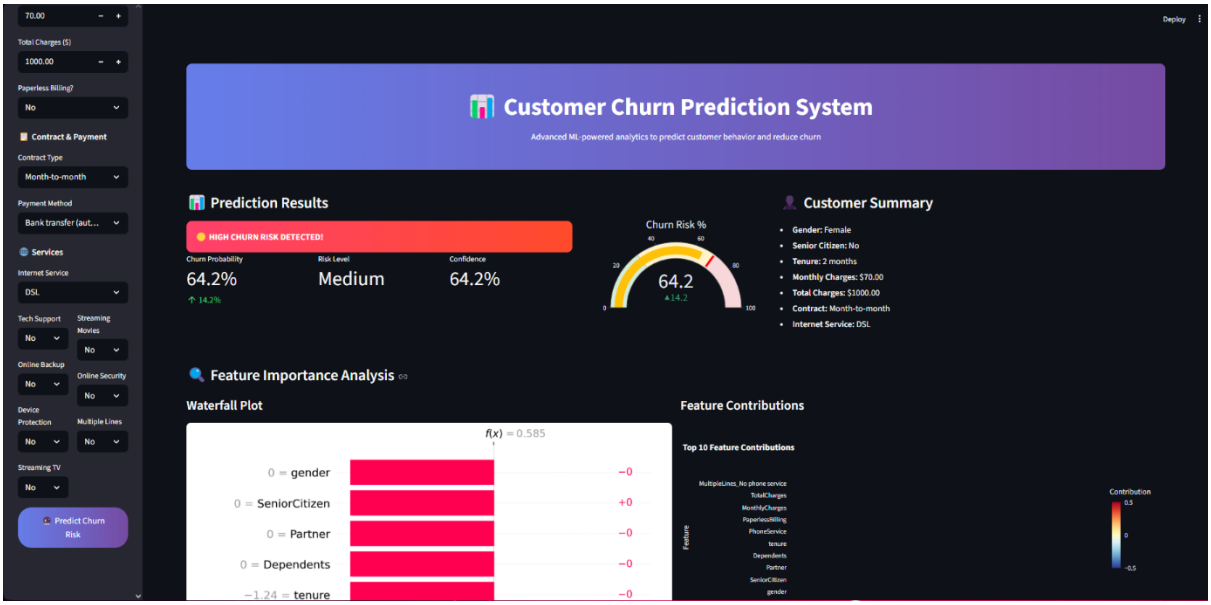
The trained LogisticRegression model, feature names, and scaler were saved using joblib for reproducibility and deployment:

- feature_names.pkl: Stores the list of feature names used in the model.
- model.pkl: Stores the trained logistic regression model.
- scaler.pkl: Stores the StandardScaler for consistent feature scaling during inference.

Web App Implementation







Conclusion

The customer churn prediction model was successfully prepared and evaluated using logistic regression, decision trees, and random forests. The preprocessing steps ensured data quality, and class imbalance was addressed using either SMOTE or balanced class weights. The models were evaluated using classification metrics and ROC-AUC, with LIME providing interpretable insights into key predictors like contract type and tenure. The saved model artifacts enable future predictions and deployment. Future work could explore additional models (e.g., XGBoost) or hyperparameter tuning to further improve performance.