Customer Churn Prediction

Sara Abdelghany

1. Overview

The goal of this project was to build a predictive model to identify customer churn, using The Orange Telecom's Churn Dataset. This report outlines the methodology, key findings, and model performance.

2. Data

The analysis utilizes an 80/20 split dataset, which includes customer-level information such as geographic location and service usage patterns. The dataset consists of 3333 samples and 20 columns.

3. Approach

3.1 Data Exploration and Preprocessing

- **Exploratory Data Analysis (EDA)**: Identified trends, checked for missing values, duplicates, outliers, and the distribution of features. Key observations included:
  - Imbalanced target variable (churn vs. non-churn customers).
  - Features with high multicollinearity.
  - Features with different scales.
- **Preprocessing Steps**:
  - Handled multicollinearity by dropping highly correlated features.
  - Encoded categorical variables using target and one-hot encoding.
  - Addressed class imbalance through SMOTE oversampling and class-weighting.
  - Scaled features using StandardScaler.

3.2 Model Building and Evaluation

- **Models Trained**: Ensemble methods, such as **Random Forest** and **XGBoost**, were chosen because they are effective in handling imbalanced datasets, **Decision Tree** and **Logistic Regression** were included for their simplicity and interpretability.

- **Hyperparameter Tuning**: GridSearchCV and Stratified K-Fold cross-validation were used to optimize hyperparameters for each model.
- **Evaluation Metrics**: Since the dataset is imbalanced, accuracy alone is not sufficient to evaluate model performance, so the following metrics were used
  - Accuracy
  - Recall
  - Precision
  - F1 Score
  - ROC-AUC

4. Findings

Model Comparison

Compared model performance across different data handling approaches (unbalanced, SMOTE, class-weighted)

Model Performance

- **XGBoost** consistently delivered the best results across all datasets, particularly with class weighting on the unbalanced dataset, achieving the highest F1-score (0.95), Accuracy (0.95), and strong ROC AUC (0.93).
- **Random Forest** also peformed best with class weighting (F1-score: 0.94, Accuracy: 0.94), and achieved the highest ROC AUC (0.94) among all models.
- **Decision Tree** performed best on the unbalanced dataset without weight adjustments (F1-score: 0.94, Accuracy: 0.94), following closely behind XGBoost in the same scenario, but its performance dropped significantly on the balanced dataset.
- **Logistic Regression** consistently underperformed compared to tree-based models, even after balancing and class weighting, with its best F1-score (0.83) and Accuracy (0.86) observed without class weighting on the unbalanced dataset.
- **Impact of Class Imbalance Handling:** Class-weighting techniques generally proved more effective than SMOTE in mitigating the impact of class imbalance and improving model performance, especially for XGBoost and Random Forest.

5. Conclusion

- **Model Recommendation**: The XGBoost Classifier, particularly with class-weighting, is recommended for deployment due to its superior performance with an F1-score of 0.95 and accuracy of 0.95.
- **Importance of Data Preprocessing**: Proper data preprocessing, including techniques to address class imbalance, considerably impacts model performance.