Yangruonan Lin, Cherie Xu, Jason Cheung, Eric Xi, David Vivish (Group #13)
COMS 4995: Applied Machine Learning (Project Deliverable #3)
Fall Semester 2024

Problem Statement
In an industry with low product differentiation and increasing competition from emerging low-cost and digital players, telcos today face significant challenges in customer retention. This pressure on retention is further compounded by shrinking margins, which limit marketing budgets. To overcome this, telcos must implement cost-effective strategies that boost customer loyalty and proactively reduce churn.

This project focuses on churn detection by leveraging datasets from a telco to identify early indicators of churn. Using predictive modeling, we aim to pinpoint high-value, at-risk customers early, enabling the implementation of targeted strategies to reduce churn and improve retention.

Datasets Analysis and Cleaning
Multiple datasets were imported from Excel, each capturing distinct aspects of the customer, including customer demographics, location properties, subscribed services, past behavior, and customer churn status. These datasets were merged and the overlapping columns were then deduplicated.

After checking for missing values, only the columns "Total Charges," "Offer," "Churn Reason," and "Churn Category" contained missing data. For "Total Charges," rows with missing values were removed entirely, as they represented only 11 entries. In the "Offer" column, missing values were filled with 'None' to indicate that no offers had been received by those customers. For "Churn Reason" and "Churn Category," missing values were filled with 'Did Not Churn' to reflect that these customers had not discontinued services. This approach ensured a consistent dataset without compromising key information.

Exploratory Data Analysis (EDA)
To perform our EDA, we broke the features into two groups, categorical and numerical, and explored each feature in the group to analyze them better and to find any hidden patterns or issues that may affect our model training. To do this, we decided to use side-by-side barplots and mosaic plots for the categorical group and density histogram and side-by-side boxplots for the numerical group.

For the categorical group, the insight we gained from the side-by-side barplot and mosaic plot was further evidence that supported our initial claim that the target variable class "Churn Value" is pretty imbalanced. This can be seen from the barplot as the bar for the "Not Churn" class was always taller than the bar for the "Churned" class, and for the mosaic plot, the block size for "Not Churned" was always bigger than "Churned." Additionally, we learned that some features had weak, medium, strong, or no collinearity with the target variable class. As a result, this formed our decision to use stratified splitting and to check and remove any collinearity when pre-processing later to help improve our model's accuracy.

For the numerical group, the insight we gained from the density barplot/histogram and side-by-side boxplots is that some of the features with respect to the target variable class were relatively skewed, highly spread out, and had significant modality or clusters. Furthermore, some features had extreme outliers that could harm our model's performance. Therefore, this led us also to include the outlier removal method, feature transformation, and scaling when pre-processing later.

Pre-processing

Given our EDA analysis, we first clipped each feature in the numerical group using IQR to not include extreme outliers. Next, we performed an 80 - 20 stratified split on our dataset and mixed encoded the categorical group using One Hot and Ordinal Encoding to convert them to numerical. The reason for these encoding choices is that we noticed some categorical features had very few unique values, which makes them suitable to one hot encode as it will not significantly increase dimensionality, and that some of the feature values had a unique sequential order to maintain which is best suited for ordinal encoding.

Then, we performed various methods on each feature of the numerical groups, like KNN imputation, to handle any missing values and the “Yeo Johnson” method to handle skewness and get them closer to normal distribution. The reason for these methods came from other people who had worked on this dataset and used the following process. Afterward, we performed min-max scaling to ensure all features in the numerical group are on the same scale to help prevent feature dominance and checked for collinearity using a correlation matrix. If any pair of features had a correlation threshold value greater than 0.78, we removed one of the two features in that pair. Lastly, the target labels were encoded back into numeric values (0 and 1 to represent Churned and Not Churned).

Model Building

Decision Tree with Random Oversampling (Baseline)

Given the class imbalance identified during our EDA, we created a baseline decision tree model using random oversampling (ROS), which created an equal distribution of two classes in the training dataset, with 4139 instances for both classes. Afterward, we performed cross-validation on the development dataset, which produced promising results with an ROC AUC score of 0.842. Additionally, the classification report indicated a strong recall for class 0 (Not Churn) at 1, but the recall for class 1 (Churn) was moderate at 0.58. Next, on the test set, the model achieved an overall accuracy of 0.88 and a test ROC AUC score of 0.835. Furthermore, precision for class 1 was excellent at 1, with no false positives, but the recall was relatively low at 0.56, leading to an F1 score of 0.72 for class 1. Conversely, class 0 showed strong performance, with both precision and recall at high levels, resulting in an F1 score of 0.93.

All in all, the confusion matrix shows strong precision for class 1 with no false positives but struggles with recall (56%), missing 164 true positives. While the overall accuracy of 88% is high, the low recall for class 1 reduces the F1-score to 72%. In contrast, the model handled class 0 exceptionally well, perfectly classifying all 1035 instances.

Decision Tree with SMOTE (Baseline)

We created another baseline decision tree with SMOTE instead, which significantly improved recall for class 1 on the development set to 0.96, but at the cost of class 0 recall dropping to 0.29, which indicates difficulty in identifying true negatives. Furthermore, the overall training accuracy was 0.63, with a moderate ROC AUC of 0.625; the precision for class 1 was 0.58, while the precision for class 0 remained high at 0.87, showing relatively fewer false positives for class 0. On the test set, class 1 recall was high at 0.93, but precision dropped to 0.33, resulting in many false positives. Class 0 recall was very low at 0.3, though precision stayed high at 0.93, leading to an overall test accuracy of 0.47, highlighting poor performance for class 0 despite better identification of churners.

While SMOTE improved class 1 recall, it introduced issues with class 0, particularly in recall. Addressing these challenges may involve using hybrid approaches like Random Forest, adjusting the decision threshold, or refining the feature set to better balance precision and recall.

Random Forest
Using a Random Forest model, which effectively handles complex, non-linear relationships, we followed the Decision Tree model and applied SMOTE and ROS to the Random Forest model. The initial model performed exceptionally well on both development and test sets, with ROC-AUC scores and classification reports showing a perfect score of 1. However, this indicated potential overfitting. To address this, we examined feature importance and removed features contributing to overfitting.

We first identified the top 10 important features and analyzed the correlation with the target variable. Then, we removed extremely high correlations features we discovered like Churn Score and Customer Status_ord to help reduce overfitting, eliminate redundancy, and improve model's generalization ability. We then retrained the model with the reduced feature set where, in the SMOTE-enhanced model, the ROC-AUC score reached 0.994, with a weighted average precision and recall of 96% for churners. For the model with Random Oversampling, the ROC-AUC score achieved 0.995, with a precision of 98% and a recall of 98% for churners in the weighted average.

XGBoost
Using an XGBoost model because it can handle structured data and capture complex patterns, we constructed an initial model using the stratified training and test dataset from the "Pre-processing" section. Although the initial model performed well, the metrics were "too" high indicating overfitting. To address this, we analyzed correlations between features and the target variable. Features like Churn Score and Customer Status_ord were noticed because they had high correlations with the target, suggesting they might be encoding direct information about churn. After removing these features, the model's performance became more realistic and reliable.

We performed hyperparameter tuning to find the best combination of parameters for the model. The final model achieved an AUC-ROC of 91.38%, with a precision of 87% and a recall of 93% for churners. The model effectively identifies customers likely to churn while maintaining a good balance between false positives and false negatives.

Conclusion
Incorporating class weights in the XGBoost model effectively improves recall for class 1 by penalizing misclassification of the minority class, reducing the need for oversampling and lowering the risk of overfitting. For Decision Tree models, Random Oversampling performed well for both churn and non-churn predictions, while SMOTE was more effective for the non-churn group. In Random Forest models, ROS slightly outperformed SMOTE with a marginally higher ROC-AUC score, but both methods predicted churned and non-churned groups effectively. Adjusting the decision threshold in both ROS and SMOTE models can further balance precision and recall, particularly for applications where missing churn cases is more costly.