Project: Customer Churn Prediction

Approach:

Data Collection: We started by collecting historical customer data, including features such as age, subscription length, monthly bill, total usage, gender, and location. The target variable, 'Churn,' indicates whether a customer churned (1) or not (0).

Data Preprocessing:

- We checked for missing values, and fortunately, the dataset was clean, with no missing data.
- We converted categorical variables like 'Gender' and 'Location' into binary numerical features.
- We handled categorical features like 'Location' differently, creating binary columns for each unique location, indicating whether a customer is from that location or not.

Feature Engineering:

- We converted the 'Gender' feature into a binary variable ('Gender_Male') where 1 indicates male and 0 indicates female.
- We used one-hot encoding to create binary columns for each unique location ('Location_Los Angeles,' 'Location_New York,' 'Location_Miami,' 'Location_Chicago,' 'Location_Houston').

Model Selection:

We experimented with various machine learning models, including:

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- AdaBoost Classifier
- XGBoost Classifier

Our primary goal was to identify the model that would provide the highest predictive accuracy while avoiding overfitting.

Dimensionality Reduction with PCA

In an attempt to enhance model performance and mitigate multicollinearity among features, we applied Principal Component Analysis (PCA). The objective was to transform the feature space into a lower-dimensional subspace while preserving the most relevant information. However, our exploration with PCA did not yield a significant improvement in model performance.

We evaluated each model's performance using accuracy, precision, recall, and F1-score on both training and test data.

Model Performance Metrics and Visualizations:

Model Evaluation:

- Logistic Regression and AdaBoost Classifier had lower accuracy and F1-scores on both training and test data, suggesting they might not be the best choices for this problem.
- Decision Tree Classifier and Random Forest Classifier achieved 100% accuracy on the training data, indicating potential overfitting.
- XGBoost Classifier performed well in terms of accuracy and F1-score on both training and test data.

Final Model Selection: XGBoost

After thorough evaluation, we selected the XGBoost classifier as our final model. The XGBoost model consistently demonstrated strong performance, both with and without hyperparameter tuning. It struck a balance between accuracy, precision, recall, and F1-score, making it the ideal choice for identifying potential customer churners.

Hyperparameter Tuning

To further optimize the XGBoost model's performance, we employed hyperparameter tuning techniques. The best hyperparameters were determined to be learning_rate=0.05, max_depth=3, and n_estimators=600, enhancing the model's predictive capabilities.

But the results didn’t had significant improvement so we choose model without hyperparameter tunning to avoid time consumption.

Threshold Optimization

We also optimized the probability threshold for the XGBoost Classifier to strike a balance between precision and recall. The best threshold was found to be 0.44.

Receiver Operating Characteristic (ROC) Curve

As a final step, we visualized the ROC curve for the XGBoost Classifier, which yielded an AUC-ROC score of approximately 0.90, highlighting the model's discriminative power.

Conclusion:

Based on the results, we selected the XGBoost Classifier as the final model for customer churn prediction. The optimized model achieved a balance between accuracy, precision, recall, and F1-score, making it suitable for identifying potential churners. By using this model in production, we can proactively address customer churn and retain valuable customers.

Please note that further fine-tuning and monitoring of the model in a real-world deployment may be necessary to maintain its predictive accuracy over time.

For any further analysis or deployment, please refer to the provided code and documentation.