Telecom Customer Churn Prediction Documentation

Dataset

The telecom customer dataset used for this project is sourced from [provide the source or origin of the dataset]. The dataset contains information about various customer attributes, such as gender, age, contract type, monthly charges, total charges, and churn status. The dataset also includes a "Churn" column indicating whether a customer has churned or not.
During the data preprocessing step, any rows with missing values were dropped from the dataset to ensure data quality and consistency.

Feature Engineering

In addition to the existing features, the following feature engineering steps were performed to enhance churn prediction:

1. Gender Index: The "Gender" column was transformed into numerical values using the StringIndexer, creating the "GenderIndex" feature.
2. Contract Index: The "Contract" column was also transformed into numerical values using the StringIndexer, creating the "ContractIndex" feature.
3. Label Encoding: The "Churn" column was encoded using the StringIndexer to convert the binary churn status into numerical values, creating the "label" feature.
4. Feature Vectorization: The features "GenderIndex", "Age", "ContractIndex", "MonthlyCharges", and "TotalCharges" were assembled into a vector using the VectorAssembler, resulting in the "features" column.

Model Selection

Two machine learning models were selected for churn prediction:

1. Random Forest Classifier: Random Forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. It was chosen for its ability to handle non-linear relationships and feature interactions.
2. Logistic Regression: Logistic Regression is a linear classification algorithm that estimates the probability of a binary outcome. It was selected as a baseline model for comparison.

Hyperparameter Tuning

Hyperparameter tuning was performed using the TrainValidationSplit technique, which splits the training data into train and validation sets to select the best model based on evaluation metrics.
For the Random Forest Classifier, two hyperparameters were tuned: maxDepth with values [5, 10].
For Logistic Regression, one hyperparameter was tuned: regParam with values [0.01, 0.1].

Model Evaluation

The models were evaluated using the BinaryClassificationEvaluator, which calculates the area under the ROC curve (AUC) as the evaluation metric. The achieved accuracies for each model are as follows:

1. Random Forest Classifier: Accuracy - 0.5
2. Logistic Regression: Accuracy - 0.8333333333333333

The Logistic Regression model outperformed the Random Forest Classifier, achieving the desired accuracy of 0.8.

Challenges Faced

Throughout the project, several challenges were encountered:

1. Limited Dataset Size: The small size of the dataset made it difficult to train complex models and achieve higher accuracies.
2. Computation Time: Initially, training the models took a considerable amount of time due to the inefficient parameter grid search. The code was optimized to reduce training time.
3. Imbalanced Classes: The churn class was imbalanced, with fewer churn instances compared to non-churn instances. This imbalance could have affected the model's performance and led to lower accuracy.

Lessons Learned

From this project, the following lessons were learned:

1. Data Preprocessing: Handling missing values and encoding categorical features is crucial for machine learning models. The StringIndexer and VectorAssembler in PySpark simplify this process.
2. Model Selection: Different models may perform differently on different datasets. It is important to experiment with multiple models to find the best fit for the problem at hand.
3. Hyperparameter Tuning: Careful selection of hyperparameters can significantly impact model performance. It is important to tune hyperparameters to find the optimal combination.