Project Brief: Telecom Customer Churn Prediction using PySpark

Background Information

Customer churn is a significant challenge in the telecom industry. Identifying customers who are likely to churn is crucial for implementing proactive measures to retain them. By leveraging PySpark, we can take advantage of its distributed computing capabilities to handle large volumes of data efficiently and build an accurate machine learning model for churn prediction.

Problem Statement

The goal of this project is to develop a machine learning model using PySpark that accurately predicts customer churn in a telecom company. The model should achieve a minimum accuracy of 0.8, enabling the company to proactively identify and retain customers at risk of leaving. By effectively predicting churn, the company can implement targeted retention strategies, reduce customer attrition, and improve overall business performance.

Guidelines

- **Dataset:** Obtain a telecom customer dataset that includes relevant features such as customer demographics, usage patterns, service plans, call details, customer complaints, and churn status. You can use this dataset.
- **Data Preprocessing:** Perform necessary preprocessing steps on the dataset, including handling missing values, feature scaling, encoding categorical variables, and splitting the data into training and testing sets. Consider using PySpark's DataFrame API for efficient data manipulation.
- **Feature Engineering:** Create new features from the existing dataset that might be helpful for predicting churn. For example, you could calculate metrics such as call duration, average monthly spend, customer tenure, or customer satisfaction scores.
- **Model Selection and Training:** Choose an appropriate machine learning algorithm for churn prediction, considering the nature of the problem and the dataset characteristics. PySpark provides various algorithms such as logistic regression, random forests, gradient boosting, and support vector machines. Experiment with different models and hyperparameter configurations to achieve the desired accuracy of 0.8.
- **Model Evaluation:** Assess the performance of the trained models using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score.
- **Documentation and Reporting:** Maintain clear documentation throughout the project, including details about the dataset, preprocessing steps, feature engineering, model selection, and evaluation results. Summarize the project findings, challenges faced, and lessons learned in a final report.