Project Title: Customer Churn Prediction

by Sathish Vanga.

Project Methodology: CRISP-ML(Q)

Phase 1: Business Understanding

1.1 Business Understanding

Problem Statement:

Develop a predictive model to estimate the likelihood of customer churn for a telecommunications company. The model should consider various factors such as customer demographics, usage patterns, billing history, and customer service interactions.

Objective:

The primary objective of this project is to build a predictive model that accurately identifies customers who are likely to churn. By doing so, the telecommunication company can implement targeted retention strategies to reduce churn rates, enhance customer satisfaction, and improve overall profitability.

1.2 Data Understanding

- I described the dataset columns and their meanings.
- Checked for missing values and duplicated records, but there no missing and duplicate values in my data
- Checked the data types of columns and handled issues such as incorrect data types (e.g., 'TotalCharges').
- Conducted exploratory data analysis (EDA) to gain insights into the dataset's characteristics, including statistical summaries and visualizations.

Phase 2: Data Preparation/Data Engineering

2.1 Exploratory Data Analysis

- Conducted statistical analysis on key numerical features like tenure, MonthlyCharges, and TotalCharges.
- Visualized distributions and relationships between variables using various plots.

- Performed hypothesis testing to understand the association between variables and churn.

2.2 Data Cleaning

- Handled missing values in the 'TotalCharges' column by replacing them with the median.
- Grouped related service columns into a single feature to reduce dimensionality.
- Transformed categorical variables into binary features for better modeling.

Phase 3: Model Building

- The dataset is split into training and testing sets, with 75% of the data used for training and 25% for testing.
- StandardScaler is used to normalize numerical features.
- OneHotEncoder is used for one-hot encoding of binary categorical features, and OrdinalEncoder is used for ordinal encoding of multi-class categorical features.
- A preprocessing pipeline is created to apply the respective transformations to numerical and categorical features, ensuring consistency and efficiency in data preparation.

Each model has a set of hyperparameters that are fine-tuned using GridSearchCV. The hyperparameters include:

- For Logistic Regression: Regularization strength (C) and number of features to select.
- For SGD Classifier: Regularization parameter (alpha) and number of features to select.
- For Decision Tree Classifier: Maximum depth of the tree and number of features to select.
- For Random Forest Classifier: Number of trees (estimators) and maximum depth of the trees.

Phase 4: Model Evaluation

Model Storage and Loading

The best models for each algorithm are saved to disk using joblib for later use. The models are then reloaded to ensure they can be successfully retrieved and evaluated.

Evaluation Metrics

Each model's performance is evaluated using:

- **Accuracy**: The proportion of correctly classified instances.
- **F1 Score**: The harmonic mean of precision and recall, providing a balance between the two metrics.
- **Recall**: The proportion of actual positives correctly identified.
- **Precision**: The proportion of positive identifications that are actually correct.
- **Confusion Matrix**: A table showing the true positive, true negative, false positive, and false negative predictions.
- **Classification Report**: Detailed metrics for each class, including precision, recall, and F1 score.

Results

Logistic Regression

- **Accuracy**: 80.8%
- **F1 Score**: 80.2%
- **Recall**: 80.8%
- **Precision**: 80.4%
- **Confusion Matrix**:
  - True Negatives: 1100
  - False Positives: 227
  - False Negatives: 201
  - True Positives: 372
- **Classification Report**:
  - Precision, recall, and F1 scores for each class (0 and 1).

SGD Classification

- **Accuracy**: 80.9%
- **F1 Score**: 80.1%
- **Recall**: 80.9%
- **Precision**: 80.5%
- **Confusion Matrix**:
  - True Negatives: 1111
  - False Positives: 216
  - False Negatives: 186
  - True Positives: 387
- **Classification Report**:
  - Precision, recall, and F1 scores for each class (0 and 1).

Decision Tree Classification

- **Accuracy**: 75.5%

- **F1 Score**: 74.2%
- **Recall**: 75.5%
- **Precision**: 74.8%
- **Confusion Matrix**:
    - True Negatives: 999
    - False Positives: 328
    - False Negatives: 168
    - True Positives: 405
- **Classification Report**:
    - Precision, recall, and F1 scores for each class (0 and 1).

Random Forest Classification

- **Accuracy**: 80.5%
- **F1 Score**: 79.7%
- **Recall**: 80.5%
- **Precision**: 79.9%
- **Confusion Matrix**:
    - True Negatives: 1101
    - False Positives: 226
    - False Negatives: 207
    - True Positives: 366
- **Classification Report**:
    - Precision, recall, and F1 scores for each class (0 and 1).

Conclusion

The evaluation results indicate that the **SGD Classifier** achieved the highest accuracy (80.9%) and balanced performance across other metrics. The **Logistic Regression** model also performed well with an accuracy of 80.8%. The **Decision Tree Classifier** had the lowest performance with an accuracy of 75.5%, while the **Random Forest Classifier** demonstrated good performance but had a larger model size.

Recommendation

Based on the evaluation metrics, the **SGD Classifier** is recommended for deployment due to its high accuracy, efficient model size, and balanced performance. The **Logistic Regression** model is a close second and can be considered as an alternative, especially if model interpretability is important.

Phase 5: Deployment

The Streamlit application is designed to predict customer churn for a telecommunication company based on various input features. The application utilizes a trained machine-learning model to make predictions. Below are the steps to interact with the application:

Steps:

1. **Input Features:** Provide necessary information about the customer demographics, usage patterns, billing history, and service interactions.
2. **Click on Predict:** After entering the required information, click on the "Predict" button.
3. **View Prediction:** The application will display whether the customer is likely to churn or not based on the provided input.