## Documentation: Telecom Customer Churn Prediction Models

### 1. Introduction

This documentation outlines the steps involved in processing and analyzing the Telecom Customer Churn dataset. The goal is to predict customer churn using various machine learning models. The analysis includes data cleaning, feature engineering, model training, evaluation, and visualization of results.

### 2. Dependencies

The following Python packages are required for executing the script:

* **pandas**: Data manipulation and analysis.
* **numpy**: Numerical operations.
* **matplotlib**: Plotting and visualization.
* **seaborn**: Data visualization.
* **scikit-learn**: Machine learning model implementation, preprocessing, and evaluation.
* **warnings**: For suppressing unnecessary warnings.

Install the dependencies using pip:

pip install pandas numpy matplotlib seaborn scikit-learn

### 3. Dataset

* **Input Data**: The script uses a CSV file named Telecom\_customer\_churn.csv.
* **Target Variable**: The Churn column is the target variable, indicating whether a customer has churned (Yes or No).
* **Features**: Various customer-related features, including tenure, MonthlyCharges, TotalCharges, and several categorical variables such as InternetService, Contract, and PaymentMethod.

### 4. Data Cleaning and Preprocessing

The data cleaning and preprocessing steps include:

* **Dropping Unnecessary Columns**: The customerID column is dropped as it does not contribute to the prediction.
* **Handling Missing Values**: The TotalCharges column is converted to numeric, forcing errors to NaN, and rows with missing values are dropped.
* **Replacing Categorical Values**: The script replaces ‘No internet service’ and ‘No phone service’ with ‘No’ for simplicity.
* **Encoding Binary Categorical Variables**: Columns with binary categorical values (e.g., Partner, Dependents, etc.) are encoded to numerical values (Yes to 1 and No to 0).
* **Encoding Gender**: The gender column is encoded (Female to 1 and Male to 0).
* **One-Hot Encoding**: For categorical variables with multiple categories, one-hot encoding is applied (e.g., InternetService, Contract, PaymentMethod).
* **Feature Scaling**: Continuous variables (tenure, MonthlyCharges, TotalCharges) are scaled using MinMaxScaler to normalize the data.

### 5. Model Definition and Training

The script defines and trains four machine learning models:

* **Logistic Regression**: A linear model for binary classification.
* **Support Vector Machine (SVM)**: A classifier that constructs hyperplanes to separate different classes.
* **Naive Bayes**: A probabilistic classifier based on Bayes’ theorem.
* **Random Forest**: An ensemble model that combines multiple decision trees.

Each model is trained on the preprocessed data, and the performance is evaluated using a test set.

### 6. Model Evaluation

The performance of each model is evaluated using the following metrics:

* **Accuracy**: The proportion of correctly predicted instances.
* **Precision**: The proportion of positive predictions that are actually correct.
* **Recall**: The proportion of actual positives that are correctly identified.
* **F1 Score**: The harmonic mean of precision and recall, providing a balanced measure.

A classification report is generated for each model, summarizing these metrics.

### 7. Results and Visualization

The results of the model evaluations are compiled into a DataFrame and visualized:

* **Model Comparison**: A bar plot is generated to compare the performance of the models across the accuracy, precision, recall, and F1 score metrics. The plot is saved as model\_comparison.png.
* **Confusion Matrix**: A confusion matrix is generated for the Logistic Regression model and visualized using a heatmap. The plot is saved as Churn Confusion Matrix.png.
* **Feature Importance**: For the Logistic Regression model, the importance of each feature is calculated and visualized in a horizontal bar chart, showing which features contribute most to the prediction. The plot is saved as Featues Importance.png.

### 8. Execution

To run the script, ensure that the Telecom\_customer\_churn.csv file is available in the working directory. The script will process the data, train the models, evaluate their performance, and generate visualizations. The results are printed to the console, and the plots are saved as PNG files.

### 9. File Output

* **model\_comparison.png**: A bar plot comparing the performance of Logistic Regression, SVM, Naive Bayes, and Random Forest models.
* **Churn Confusion Matrix.png**: A heatmap of the confusion matrix for the Logistic Regression model.
* **Featues Importance.png**: A bar chart showing the feature importance for the Logistic Regression model, indicating which features most influence the churn prediction.