Assignment: Customer Churn Prediction

Objective:

Build a predictive model to identify customers at high risk of churn for a telecom company based on synthetically generated customer data.

Computational Constraints:

- **Time Limit:** The entire project must be completed within a 4 days

Data Constraints:

- **Data Quality:** Introduce specific data quality issues like missing values, outliers, or inconsistencies.
- **Data Volume:** Reduce the dataset size to simulate real-world constraints.
- **Feature Limitations:** Restrict the number of features that can be used in the model.

Modeling Constraints:

- **Algorithm Restrictions:** Limit the choice of algorithms to specific options (e.g., only decision trees and logistic regression).
- **Hyperparameter Tuning:** Restrict the hyperparameter search space.
- **Model Complexity:** Impose limitations on model complexity (e.g., maximum depth of a decision tree).

Evaluation Constraints:

- **Metric Focus:** Prioritize a specific evaluation metric (e.g., precision over recall).
- **Imbalanced Dataset:** Create a highly imbalanced dataset to challenge the model.

Task Breakdown:

1. Data Generation:

- Generate a synthetic dataset of 5000 customer records containing the following features:
  - CustomerID
  - Age
  - Gender
  - ContractType (Month-to-month, One year, Two year)
  - MonthlyCharges

    - TotalCharges
    - TechSupport
    - InternetService (DSL, Fiber optic, No)
    - Tenure
    - PaperlessBilling
    - PaymentMethod
    - Churn (Yes/No)
- Introduce realistic distributions, correlations, and outliers to the data.
- Ensure a target churn rate of approximately 20%.
- Create derived features like average_monthly_charges, customer_lifetime_value.

2. Exploratory Data Analysis (EDA):

- Perform in-depth EDA to understand the dataset characteristics.
- Calculate summary statistics for numerical columns.
- Analyze categorical data distributions.
- Visualize relationships between features and the target variable (churn).
- Identify potential correlations and patterns.

3. Data Preprocessing:

- Handle missing values (if any) using appropriate techniques.
- Encode categorical features into numerical format.
- Split the dataset into training, validation, and testing sets.
- Consider techniques to handle imbalanced data (if necessary).

4. Feature Engineering:

- Create additional features based on domain knowledge and EDA insights.
- Explore feature interactions and transformations.

5. Model Building:

- Experiment with various classification algorithms (Logistic Regression, Random Forest, Gradient Boosting, XGBoost, etc.).
- Optimize model hyperparameters using techniques like Grid Search or Randomized Search.
- Evaluate model performance using metrics like accuracy, precision, recall, F1-score, ROC curve, AUC.
- Consider ensemble methods for improved performance.

6. Model Selection and Evaluation:

- Select the best-performing model based on evaluation metrics and explainability.
- Create a confusion matrix to analyze model predictions.
- Calculate feature importance to understand key drivers of churn.

7. Model Deployment (Optional):

- Develop a plan for deploying the model into a production environment.
- Consider real-time or batch prediction scenarios.
- Create a user-friendly interface for model consumption.

Deliverables:

- A Jupyter Notebook or Python script containing the code for all steps.
- A comprehensive report summarizing the findings, including EDA results, model performance, and insights.
- Visualizations to support the analysis.
- A well-structured and commented codebase.

Evaluation Criteria:

- Data generation quality and realism.
- Depth of EDA and data preprocessing.
- Model performance and selection.
- Code quality, readability, and efficiency.
- Report clarity and comprehensiveness.

Additional Considerations:

- Explore techniques for model interpretability (e.g., SHAP values).
- Consider the ethical implications of using customer data.
- Document the entire process for reproducibility.

By completing this assignment, you will demonstrate your ability to handle the entire machine learning lifecycle, from data generation to model deployment, and effectively solve a real-world problem.