**Assignment: Machine Learning Model Comparison and Feature Selection**

Objective

This assignment evaluates your ability to implement, compare, and analyze machine learning models (Logistic Regression, Random Forest, and XGBoost) while investigating the impact of feature selection on model performance. The goal is to develop a structured workflow for model evaluation and feature engineering, grounded in real-world scenarios.

Dataset Requirements

1. **Domain:** Use a binary classification dataset (e.g., healthcare diagnostics, loan default prediction, equipment failure detection).
2. **Size:** 200–300 samples with 15–20 features.
3. **Public Sources:**
   * Kaggle (e.g., Titanic Survival, Pima Indians Diabetes)
   * UCI Machine Learning Repository
   * Synthetic data generation (using sklearn.datasets.make\_classification)

Tasks

Part 1: Model Comparison

1. **Data Preprocessing**
   * Handle missing values and outliers.
   * Encode categorical variables (e.g., one-hot encoding).
   * Split data into training (75%) and testing (25%) sets.
2. **Model Implementation**
   * Train three models:
     + Logistic Regression (sklearn.linear\_model.LogisticRegression)
     + Random Forest (sklearn.ensemble.RandomForestClassifier)
     + XGBoost (xgboost.XGBClassifier)
   * Optimize hyperparameters via grid/random search (e.g., max\_depth for trees, regularization for logistic regression).
3. **Evaluation Metrics**
   * Calculate for each model:
     + Accuracy
     + Precision
     + Recall
     + F1-score
     + AUC-ROC curve
4. **Analysis**
   * Compare results in a table.
   * Justify which model performs best for your dataset (e.g., "XGBoost achieved 94.1% accuracy due to gradient boosting’s iterative error correction").

Part 2: Feature Selection Impact

1. **Feature Engineering**
   * Reduce features to 10 using one of:
     + Filter methods (e.g., correlation analysis, chi-square)
     + Wrapper methods (e.g., recursive feature elimination)
     + Embedded methods (e.g., Lasso regularization, tree-based importance)
2. **Re-evaluate Models**
   * Retrain models on the reduced feature set.
   * Compare metrics with Part 1 results.
3. **Critical Analysis**
   * Discuss how feature reduction impacted performance (e.g., "Precision dropped by 12% in Random Forest due to loss of critical interaction terms").
   * Recommend optimal feature count for each model.

Deliverables

1. **Code Submission**
   * Jupyter Notebook/Python script with:
     + Data preprocessing steps
     + Model training/evaluation code
     + Visualization (e.g., ROC curves, feature importance plots)
2. **Report**
   * Summary of methodology (1 page).
   * Comparative analysis tables/graphs.
   * Justification of model recommendations (e.g., "XGBoost is ideal for high-recall scenarios like disease detection").
3. **Presentation**
   * 5-minute summary highlighting key findings and technical challenges.

References

1. Comparative analysis of LR, RF, and XGBoost
2. Feature selection techniques
3. Model evaluation metrics

**Note:**  
Focus on reproducibility-include requirements.txt for dependencies. Use comments to explain non-trivial code decisions (e.g., "Chose ANOVA F-value for feature selection due to numeric-class relationships").