Project Two: Supervised Algorithms for Classification and Prediction

STA 551 Foundations of Data Science

Introduction

Supervised classification and prediction models are a core part of machine learning (ML) and statistical modeling, where the goal is to learn a mapping from input features to an output label based on labeled training data. These models are widely used in applications like spam detection, medical diagnosis, credit scoring, and more.

The goal of this project is to implement several commonly used, simple supervised learning algorithms to solve binary classification/prediction problems. The supervised learning algorithms selected for classification in this project include:

- **Perceptron** (single-layer neural network)
- **Decision tree method** (including penalized trees)
- **Ensemble trees—Bagging** (with encouragement to explore other ensemble methods)

Additionally, a logistic regression model will serve as a baseline for performance comparison.

Data Requirements

The following is a clear and structured breakdown of the data requirements needed to perform EDA, feature engineering, and implement the specified models (logistic regression, perceptron, decision tree, and bagging):

- **General Data Requirements**
  - **Sufficient Sample Size**
    - At least 100 - 1,000+ samples (for reliable training/testing splits).
    - More data improves model robustness, especially for ensemble methods like bagging.
  - **Binary Target Variable**
    - Clearly labeled 0/1 or True/False for classification (e.g., "Default/No Default").
  - **Feature Variety**
    - Mix of numeric (continuous/discrete) and categorical features (if applicable).
    - Avoid datasets with only one feature type (limits feature engineering).
  - **No Severe Class Imbalance**
    - Ideally, the minority class should be $\geq 20\%$ of the data.
    - If imbalanced, ensure techniques like resampling or class weights can be applied.
- **Data Quality Requirements**
  - **Handling Missing Values**
    - Missing data should be $< 10\%$ per feature (or a strategy for imputation, e.g., mean/median).
  - **No Severe Multicollinearity**
    - High correlation between features ($|r| > 0.8$) can distort logistic regression/decision trees.
  - **No Leakage**
    - Ensure no feature directly reveals the target (e.g., "Loan_Approved_Flag" in features).
- **Feature Engineering Needs**

  - **Numeric Features**
    * Should be scaled (e.g., standardization for perceptron/logistic regression).
    * Potential for polynomial/interaction terms (if nonlinear relationships exist).
  - **Categorical Features**
    * Low cardinality (e.g., < 10 categories) to avoid excessive dummy variables.
    * Encoding options: One-Hot (for a few categories) or Ordinal (if ordered).
  - **Feature Importance Flexibility**
    * Some features should be redundant/noisy to test model robustness (e.g., decision trees vs. bagging).
- **Model-Specific Requirements**
  - **Logistic Regression** - Linearly separable features (or engineered terms).
  - **Perceptron** - Scaled features (sensitive to magnitude).
  - **Decision Tree** - Handles mixed data types; benefits from outliers.
  - **Bagging (e.g., A Special Random Forest)** - Larger datasets to reduce overfitting.
- **Suggested data sites**:
  - My teaching data repository (https://pengdsci.github.io/datasets/),
  - UCI Machine Learning Repository (https://archive.ics.uci.edu/), and
  - Kaggle (https://www.kaggle.com/datasets?fileType=csv).

Methodology

It is crucial to follow the logical process for developing predictive models using logistic regression, decision trees, and neural networks, including exploratory data analysis (EDA), feature engineering, and model comparison via ROC analysis.

- **Data Description** (source, background, list of variables with descriptions/definitions)
- **Formulate practical questions** and convert them into analytical questions
- Specify relevant candidate models to address the analytical questions and related assumptions
- **EDA**
  - identifying new patterns to enhance modeling
  - detecting abnormal patterns or potential violations of model assumptions for feature engineering
    * Missing value handling
    * feature encoding and discretization
    * Feature transformations
    * Feature creation
    * Feature selection
- **Model development**: using a training-validation-testing mechanism and the global and local performance metrics (ROC-AUC, accuracy, sensitivity, specificity, and F1 scores) for model selection and hyperparameter tuning.
  - Logistic regression (reference model)
    * Select initial models based on relevancy
    * Use ROC-AUC to select the best one to compete with other classification algorithms
  - Perceptron (single-layer neural networks)
    * Use different activation functions to create candidate perceptrons
    * Best perceptron selection using ROC-AUC
  - Decision tree
    * create candidate models based on complexity parameters and/or penalty of misclassification, etc.
    * Select the best decision tree with ROC-AUC.
  - Ensemble trees-BAGGING

    - Create candidate BAGGING models using hyperparameters such as cost-sensitive penalty and size of bagging, etc..
    - Use ROC-AUC to identify the best candidate model
  - Perform a comparison among the optimal models from each of the above families using ROC-AUC.
  - Final models
    - **Model Selection Criteria**: performance, interpretability needs, and computational requirements.
    - **Threshold Optimization**: maximizing business metrics and cost considerations.

Reporting and Submission

- **Report Format**
  - The report should follow the same structure and components as those in project one.