Comparative Study of Parametric and Non-Parametric Classifiers for Binary Classification on Numerical Datasets

- Comparative analysis of 4 classifiers on 5 numerical datasets
- Classifiers: Logistic Regression, Linear SVM (parametric), Decision Tree, Weighted KNN (non-parametric)

Overview

- Implements 4 classifiers from scratch
- 2 parametric: Logistic Regression, Linear SVM
- 2 non-parametric: Decision Tree, Weighted KNN
- Evaluated on 5 binary classification datasets
- Metrics: Accuracy, Precision, Recall, F1-Score

Setup

- Requirements: Python 3.8+, pip
- Install dependencies:
  cd C:\Users\saket\SEM5-PROJECTS\ML
  pip install numpy pandas matplotlib ydata-profiling pydantic-settings

Structure

- Classifiers/
  - 30_DecisionTree.py # CART with Gini impurity
  - 30_LinearSVM.py # Support Vector Machine
  - 30_LogisticRegression.py # Gradient descent
  - 30_WeightedKNN.py # Distance-weighted neighbors
  - output.txt # Results
- Datasets/ # 5 cleaned datasets
- Cleanup.py # Data preprocessing
- DataProfiling.py # Dataset profiles
- plot.py # Decision boundary visualization
- Outputs/ # Generated visualizations

How to Run :

Individual Classifiers

- cd Classifiers
- python 30_DecisionTree.py # Non-parametric: CART, Gini
- python 30_LinearSVM.py # Parametric: margin maximization
- python 30_LogisticRegression.py # Parametric: gradient descent
- python 30_WeightedKNN.py # Non-parametric: distance-weighted

Data Processing

- python Cleanup.py # Clean datasets
- python DataProfiling.py # Generate profiles

Visualization

- python plot.py # Select dataset, view decision boundaries

Full Pipeline

- python Cleanup.py && python DataProfiling.py
- cd Classifiers && python 30_DecisionTree.py && python 30_LinearSVM.py && python 30_LogisticRegression.py && python 30_WeightedKNN.py
- cd .. && python plot.py

Classifiers

- Decision Tree: Non-parametric, CART + Gini, interpretable rules
- Linear SVM: Parametric, margin maximization, global optimization
- Logistic Regression: Parametric, gradient descent, fast & probabilistic
- Weighted KNN: Non-parametric, distance-weighted, adaptive to data

Datasets (Binary Classification)

- Bank: 4,521 samples, 16 features, Finance
- Heart: 303 samples, 13 features, Healthcare
- Wine: 178 samples, 13 features, Food science
- Breast: 569 samples, 30 features, Healthcare
- Churn: 10,000 samples, 20 features, Telecom

Output

- Results saved in: Classifiers/output.txt
- Contains Accuracy, Precision, Recall, F1-Score for each classifier

Troubleshooting

- Install missing modules:
  pip install numpy pandas matplotlib ydata-profiling pydantic-settings
- Verify datasets:
  ls Datasets/
- Check Python version:
  python --version

Metrics

- Accuracy: (TP + TN) / Total
- Precision: TP / (TP + FP)
- Recall: TP / (TP + FN)
- F1-Score: 2 × (Precision × Recall) / (Precision + Recall)