Assignment-5: Other Classical Supversied Learning

Munem Shahriar 2021251042
Tausif Mushtaque 2031020642
Email addresses: munem.shahriar07@nothsouth.edu
tausif.mushtaque@northsouth.edu

December 8, 2024

I. Introduction

Other classical supervised learning algorithms, including ZeroR, OneR, K-Nearest-Neighbor, Naive Bayesian, Support Vector Machine, and Support Vector Regression, provide a range of methods for tackling classification and regression tasks, each employing distinct assumptions and techniques to analyze data effectively.

ZeroR Classifier

ZeroR is a simple classification algorithm that makes predictions based solely on the majority class in the dataset. It doesn't consider any features but only predicts the most frequent class, making it a baseline model for comparison with more complex algorithms.

OneR Classifier

OneR is a basic classification algorithm that builds a single rule using one feature. It evaluates all features and selects the one that produces the simplest rule with the least error, providing a simple but effective model for classification tasks.

K-Nearest-Neighbor Classifiers

K-Nearest-Neighbor (KNN) is a non-parametric algorithm that classifies new instances based on the majority class of their nearest neighbors in the feature space. It is simple, intuitive, and effective for many classification problems, but computationally expensive for large datasets.

Naive Bayesian Classifier

The Naive Bayesian Classifier is a probabilistic classifier based on Bayes' Theorem, assuming feature independence. It computes the probability of each class given the features and predicts the class with the highest likelihood. It's fast and effective, especially for text classification tasks.

Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm that finds an optimal hyperplane to separate data into different classes. It maximizes the margin between classes, making it highly effective for high-dimensional data and classification problems with clear boundaries.

Support Vector Regression (SVR)

Support Vector Regression (SVR) extends SVM to regression tasks, where the goal is to predict continuous values. It works by finding a hyperplane that best fits the data while allowing for some margin of error, making it suitable for predicting real-valued outputs with minimal overfitting.

II. Method

Here we will see some parts from the a dataset that was used to demonstrate certain techniques. Also note that these results are recreatable for each dataset by using the same code:

A. ZeroR on DT-credit dataset

```
========================================
most frequent class (zeroR prediction): No
Accuracy (zeroR prediction): 0.90
========================================
```

B. OneR on DT-credit dataset

```
========================================
Feature: Income, Error Rate: 0.002500
Feature: Limit, Error Rate: 0.007500
Feature: Rating, Error Rate: 0.050000
Feature: Cards, Error Rate: 0.100000
Feature: Age, Error Rate: 0.097500
Feature: Education, Error Rate: 0.100000
Feature: Own, Error Rate: 0.100000
Feature: Married, Error Rate: 0.100000
Feature: Region, Error Rate: 0.100000
Feature: Balance, Error Rate: 0.010000
Best Feature: Income, Rule: Income
========================================
```

C. KNN on DT-credit dataset

```
========================================
Binary Classification (Student column)
KNN Binary Classification Accuracy: 0.8833
========================================
Multi-Class Classification (Region column)
KNN Multi-Class Classification Accuracy: 0.3000
========================================
Regression (Balance column)
KNN Regression Mean Squared Error: 52878.5370
KNN Regression R^2 Score: 0.6514
========================================
```

D. Naive Bayesian Classifier on DT-credit dataset

```
========================================
Binary Classification (Student column)
Naive Bayes Binary Classification Accuracy: 0.8833
========================================
Multi-Class Classification (Region column)
Naive Bayes Multi-Class Classification Accuracy: 0.4667
========================================
```

E. SVM on DT-credit dataset

```
========================================
Binary Classification (Student column)
SVM Binary Classification Accuracy: 0.9833
========================================
Multi-Class Classification (Region column)
SVM Multi-Class Classification Accuracy: 0.5000
========================================
```

F. SVR on DT-credit dataset

```
========================================
#Result for target column income
SVR Mean Squared Error: 1152.7782
SVR R^2 Score: 0.1445
========================================
```

Mathematical Terminology for Classifiers

ZeroR Classifier

The ZeroR classifier predicts the most frequent class in the training data, regardless of the input features. The error rate is given by:

$$\text{Error}(ZeroR) = 1 - P(\hat{y} = y_{\text{majority}})$$

where $P(\hat{y} = y_{\text{majority}})$ is the probability of predicting the majority class correctly.

OneR Classifier

The OneR classifier builds a decision rule based on one feature and creates a partition of the feature space that minimizes classification error. The error rate is given by:

$$\text{Error}(OneR) = \sum_{i=1}^{n} \mathbf{1}(y_i \neq \hat{y}_i)$$

where $\hat{y}_i$ is the predicted class based on the selected feature, and $\mathbf{1}$ is the indicator function.

K-Nearest-Neighbor Classifiers (KNN)

The KNN classifier assigns the class $\hat{y}$ by taking the majority vote from the k nearest neighbors of the test point x:

$$\hat{y} = \text{mode}(y_1, y_2, \ldots, y_k)$$

where $y_1, y_2, \ldots, y_k$ are the classes of the k nearest neighbors of x, and "mode" refers to the most frequent class among them.

Naive Bayesian Classifier

The Naive Bayes classifier applies Bayes' theorem, assuming feature independence given the class. The class $\hat{y}$ is computed as:

$$\hat{y} = \arg\max_{y \in Y} P(y) \prod_{i=1}^{d} P(x_i|y)$$

where $P(y)$ is the prior probability of class y, and $P(x_i|y)$ is the likelihood of feature x_i given class y.

Support Vector Machine (SVM)

The SVM classifier aims to find a hyperplane $w \cdot x + b = 0$ that maximizes the margin between the classes. The optimization problem is formulated as:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{subject to} \quad y_i(w \cdot x_i + b) \geq 1 \quad \forall i$$

where y_i is the class label, x_i is the feature vector, and w and b are the hyperplane parameters.

Support Vector Regression (SVR)

In SVR, the goal is to find a function $f(x) = w \cdot x + b$ that approximates the data while staying within a margin of error ϵ. The objective is:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{subject to} \quad |y_i - (w \cdot x_i + b)| \leq \epsilon \quad \forall i$$

where ϵ defines the margin of tolerance for the error.

Conclusion

In conclusion, while binary trees like those used in basic decision tree models excel in interpretability and simplicity, advanced algorithms such as XGBoost significantly outperform them in terms of accuracy and stability. XGBoost's use of rectified linear activation-like functions and fine-tuned hyperparameters, such as learning rate and iteration count, enables superior performance and convergence. As demonstrated, optimizing these parameters is crucial for achieving high accuracy, underscoring XGBoost's robustness and adaptability compared to traditional binary tree approaches.