Iran University of Science and Technology

Faculty of Computer Engineering

Artificial intelligence and robotics group

Pattern Recognition project

Course teacher:

Dr. Morteza Analoui

Second project

Teacher Assistants:

Shide Maleki

Erfan Mousavi

First semester of 1402 - 1403

Abstract

The goal of this project is to familiarize with classification concepts. In this project, we ask you to implement classification algorithms on real medical datasets so that, in addition to becoming familiar with and mastering the algorithms, you also become acquainted with the challenges present in real datasets. Data preprocessing in the dataset is a fundamental step in machine learning that significantly affects the model's performance. Paying attention to the quality of the data and preprocessing techniques is crucial for the success of machine learning projects.

Dataset

The "UCI Heart Disease Data" dataset is a widely used dataset in machine learning and healthcare. It includes various features related to heart health, such as age, cholesterol levels, and exercise habits, along with a target variable indicating the presence or absence of heart disease. This dataset is valuable for practicing preprocessing algorithms and data classification in the field of cardiovascular health research.

Project explanation

This project consists of three sections, and we will now elaborate on each of them. The third section is optional.

First section

In this section, we want to use the "UCI Heart Disease Data" dataset. You are expected to determine the optimal number of estimators (n_estimators) for the AdaBoost algorithm using the GridSearchCV technique. Then, based on the best n_estimators, compare the accuracy obtained from implementing the AdaBoost algorithm on this dataset using Cross-Validation with k values of 3, 5, and 7. What does n_estimators specify? (Your final analysis should provide insights into the impact of cross-validation folds and the ideal number of estimators on the algorithm's accuracy for this specific dataset.)

Consider the following points for implementation:

- Examine the provided dataset. If there are missing values, fill them in (do not use the dropna command for deletion).
- Some features are categorical; transform them into numerical representations suitable for machine learning algorithms.
- Also, use normalization for the data
- Keep in mind that dividing the dataset into training and testing sets is necessary for an accurate evaluation of the model's performance.
- As you prepare the "UCI Heart Disease Data" for analysis, ensure that the data is well-prepared for model training. Pay attention to the distribution of your data to ensure the model effectively learns from a representative set of samples.
- The "num" column, which is the target, is structured as [0=no heart disease; 1,2,3,4 = stages of heart disease]. For binary classification, consider individuals without heart disease as 0 (no heart disease) and stages 1 to 4 as 1 (having heart disease).

Dataset link: UCI Heart Disease Data | Kaggle

Second section

1. Explain the Stacking method in the field of machine learning.
2. Implement the Stacking method on the dataset in the first section. Use the following three base models:
    a. SVM
    b. Decision Tree
    c. GradientBoosting

(Hint: sklearn.ensemble.StackingClassifier — scikit-learn 1.3.0 documentation)

3. The link below outlines 8 steps for Stacking. Specify these 8 steps in your code.

(Stacking to Improve Model Performance: A Comprehensive Guide on Ensemble Learning in Python | by Brijesh Soni | Medium)

4. Provide an example explaining how Stacking can be used to assess the importance of each base model.
5. Explain how Stacking prevents Overfitting.
6. Explain the differences between the three methods: Bagging, Boosting, and Stacking.

Third section (optional)

In this section, we leverage the knowledge and understanding gained from the first section, which covers AdaBoost, and explore the fundamental principles of various boosting techniques, such as XGBoost. The aim of this introduction is to familiarize ourselves with ensemble methods based on gradient boosting, a powerful technique in machine learning.

gradient boosting Algorithm:

Gradient Boosting Algorithm

1. Initialize model with a constant value:

$$F_0(x) = \underset{\gamma}{argmin} \sum_{i=1}^{n} L(y_i, \gamma)$$

2. for $m = 1$ *to* M:

2-1. Compute residuals $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ $for\ i = 1,\ldots, n$

2-2. Train regression tree with features x against r and create terminal node reasions R_{jm} for $j = 1,\ldots, J_m$

2-3. Compute $\gamma_{jm} = \underset{\gamma}{argmin} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma)$ $for\ j = 1,\ldots, J_m$

2-4. Update the model:

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} 1(x \in R_{jm})$$

Now, we will delve into specific gradient boosting methods, such as XGBoost.

Regarding to this paper 2939672.2939785 (acm.org) (XGBoost: A Scalable Tree Boosting System) answer follow questions:

a) In machine learning, what is "ensemble learning," and how does XGBoost fit into this concept?
b) Explain the concept of "sparsity-aware" learning within the framework of the XGBoost algorithm. How does the algorithm handle sparse data differently from dense data?
c) How does XGBoost handle missing values in input data? Explain the role of a missing value in the optimization process.
d) Compare the main goals of XGBoost and AdaBoost algorithms. What are the similarities and differences in their objectives?
e) In terms of system design, optimization techniques, and experimental results, it describes the main advantages of XGBoost over AdaBoost. Are there specific scenarios where AdaBoost still has an advantage?
f) Implement XGBoost using the dataset from the first section and report the results. Use evaluation metrics such as accuracy, precision, recall, and F1-score. Compare the results with the first section.