2024 Spring Pattern Recognition Homework 3 Announcement

Release Date: 2024/05/01 12:00

Homework 3

- Deadline: 23:59, May. 15th (Wed), 2024
- **Coding** (60%): Implement ensemble methods with Numpy and PyTorch**.**
  - Submit your code in executable python files (.py).
  - Report the outcome and parameters by screenshots to the questions.
- **Handwritten Questions** (40%): Answer questions about ensemble methods.
  - Answer the questions in the report.
  - You **must use the template** and in **digital-typed** (**no handwritten scan**)
  - In English

Links

- Questions and Report template
- Sample code / Dataset

Coding Environment

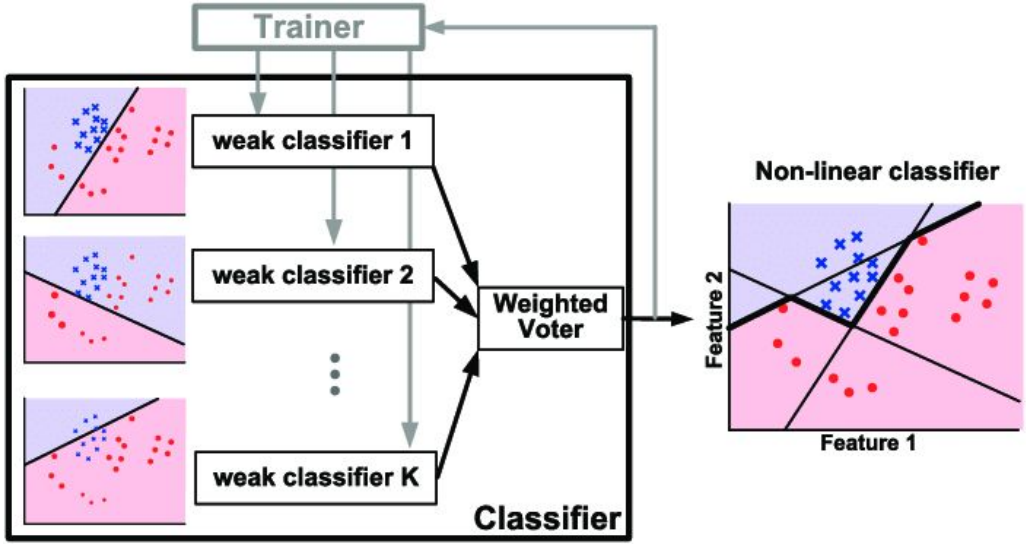
- Recommnedation: Python 3.9 or higher
- Tips
  - We recommend you to use **virtual environments** when implementing your homework assignments.
  - Here are some popular virtual environment management tools
    - Poetry
    - Conda
    - Virtualenv

Numpy & PyTorch

- Numpy Tutorial: Link
- PyTorch Tutorial: Link
  - Allowed to use any optimizer
  - Not allowed to used built-in loss function (Please implement it by your self!)

Adaboost

- AdaBoost is a boosting technique in machine learning. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances.



https://www.researchgate.net/figure/Illustration-of-AdaBoost-algorithm-for-creating-a-strong-classifier-based-on-multiple_fig9_288699540

Bagging

- Train multiple classifiers, each with a proportion of data
- Data is sampled from the whole training set with a sampling with replacement strategy.
- Majority votes for the final prediction

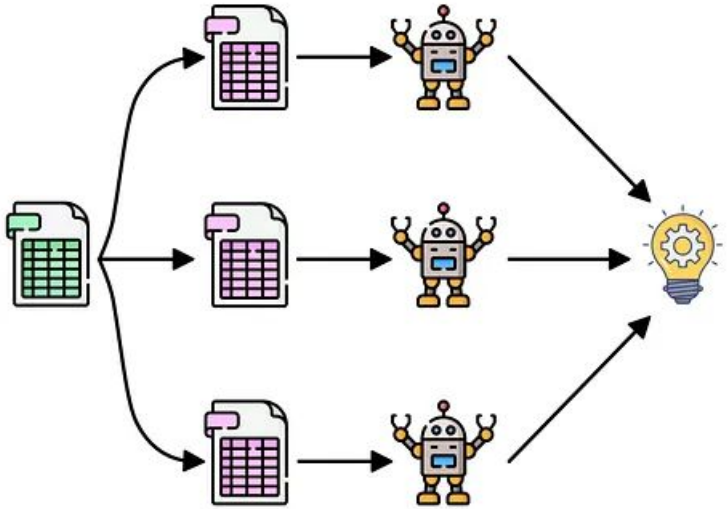


image credit: Fernando López

Decision Tree

- Decision tree is a non-parametric supervised learning algorithm which has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

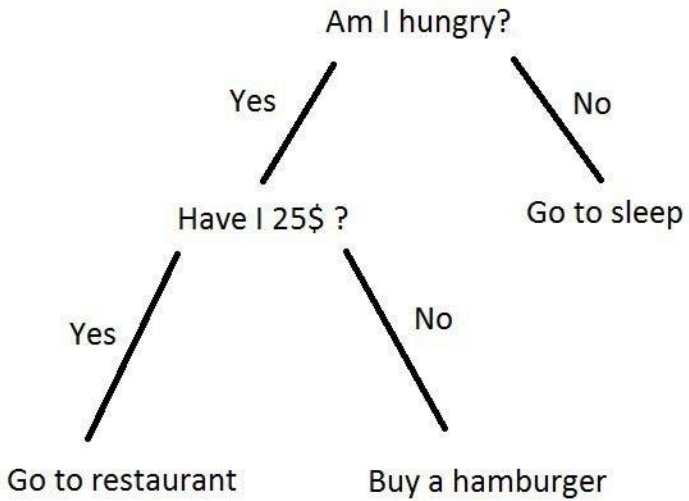


image credit: Rishabh Mall

Dataset and Environment

- Heart Attack Dataset
- Features (ncols=6)
  - age
  - sex
  - cp: chest pain type (4 values)
  - fbs: fasting blood sugar > 120 mg/dl
  - thalach: maximum heart rate achieved
  - thal: 0 = normal; 1 = fixed defect; 2 = reversable defect
- Target
  - target (0 = no heart attack, 1 = heart attack)

- Required packages: `numpy`, `pandas`, `matplotlib`, `loguru`, `flake8`, `scikit-learn`, PyTorch

AdaBoost (20%)

Accuracy	Score (pt)
>=0.77	10
>=0.75	8
>=0.7	5
<0.7	0

- Requirements
  - Implement the AdaBoost method with weak linear classifiers
  - Plot the AUC curve of each weak classifier
  - Plot the feature importance
- Grading Criteria
  - (10%) Show your accuracy of the testing data with 10 estimators. (n_estimators=10)
  - (5%) Plot the AUC curves of each weak classifier.
  - (5%) Plot the feature importance of the AdaBoost method. Also, you should snapshot the implementation of the feature importance estimation.

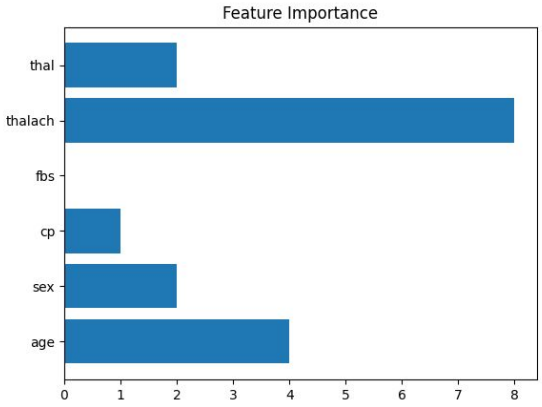
Bagging (20%)

Accuracy	Score (pt)
>=0.77	10
>=0.75	8
>=0.7	5
<0.7	0

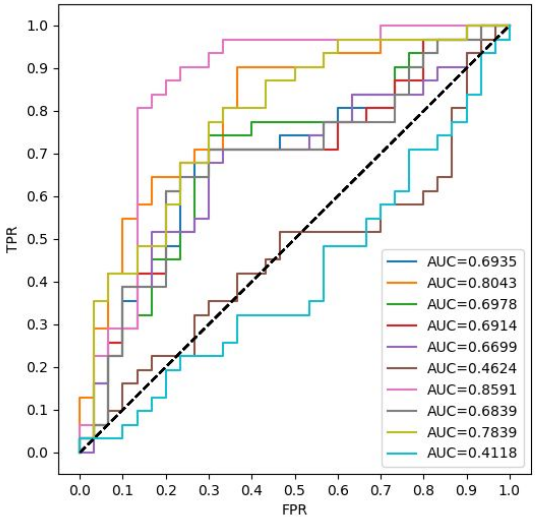
- Requirements
  - Implement the Bagging method with weak linear classifiers
  - Plot the AUC curve of each weak classifier
  - Plot the feature importance
- Grading Criteria
  - (10%) Show your accuracy of the testing data with 10 estimators. (n_estimators=10)
  - (5%) Plot the AUC curves of each weak classifier.
  - (5%) Plot the feature importance of the Bagging method. Also, you should snapshot the implementation of the feature importance calculation.

Plot examples

Examples of figures and code snippet. (Only for reference, not answer)



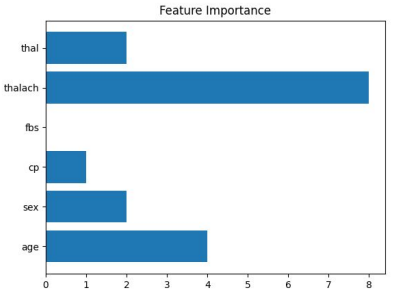

```
def compute_feature_importance(self) -> t.Sequence[float]:
    feature_importance = []
    for
                                                    lphas):
                                                    y()
                                                    e_)
    feat                                            e)
    # su
    retu
```



Decision Tree (15%)

Accuracy	Score (pt)
>=0.72	10
>=0.7	5
<0.7	0

- Requirements
  - Implement entropy for measuring the best split of the data.
  - Implement the decision tree classifier with the argument `max_depth`
- Tips
  - Your model should produce the same results when rebuilt with the same arguments.
- Grading Criteria
  - (5%) Compute the gini index and the entropy of the array [0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1].
  - (5%) Show your accuracy of the testing data with a max-depth = 7
  - (5%) Plot the feature importance of the decision tree.



Only a reference, not answer.

Python Coding Style Guide Reference

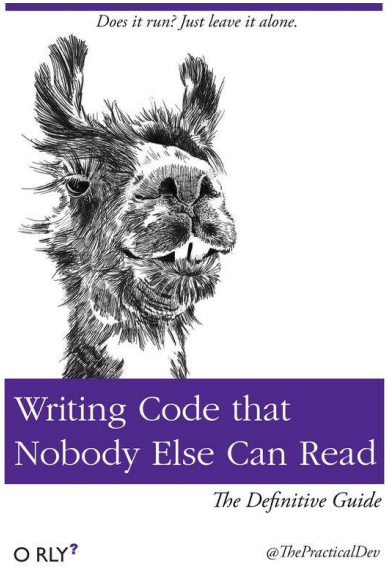
1. PEP8
2. Google Python Style

Additional Requirements (5%)

Code Check and Verification: **Lint** the code (5%)

- Code linting: `$ flake8 main.py`
  - -5pt per warning / error (any issue will make you get no point)

```
./main.py:103:1: W391 blank line at end of file
1     W391 blank line at end of file
```



Handwritten Questions (40%)

2-1 (10%)

We have three distinct binary classifiers, and our goal is to leverage them in creating an ensemble classifier through the majority voting strategy to make decisions. Assuming each individual binary classifier operates independently of the others with an accuracy of 60%, what would be the accuracy of the ensemble classifier?

2-2 (15%)

For the decision tree algorithm, we can use the “pruning” technique to avoid overfitting. Does the random forest algorithm also need pruning?

Please explain in detail.

Handwritten Questions (40%)

2-3 (15%)

Activation functions are core components of neural networks. They need to be differentiable to ensure backpropagation works correctly. Please calculate the derivatives of the following commonly used activation functions.

1. f(x) = relu(x), df(x)/dx = ?
2. f(x) = leaky_relu(x) with negative_slope=0.01, df(x)/dx = ?
3. f(x) = sigmoid(x), df(x)/dx = ?
4. f(x) = silu(x), df(x)/dx = ?
5. f(x) = tanh(x), df(x)/dx = ?

Reference & Notes:

https://pytorch.org/docs/stable/nn.functional.html#non-linear-activation-functions

For questions 1. and 2., consider the cases where $x > 0$ and $x \leq 0$

Report

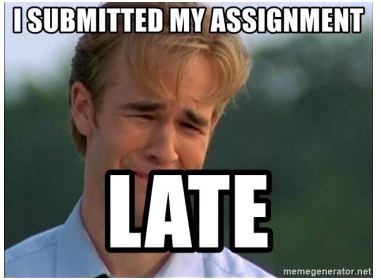
- Please follow the report template format. (-5pts if not use the template)
- Link

Submission

- Compress your **code** and **report** into a **.zip file** and submit it to E3.
- Report should be written in English. (-5 pts if not English)
- <STUDENT ID>_HW3.zip
  - main.py
  - src/
  - setup.cfg
  - <STUDENT ID>_HW3.**pdf** (NO .doc, .docx or others format)
- Don’t put the data (e.g. train.csv / test.csv) into submission file

Other rules

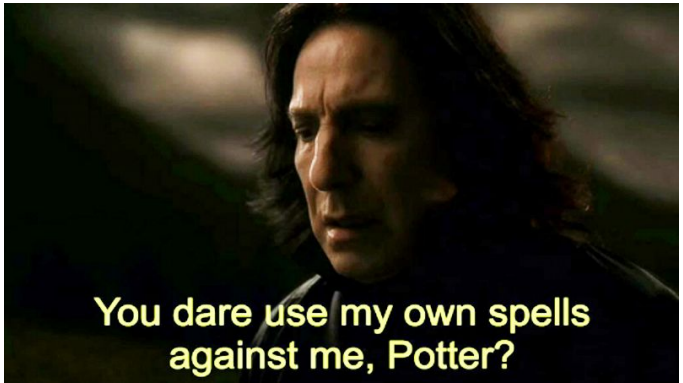
- **Late Policy**: A penalty of **20 points** per additional late day. (-20pt / delayed.day)
  - For example, If you get 90 points but delay for two days, your will get only 50 points!



- **No Plagiarism**: You should complete the assignment by yourself. Students engaged in plagiarism will be penalized heavily. Super serious penalty.
  - e.g. -100pt for the assignment or failed this course, etc
  - Report to academic integrity office

AI-Assistant

- Not recommended but no forbidden
- Copy-and-Paste answers from the AI-Assiant will be seen as Plagiarism
  - However, you can have your own answer first then rephrase it by AI-Assiant.
- Some questions might be parts of final exam, make sure you understand the concept



FAQs

- If you have other questions, ask on **E3 forum** first! We will reply as soon as possible.
  - If the E3 discussion area still have issues, feel free to write email to TAs (And remember to cc all TAs).

Have Fun!