**Fundamentals of Machine Learning (Fall 2022)**

**Homework #4 (100 Pts, Due date: Nov 20)**

**Student ID 2017314712**

**Name 김우진**

**Instruction:** We provide all codes and datasets in Python. Please write your code to complete Decision Tree and BaggingEnsemble(‘models/DecisionTree.py’, ‘models/Bagging.py). You should submit two files as follows:

* ‘STUDENTID\_HW4.zip’: ./models/\*.py and your document. Your document must be converted into a pdf.

**Note**: **You should write your source code in the ‘EDIT HERE’ and not edit other parts.**

1. **[40 pts]** We provide all template codes and datasets in Python. Write your code to implement the decision tree. You must install NumPy, Scikit-learn (sklearn), and Matplotlib libraries.
2. **[10 pts]** Implement the function ‘compute\_entropy**’** in ‘model/DecisionTree.py.’ The entropy is defined as follows:

**Answer: Write your code here. You also have to submit your code to i-campus.**

|  |
| --- |
|  |

1. **[20 pts]** Implement the function ‘selection\_criteria**’** in ‘model/DecisionTree.py.’ The conditional entropy and information gain are defined as follows:

**Answer: Write your code here. You also have to submit your code to i-campus.**

|  |
| --- |
|  |

1. **[10 pts]** After training the model, fill in the blank using the code provided in ‘0\_DecisionTree\_main.py’.

**Answer: Fill in the blank in the table.**

|  |  |
| --- | --- |
| **Max depth** | **Accuracy** |
| 3 | 0.7696 |
| 4 | 0.7958 |
| 5 | 0.8586 |

1. **[40 pts]** We provide all template codes and datasets in Python. Write your code to implement the bagging ensemble. You may need to install NumPy, Scikit-learn (sklearn), and Matplotlib libraries.
2. **[10 pts]** Implement the function ‘create\_datasets’ in ‘model/BaggingSampler.py’. You need to generate new datasets by randomly sampling the raining dataset with a given sampling ratio in this function.

**Answer: Write your code here. You also have to submit your code to i-campus.**

|  |
| --- |
|  |

1. **[10 pts]** Implement the function ‘random\_feature\_selection’ in ‘model/DecisionTree.py’.

**Answer: Write your code here. You also have to submit your code to i-campus.**

|  |
| --- |
|  |

1. **[20 pts]** Fill in the blank using the code provided in ‘1\_RandomForest\_main.py.’ Both depth and number of features are fixed as 3.

**Answer: Fill in the blank in the table.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Sampling ratio** | **Num datasets** | **Accuracy** |
| Decision tree | - | - |  |
| Random Forest | 0.6 | 1 | 0.6545 |
| 10 | 0.8010 |
| 50 | 0.8325 |
| 100 | 0.8325 |
| 0.1 | 50 | 0.8272 |
| 0.3 | 0.8325 |
| 0.5 | 0.8325 |
| 0.7 | 0.7958 |

1. **[20 pts]** Given the “breast\_cancer” data and ‘2\_Performance\_Comparison.py’, draw a plot that depicts performances of decision tree, random forest, AdaBoost. Briefly explain your result. You can use the scikit-learn library

**Answer:**

|  |
| --- |
| **Decision Tree는 값이 항상 동일하다는 것을 알수있다**  **Random Forest는 estimator 개수가 늘어나면 보통 acc가 늘어난다.**  **이 seed에서는 동일하게 나온부분이있지만 다른seed로 해봤을 때 늘어났다**  **Adaboost도 estimator 개수가 늘어날수록 성능이 올라갔다**  **Adaboost가 가장 뛰어난 성능을 보였다.** |