Statement PDF

1.) Is the code your own implementation, an adaptation of existing code, or
a direct copy of existing code? Note that, only changing the name of variables, functions and methods is still considered a direct copy.

a) Own implementation
b) Adaptation of existing code
c) Direct copy

If you adapted existing code, how exactly did you modify the code? Did you implement any new feat ures? Which part of the code is your own work?

- implementing the Gini index in decision tree and using NumPy

- used existing code only as a base template to understand the overall structure of a decision tree (building one from scratch)

2.) Please list all the external resources used, and how they were used. An external resource includes e xisting code, libraries, web pages, books, web forums, friends, etc. Note that this question is required even if you worked on your own implementation.

Libraries

- NumPy library was used for the custom implementation of the decision tree- used for unique labels (np.unique- getting the unique elements) and np.sum- for the sum of elements in an array, np.argmax- to get the elements with maximum values

- NumPy was also used for the comparison of the sklearn decision tree and custom implementation- when converting X and y to numpy arrays for the datasets.

- Pandas library was used for the comparison of the sklearn decision tree and custom implementation- to convert dataframe of metric results to .csv

Webpages

https://github.com/tanvipenumudy/Winter-Internship-Internity/blob/4dd1795716a2506562d86e13d2d242033a814f72/Day%2009/Day-9%20Notebook-1%20(Decision%20Tree).ipynb#L4

- **Git-hub example of a decision implementation**
- Seeing an example of a decision tree and how it was formed and structured helped me in my first steps of building a decision tree- looking at what functions they used
- They used entropy instead of gini index

https://anderfernandez.com/en/blog/code-decision-tree-python-from-scratch/

- "**How to program a decision tree in Python from 0"**
- Understanding what hyperparameters I need to use – max_depth and min_sample_split is what I used
- Understanding how gini index works

https://www.datacamp.com/tutorial/decision-tree-classification-python

- **“Decision Tree Classification in Python Tutorial”**
- Understanding the theory behind a decision tree
- Explaining gini index
- How to import the sklearn decision tree classifier and metrics
- Understanding how to split the data as X_train, X_test, y_train, y_test
- Learning how to build the sklearn decision tree : clf = DecisionTreeClassifier()
- Understanding how to include hyperparameters and criteria like “gini” and “max_depth” and “min_sample_split”

https://pandulaofficial.medium.com/implementing-cart-algorithm-from-scratch-in-python-5dd00e9d36e

- “**Implementing a classification tree with Gini Impurity from scratch in Python**”
- How to implement gini in decision tree custom implementation

https://inria.github.io/scikit-learn-mooc/python_scripts/trees_hyperparameters.html#:~:text=The%20hyperparameter%20max_depth%20controls%20the,the%20impact%20of%20the%20parameter.

- Hyperparameters and understanding how to include them in my code for sklearn decision tree and custom

https://docs.pytest.org/en/stable/how-to/assert.html

- **Understanding what assertions are and how to use them**

https://stackoverflow.com/questions/22591297/run-same-test-on-multiple-datasets

- **Helpful for understanding pytest more and @pytest.fixture**

Videos

https://youtu.be/wxS5P7yDHRA?si=Y0REJyMFGC9OwKwv

- “**How to Implement Decision Trees in Python (Train, Test, Evaluate, Explain)”**
- Helpful to see someone training sklearn decision tree and testing it
- Helped when looking at what performance metrics other people used
- Useful to see how they used parameters
- How to do test_sizes

https://youtu.be/YkYpGhsCx4c?si=5LO6ovLifMXtKs2J

- “how to build your first decision tree in python”
- Understanding what a decision tree is
- How to import sklearn decision tree and importing train_test_split
- Using f1 score metric

https://youtu.be/NxEHSAfFlK8?si=L-B2NkzOizBgnyCe

- How to implement decision trees from scratch with python
- Understanding how decision trees work a more in-depth video
- What needs to be included in a decision tree: split feature and split point
- Giving an example writing an own implementation of a decision tree
    - Including stopping criteria
    - Was helpful because they used numpy as well
    - How to structure a decision tree and what to include
    - They used entropy instead of gini index

https://youtu.be/mzlH8lp4ISA?si=dbmVPhd5vNaqnaCf

- Video explaining how to run pytests
- How to use assert
- function tests better than inheritance which allowed me to understand how function tests would work

Other external resources

- Someone that I know helped me- they study computer science and have taken a module on Artificial Intelligence where they touch on Decision Trees- helped me with the overall structure of coding, helped with the memory profiler/tracemalloc (memory-usage metric) and other performance metrics. Also helped with debugging.