

Machine Learning

Assignment 2

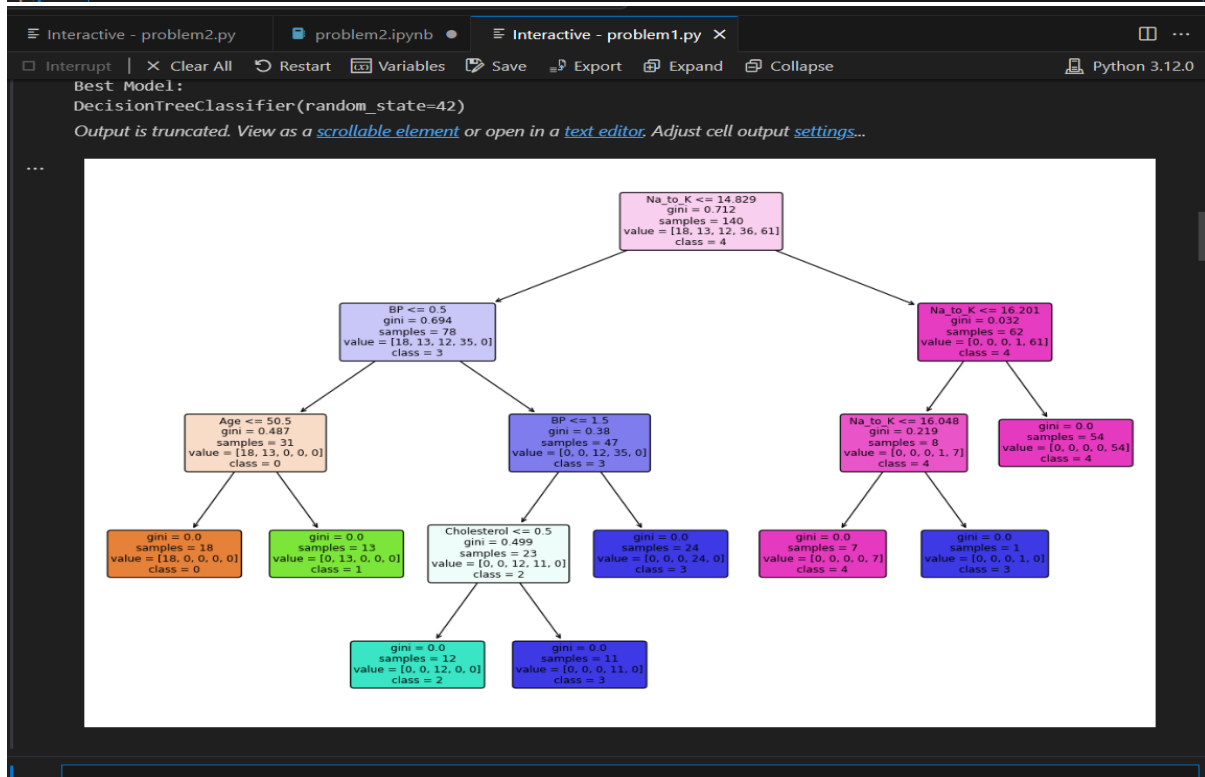
Name	ID
Eman Fathy Abo Alhassan	20200105
Sara Ahmed Sayed	20200214
Fatma Mahmoud Ramadan	20201134
Basmala Magdy Mohamed	20201045
Esraa Osama Mohamed	20201014

Output Screenshots

Problem 1 :

```
Assignment2
Interactive - problem2.py  problem2.ipynb  Interactive - problem1.py x
Interrupt | x Clear All | Restart | Variables | Save | Export | Expand | Collapse | Python 3.12.0

... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Age         200 non-null   int64
1   Sex         200 non-null   object
2   BP          198 non-null   object
3   Cholesterol  198 non-null   object
4   Na_to_K     199 non-null   float64
5   Drug        200 non-null   object
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
Missing Values:
Age      0
Sex      0
BP       2
Cholesterol  2
Na_to_K  1
Drug     0
dtype: int64
After Handling Missing Values:
Age      0
Sex      0
BP       0
...
Best Model Accuracy: 1.0
Best Model Tree Size: 15
Best Model:
DecisionTreeClassifier(random_state=42)
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```



```

Assignment2
Interactive - problem2.py  problem2.ipynb  Interactive - problem1.py X
Interrupt | X Clear All | Restart | Variables | Save | Export | Expand | Collapse | Python 3.12.0

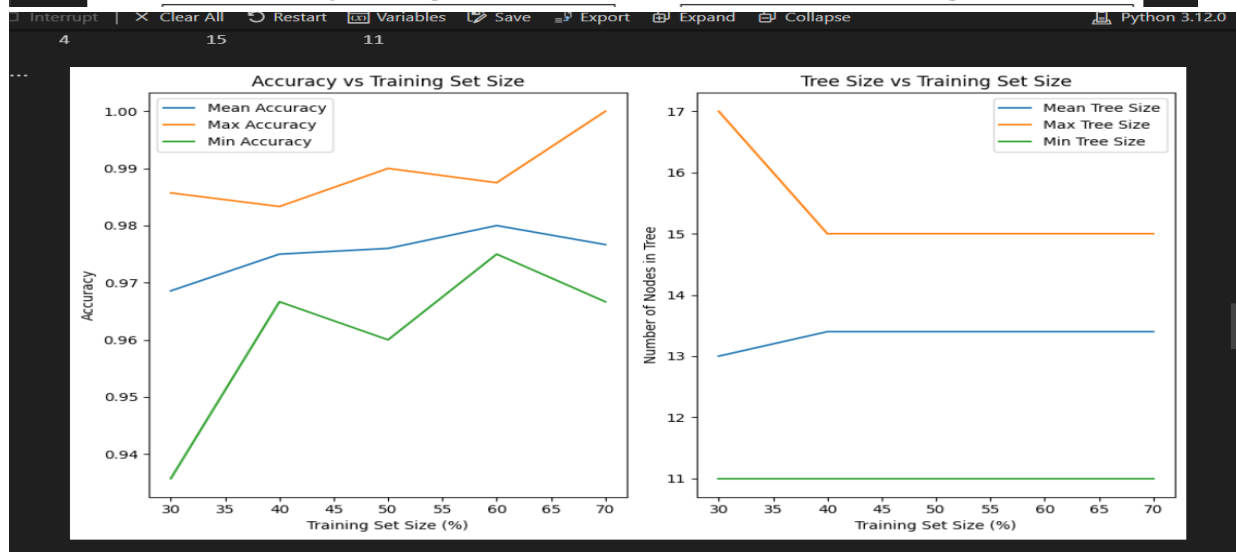
[2] ✓ 0.4s

... Experiment Report:
  Train_Size  Mean_Accuracy  Max_Accuracy  Min_Accuracy  Mean_Tree_Size \
0           30         0.968571         0.985714         0.935714         13.0
1           40         0.975000         0.983333         0.966667         13.4
2           50         0.976000         0.990000         0.960000         13.4
3           60         0.980000         0.987500         0.975000         13.4
4           70         0.976667         1.000000         0.966667         13.4

  Max_Tree_Size  Min_Tree_Size
0              17             11
1              15             11
2              15             11
3              15             11
4              15             11

...

```

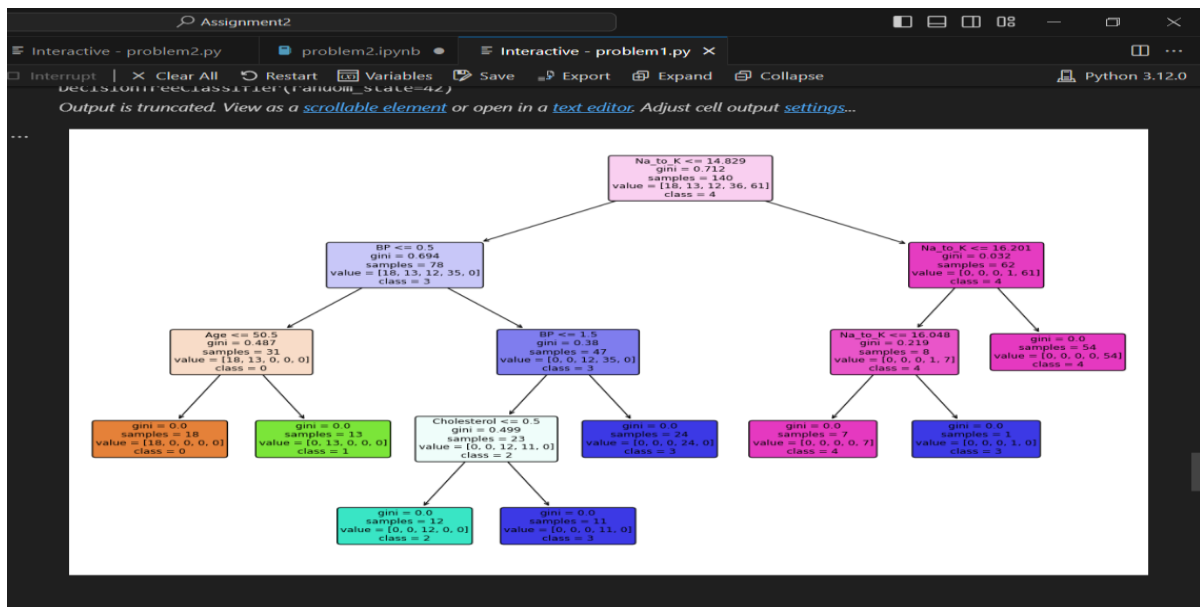


```

[3] ✓ 0.7s

... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Age         200 non-null   int64
1   Sex         200 non-null   object
2   BP          198 non-null   object
3   Cholesterol  198 non-null   object
4   Na_to_K     199 non-null   float64
5   Drug        200 non-null   object
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
Missing Values:
Age      0
Sex      0
BP       2
Cholesterol  2
Na_to_K  1
Drug     0
dtype: int64
After Handling Missing Values:
Age      0
Sex      0
BP       0
...
Best Model Accuracy: 1.0
Best Model Tree Size: 15
Best Model:
DecisionTreeClassifier(random_state=42)

```



Interactive - problem2.py problem2.ipynb Interactive - problem1.py

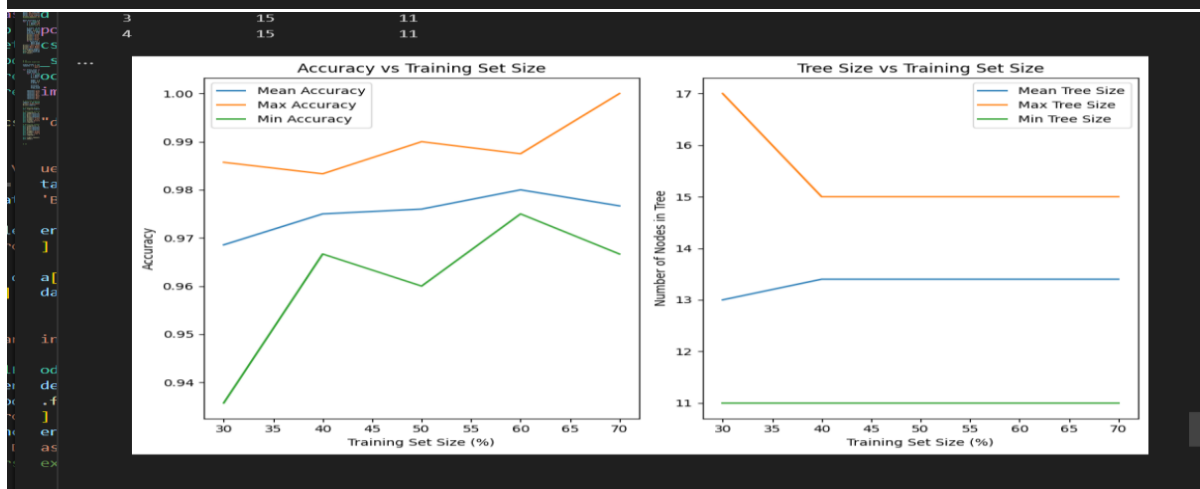
Interrupt | X Clear All | Restart | Variables | Save | Export | Expand | Collapse

Python 3.12.0

Experiment Report:

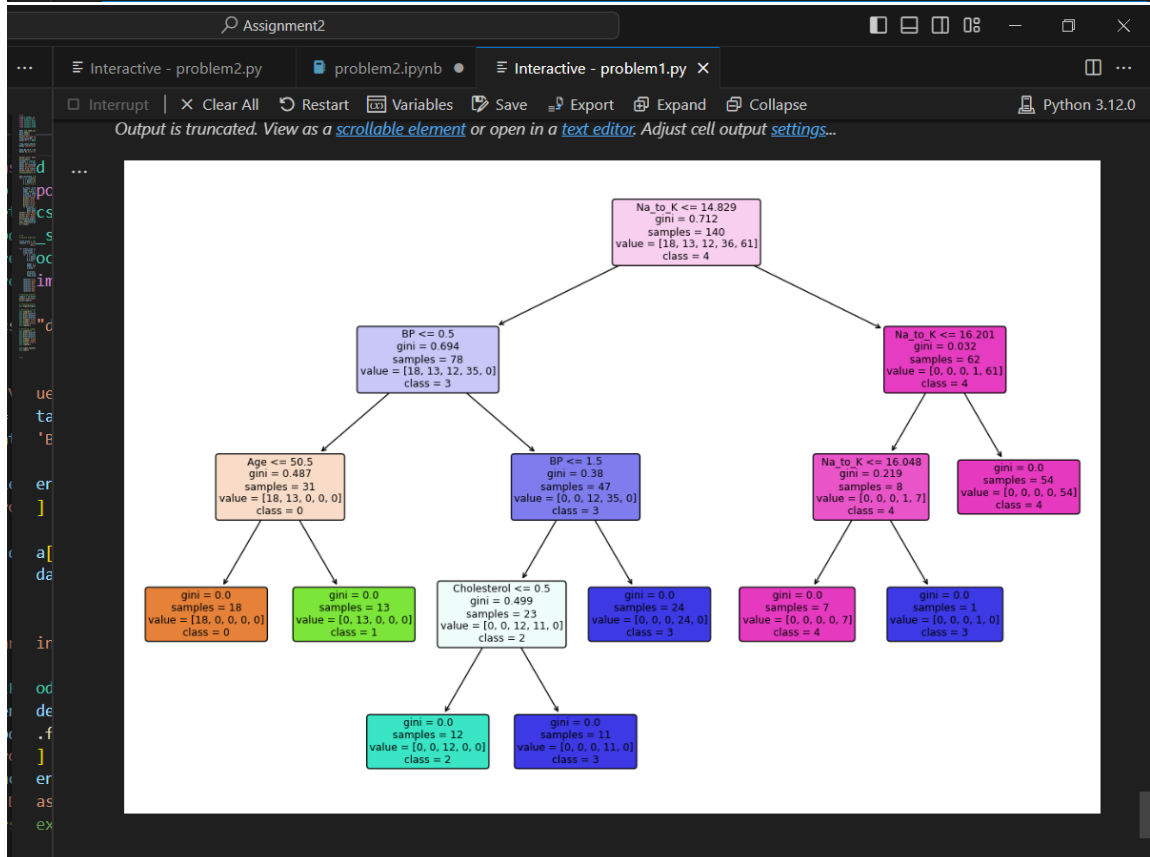
	Train_Size	Mean_Accuracy	Max_Accuracy	Min_Accuracy	Mean_Tree_Size \
0	30	0.968571	0.985714	0.935714	13.0
1	40	0.975000	0.983333	0.966667	13.4
2	50	0.976000	0.990000	0.960000	13.4
3	60	0.980000	0.987500	0.975000	13.4
4	70	0.976667	1.000000	0.966667	13.4

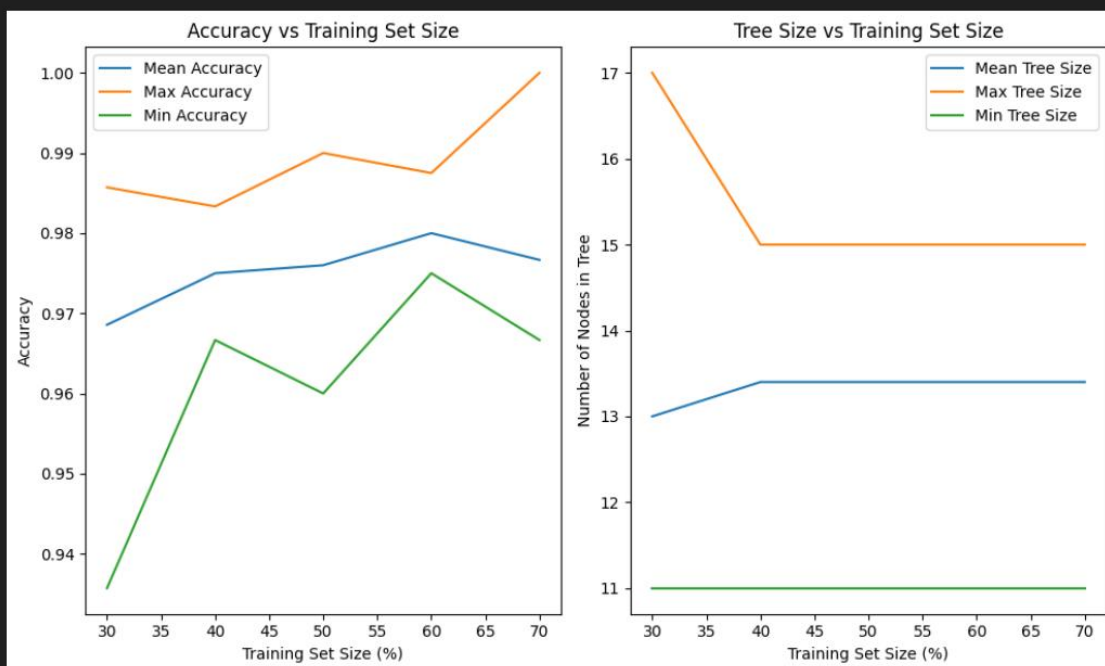
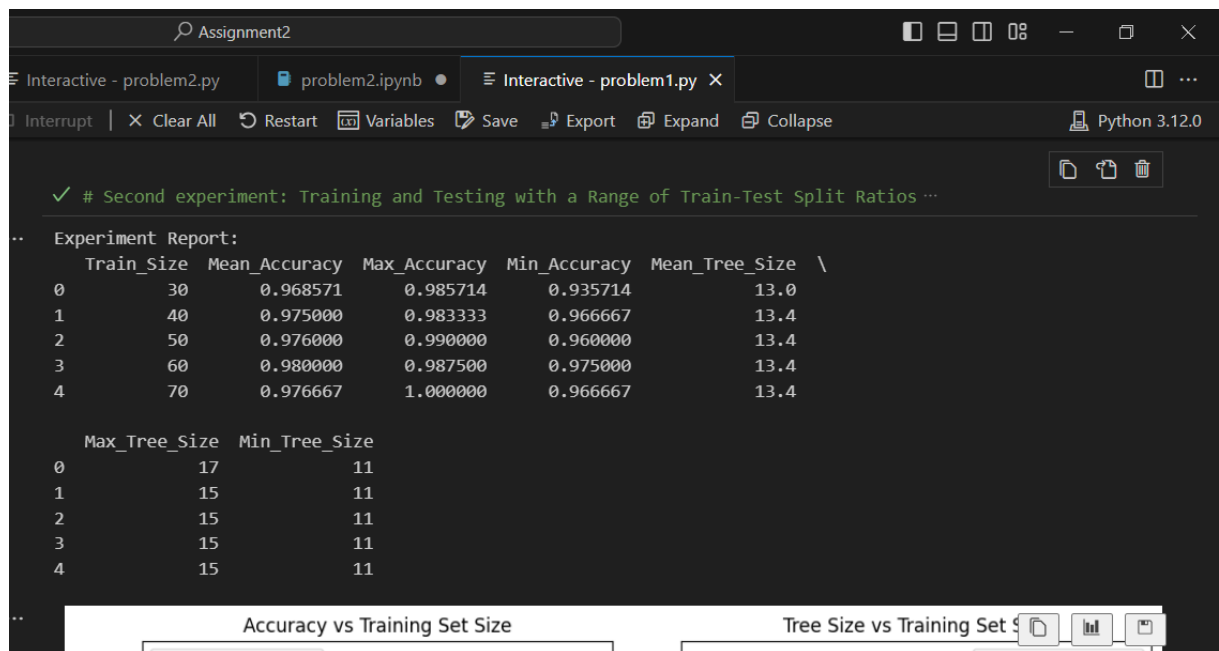
	Max_Tree_Size	Min_Tree_Size
0	17	11
1	15	11
2	15	11
3	15	11
4	15	11



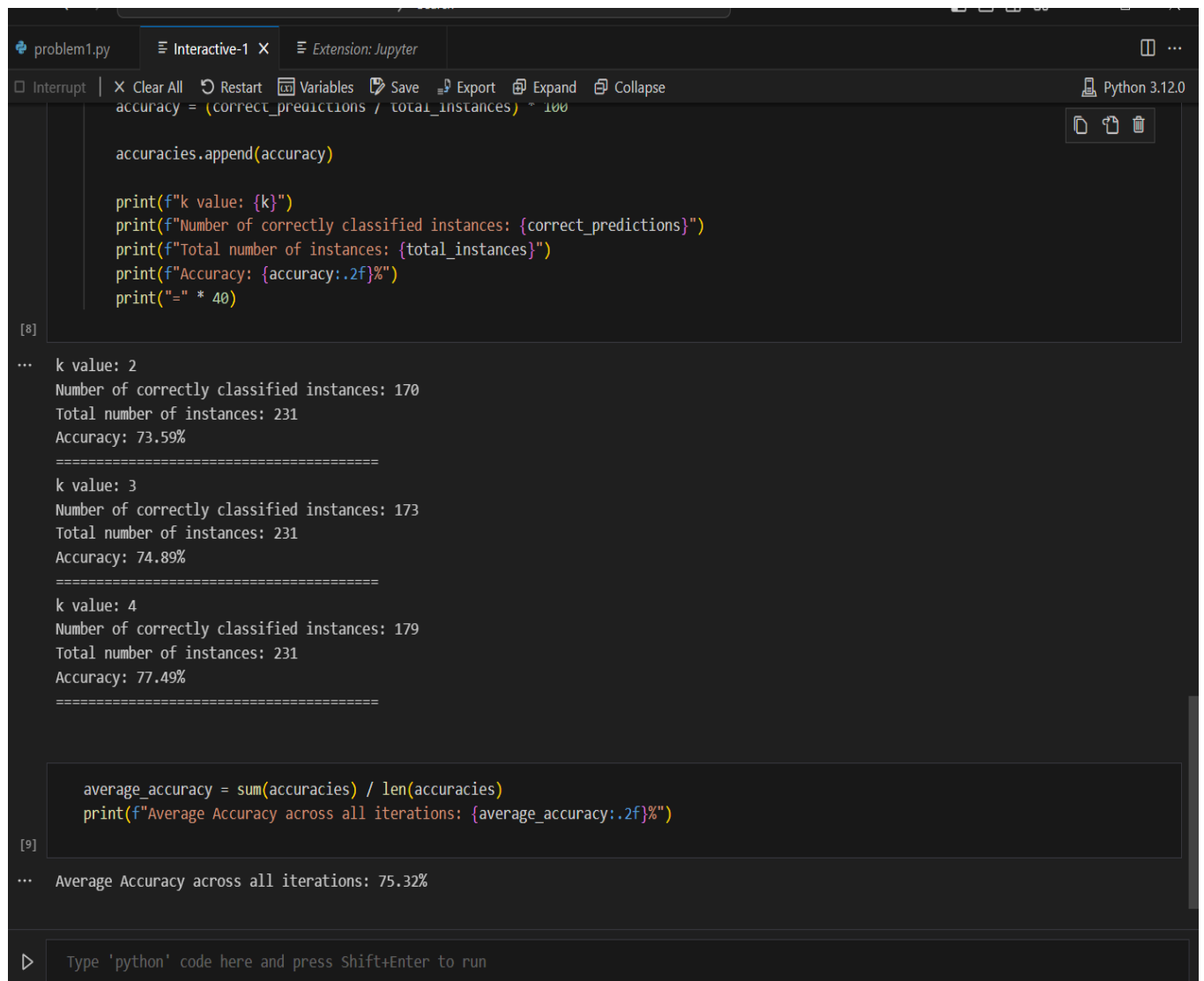
```
Interrupt | X Clear All | Restart | Variables | Save | Export | Expand | Collapse | Python 3.12.0
import pandas as pd

... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Age          200 non-null    int64
1   Sex          200 non-null    object
2   BP           198 non-null    object
3   Cholesterol  198 non-null    object
4   Na_to_K      199 non-null    float64
5   Drug         200 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
Missing Values:
Age          0
Sex          0
BP           2
Cholesterol  2
Na_to_K      1
Drug         0
dtype: int64
After Handling Missing Values:
Age          0
Sex          0
BP           0
...
Best Model Accuracy: 1.0
Best Model Tree Size: 15
Best Model:
DecisionTreeClassifier(random_state=42)
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```





Problem 2:



The image shows a Jupyter Notebook interface with a dark theme. The top bar includes a file tab 'problem1.py', a tab 'Interactive-1', and an extension 'Jupyter'. Below the top bar is a toolbar with icons for 'Interrupt', 'Clear All', 'Restart', 'Variables', 'Save', 'Export', 'Expand', and 'Collapse'. The right side of the toolbar shows 'Python 3.12.0'. The main area contains a code cell with the following Python code:

```
accuracy = (correct_predictions / total_instances) * 100

accuracies.append(accuracy)

print(f"k value: {k}")
print(f"Number of correctly classified instances: {correct_predictions}")
print(f"Total number of instances: {total_instances}")
print(f"Accuracy: {accuracy:.2f}%")
print("=" * 40)
```

The output of the code cell is as follows:

```
[8]
... k value: 2
    Number of correctly classified instances: 170
    Total number of instances: 231
    Accuracy: 73.59%
    =====
    k value: 3
    Number of correctly classified instances: 173
    Total number of instances: 231
    Accuracy: 74.89%
    =====
    k value: 4
    Number of correctly classified instances: 179
    Total number of instances: 231
    Accuracy: 77.49%
    =====
```

Below the code cell is a new code cell with the following Python code:

```
average_accuracy = sum(accuracies) / len(accuracies)
print(f"Average Accuracy across all iterations: {average_accuracy:.2f}%")
```

The output of this code cell is:

```
[9]
... Average Accuracy across all iterations: 75.32%
```

At the bottom of the notebook, there is a prompt: 'Type 'python' code here and press Shift+Enter to run'.