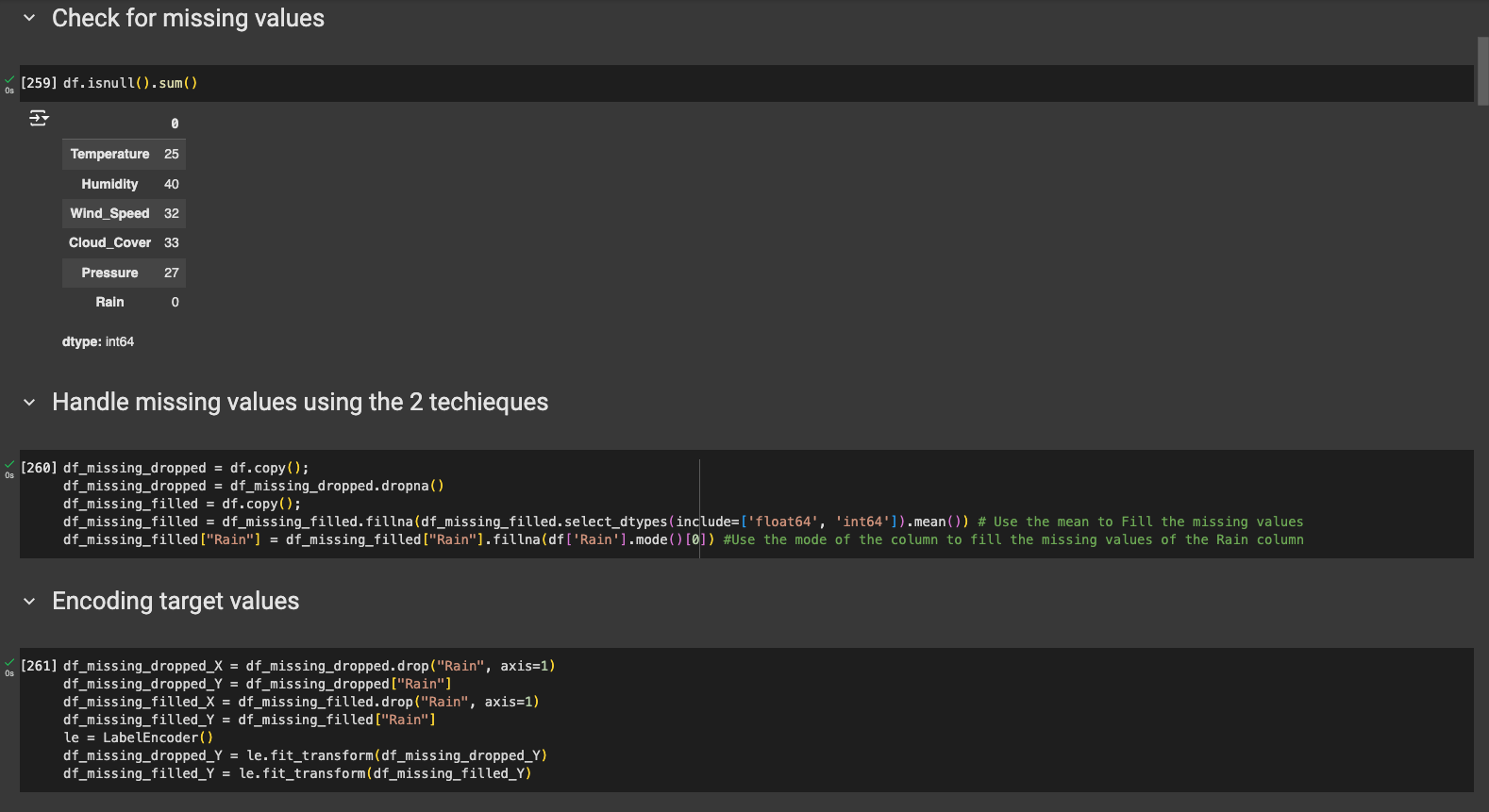
**Machine Learning**

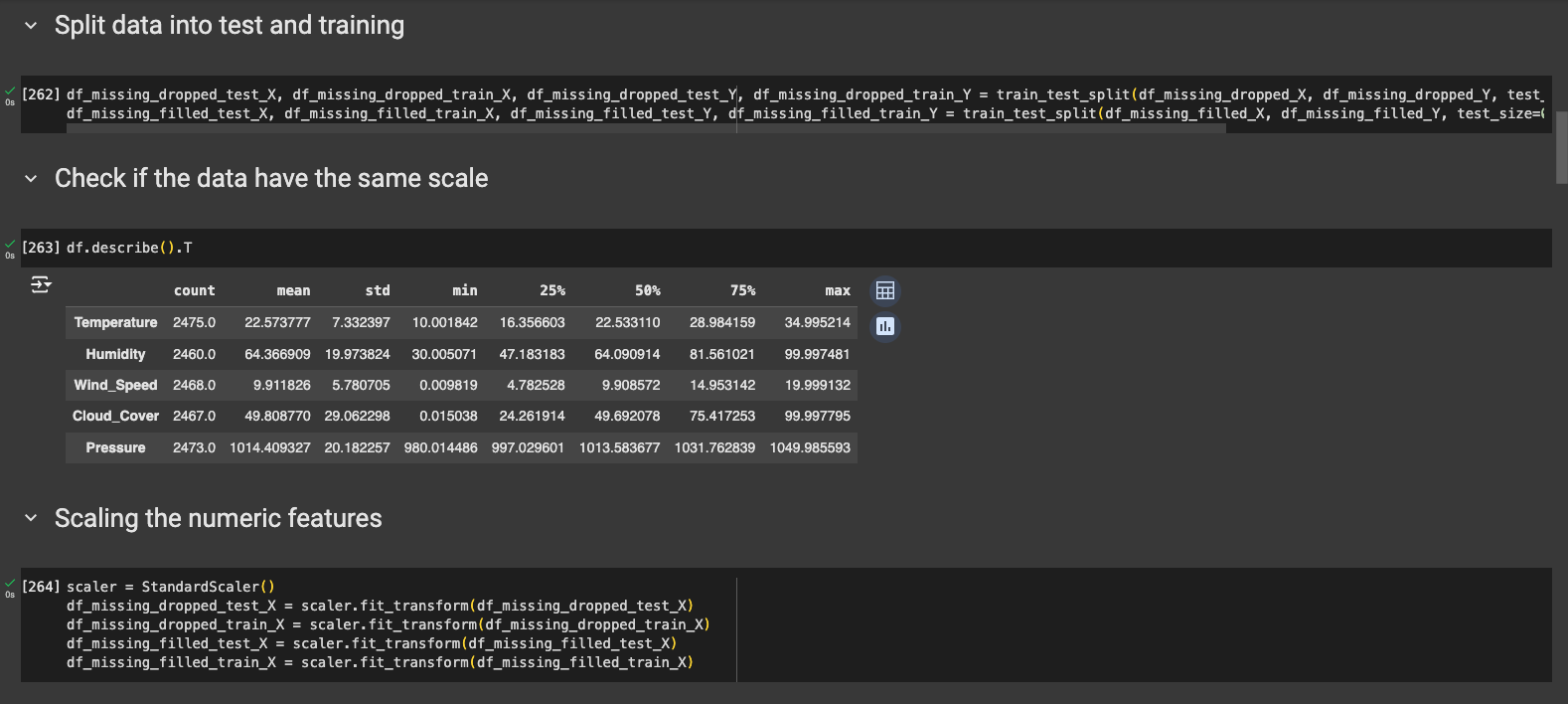
**Assignment 2**

Group: IS-S5 & IS-S6

|  |  |
| --- | --- |
| **Name** | **ID** |
| Mazen Khaled Ahmed | 20210314 |
| Mohamed Hany Saad | 20210358 |
| Ahmed Ezzat Hammed | 20210026 |
| Ahmed Yasser Badawy | 20210047 |
| Mazen Mahmoud Adly | 20210318 |

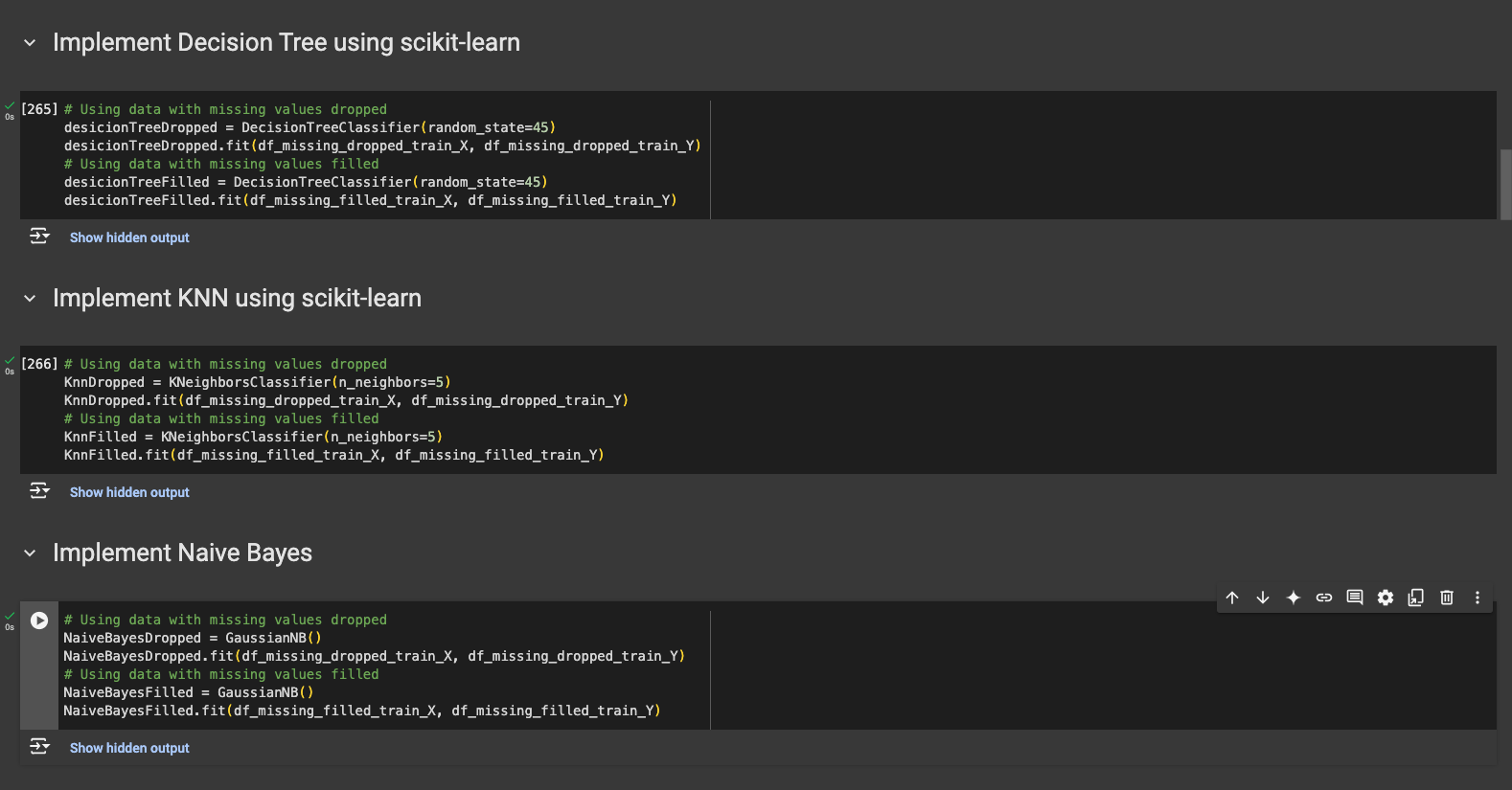
Task 1:



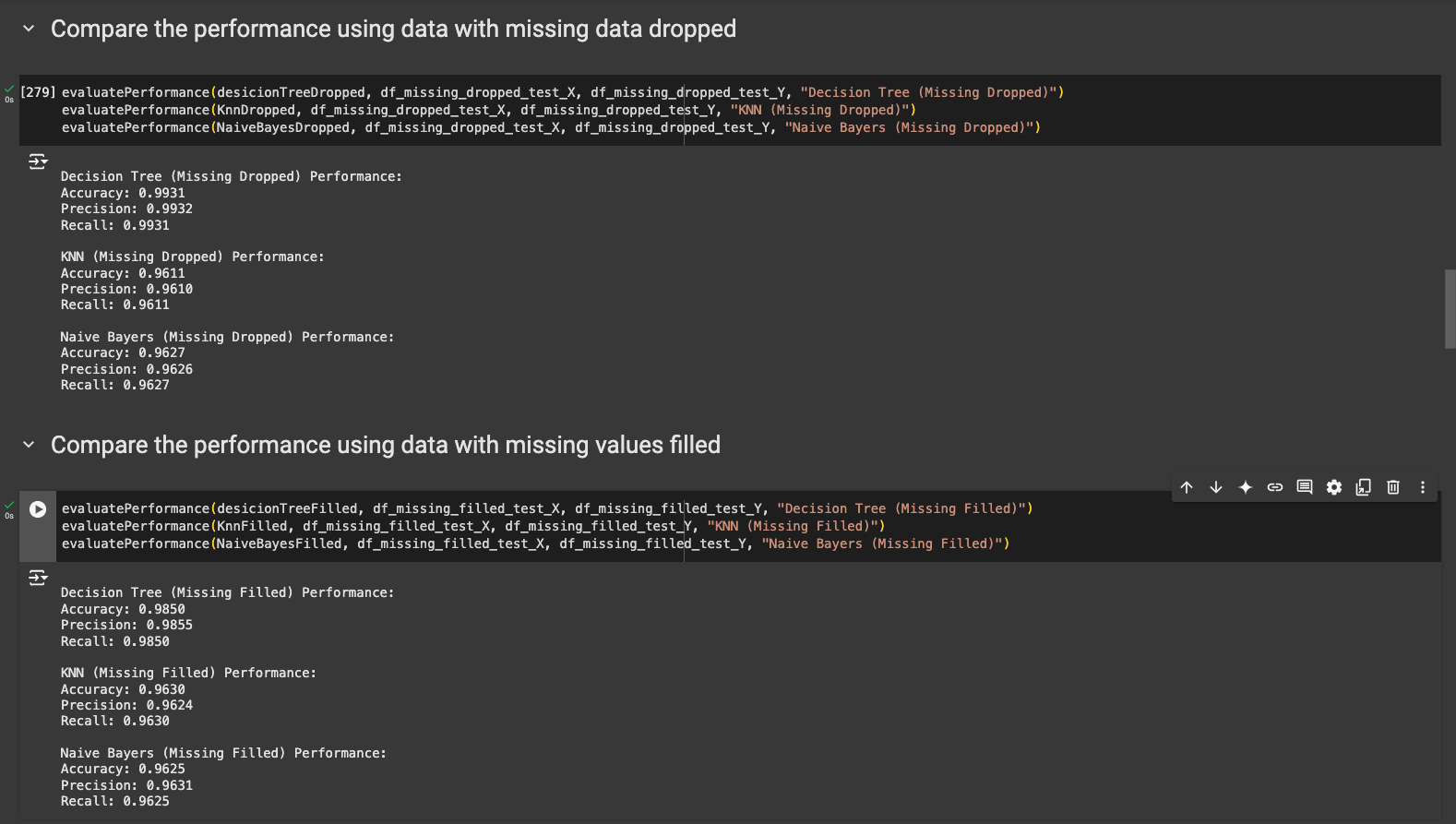


Task 2:

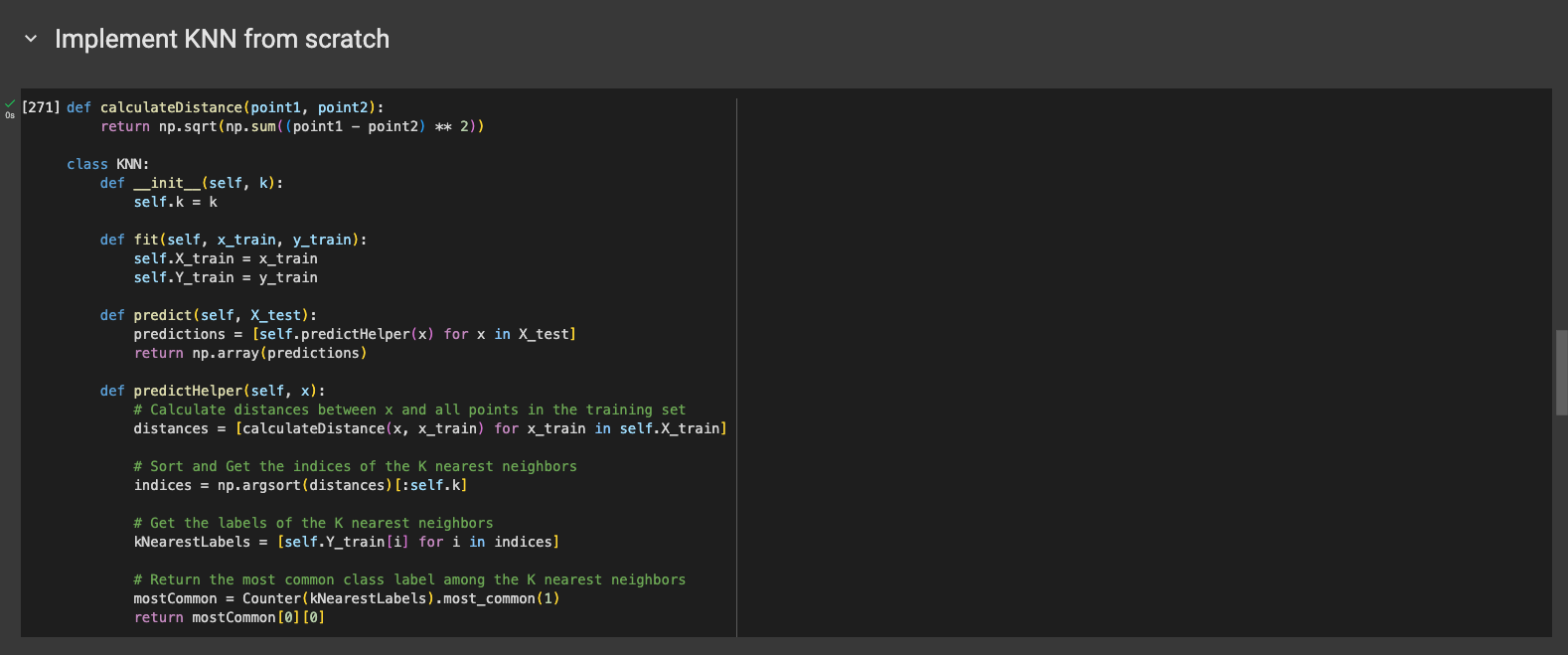
1.



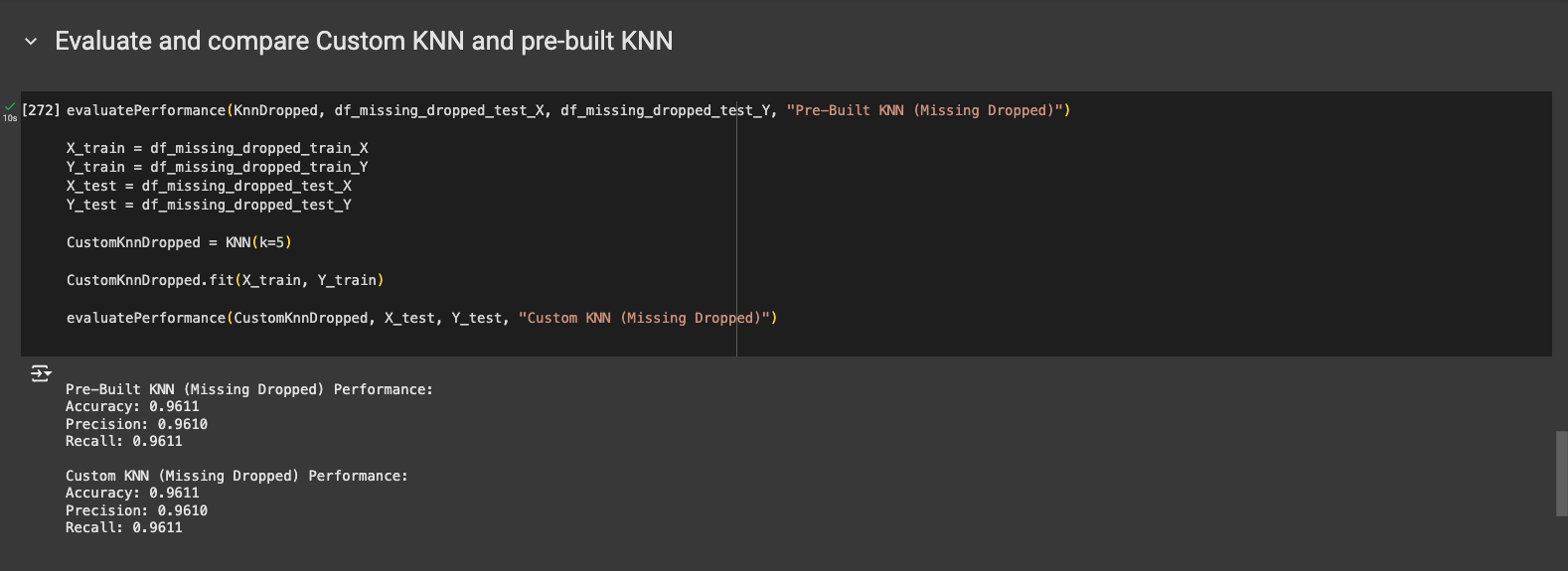
2.



3.



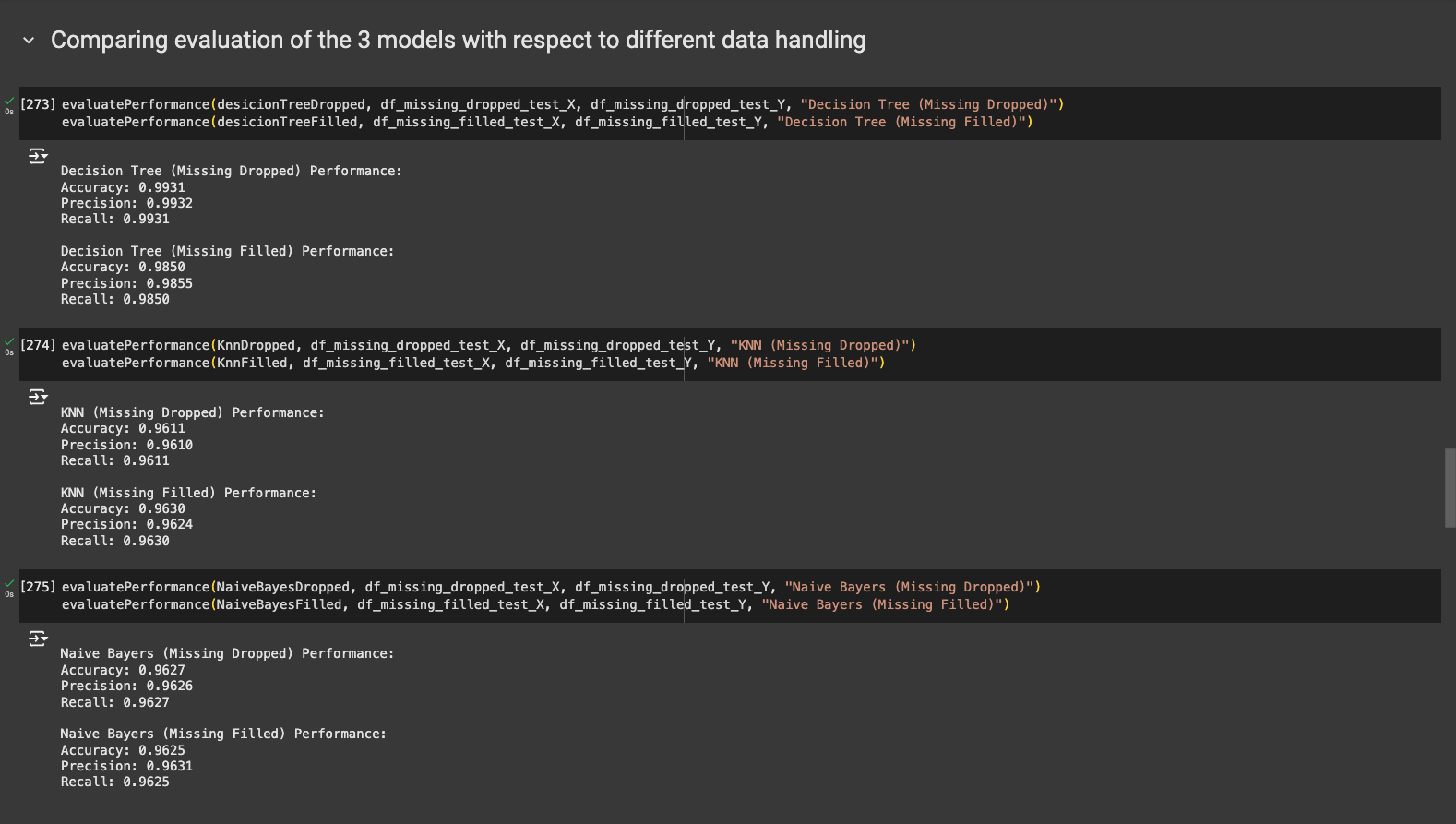
4.



After evaluating both the custom and pre-built KNN models using Accuracy, Precision and Recall the results show that both models accomplished the same performance.

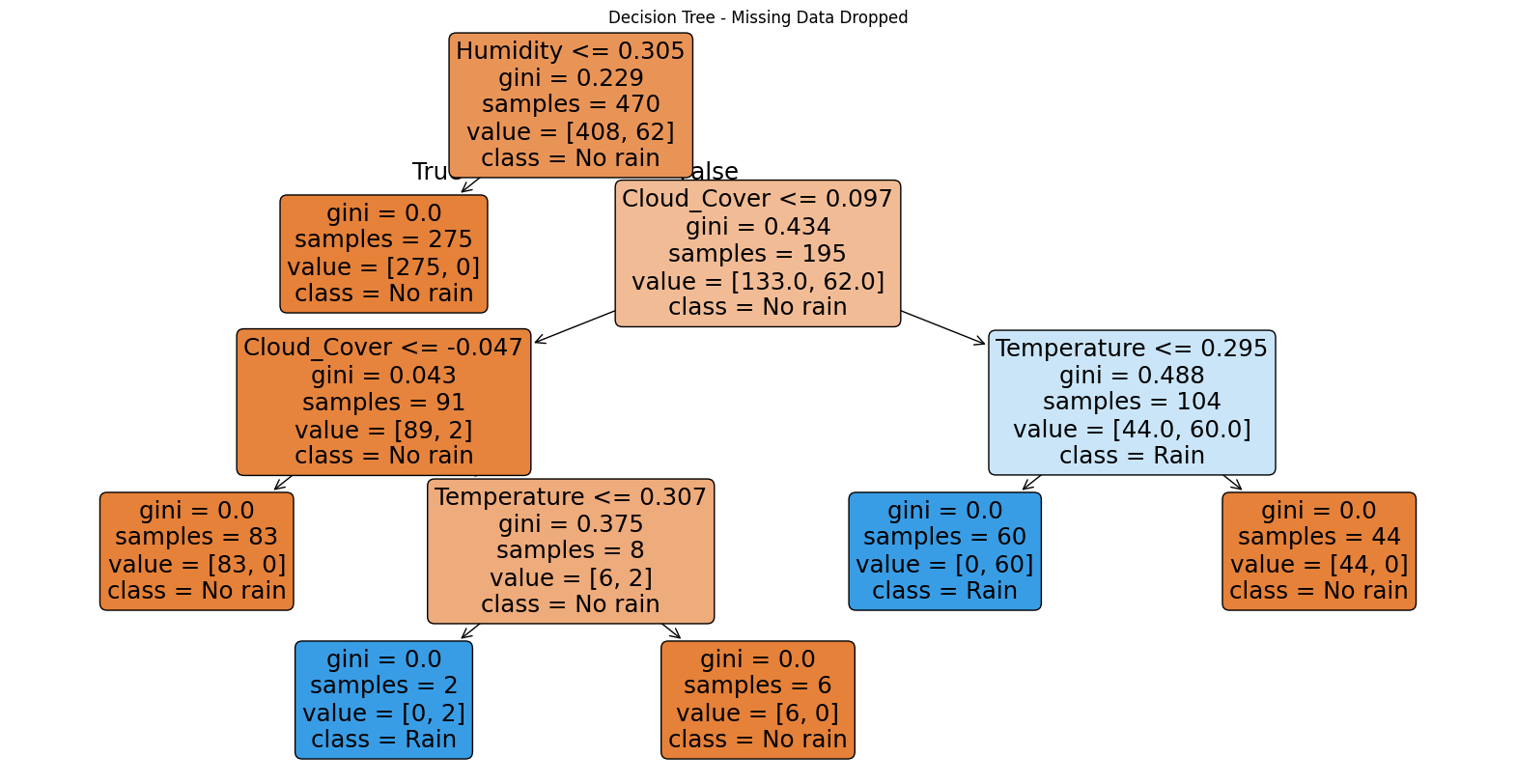
Task 3:

1.

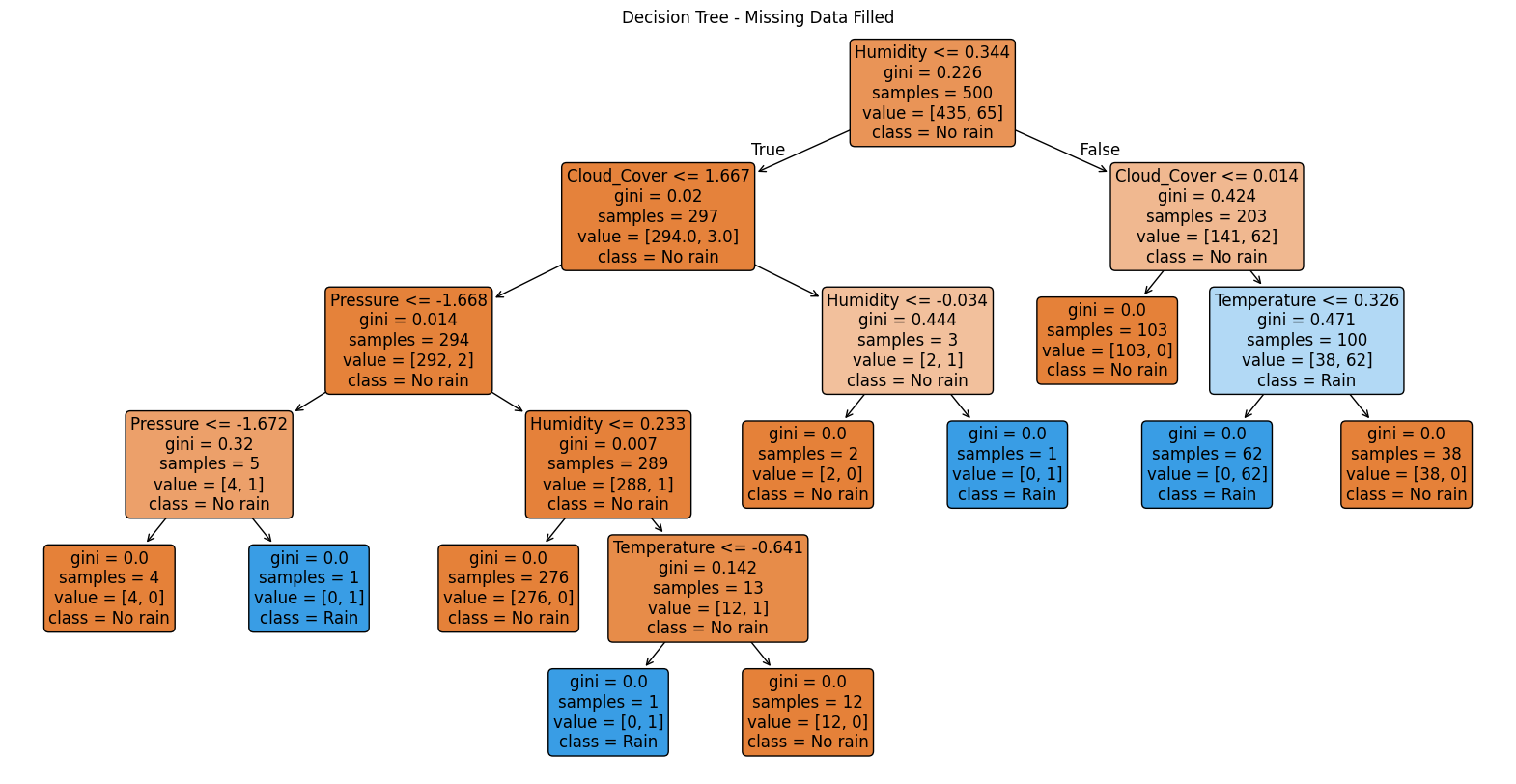
After comparing the performance of the 3 models with respect to different missing values handling techniques, the results show that there is no huge difference in performance of each model however it shows that the performance was slightly better when trained on data with the missing values dropped except for the Naive Bayes model that shows almost no difference.

2.

Decision tree on data with missing values dropped:



Decision Tree on data with missing values filled:



The decision tree predicts outcomes by traversing from the root node to a leaf node based on decision rules defined at each node. Here's how the tree makes predictions:

1. Start at the Root Node:

The root node splits based on some condition, for example in the decision tree on data with missing values dropped is Humidity <= 0.305.

If the condition is True, move to the left child node If False, move to the right child node.

1. Traverse Based on Conditions:

At each node, a condition is evaluated, which determines the next branch to follow.

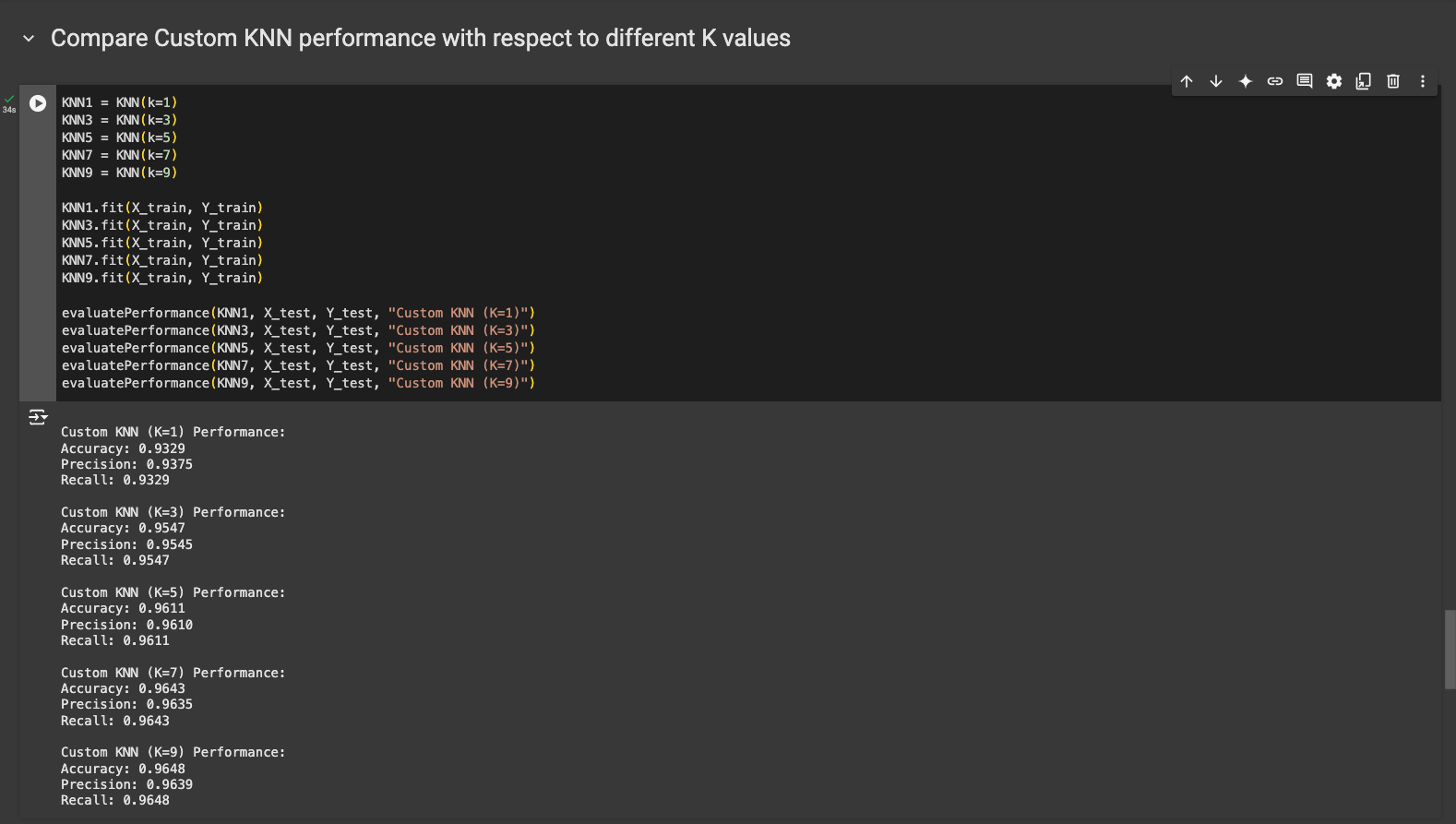
This process continues until a leaf node is reached, where the prediction is made.

1. Predictions at Leaf Nodes:

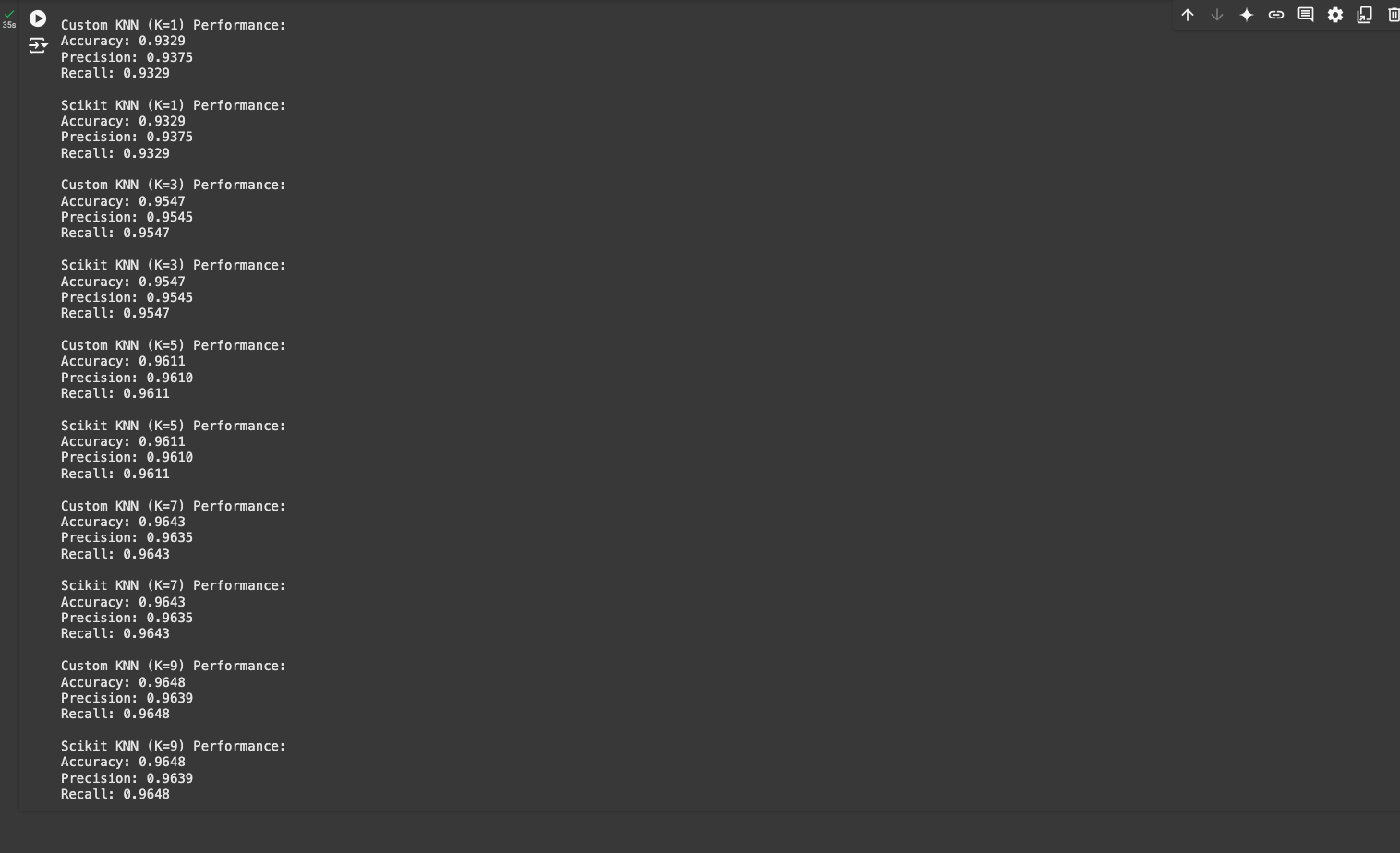
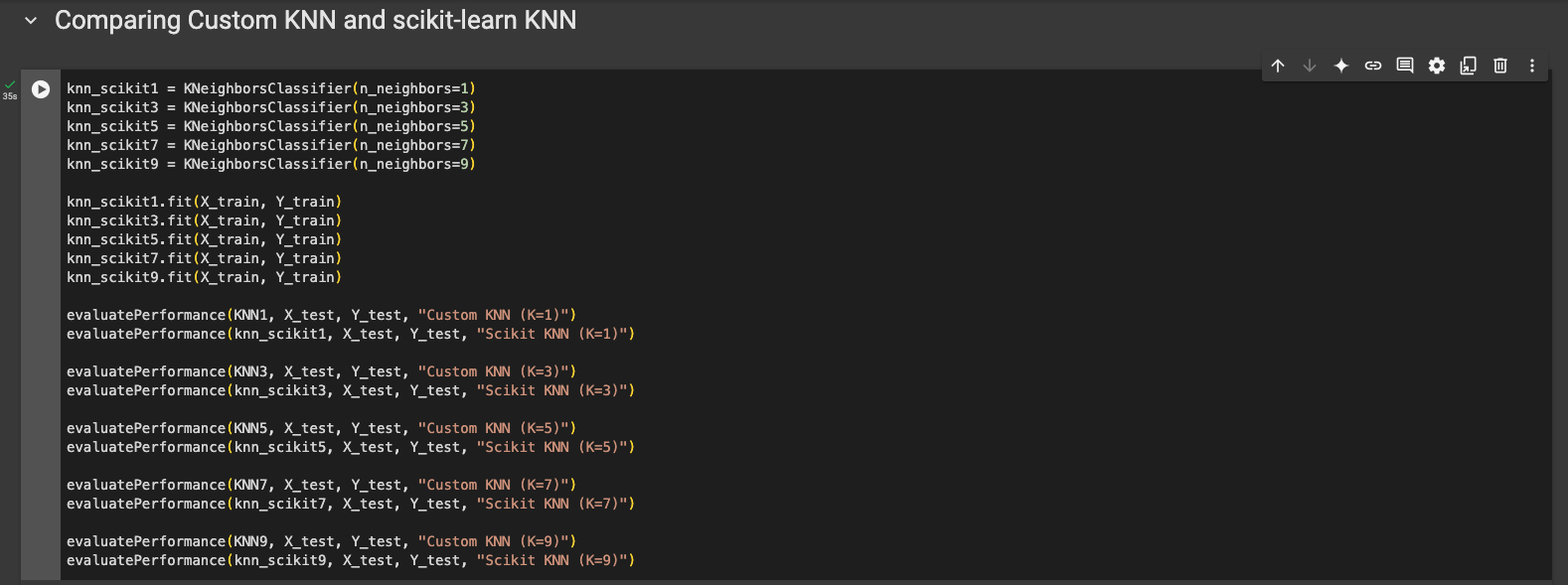
Each leaf node represents a class (in our example either "Rain" or "No rain").

For example: If the path ends at a node with class = No rain, the prediction for that instance is No rain. And, if class = Rain, the prediction is Rain.

1. The decision tree uses impurity as the criterion to determine the best splits. Impurity measures the impurity of classes at a node, and the algorithm goal is to minimize this value with each split.

3. 

After evaluating the performance of the created from scratch KNN model with 5 different K values and comparing the results, it showed that the model performed better each time the value of K increased.



After evaluating the performance of the created from scratch KNN model and the one implemented using Scikit-Learn and comparing the results, it showed no difference since the 2 models accomplished the same performance.