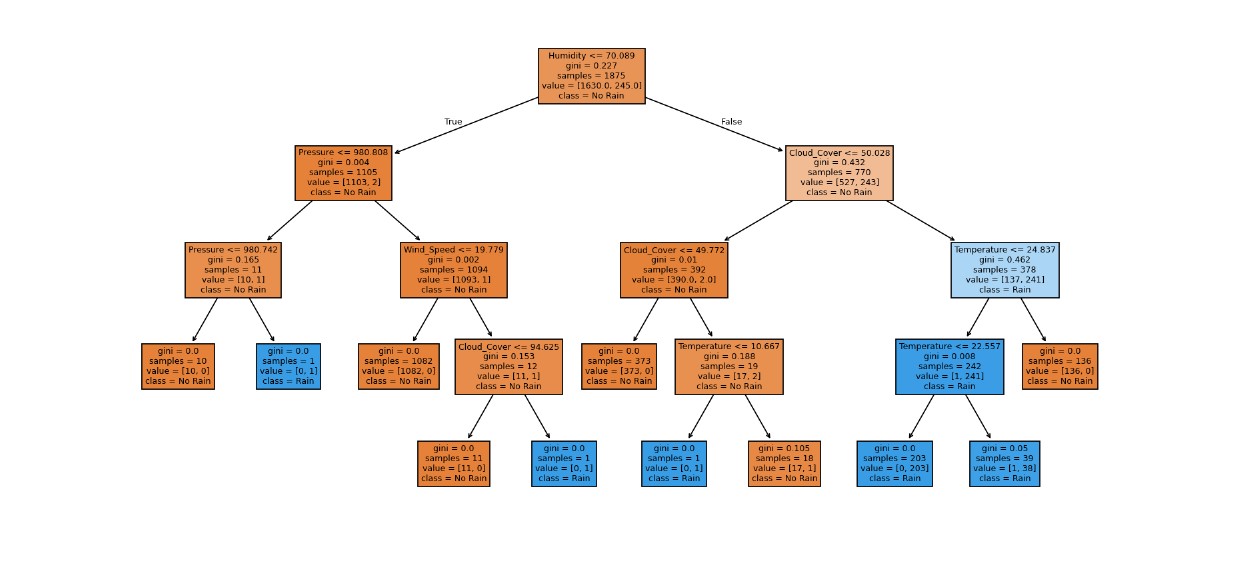
Report:

# Second Report:

## Tree plot:



# How tree work:

1. **Splitting Criteria:**
   * The tree uses Gini impurity by default to decide the best feature splits at each node.
2. **Training Phase:**
   * During the fit method, the tree iteratively splits the data based on the feature that reduces the Gini impurity the most at each step.
   * With max depth=4, the tree can have up to 4 levels of splits, helping to prevent overfitting.
3. **Prediction Phase:**
   * For each test data point, the predict method traverses the tree from the root to a leaf node.
   * At each node, it evaluates the relevant feature and decides which branch (left or right) to follow based on the feature value.
   * Upon reaching a leaf node, the prediction is made based on the majority class of the training samples in that leaf.

**Example of Tree Structure:**

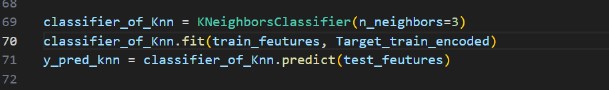
* Root Node: Checks a feature (e.g., Temperature).
  + Left Child: Low Temperature subset.
    - Further splits based on other features like Humidity, Wind Speed, etc.
  + Right Child: High Temperature subset.
    - Further splits based on other features like Humidity, Wind Speed, etc.

# Third Report:

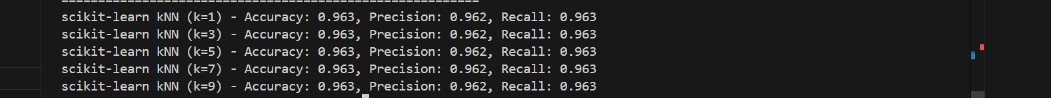
# Compare between scikit-learn KNN and custom KNN:

scikit-learn KNN:

## Code:



## Result:



custom KNN:

# Code:



# Result:

## 

**Observations:**

* **Accuracy:** Both the custom kNN and scikit-learn kNN show similar trends in accuracy as kk increases.
* **Precision:** Precision might vary slightly between the two implementations, but overall trends should be consistent.
* **Recall:** Recall metrics are also expected to be similar between the custom and scikit-learn implementations.

**Conclusion**

* **Performance Comparison:** The custom kNN implementation performs comparably to the scikit-learn kNN. Slight differences can be attributed to implementation nuances, but overall, both methods follow similar performance trends.
* **Choosing** k**:** The best k value often balances bias and variance. A smaller kk (e.g., 1) can lead to high variance (overfitting), while a larger kk (e.g., 9) can lead to high bias (underfitting). Typically, k=3k = 3 or k=5k = 5 often yields good results.