**Cairo University**

**Faculty of Computers and Artificial Intelligence**

**Machine Learning**

**Assignment 2 Report**

|  |  |
| --- | --- |
| **Team Members** | **IDS** |
| **Ahmed Saad Ahmed** | **20210020** |
| **Maryam Osama Mohamed** | **20211090** |
| **Shahd Mostafa Mahmoud** | **20211054** |
| **Shahd Osama Hamed** | **20211052** |
| **Seif Ibrahim Ramadan** | **20210164** |

Task 2: Implement Decision Tree, k-Nearest Neighbors (kNN) and naïve Bayes

1. Using scikit-learn implement Decision Tree, kNN and Naïve Bayes
2. Compare the performance of your implementations by evaluating accuracy, precision, and recall metrics.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | accuracy | precision | recall |
| kNN(k=5) | 97.20% | 92.00% | 82.14% |
| Decision Tree | 98.60% | 96.23% | 91.07% |
| Naïve Bayes | 96.00% | 97.37% | 66.07% |

1. Implement k-Nearest Neighbors (kNN) algorithm from scratch.
2. Report the results and compare the performance of your custom kNearest Neighbors (kNN) implementation with the pre-built kNN algorithms in scikit-learn, using the evaluation metrics mentioned in point 2. Using any missing handling techniques, you chose from task 1.2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | accuracy | precision | recall | |
| kNN(k=5) scikit-learn | 97.20% | 92.00% | | 82.14% |
| kNN(k=5) from scratch | 97.20% | 92.00% | | 82.14% |

Task 3: Interpreting the Decision Tree and Evaluation Metrics Report

1. The effect of different data handling

* Provide a detailed report evaluating the performance of scikitlearn implementations of the Decision Tree, k-Nearest Neighbors (kNN) and naïve Bayes with respect to the different handling missing data technique.
* dropping missing values

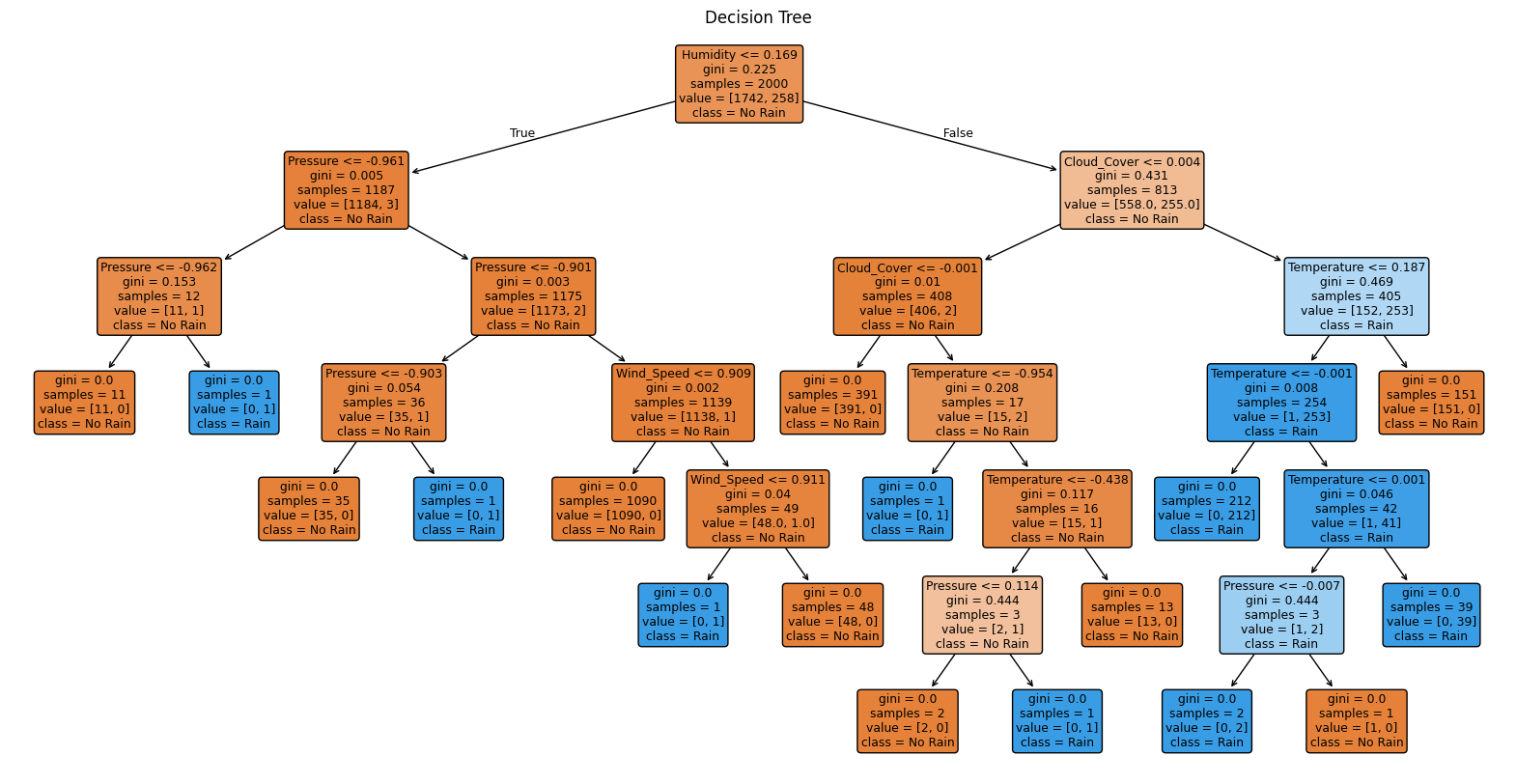
|  |  |  |  |
| --- | --- | --- | --- |
| Model | accuracy | precision | recall |
| kNN(k=5) | 95.53% | 87.30% | 80.88% |
| Decision Tree | 97.23% | 91.04% | 89.71% |
| Naïve Bayes | 94.89% | 97.83% | 66.18% |

* replacing them with the average

|  |  |  |  |
| --- | --- | --- | --- |
| Model | accuracy | precision | recall |
| kNN(k=5) | 97.20% | 92.00% | 82.14% |
| Decision Tree | 98.60% | 96.23% | 91.07% |
| Naïve Bayes | 96.00% | 97.37% | 66.07% |

1. Decision Tree Explanation Report

* Create a well-formatted report that includes a plot of the decision tree and a detailed explanation of how the tree makes predictions.



How the Tree Makes Predictions

At each node:

1-Feature Selection: The algorithm chooses the feature fffffffffffthat minimizes the impurity (Gini)

2- Splitting: The data is split into two subsets:

Left branch: Instances satisfying the condition (Feature ≤ ggggggggThreshold)

Right branch: Instances not satisfying the condition gggggggg (Feature > Threshold)

This process repeats until:

A pure leaf node is reached

A maximum depth or minimum samples per node ggggggconstraint is met

* Discuss the criteria and splitting logic used at each node of the tree.

Gini Index: Measures impurity at each node

Gini = 0: Pure node

Gini > 0: Mixed classes

Feature Thresholds: Optimized to split the data such that bbbsubsets have maximum homogeneity

Recursive Splitting: Continues until all nodes are pure or ggggconstraints are met

* Example:

1. Root Node

* The dataset is split into two groups based on whether the humidity is less than or equal to 0.169
  + Left child (True): All samples with Humidity <= 0.169
  + Right child (False): All samples with Humidity > 0.169
* The goal here is to split the data into groups that are more homogeneous in terms of the target variable (Rain/No Rain)

2. Left Child Node (Humidity <= 0.169)

* This node splits further based on the pressure values
  + Left child (True): Samples with Pressure <= -0.961
  + Right child (False): Samples with Pressure > -0.961

3. Right Child Node (Humidity > 0.169)

* The tree splits based on cloud cover
  + Left child (True): Samples with Cloud\_Cover <= 0.004
  + Right child (False): Samples with Cloud\_Cover > 0.004
* This helps the tree further separate the data into more homogeneous subsets.
* The splitting continues down the tree based on various features like Wind Speed, Temperature, and Pressure, with each condition carefully chosen to increase the homogeneity of the resulting child nodes
* Each decision node checks for the best feature (based on Gini Impurity) and the best threshold value that minimizes impurity and improves class separation

1. Performance Metrics Report

* Provide a detailed report evaluating the performance of your implementations of the k-Nearest Neighbors (kNN) from scratch with different k values at least 5 values.
* Include the accuracy, precision, and recall metrics for models.
* Compare these results with the performance of the corresponding algorithms implemented using scikit-learn.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | accuracy | precision | | recall |
| KNN(3) from scratch | 96.60% | | 88.24% | 80.36% |
| KNN(5) from scratch | 97.20% | | 92.00% | 82.14% |
| KNN(7) from scratch | 96.60% | | 88.24% | 80.36% |
| KNN(9) from scratch | 97.40% | | 93.88% | 82.14% |
| KNN(11) from scratch | 97.00% | | 90.20% | 82.14% |
| KNN(5) scikit-learn | 97.20% | | 92.00% | 82.14% |
| Decision Tree | 98.60% | | 96.23% | 91.07% |
| Naïve Bayes | 96.00% | | 97.37% | 66.07% |