Bronco ID: 014429779 CS 4210.01 – Assignment #2

Last Name: Koepke

First Name: Christopher

Timeline

Description automatically generated with low confidence

1. Considering that ID3 built the decision tree below after analyzing a given training set, answer the following questions:
2. What is the accuracy of this model if applied to the test set below? You must **identify** **each** True Positive, True Negative, False Positive, and False Negative for full credit.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Age | Spectacle | Astigmatism | Tear | Lenses (ground truth) |  |
| Young | Hypermetrope | Yes | Normal | Yes | FALSE NEGATIVE |
| Young | Hypermetrope | No | Normal | Yes | TRUE POSITIVE |
| Young | Myope | No | Reduced | No | TRUE NEGATIVE |
| Presbyopic | Hypermetrope | No | Reduced | No | TRUE NEGATIVE |
| Presbyopic | Myope | No | Normal | No | FALSE POSITIVE |
| Presbyopic | Myope | Yes | Reduced | No | TRUE NEGATIVE |
| Prepresbyopic | Myope | Yes | Normal | Yes | TRUE POSITIVE |
| Prepresbyopic | Myope | No | Reduced | No | TRUE NEGATIVE |

1. What is the precision, recall, and F1-measure of this model when applied to the same test set?

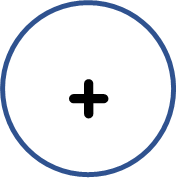
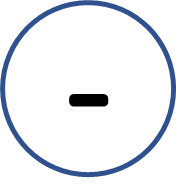
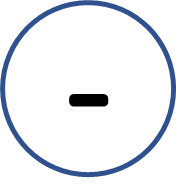
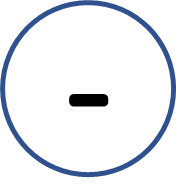
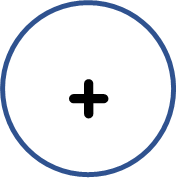
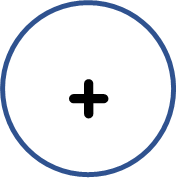
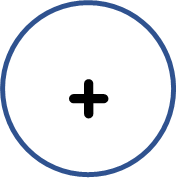
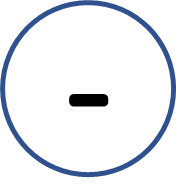
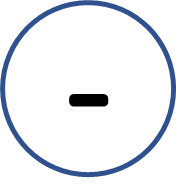
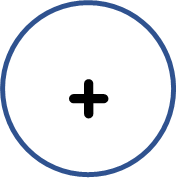
* PRECISION
* RECALL
* F1-measure

1. Complete the Python program (decision\_tree\_2.py) that will read the files contact\_lens\_training\_1.csv, contact\_lens\_training\_2.csv, and contact\_lens\_training\_3.csv. Each of those training sets has a different number of instances. You will observe that now the trees are being created setting the parameter *max\_depth = 3*, which it is used to define the maximum depth of the tree (pre-pruning strategy) in *sklearn*. Your goal is to train, test, and output the performance of the **3 models created by using each training set** on the test set provided (contact\_lens\_test.csv). **You must repeat this process 10 times** (train and test by using a different training set), choosing the lowest accuracy as the **final classification performance of each model**.

* Final accuracy of contact\_lens\_training\_1.csv is: 0.5
* Final accuracy of contact\_lens\_training\_2.csv is: 0.75
* Final accuracy of contact\_lens\_training\_3.csv is: 0.875

https://github.com/chris-k87/CS\_4210.01/tree/main/Assignment\_2/Decision\_Tree

1. Consider the dataset below to answer the following questions:



y

* 1. What is the leave-one-out cross-validation error rate (LOO-CV) for **1NN**? Use Euclidean distance as your distance measure and the error rate calculated as:
* Number of wrong predictions = 4

x

* Total number of predictions = 10
  1. What is the leave-one-out cross-validation error rate (LOO-CV) for **3NN**?
* Number of wrong predictions = 2
* Total number of predictions = 10
  1. What is the leave-one-out cross-validation error rate (LOO-CV) for **9NN**?
* Number of wrong predictions = 10
* Diagram

  Description automatically generated with low confidenceTotal number of predictions = 10
  1. Draw the **decision boundary** learned by the 1NN algorithm.
  2. Complete the Python program (knn.py) that will read the file binary\_points.csv and output the LOO-CV error rate for 1NN (**same answer of part a**).

https://github.com/chris-k87/CS\_4210.01/tree/main/Assignment\_2/KNN

1. Find the class of instance #10 below following the 3NN strategy. Use Euclidean distance as your distance measure. You must **show all your calculations** for full credit.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Red | Green | Blue | Class |
| #1 | 220 | 20 | 60 | 1 |
| #2 | 255 | 99 | 21 | 1 |
| #3 | 250 | 128 | 14 | 1 |
| #4 | 144 | 238 | 144 | 2 |
| #5 | 107 | 142 | 35 | 2 |
| #6 | 46 | 139 | 87 | 2 |
| #7 | 64 | 224 | 208 | 3 |
| #8 | 176 | 224 | 23 | 3 |
| #9 | 100 | 149 | 237 | 3 |
| #10 | 154 | 205 | 50 | ? |

* From the distance calculations shown, data points #4, #5, and #8 have the shortest distance to data point #10. The class labels for those three data points are 2, 2, and 3, respectively. From those three class labels, data point # 10 will be assigned the class label of 2.

5. Use the dataset below to answer the next questions:

Table

Description automatically generated

1. Classify the instance ‹D15, Sunny, Mild, Normal, Weak› following the Naïve Bayes strategy. **Show all your calculations** until the final normalized probability values.

* After Normalization
* The most probable classification for instance D15 when using the Naïve Bayes strategy would be Yes, based on the highest probability value after normalization that corresponds to the class label of Yes.

1. Complete the Python program (naïve\_bayes.py) that will read the file weather\_training.csv (training set) and output the classification of each test instance from the file weather\_test (test set) **if the classification confidence is >= 0.75**.

* Program Output:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Day | Outlook | Temperature | Humidity | Wind | PlayTennis | Confidence |
| D15 | Sunny | Hot | Normal | Weak | YES | 0.8380769978882107 |
| D18 | Overcast | Hot | High | Strong | NO | 0.7916957810518334 |
| D21 | Rain | Mild | Normal | Strong | YES | 0.8690250284177714 |
| D22 | Rain | Hot | Normal | Strong | YES | 0.7863528796684012 |

https://github.com/chris-k87/CS\_4210.01/tree/main/Assignment\_2/Naive\_Bayes