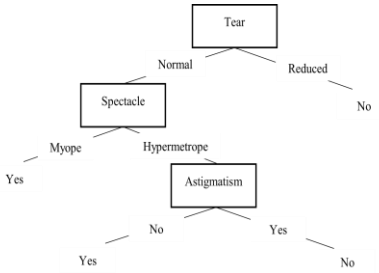
Bronco ID: 014429779
Last Name: Koepke
First Name: Christopher

CS 4210.01 – Assignment #2

1. Considering that ID3 built the decision tree below after analyzing a given training set, answer the following questions:



   a) 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

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

- PRECISION

$$p = \frac{TP}{TP + FP}$$
$$P = \frac{2}{2+1} = \frac{2}{3} \approx 0.6666$$

- RECALL

$$r = \frac{TP}{TP + FN}$$
$$r = \frac{2}{2+1} = \frac{2}{3} \approx 0.66667$$

- F1-measure

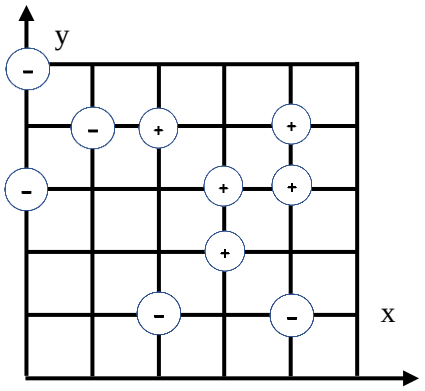
$$F_1 = \frac{2}{\frac{1}{r} + \frac{1}{p}} = \frac{2rp}{r+p}$$
$$F_1 = \frac{2(\frac{2}{3})(\frac{2}{3})}{\frac{2}{3} + \frac{2}{3}} = \frac{2}{3} \approx 0.6666$$

2. 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

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



   a. 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:

$$error\ rate = \frac{number\ of\ wrong\ predictions}{total\ number\ of\ predictions}$$

   - Number of wrong predictions = 4
   - Total number of predictions = 10

$$\boxed{error\ rate = \frac{4}{10} = 0.4}$$

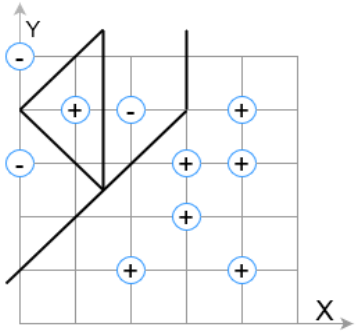
   b. What is the leave-one-out cross-validation error rate (LOO-CV) for **3NN**?
   - Number of wrong predictions = 2
   - Total number of predictions = 10

$$\boxed{error\ rate = \frac{2}{10} = 0.2}$$

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

$$\boxed{error\ rate = \frac{10}{10} = 1}$$

   d. Draw the **decision boundary** learned by the 1NN algorithm.



   e. 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**).

$$\boxed{error\ rate = 0.4}$$

https://github.com/chris-k87/CS_4210.01/tree/main/Assignment_2/KNN

4. 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	?

- $d(\#1, \#10) = \sqrt{(220-154)^2 + (20-205)^2 + (60-50)^2} \approx 196.6748586$
- $d(\#2, \#10) = \sqrt{(255-154)^2 + (99-205)^2 + (21-50)^2} \approx 149.2581656$
- $d(\#3, \#10) = \sqrt{(250-154)^2 + (128-205)^2 + (14-50)^2} \approx 128.2224629$
- $d(\#4, \#10) = \sqrt{(144-154)^2 + (238-205)^2 + (144-50)^2} \approx 100.124922$
- $d(\#5, \#10) = \sqrt{(107-154)^2 + (142-205)^2 + (35-50)^2} \approx 80.0187478$
- $d(\#6, \#10) = \sqrt{(46-154)^2 + (139-205)^2 + (87-50)^2} \approx 131.8673576$
- $d(\#7, \#10) = \sqrt{(64-154)^2 + (224-205)^2 + (208-50)^2} \approx 182.825053$
- $d(\#8, \#10) = \sqrt{(176-154)^2 + (224-205)^2 + (23-50)^2} \approx 39.67366885$
- $d(\#9, \#10) = \sqrt{(100-154)^2 + (149-205)^2 + (237-50)^2} \approx 202.5364165$

- 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:

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

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

- $P(class = No | A1 = Sunny, A2 = Mild, A3 = Normal, A4 = Weak)$

$$\left(\prod_i P(Ai = Xi |\ Class = No)\right) * P(Class = No) =$$

$$\left(P(A1 = Sunny|Class = No) * (A2 = Mild|Class = No) * (A3 = Normal|Class = No) * (A4 = Weak|Class = No)\right) * P(Classs = No) =$$

$$\left(\left(\frac{3}{5}\right) * \left(\frac{2}{5}\right) * \left(\frac{1}{5}\right) * \left(\frac{2}{5}\right)\right) * \left(\frac{5}{14}\right) = \frac{4}{7} \approx 0.5714285714$$

- $P(class = Yes | A1 = Sunny, A2 = Mild, A3 = Normal, A4 = Weak)$

$$\left(\prod_i P(Ai = Xi |\ Class = Yes)\right) * P(Class = Yes) =$$

$$\left(P(A1 = Sunny|Class = Yes) * (A2 = Mild|Class = Yes) * (A3 = Normal|Class = Yes) * (A4 = Weak|Class = Yes)\right) * P(Classs = Yes) =$$

$$\left(\left(\frac{2}{9}\right) * \left(\frac{4}{9}\right) * \left(\frac{6}{9}\right) * \left(\frac{6}{9}\right)\right) * \left(\frac{9}{14}\right) = \frac{9}{7} \approx 1.285714286$$

- After Normalization

$$P(class = No | A1 = Sunny, A2 = Mild, A3 = Normal, A4 = Weak) = \frac{0.5714285714}{(0.5714285714 + 1.285714286)} \approx 0.3076923076$$

$$P(class = Yes | A1 = Sunny, A2 = Mild, A3 = Normal, A4 = Weak) = \frac{1.285714286}{(0.5714285714 + 1.285714286)} \approx 0.6923076924$$

- 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.

b) 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