CS 4210 – Assignment #2

Maximum Points: 100 pts.

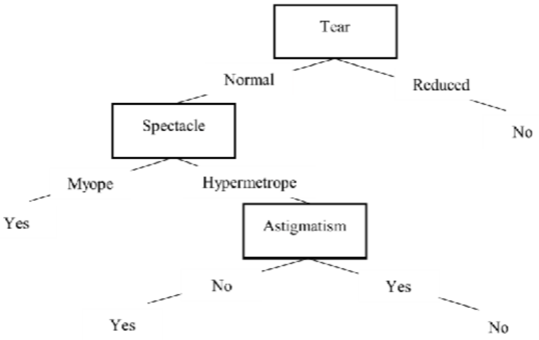
Bronco ID: 016414437

Last Name: Yen

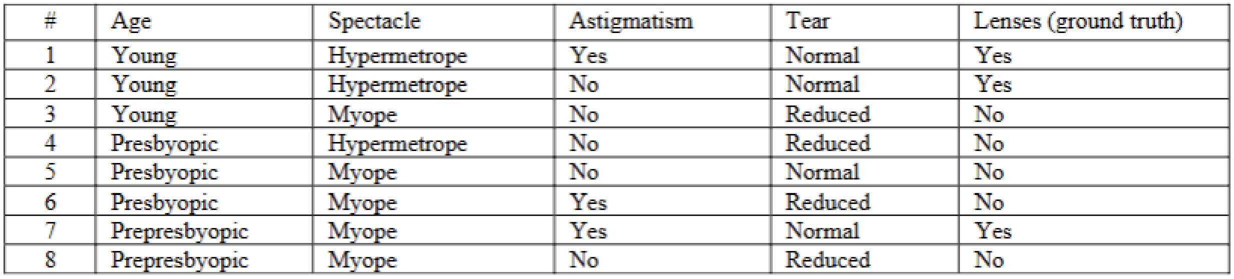
First Name: Kaitlin

Note 1: Your submission header must have the format as shown in the above-enclosed rounded rectangle.
Note 2: Homework is to be done individually. You may discuss the homework problems with your fellow students, but you are NOT allowed to copy – either in part or in whole – anyone else's answers.
Note 3: Your deliverable should be a .pdf file submitted through Gradescope until the deadline. Do not forget to assign a page to each of your answers when making a submission. In addition, source code (.py files) should be added to an online repository (e.g., github) to be downloaded and executed later.
Note 4: All submitted materials must be legible. Figures/diagrams must have good quality.
Note 5: Please use and check the Canvas discussion for further instructions, questions, answers, and hints. The bold words/sentences provide information for a complete or accurate answer.

1. [16 points] Considering that ID3 built the decision tree below after analyzing a given training set, answer the following questions:



a. [12 points] 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. For instance: TP = 1,5 | TN = 2,3 ...



#	Age	Spectacle	Astigmatism	Tear	Lenses (ground truth)
1	Young	Hypermetrope	Yes	Normal	Yes
2	Young	Hypermetrope	No	Normal	Yes
3	Young	Myope	No	Reduced	No
4	Presbyopic	Hypermetrope	No	Reduced	No
5	Presbyopic	Myope	No	Normal	No
6	Presbyopic	Myope	Yes	Reduced	No
7	Prepresbyopic	Myope	Yes	Normal	Yes
8	Prepresbyopic	Myope	No	Reduced	No

TP = 2,7 | TN = 3,4,6,8 | FP = 1 | FN = 5

Accuracy: $\frac{TP+TN}{TP+FP+TN+FN} = \frac{2+4}{2+1+4+1} = \frac{6}{8} = \frac{3}{4}$

b. [4 points] What is the precision, recall, and F1-measure of this model when applied to the same test set?

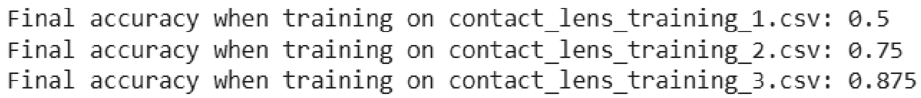
Precision: $P = \frac{T_P}{T_P+F_P} = \frac{2}{2+1} = \frac{2}{3}$

Recall: $R = \frac{T_P}{T_P+F_N} = \frac{2}{2+1} = \frac{2}{3}$

F1-measure: $F1 = \frac{2\times R\times P}{R+P} = \frac{2\times\frac{2}{3}\times\frac{2}{3}}{\frac{2}{3}+\frac{2}{3}} = \frac{\frac{8}{9}}{\frac{4}{3}} = \frac{2}{3}$

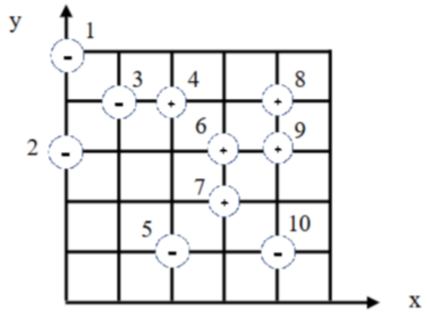
2. [15 points] Complete the Python program (decision_tree_2.py) that will read the files contact_lens_training_1.csv, contact_lens_training_2.csv, and contact_lens_training_3.csv. Each training set has a different number of instances (10, 100, 1000 samples). You will observe that the trees are being created by setting the parameter max_depth = 5, which 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 using a different training set), choosing the average accuracy as the **final classification performance of each model**.

Github: https://github.com/kaiyen-pepper/4210Homework2



```
Final accuracy when training on contact_lens_training_1.csv: 0.5
Final accuracy when training on contact_lens_training_2.csv: 0.75
Final accuracy when training on contact_lens_training_3.csv: 0.875
```

3. [32 points] Consider the dataset below to answer the following questions:



a. [4 points] What is the leave-one-out cross-validation error rate (LOO-CV error rate) for **1NN**? Use Euclidean distance as your distance measure, and the error rate calculated as:

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

Requirement. Identify the data point(s) misclassified for full marks.

Node	INN	Class Prediction	Class Actual	Prediction
1	3	-	-	Correct
2	3	-	-	Correct
3	4	+	-	Incorrect
4	3	-	+	Incorrect
5	7	+	-	Incorrect

6	7	+	+	Correct
7	6	+	+	Correct
8	9	+	+	Correct
9	6	+	+	Correct
10	7	+	-	Incorrect

$$error\ rate = \frac{number\ of\ wrong\ predictions}{total\ number\ of\ predictions} = \frac{4}{10} = .4$$

b. [4 points] What is the leave-one-out cross-validation error rate (LOO-CV) for **3NN**? **Requirement**. Identify the data point(s) misclassified for full marks.

Node	3NN	Class Prediction	Class Actual	Prediction
1	3,2,4	-,-,+ = -	-	Correct
2	3,1,4	-.-.+ = -	-	Correct
3	4,1,2	+,-,- = -	-	Correct
4	3,6,8	-,+,+ = +	+	Correct
5	7,10,6	+,-,+ = +	-	Incorrect
6	7,9,4	+,+,+ = +	+	Correct
7	6,5,9	+,-,+ = +	+	Correct
8	9,6,4	+,+,+ = +	+	Correct
9	6,8,7	+,+,+ = +	+	Correct
10	7,5,9	+,-,+ = +	-	Incorrect

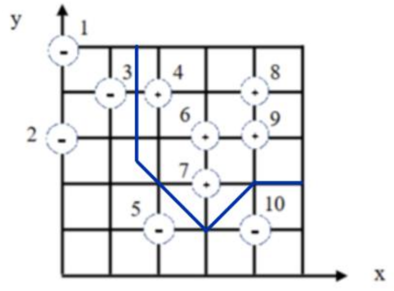
$$error\ rate = \frac{number\ of\ wrong\ predictions}{total\ number\ of\ predictions} = \frac{2}{10} = .2$$

c. [4 points] What is the leave-one-out cross-validation error rate (LOO-CV) for **9NN**? **Requirement**. Identify the data point(s) misclassified for full marks.

Node	9NN	Class Prediction	Class Actual	Prediction
1	2,3,4,5,6,7,8,9,10	4 -, 5 + = +	-	Incorrect
2	1,3,4,5,6,7,8,9,10	4 -, 5 + = +	-	Incorrect
3	1,2,4,5,6,7,8,9,10	4 -, 5 + = +	-	Incorrect
4	1,2,3,5,6,7,8,9,10	5 -, 4 + = -	+	Incorrect
5	1,2,3,4,6,7,8,9,10	4 -, 5 + = +	-	Incorrect
6	1,2,3,4,5,7,8,9,10	5 -, 4 + = -	+	Incorrect
7	1,2,3,4,5,6,8,9,10	5 -, 4 + = -	+	Incorrect
8	1,2,3,4,5,6,7,9,10	5 -, 4 + = -	+	Incorrect
9	1,2,3,4,5,6,7,8,10	5 -, 4 + = -	+	Incorrect
10	1,2,3,4,5,6,7,8,9	4 -, 5 + = +	-	Incorrect

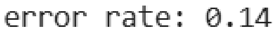
$$error\ rate = \frac{number\ of\ wrong\ predictions}{total\ number\ of\ predictions} = \frac{10}{10} = 1.0$$

d. [5 points] Draw the **decision boundary** learned by the 1NN algorithm.



e. [15 points] Complete the Python program (knn.py) to read the file email_classification.csv and compute the LOO-CV error rate for a 1NN classifier on the spam/ham classification task. The dataset consists of email samples, where each sample includes the counts of 20 specific words (e.g., “agenda” or “prize”) representing their frequency of occurrence.

Github: https://github.com/kaiyen-pepper/4210Homework2



```
error rate: 0.14
```

4. [12 points] 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	?

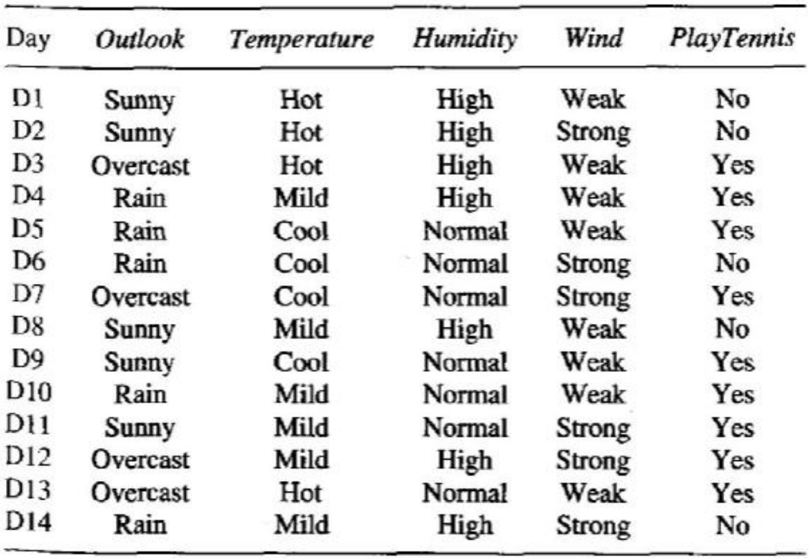
Distance from 10:

ID	Red	Green	Blue	Total Distance	Class
1	66	185	10	$\sqrt{(66)^2+(185)^2+(10)^2}=196.674$	1
2	101	106	29	$\sqrt{(101)^2+(106)^2+(29)^2}=149.258$	1
3	96	77	36	$\sqrt{(96)^2+(77)^2+(36)^2}=128.222$	1
4	10	33	94	$\sqrt{(10)^2+(33)^2+(94)^2}=100.124$	2
5	47	63	15	$\sqrt{(47)^2+(63)^2+(15)^2}=80.018$	2
6	108	66	37	$\sqrt{(108)^2+(66)^2+(37)^2}=131.867$	2
7	90	19	158	$\sqrt{(90)^2+(19)^2+(158)^2}=182.825$	3
8	22	19	27	$\sqrt{(22)^2+(19)^2+(27)^2}=39.673$	3
9	54	56	187	$\sqrt{(54)^2+(56)^2+(187)^2}=202.536$	3

Distance Formula = $\sqrt{(x_0-x_1)^2+(y_0-y_1)^2+(z_0-z_1)^2}$

Class Prediction: 2, 2, 3 = 2

5. [25 points] 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. [10 points] Classify the instance ‹D15, Sunny, Mild, Normal, Weak› following the Naïve Bayes strategy. **Show all your calculations** until the final **normalized probability values**. Hint. No smoothing needed.

Using the Bayes theorem and assuming conditional independence

P(Class = No | Outlook, Temperature, Humidity, Wind)

P(Class = Yes | Outlook, Temperature, Humidity, Wind)

Day	**Outlook**	**Temperature**	**Humidity**	**Wind**	**Play Tennis**
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**	**Cold**	**Normal**	**Weak**	**Yes**
D6	**Rain**	**Cold**	**Normal**	**Strong**	**No**
D7	**Overcast**	**Cold**	**Normal**	**Strong**	**Yes**
D8	**Sunny**	**Mild**	**High**	**Weak**	**No**
D9	**Sunny**	**Cold**	**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**

P(Class = No | Outlook, Temperature, Humidity, Wind)

= ($\prod_i$ P(Ai = xi | Class = No)) * P(Class = No)

= (P(Outlook = Sunny | Class = No) * P(Temperature = Mild | Class = No) * P(Humidity = Normal | Class = No) * P(Wind = Weak | Class = No)) * P(Class = No)

= (3/5) * (2/5) * (1/5) * (2/5) * (5/14) = 0.0068

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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	Cold	Normal	Weak	Yes
D6	Rain	Cold	Normal	Strong	No
D7	Overcast	Cold	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	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

P(Class = Yes | Outlook, Temperature, Humidity, Wind)

= ($\prod_i$ P(Ai = xi | Class = Yes)) * P(Class = Yes)

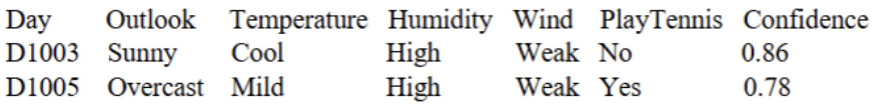
= (P(Outlook = Sunny | Class = Yes) * P(Temperature = Mild | Class = Yes) * P(Humidity = Normal | Class = Yes) * P(Wind = Weak | Class = Yes)) * P(Class = Yes)

= (2/9) * (4/9) * (6/9) * (6/9) * (9/14) = 0.0282

Normalization: $\frac{0.0282}{0.0282+0.006} = 0.804$ **Yes** $\frac{0.0068}{0.0282+0.0068} = 0.195$ **No**

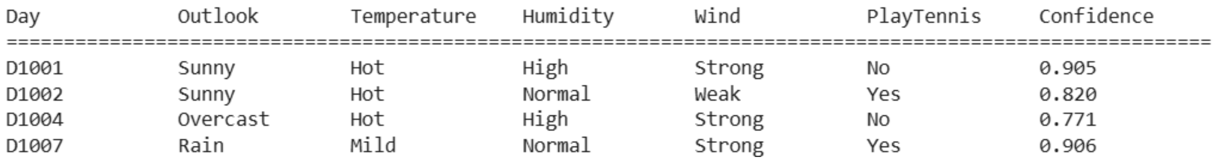
Classification Prediction: Yes (Play Tennis)

b. [15 points] Complete the Python program (naïve_bayes.py) that will read the file weather_training.csv (training set) and output the classification of each of the 10 instances from the file weather_test (test set) **if the classification confidence is >= 0.75**. Sample of output:



```
Day    Outlook   Temperature  Humidity  Wind  PlayTennis  Confidence
D1003  Sunny     Cool         High      Weak  No          0.86
D1005  Overcast  Mild         High      Weak  Yes         0.78
```

Github: https://github.com/kaiyen-pepper/4210Homework2



```
Day             Outlook         Temperature     Humidity        Wind            PlayTennis      Confidence
==========================================================================================================
D1001           Sunny           Hot             High            Strong          No              0.905
D1002           Sunny           Hot             Normal          Weak            Yes             0.820
D1004           Overcast        Hot             High            Strong          No              0.771
D1007           Rain            Mild            Normal          Strong          Yes             0.906
```

Important Note: Answers to all questions should be written clearly, concisely, and unmistakably delineated. You may resubmit multiple times until the deadline (the last submission will be considered).

NO LATE ASSIGNMENTS WILL BE ACCEPTED. ALWAYS SUBMIT WHATEVER YOU HAVE COMPLETED FOR PARTIAL CREDIT BEFORE THE DEADLINE!