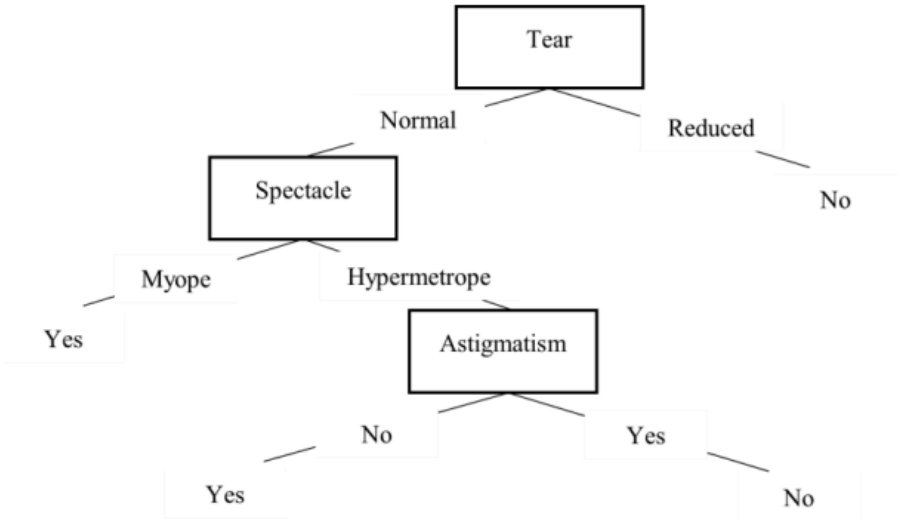
Brandon Tang
CS4210.01
Assignment 2

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. 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 = 5
FN = 1

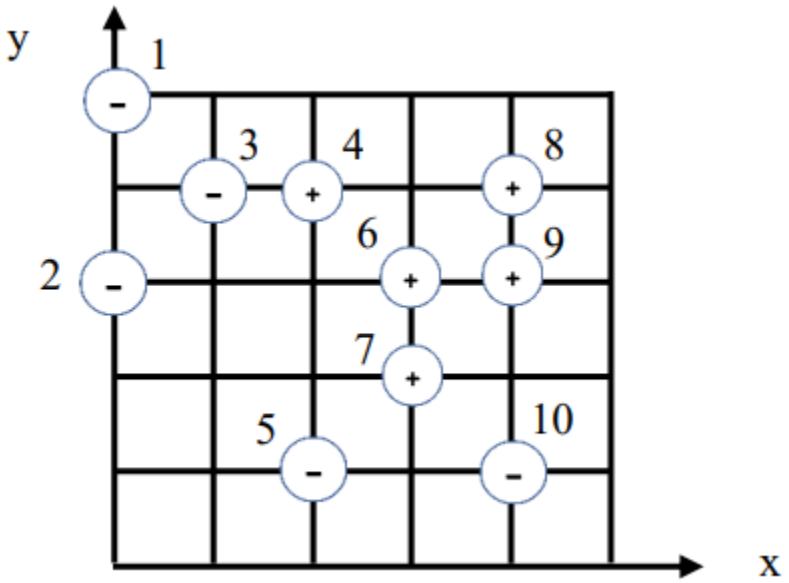
Accuracy = (TP + TN) / (TP + TN + FP + FN) = (2 + 4) / (2 + 4 + 1 + 1) = 0.75

b) What is the precision, recall, and F1-measure of this model when applied to the same test set?
Precision = TP / (TP + FP) = 2 / (2 + 1) = 0.67
Recall = TP / (TP + FN) = 2 / (2 + 1) = 0.67
F1 = 2 * P * R / (P + R) = (2 * 0.67 * 0.67) / (0.67 + 0.67) = 0.67

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 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.

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) Consider the dataset below to answer the following questions:



a) 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:

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

Distance = $\sqrt{[(x2 - x1)^2 + (y2 - y1)^2]}$

Test	1-NN	Predicted	True Label	Correct?
1	2	n	n	y
2	1	n	n	y
3	4	p	n	n
4	3	n	p	n
5	7	p	n	n

6	7	p	p	y
7	6	p	p	y
8	9	p	p	y
9	8	p	p	y
10	5	n	n	y

Error rate = 3/10 = 0.3
Misclassified = 3, 4, 5

b) What is the leave-one-out cross-validation error rate (LOO-CV) for 3NN?

Test	3-NN	Predicted	True Label	Correct?
1	2,3,4	n	n	y
2	1,3,4	n	n	y
3	1,2,4	n	n	y
4	3,6,9	p	p	y
5	2,7,10	n	n	y
6	4,7,9	p	p	y
7	5,6,10	p	p	y
8	4,6,9	p	p	y
9	4,6,8	p	p	y
10	5,7,9	p	n	n

Error rate = 1/10 = 0.1
Misclassified = 10

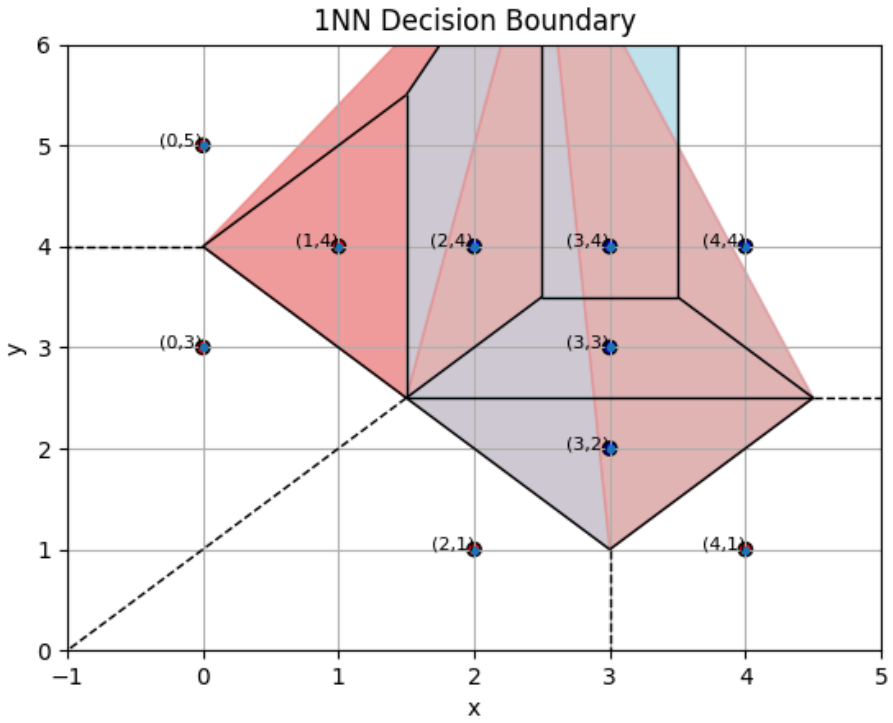
c) What is the leave-one-out cross-validation error rate (LOO-CV) for 9NN?

Test	3-NN	Predicted	True Label	Correct?
1	2,3,4,5,6,7,8,9,10	p	n	n
2	1,3,4,5,6,7,8,9,10	p	n	n
3	1,2,4,5,6,7,8,9,10	p	n	n
4	1,2,3,5,6,7,8,9,10	p	p	y
5	1,2,3,4,6,7,8,9,10	p	n	n
6	1,2,3,4,5,7,8,9,10	p	p	y

7	1,2,3,4,5,6,8,9,10	p	p	y
8	1,2,3,4,5,6,7,9,10	p	p	y
9	1,2,3,4,5,6,7,8,10	p	p	y
10	1,2,3,4,5,6,7,8,9	p	n	n

Error rate = 5/10 = 0.5
Misclassified = 1, 2, 3, 5, 10

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



e) 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.

Final Conclusion: LOO-CV Error Rate for 1NN: 0.14

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	

Distance to 1: $\sqrt{(154-220)^2 + (205-20)^2 + (50-60)^2} = 196.67$
Distance to 2: $\sqrt{(154-255)^2 + (205-99)^2 + (50-21)^2} = 149.26$
Distance to 3: $\sqrt{(154-250)^2 + (205-128)^2 + (50-14)^2} = 128.22$
Distance to 4: $\sqrt{(154-144)^2 + (205-238)^2 + (50-144)^2} = 100.12$
Distance to 5: $\sqrt{(154-107)^2 + (205-142)^2 + (50-35)^2} = 80.02$
Distance to 6: $\sqrt{(154-46)^2 + (205-139)^2 + (50-87)^2} = 131.87$
Distance to 7: $\sqrt{(154-64)^2 + (205-224)^2 + (50-208)^2} = 182.83$
Distance to 8: $\sqrt{(154-176)^2 + (205-224)^2 + (50-237)^2} = 39.68$
Distance to 9: $\sqrt{(154-100)^2 + (205-149)^2 + (50-50)^2} = 202.54$

3-NN:
Instance 8 (Distance = 39.68, Class = 3)
Instance 5 (Distance = 80.02, Class = 2)
Instance 4 (Distance = 100.12, Class = 2)

Majority = Class 2
Therefore, the predicted class of Instance #10 is **Class 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. Hint. No smoothing needed.

Total Instances = 14
PlayTennis Yes = 9
PlayTennis No = 5

P(Yes) = 9/14 = 0.6429
P(No) = 5/14 = 0.3571

<Sunny, Mild, Normal, Weak>

Likelihood of Yes:
P(Outlook = Sunny | Yes) = 2/9 = 0.2222
P(Temperature = Mild | Yes) = 4/9 = 0.4444
P(Humidity = Normal | Yes) = 6/9 = 0.6667
P(Wind = Weak | Yes) = 6/9 = 0.6667

Likelihood of No:
P(Outlook = Sunny | No) = 3/5 = 0.6
P(Temperature = Mild | No) = 2/5 = 0.4
P(Humidity = Normal | No) = 1/5 = 0.2
P(Wind = Weak | No) = 2/5 = 0.4

Posterior Probabilities:
P(Yes | Features) = 0.6429 * 0.2222 * 0.4444 * 0.6667 * 0.6667 = 0.0282
P(No | Features = 0.3571 * 0.6 * 0.4 * 0.2 * 0.4 = 0.0068

Normalized Probabilities
P(Yes | Features) = 0.0282 / (0.0282 + 0.0068) = 0.8057
P(No | Features) = 0.0068 / (0.0068 + 0.0282) = 0.1943

Final Decision:
P(Yes) = 0.8057
P(No) = 0.1943

P(Yes) > P(No), therefore the predicted class is **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 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**

Day	Outlook	Temperature	Humidity	Wind	PlayTennis	Confidence
D1001	Sunny	Hot	High	Strong	No	0.90
D1002	Sunny	Hot	Normal	Weak	Yes	0.82
D1004	Overcast	Hot	High	Strong	No	0.77
D1007	Rain	Mild	Normal	Strong	Yes	0.91

Code Implementation (assignment2)
https://github.com/BrandonTang95/cs4210.git