Homework 1
Automated Learning and Data Analysis
Team 41
Rajshree Jain, rjain27
Srujana Rachakonda, srachak

Q1.

a) Information Gain Decision Tree:

Entropy calculation formula = $-\frac{p}{(p+n)}log_2(\frac{p}{p+n}) - \frac{n}{(p+n)}log_2(\frac{n}{p+n})$
We will be using this formula to calculate all entropy values

Total Entropy = -(9/16)log(9/16) -(7/16)log(7/16) = 0.988

H(V2 = Mild) = -(5/10)log(5/10) - (5/10)log(5/10) = 1
H(V2 = Strong) = -(2/6)log(2/6) - (4/6)log(4/6) = 0.92
H(V2) = 1 * (10/16) + 0.92 * (6/16) = 0.97
IG(V2) = H(S) - H(V2), where H(S) is total entropy
IG(V2) = 0.988 - 0.97 = 0.019

Similarly,

H(V3 = Raleigh) = 1
H(V3 = Durham) = 0.92
IG(V3) = 0.019

Since V4 is a continuous variable we calculate the information gain for each possible value it can take and split on the maximum IG() value. We split on each value similar to how we did above.

H(V4<=1) = 0
H(V4>1) = 0.971
IG(V4=1) = 0.077

Similarly Information Gains of all values of V4 are listed as follows:

V4	IG(V4)
1	0.077
2	0.1665
3	0.036
4	0.09
6	0.035
8	0.007
9	0.0405
11	0.0115

12	0.0018
13	0.007
17	0.0025
18	0.036
19	0.008
21	0.113
22	0.055
27	0

Since IG(V4 = 2) is the max of all the IG(V4 =) values we pick <=2 and >2 as the split for V4

H(V5 = Hot) = 0.98
H(V5 = Cool) = 0.92
IG(V5) = 0.0413

Now the IG values of all the attributes is as follows:

Attribute	IG
V1	0.05
V2	0.019
V3	0.019
V4	0.1665
V5	0.0413

Since V4 gives us the highest IG we split the dataset on that and we get a clear distinction on splitting as all the values for V4 <= 2 are resulting in false, we can classify these values as false and remove these values from the dataset for the further iterations.

Resultant Dataset:

V1	**V2**	**V3**	**V4**	**V5**	**Class**
Sunny	Mild	Raleigh	1	Cool	F
Sunny	Strong	Raleigh	9	Cool	F
Overcast	Strong	Durham	11	Cool	T
Sunny	Strong	Durham	3	Cool	T
Rain	Strong	Durham	12	Cool	T
Rain	Mild	Raleigh	2	Hot	F
Rain	Mild	Raleigh	19	Hot	T

Overcast	Mild	Durham	4	Cool	F
Overcast	Strong	Raleigh	18	Hot	F
Sunny	Mild	Raleigh	27	Cool	T
Sunny	Mild	Raleigh	8	Cool	T
Overcast	Strong	Durham	6	Cool	T
Sunny	Mild	Durham	21	Hot	F
Overcast	Mild	Raleigh	17	Hot	F
Rain	Mild	Raleigh	22	Hot	T
Sunny	Mild	Raleigh	13	Hot	T

In our second iteration we do not consider the highlighted instances.

Iteration 2:

Total Entropy of the resultant dataset = 0.9395

Information Gain of each of the resultant attribute is as follows:

Attribute	IG(Attribute)
V1	0.198
V2	0.1511
V3	0.0018
V5	0.1595

Since V1 offers the most information gain we split on V1 next.

In this case, we get a clear split for Rain but we do not get a clear split for Sunny and overcast

The values are as follows:

V1 =	Class values
Sunny	T=4, F=2
Overcast	T=2, F=3
Rain	T=3 , F=0

So we can eliminate all the rain instances as they correspond to true

The reduced dataset is as follows:

V1	V2	V3	V4	V5	Class
Sunny	Mild	Raleigh	1	Cool	F
Sunny	Strong	Raleigh	9	Cool	F
Overcast	Strong	Durham	11	Cool	T
Sunny	Strong	Durham	3	Cool	T
Rain	Strong	Durham	12	Cool	T
Rain	Mild	Raleigh	2	Hot	F
Rain	Mild	Raleigh	19	Hot	T
Overcast	Mild	Durham	4	Cool	F
Overcast	Strong	Raleigh	18	Hot	F
Sunny	Mild	Raleigh	27	Cool	T
Sunny	Mild	Raleigh	8	Cool	T
Overcast	Strong	Durham	6	Cool	T
Sunny	Mild	Durham	21	Hot	F
Overcast	Mild	Raleigh	17	Hot	F
Rain	Mild	Raleigh	22	Hot	T
Sunny	Mild	Raleigh	13	Hot	T

All the pink shaded instances will be considered for the v1= sunny reduced dataset and unshaded instances will be considered for v1 = overcast reduced dataset.

Third Iteration: (Sunny)

Total Entropy of the resultant dataset = 0.92

Information Gain of each of the resultant attribute is as follows:

Attribute	IG(Attribute)
V2	0.046
V3	0.046
V5	0.046

We split sunny on V2 since it's the leftmost attribute

Third iteration: (Overcast)

Total Entropy of the resultant dataset = 0.97

Information Gain of each of the resultant attribute is as follows:

Attribute	IG(Attribute)
V2	0.418
V3	0.418
V5	0.418

We split overcast on V2 since it's the leftmost value

Now, we get a clear split on V2 = mild for V1 = mild and V4 >2 as false and we can eliminate that from our resultant dataset.

4th Iteration:

Resultant Dataset:

V1	**V2**	**V3**	**V4**	**V5**	**Class**
Sunny	Mild	Raleigh	1	Cool	F
Sunny	Strong	Raleigh	9	Cool	F
Overcast	Strong	Durham	11	Cool	T
Sunny	Strong	Durham	3	Cool	T
Rain	Strong	Durham	12	Cool	T
Rain	Mild	Raleigh	2	Hot	F
Rain	Mild	Raleigh	19	Hot	T
Overcast	Mild	Durham	4	Cool	F
Overcast	Strong	Raleigh	18	Hot	F
Sunny	Mild	Raleigh	27	Cool	T
Sunny	Mild	Raleigh	8	Cool	T
Overcast	Strong	Durham	6	Cool	T
Sunny	Mild	Durham	21	Hot	F
Overcast	Mild	Raleigh	17	Hot	F
Rain	Mild	Raleigh	22	Hot	T
Sunny	Mild	Raleigh	13	Hot	T

For V1 = overcast, V2 = Strong, V4>2

We have the following entropies:

Total entropy = 0.92

Attribute	IG

V3	0.92
V5	0.92

We split on V3 since it's the leftmost attribute.
We get a clear split as for V1 = overcast, V2 = Strong, V4>2 and V3 = Raleigh → false
And for V1 = overcast, V2 = Strong, V4>2 and V3 = Durham → True

For V1 = sunny, V2 = Strong, V4>2

We have the following entropies:

Total entropy = 1

Attribute	IG
V3	1
V5	0

So we split on V3

For V1 = sunny, V2 = Mild, V4>2

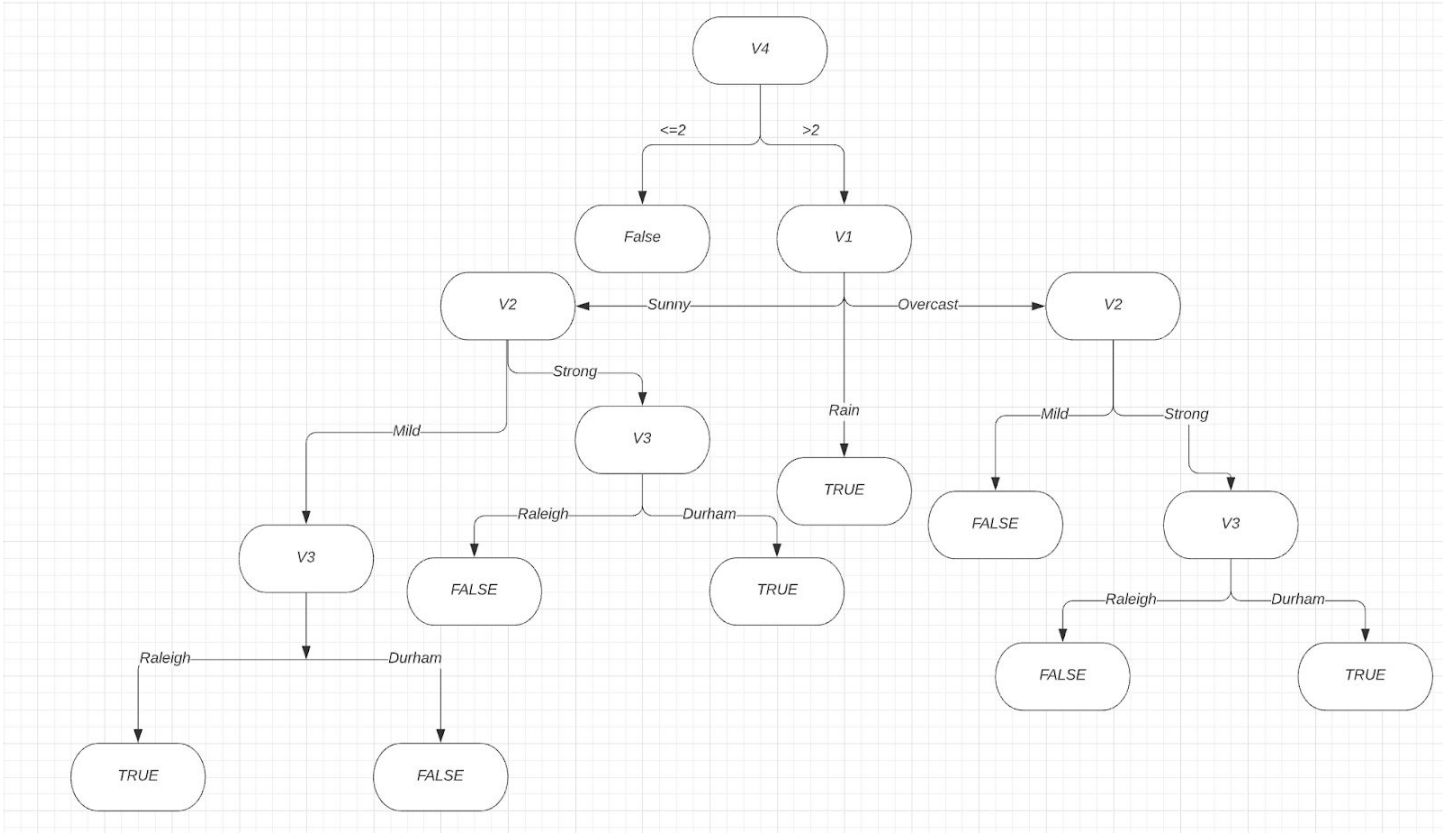
We have the following entropies:

Total entropy = 0.81

Attribute	IG
V3	0.81
V5	0.31

So we split on V3

Resultant Decision Tree is as follow:



b) Construct the tree manually using the Gini index. The maximum depth of the tree should be 2.

The Gini Index before Splitting the Tree at all there are 7 False and 9 True in the Class Labels so, the Gini Index will be like

--> $1 - (7/16)^2 - (9/16)^2 = 0.4921$

Now when I try to split on the basis of the attribute V1

Sunny (7 total)	3T	4F	$1 - (3/7)^2 - (4/7)^2 = 0.4897$
Over Cast (5 total)	2T	3F	$1 - (2/5)^2 - (3/5)^2 = 0.48$
Rain (4 rain)	3T	1F	$1 - (3/4)^2 - (1/16)^2 = 0.375$

Gini(V1)=7/16*0.4897+5/16*0.48+4/16*0.375=0.45799

Now we will split on the attribute V2

Strong(6 total)	4T	2F	$1 - (4/6)^2 - (2/6)^2 = 0.4444$
Mild (10 total)	5T	5F	$1 - (5/10)^2 - (5/10)^2 = 0.5$

Gini(V2)=6/16*0.44444+10/16*0.5=0.47915

Now we will split on the attribute V3

Raleigh (10 total)	5T	5F	$1-(5/10)^2-(5/10)^2=0.5$
Durham (6 total)	4T	2F	$1-(4/6)^2-(2/6)^2=0.4444$

Gini(V3)=10/16*0.5+6/16*0.4444=0.47915

Now we will split on the attribute V4
Since the attribute V4 is continuous, we will split on all the possible values and try to find out the Gini Index on all the possible splits.

1	F
2	F
3	T
4	F
6	T
8	T
9	F
11	T
12	T
13	T
17	T
18	F
19	T
21	F
22	T
27	T

Split on V4=1 such that V4<1 and V4>=1

V4<1(0 total)	0T	0F	$1-(0/0)^2-(0/0)^2=0$
V4>=1(16 total)	9T	7F	$1-(7/16)^2-(9/16)^2=0.4921$

Gini(V4=1)=16/16*0.4921

Split on V4=2 such that V4<2 and V4>=2

V4<2(0 total)	0T	1F	$1-(1/1)^2-(0/1)^2=0$
V4>=2(16 total)	9T	6F	$1-(6/15)^2-(9/15)^2=0.48$

Gini(V4=2)=1/16*0+15/16*0.48=0.45

Split on V4=3 such that V4<3 and V4>=3

V4<3(2 total)	0T	2F	$1-(2/2)^2-(0/2)^2=0$
V4>=3(14 total)	9T	5F	$1-(9/14)^2-(5/14)^2=0.4591$

Gini(V4=3)=14/16*0.4591=0.4017

Split on V4=4 such that V4<4 and V4>=4

V4<4(3 total)	1T	2F	$1-(1/3)^2-(2/3)^2=0.4444$
V4>=4(13 total)	8T	5F	$1-(8/13)^2-(5/13)^2=0.47337$

Gini(V4=4)=3/16*0.4444+13/16*0.47337=0.46793

Split on V4=6 such that V4<6 and V4>=6

V4<6(4 total)	1T	3F	$1-(1/4)^2-(3/4)^2=0.375$
V4>=6(12 total)	8T	4F	$1-(8/12)^2-(4/12)^2=0.44444$

Gini(V4=6)=4/16*0.375+12/16*0.44444=0.42705

Split on V4=8 such that V4<8 and V4>=8

V4<8(5 total)	2T	3F	$1-(2/5)^2-(3/5)^2=0.48$
V4>=8(11 total)	7T	4F	$1-(7/11)^2-(4/11)^2=0.462$

Gini(V4=8)=5/16*0.48+11/16*0.4628=0.468175

Split on V4=9 such that V4<9 and V4>=9

V4<9(6 total)	3T	3F	$1-(3/6)^2-(3/6)^2=0.5$
V4>=9(10 total)	6T	4F	$1-(6/10)^2-(4/10)^2=0.48$

Gini(V4=9)=6/16*0.5+10/16*0.48=0.4875

Split on V4=11 such that V4<11 and V4>=11

V4<11(7 total)	3T	4F	$1-(3/7)^2-(4/7)^2=0.4897$
V4>=11(9 total)	6T	3F	$1-(6/9)^2-(3/9)^2=0.4444$

Gini(V4=11)=7/16*0.4897+9/16*0.4444=0.464218

Split on V4=12 such that V4<12 and V4>=12

V4<12(8 total)	4T	4F	$1-(4/8)^2-(4/8)^2=0.5$
V4>=12(8 total)	5T	3F	$1-(5/8)^2-(3/8)^2=0.46875$

Gini(V4=12)=8/16*0.5+8/16*0.46875=0.484375

Split on V4=13 such that V4<13 and V4>=13

V4<13(9 total)	5T	4F	$1-(5/9)^2-(4/9)^2=0.4938$
V4>=13(7 total)	4T	3F	$1-(4/7)^2-(3/7)^2=0.4897$

Gini(V4=13)=9/16*0.4938+7/16*0.4897=0.49200625

Split on V4=17 such that V4<17 and V4>=17

V4<17(10 total)	6T	4F	$1-(6/10)^2-(4/10)^2=0.48$
V4>=17(6 total)	3T	3F	$1-(3/6)^2-(3/6)^2=0.5$

Gini(V4=13)=10/16*0.48+7/16*0.5=0.4875

Split on V4=18 such that V4<18 and V4>=18

V4<18(11 total)	6T	5F	$1-(6/11)^2-(5/11)^2=0.4958$
V4>=18(5 total)	3T	2F	$1-(3/5)^2-(2/5)^2=0.48$

Gini(V4=18)=11/16*0.4958+5/16*0.48=0.490862

Split on V4=19 such that V4<19 and V4>=19

V4<19(12 total)	6T	6F	0.5
V4>=19(4 total)	3T	1F	$1-(3/4)^2-(1/4)^2=0.375$

Gini(V4=19)=12/16*0.5+4/16*0.375=0.46875

Split on V4=21 such that V4<21 and V4>=21

V4<19(13 total)	7T	6F	$1-(7/13)^2-(6/13)^2=0.4970$
V4>=19(3 total)	2T	1F	$1-(2/3)^2-(1/3)^2=0.4444$

Gini(V4=21)=13/16*0.4970+3/16*0.4444=0.48713

Split on V4=22 such that V4<22 and V4>=22

V4<22(14 total)	7T	7F	0.5
V4>=22(2 total)	2T	0F	0

Gini(V4=22)=14/16*0.5=0.4375

Split on V4=27 such that V4<27 and V4>=27

V4<22(14 total)	7T	8F	$1-(7/15)^2-(8/15)^2=0.49777$
V4>=22(2 total)	1T	0F	0

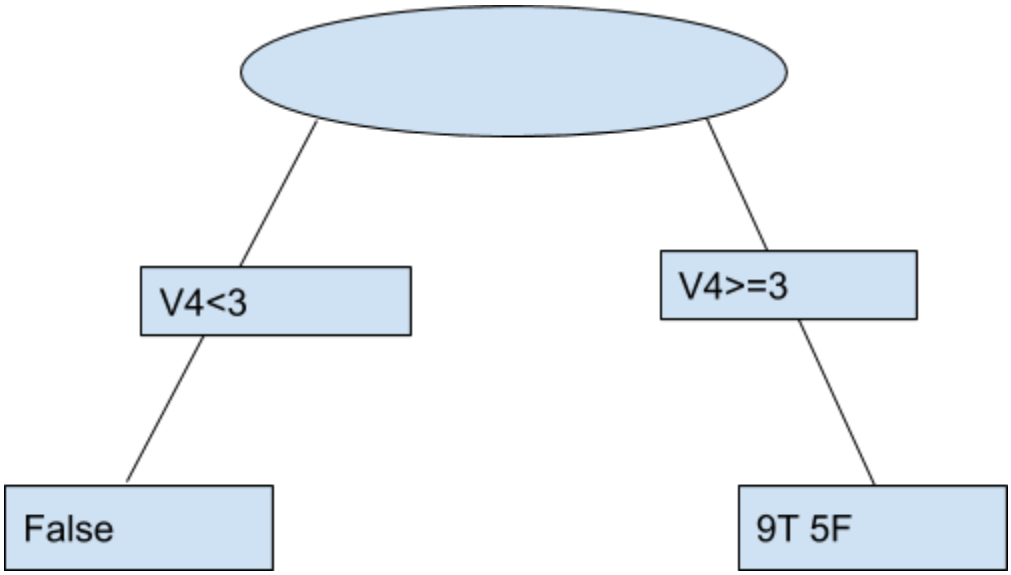
Gini(V4=22)=15/16*0.49777=0.4666

Now we will split on the attribute V5

cool(9 total)	6T	3F	$1-(6/9)^2-(3/9)^2=0.4444$
hot(7 total)	3T	4F	$1-(3/7)^2-(4/7)^2=0.48979$

Gini(V2)=9/16*0.44444+7/16*0.48979=0.4642

We will split the tree on the basis of attribute V4 where it had V4=3 as the Gini Index, in that case, was the least that is 0.4017



Now to Split further for the leftover Dataset which is the following:

V1	V2	V3	V4	V5	Class
Sunny	Strong	Raleigh	9	Cool	F
Overcast	Strong	Durham	11	Cool	T
Sunny	Strong	Durham	3	Cool	T
Rain	Strong	Durham	12	Cool	T
Rain	Mild	Raleigh	19	Hot	T
Overcast	Mild	Durham	4	Cool	F

Overcast	Strong	Raleigh	18	Hot	F
Sunny	Mild	Raleigh	27	Cool	T
Sunny	Mild	Raleigh	8	Cool	T
Overcast	Strong	Durham	6	Cool	T
Sunny	Mild	Durham	21	Hot	F
Overcast	Mild	Raleigh	17	Hot	F
Rain	Mild	Raleigh	22	Hot	T
Sunny	Mild	Raleigh	13	Hot	T

Now when I try to split on the basis of the attribute V1

Sunny (6 total)	4T	2F	$1-(4/6)^2-(2/6)^2=0.4444$
Over Cast (5 total)	2T	3F	$1-(2/5)^2-(3/5)^2=0.48$
Rain (3 rain)	3T	0F	$1-(3/3)^2=0$

Gini(V1)=6/14*0.4444+5/14*0.48+3/14*0=0.3618

Now we will split on the attribute V2

Strong(6 total)	4T	2F	$1-(4/6)^2-(2/6)^2=0.4444$
Mild (8 total)	5T	3F	$1-(5/8)^2-(3/8)^2=0.46875$

Gini(V2)=6/14*0.44444+8/14*0.46875=0.0.458314

Now we will split on the attribute V3

Raleigh (8 total)	5T	3F	$1-(5/8)^2-(3/8)^2=0.46875$
Durham (6 total)	4T	2F	$1-(4/6)^2-(2/6)^2=0.4444$

Gini(V3)=8/14*0.46875+6/14*0.4444=0.458314

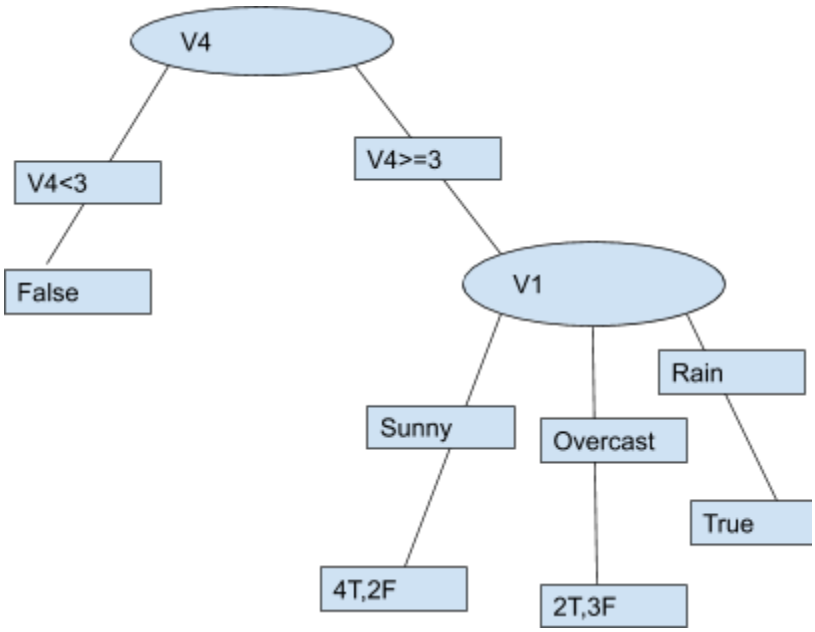
Now we will split on the attribute V5

cool(8 total)	6T	2F	$1-(6/8)^2-(2/8)^2=0.375$
hot(6 total)	3T	3F	0.5

Gini(V2)=8/14*0.375+6/14*0.5=0.42857

Now that the value of Gini when splitting on the attribute V1

The tree will look like the below:



c) How are the trees different? As part of your explanation, give 2 examples of data objects that would be classified differently by the two trees.

The first tree when checked for the training data gives 100 percent accuracy. However, we have made the second tree only for depth 2 so there are a few data points in the second tree that will be misclassified.

2 examples of data points that are wrongly classified are as follows:

Sunny	Mild	Durham	21	Hot	F

This datapoint from the training dataset will be predicted as True if we use the Gini Index based Decision Tree.

Overcast	Strong	Durham	11	Cool	T

This datapoint from the training dataset will be predicted as False if we consider the Gini Index based Decision Tree.

However, if we take the Information Gain based decision trees both the data points are correctly classified as False and True.

d) Which decision tree will perform better on the training dataset (hw2q1.csv)? Which will perform better on a test dataset? Can we know the answer?

When we look into the training data set, we can see that none of the data points will be misclassified when we use the Information Gain based decision Tree. So the classification error is 0/16.

However, when we consider the Gini Index based Decision tree the number of datapoints that go misclassified are 4 out of 16. So, the classification error is 4/16.

Hence we can say that for the training data Information Gain based decision tree performs better as all the data points in the dataset are classified correctly.

Now, let us check for the test dataset. We take into consideration the Generalization Error. Further, the Information Gain decision Tree might lead to overfitting as it has used the entire training data set for the making of the decision tree model. But when we look at the Gini Index decision tree, there might be underfitting because there are parts of the training dataset that are not taken into consideration while making the decision tree model.

Now, the data points in the test data set would determine if the errors are more or less.

Now we can also attempt to find the pessimistic error for both the trees, that is the Information Gain decision tree and Gini Index.

The pessimistic error for the Information Gain decision tree is 0+(9*0.5)/16=0.28125. The pessimistic error for the Gini Index decision tree is 0+(4+4*0.5)/16=0.375. Since the pessimistic error for the Gini Index based decision tree is less, the tree might be underfitting.

Q2.

a) Use the decision tree above to classify the provided dataset. hw2q2.csv. Construct a confusion matrix and report the test Accuracy, Error Rate, Precision, Recall, and F1 score. Use “Yes” as the positive class in the confusion matrix.

City	Pollution	Size	Precipitation	Label	Predicted Labels according to the given Decision tree
A	High	Large	Mild	N	N
A	High	Large	Mild	N	N
A	Low	Small	Low	N	N
A	Low	Medium	High	N	N
B	High	Meta	High	N	Y
B	High	Meta	Low	N	N
B	Low	Large	Mild	Y	Y
B	Low	Small	High	Y	N
B	High	Meta	Mild	Y	Y
B	High	Medium	High	Y	Y
B	High	Large	Mild	Y	Y
B	High	Large	Low	N	N
B	Low	Meta	Low	Y	Y

B	High	Small	Low	Y	N
B	Low	Medium	High	N	N
B	Low	Small	High	N	N

Confusion Matrix:

TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

	Predicted Class (Yes)	Predicted Class(No)
Actual Class (Yes)	5 (TP)	2 (FN)
Actual Class (No)	1 (FP)	8 (TN)

Accuracy = (TP+TN)/(TP+TN+FP+FN) = (5+8)/(5+8+1+2)=0.8125

Precision (p)= (TP)/(TP+FP)=(5)/(5+1)=⅝=0.8333

Recall (r)= (TP)/(TP+FN)=(5)/(5+2)=5/7=0.714

F1 Score = (2.r.p)/(r+p)=(2*0.8333*0.714)/(0.8333+0.714)

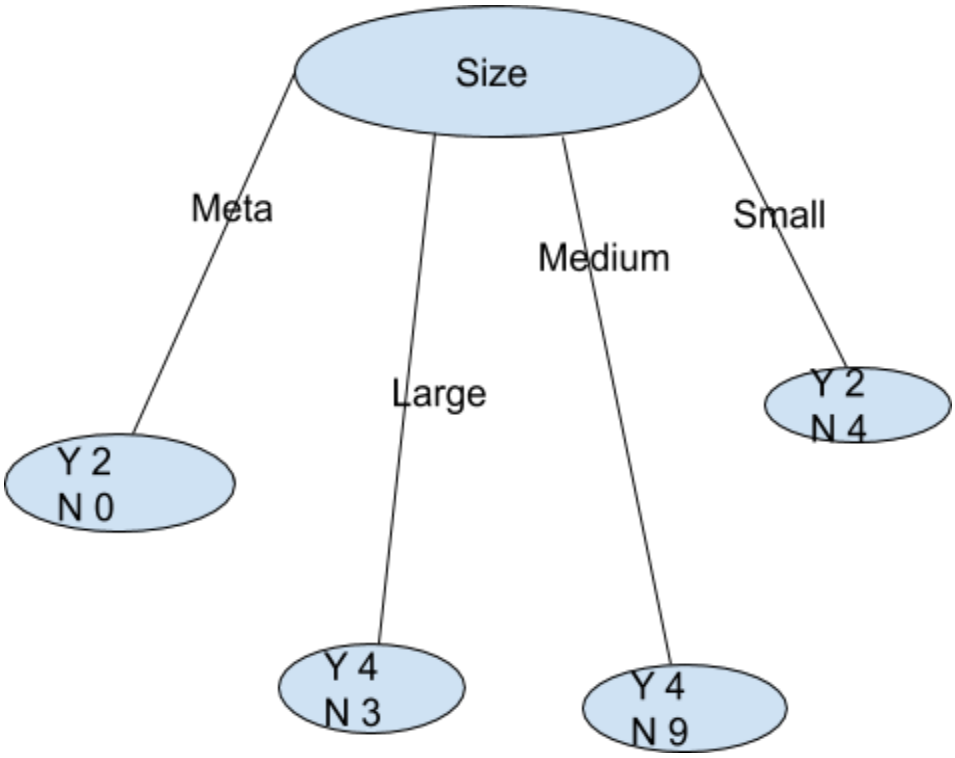
Error Rate = (FP+ FN)/(TP+FN+FN+TN)=(1+2)/(5+1+2+8)=3/16=0.1875

b) Calculate the optimistic training classification error before splitting and after splitting using Size, respectively. Consider only the subtree starting with the Size node. If we want to minimize the optimistic error rate, should the node's children be pruned?

Optimistic Error before splitting= 12/28

When you do not prune there will be 12 Yes, and 16 No.
So if we take the Supermajority then the prediction will be No.
In that case, there will be 12 wrong predictions, since all of it will be predicted as a No.
Hence, the misclassifications will be 12 out of 28.

Optimistic Error after splitting = 9/28



In the first branch, Y will be majority so all the predictions from that leaf will be Y so there will be 0 misclassifications
In the second branch Y again will be the majority so all the predictions will be Y so there will be 3 misclassifications
In the third branch, N is the majority so the prediction will be N and in that case there will be 4 misclassifications
In the fourth branch, N is the majority so prediction will be N and hence there will be 2 misclassifications.

Hence total no. of misclassifications will be 3+4+2=9

Hence the Optimistic Error will be 9/28

We can see that the Optimistic Error is decreasing from 12/28 to 9/28 so, in that case, we will not prune the tree.

c) Calculate the pessimistic training errors before splitting and after splitting using Size respectively. Consider only the subtree starting with the Size node. When calculating pessimistic error, use a leaf node error penalty of 0.8. If we want to minimize the pessimistic error rate, should the node's children be pruned?

When we talk about the Pessimistic Error, e(T) + N × 0.5 (N: number of leaf nodes)

Pessimistic Error before splitting

= (12+0.8)/28=12.8/28

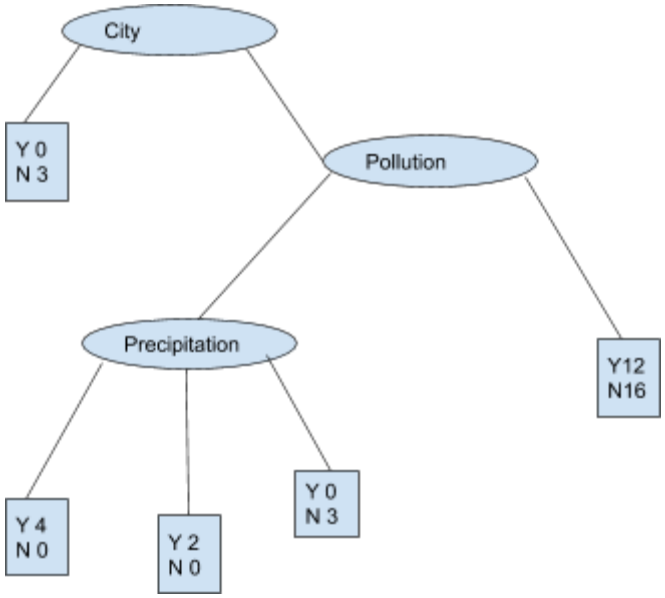
Pessimistic Error after splitting

= (9+4*0.8)/28 = 12.2/28

The pessimistic error is reducing from 12.8/28 to 12.2/28 so we should not Prune the tree.

d) Assuming that the "Size" node is pruned, recalculate the test Error Rate using hw2q2.csv. Based on your evaluation using the test dataset in hw2q2.csv, was the original tree (with the Size node) over-fitting? Why or why

not?



We can see that the tree will look like this after pruning so the new predicted and actual values will become like mentioned in the below table

City	Pollution	Size	Precipitation	Label	Predicted Labels according to the given Decision tree
A	High	Large	Mild	N	N
A	High	Large	Mild	N	N
A	Low	Small	Low	N	N
A	Low	Medium	High	N	N
B	High	Meta	High	N	Y
B	High	Meta	Low	N	N
B	Low	Large	Mild	Y	N
B	Low	Small	High	Y	N
B	High	Meta	Mild	Y	Y
B	High	Medium	High	Y	Y
B	High	Large	Mild	Y	Y
B	High	Large	Low	N	N
B	Low	Meta	Low	Y	N
B	High	Small	Low	Y	N
B	Low	Medium	High	N	N

B	Low	Small	High	N	N

Confusion Matrix:

TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

	Predicted Class (Yes)	Predicted Class(No)
Actual Class (Yes)	3 (TP)	4 (FN)
Actual Class (No)	1 (FP)	8 (TN)

Error rate according to the New decision tree and confusion Matrix:

= (FP+FN)/(TP+FN+FP+TN)
= 5/16

Based on the evaluation of the dataset we can say that the original tree was over fitting, because the error rate was less when the tree was not pruned. Since this decision tree is very flexible and there is a possibility that it must have been very specific to the particular training dataset. There can be scenarios where a new data point would come up and they go misclassified due to the overfitting of the tree.
This issue has been resolved by pruning the tree to remove the extra details and the overfit areas. Now even though the error rate has increased the tree is more likely to classify a new data point correctly.

Q3. (a) Distance Matrix:

	1	2	3	4	5	6	7	8	9
1	0	6.726812	1.802776	9.552487	23.706539	17.613915	18.027756	20.099751	11.011358
2	6.726812	0	8.485281	11.597414	17.269916	14.089003	11.629703	24.68299	7.211103
3	1.802776	8.485281	0	9.300538	25.303162	18.506756	19.830532	18.741665	12.165525
4	9.552487	11.597414	9.300538	0	23.113849	12.083046	22.276669	13.756816	8.746428
5	23.706539	17.269916	25.303162	23.113849	0	13.462912	11.7047	36.359318	14.5
6	17.613915	14.089003	18.506756	12.083046	13.462912	0	18.580904	23.753947	6.964194
7	18.0277	11.6297	19.8305	22.2766	11.7047	18.5809	0	35.9026	14.5344

	56	03	32	69		04		46	42
8	20.099751	24.68299	18.741665	13.756816	36.359318	23.753947	35.902646	0	22.388613
9	11.011358	7.211103	12.165525	8.746428	14.5	6.964194	14.534442	22.388613	0

(b) (i)
From the above table we check for the smallest distance for each of the test values amongst the remaining training values and we find the nearest neighbours as follows:

Test instance	Nearest Neighbour	Predicted Class	Actual Class
6	4	+	+
7	2	-	-
8	4	+	+
9	2	-	-

Confusion Matrix:

Actual (column) Predicted(row)	+	-
+	2	0
-	0	2

Accuracy = (2+2)/(2+2) = 1 i.e 100%

(b) (ii)

First fold: [3,6,9]

Test instance	Nearest Neighbour	Predicted Class	Actual Class
3	1	-	+
6	4	+	+
9	2	-	-

Second fold: [1,4,7]

Test instance	Nearest Neighbour	Predicted Class	Actual Class
1	3	+	-

4	9	-	+
7	2	-	-

Third fold: [2,5,8]

Test instance	Nearest Neighbour	Predicted Class	Actual Class
2	1	-	-
5	7	-	-
8	4	+	+

Confusion Matrix:

Actual (column) Predicted(row)	+	-
+	2	1
-	2	4

Accuracy: (2+4)/(2+1+2+4) = ⅔ = 0.667 i.e 66.7%

(b)(iii) LOOCV

Each test instance in the following table is considered individually and compared against the rest of the dataset of n-1 instances i.e 9-1 = 8 instances in this case.

Test instance	Nearest Neighbour	Predicted Class	Actual Class
1	3	+	-
2	1	-	-
3	1	-	+
4	9	-	+
5	7	-	-
6	9	-	+
7	2	-	-
8	4	+	+
9	6	+	-

Confusion Matrix:

Actual (column)	+	-

Predicted(row)		
+	1	2
-	3	3

Accuracy: (1+3)/(1+2+3+3) = 4/9 = 0.44

(c) LOOCV will give us 0% testing accuracy since we are using the “simple majority classifier” and we have an equal number of instances in each class originally. Each test instance we test for, will have a training majority of the wrong class and it would be classified as the wrong class instead.

For example: let us consider an instance with the actual class = positive. Now the train set will consist of the remaining 29 instances.

Now the class distribution in the training dataset is as follows: positive -> 14 and negative -> 15
The simple majority classifier will pick negative since 15>14 and it would be classified wrong. This would happen for every test instance and result in a 0% accuracy.

Q4.For the following problem, you may find it useful to fill in the following table (optional).

P (Buy = Y es) = 5/10	P(Buy = No) = 5/10
P(Size = small | Buy = Yes) = 2/5	P(Size = small | Buy = No) = 1/5
P(Size = large | Buy = Yes) = 3/5	P(Size = large | Buy = No) = 4/5
P(Color = white | Buy = Yes) = 0/5	P(Color = white | Buy = No) = 2/5
P(Color = red | Buy = Yes) = 2/5	P(Color = red | Buy = No) =2/5
P(Color = black | Buy = Yes) = 3/5	P(Color = black | Buy = No) = 1/5
P(Material = cotton | Buy = Yes) =2/5	P(Material = cotton | Buy = No) = 4/5
P(Material = fiber | Buy = Yes) =3/5	P(Material = fiber | Buy = No) = 1/5

Using the training dataset above, how would a Naive Bayes classifier classify the following data points? Show your work.

a) {Size = small, Color = black, Material = cotton}

P(A|Buy=Yes)=P(Size = small|Buy=Yes)*P(Color = black|Buy=Yes)*P(Material = cotton|Buy=Yes)=⅖*¾*⅖=12/125

P(A|Buy=Yes)(Buy=Yes)=12/125*5/10=12/250

P(A|Buy=No)=P(Size = small|Buy=No)*P(Color = black|Buy=No)*P(Material = cotton|Buy=No)=⅕*½*⅘=4/125

P(A|Buy=No)(Buy=No)=4/125*5/10=4/250

P(A|Buy=Yes)(Buy=Yes) > P(A|Buy=No)(Buy=No)

So the prediction will be Yes.

b) {Size = small, Color = red, Material = fiber}

P(A|Buy=Yes)=P(Size = small|Buy=Yes)*P(Color =red|Buy=Yes)*P(Material =fiber|Buy=Yes)=2/5*2/5*3/5=12/125

P(A|Buy=Yes)(Buy=Yes)=12/125*5/10=12/250

P(A|Buy=No)=P(Size = small|Buy=No)*P(Color =red|Buy=No)*P(Material =fiber|Buy=No)=1/5*2/5*1/5=2/125

P(A|Buy=No)(Buy=No)=2/125*5/10=2/250

P(A|Buy=Yes)(Buy=Yes) > P(A|Buy=No)(Buy=No)

So the prediction will beYes.

c) {Size = large, Color = red, Material = cotton}

P(A|Buy=Yes)=P(Size =large|Buy=Yes)*P(Color =red|Buy=Yes)*P(Material =cotton|Buy=Yes)=3/5*2/5*2/5=12/125

P(A|Buy=Yes)(Buy=Yes)=12/125*5/10=12/250

P(A|Buy=No)=P(Size = large|Buy=No)*P(Color =red|Buy=No)*P(Material =cotton|Buy=No)=4/5*2/5*4/5=32/125

P(A|Buy=No)(Buy=No)=32/125*5/10=32/250

P(A|Buy=Yes)(Buy=Yes) < P(A|Buy=No)(Buy=No)

So the prediction will be No.

d) {Size = large, Color = white, Material = fiber}

P(A|Buy=Yes)=P(Size =large|Buy=Yes)*P(Color =white|Buy=Yes)*P(Material =fiber|Buy=Yes)=3/5*0/5*3/5=0

P(A|Buy=Yes)(Buy=Yes)=0*5/10=0

P(A|Buy=No)=P(Size = large|Buy=No)*P(Color =white|Buy=No)*P(Material =fiber|Buy=No)=4/5*2/5*1/5=8/125

P(A|Buy=No)(Buy=No)=8/125*5/10=8/250

P(A|Buy=Yes)(Buy=Yes) < P(A|Buy=No)(Buy=No)

So the prediction will be No.

Q5.

PART E.

1. Comparison in terms of overall accuracy - which classifier performed best?

Decision trees perform better since overall accuracy for dtree is greater than KNN

2. Comparison in terms of precision and recall - What does the difference mean?

For given sample data we get the following results:

	Precision	Recall
Decision Tree	0.51	0.59
KNN	0.35	0.32

We see that Decision trees perform better in terms of both precision and recall.

For the given dataset precision defines that out of all the people who were diagnosed with diabetes how many of them actually have diabetes.

Recall defines out of all the people who actually have diabetes how many have been correctly detected.

3. Considering the background and target of this dataset (classifying whether a patient has diabetes), which evaluation metrics are more important? Explain your choice.

For the given dataset precision defines that out of all the people who were diagnosed with diabetes how many of them actually have diabetes.

Recall defines out of all the people who actually have diabetes how many have been correctly detected.

We want a higher value of recall because we want to maximise the correct diagnosis of having diabetes. Because, even if people were diagnosed as having diabetes who actually do not it is not risky.

Therefore, Recall is the more important evaluation metric.

4. Based on your findings, are these classifiers effective on this dataset?

It is not effective as our recall values for both the classifiers are pretty low for the given sample checker. And the resultant evaluation metrics might vary for different folds. The size of the dataset is also limited since it consists of only 100 instances. Since the dataset is pertaining to medical diagnosis, such low values of recall/accuracy are not enough to effectively classify it.