ALDA Homework 2 – 2018 :

Submitted by :
Srinivasan Balan – Unity id : sbalan
Harika Malapaka – Unity Id : hsmalapa

(40 points) [PCA] [Xi Yang] In this problem, you will perform a PCA on the provided training dataset (“hw2q1 train.csv”) and the testing dataset (“hw2q1 test.csv”), which come from the Connectionist Bench Dataset (http://archive.ics.uci.edu/ ml/datasets/connectionist+bench+(sonar,+mines+vs.+rocks)). In both datasets, each row represents a data point or sample. The first 60 columns are input features, and the last column ”Class” is the output label, with the letters ”R” and ”M” indicating if a sample is a Rock or a Mine, respectively. Write code in Matlab, R or Python to perform the following tasks. Please report your outputs and key codes in the document file, and also include your code file (end with .m, .r or .py) in the .zip file.

(a) (2 points) Load the data. Report the size of the training and testing sets. How many Rock (R) and Mine (M) samples are in the training set and the testing set, respectively?

part a)

Size of training set is : 156

Size of test set is : 52

The no of Rocks in training set is : 73

The no of Rocks in test set is : 24

The no of Minerals in training set is : 83

The no of Minearals in test set is : 28

(b) (18 points) Preprocessing Data-Normalization: Please run normalization on all input features in both the training and testing datasets to obtain the normalized training and the

normalized testing datasets. (Hint: you will need to use the min/max of the training dataset to normalize the testing dataset and do NOT normalize the output ”Class” of data.)

Use the NEW normalized datasets for the following tasks :

i. (2 points) Calculate the covariance matrix of the NEW training dataset.

part b) i (Normalized data)

Calculated Covariance matrix

Further details to see the coariance matrix, please use the code file attached.

ii. (2 points) Calculate the eigenvalues and the eigenvectors based on the covariance matrix in (i) above. Report the size of the covariance matrix and the 5 largest eigenvalues.

part b) ii (Normalized data)

Shape of covariance matrix of NEW - normalized training data: (60, 60)

Size of covariance matrix of NEW - normalized training data 60

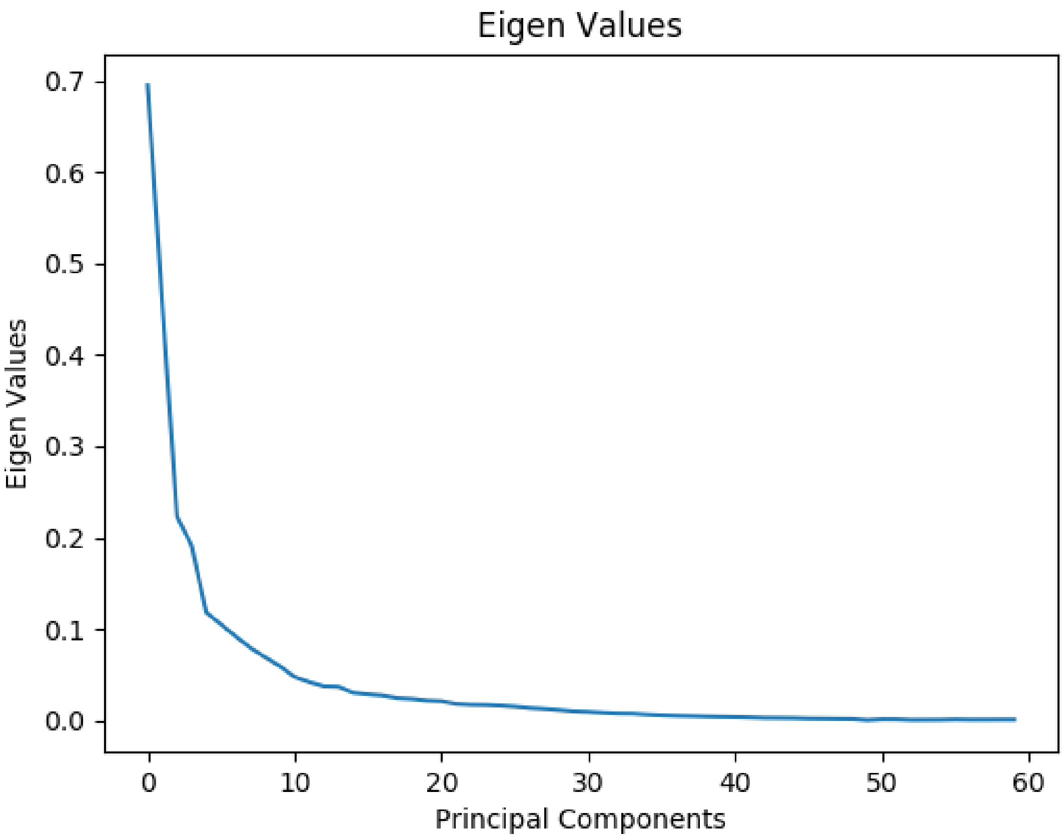
The top 5 eigen values are :

[0.695 0.457 0.223 0.191 0.118]

iii. (1 point) Display the eigenvalues using a bar graph or a plot, and choose a reasonable number(s) of eigenvectors. Justify your answer.

part b) iii (Normalized data)

Graph for Eigen Values -- Normalized data



JUSTIFICATION :

The number of eigen Vectors chosen is 10 because the graph starts to descend from that point approximately.

Although we can choose this in many ways, this would be optimal since we want to extract the components on the steep slope. The components on the shallow slope contribute little to the solution.

Although we can consider number of eigen vectors in the range 9 to 15 because the slope starts getting shallow from 9th principal vector and continues to get shallower till 15th principal and from there onwards, it's dropping by very little.

iv. (13 points) Next, you will combine PCA with a K-nearest neighbor (KNN) classifier. More specifically, PCA will be applied to reduce the dimensionality of data by transforming the original data into p (p ≤ 60) principle components; and then KNN (K = 3, euclidean distance as distance metric) will be employed to the p principle components for classification (third-party packages are allowed to use for KNN).

• (5 points) Report the accuracy of the NEW testing dataset when using PCA (p = 10) and the 3NN classifier. To show your work, please submit the corresponding csv file (including the name of csv file in your answer below). Your csv file should have 12 columns: columns 1-10 are the 10 principle components, column 11 is the original ground truth output ”Class”, and the last column is the predicted output ”Class”.

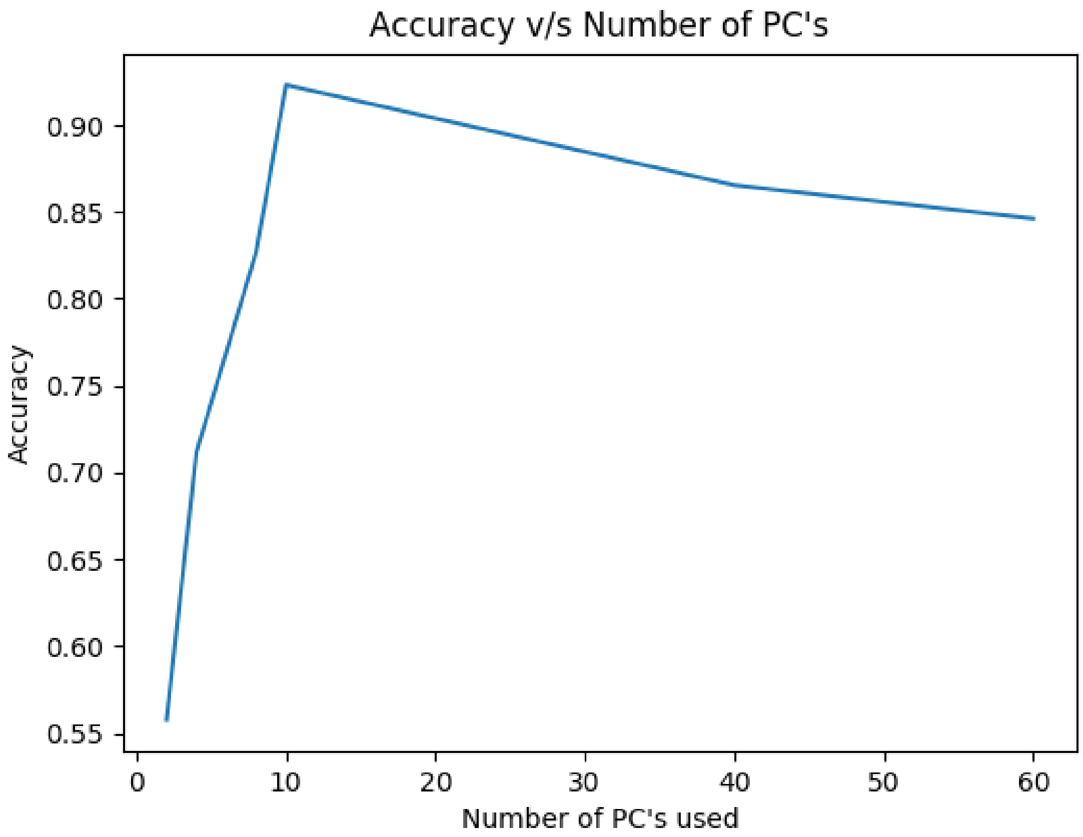
part 1 in b)iv ---> p=10 (Normalized data)

The csv file of 12 columns of Normalized dataset is saved in p10_data_norm.csv file

The accuracy when p=10for Normalized data is 0.9230769230769231

• (6 points) Plot your results by varying p: 2, 4, 8, 10, 20, 40, and 60 respectively. In your plot, the x-axis represents the number of principle components and the y-axis refers to the accuracy of the NEW testing dataset using the corresponding number of principle components and 3NN.

part 2 in b) iv all p's (Normalized dataset)



• (2 point) Based upon the PCA +3NN's results above, what is the most "reasonable" number of principle components among all the choices? Justify your answer.

part 3 in b)iv (Normalized data)

Looking at the data, no of p's is ideally as 10 as the accuracy is high for that

(c) (18 points) Preprocess Data-Standardization: Similarly, please run standardization on all input features to obtain the standardized training and the standardized testing datasets. Then repeat the four steps i-iv in (b) above on the two NEW standardized datasets.

-------------For Standardized Data------------------------------

part c) - repeating b) with standardized data

part b) i (Standardized data)

Calculated covariance matrix

USe code file to see the covariance matrix

part b) ii (Standardized data)

Shape of covariance matrix of NEW - standardized training data: (60, 60)

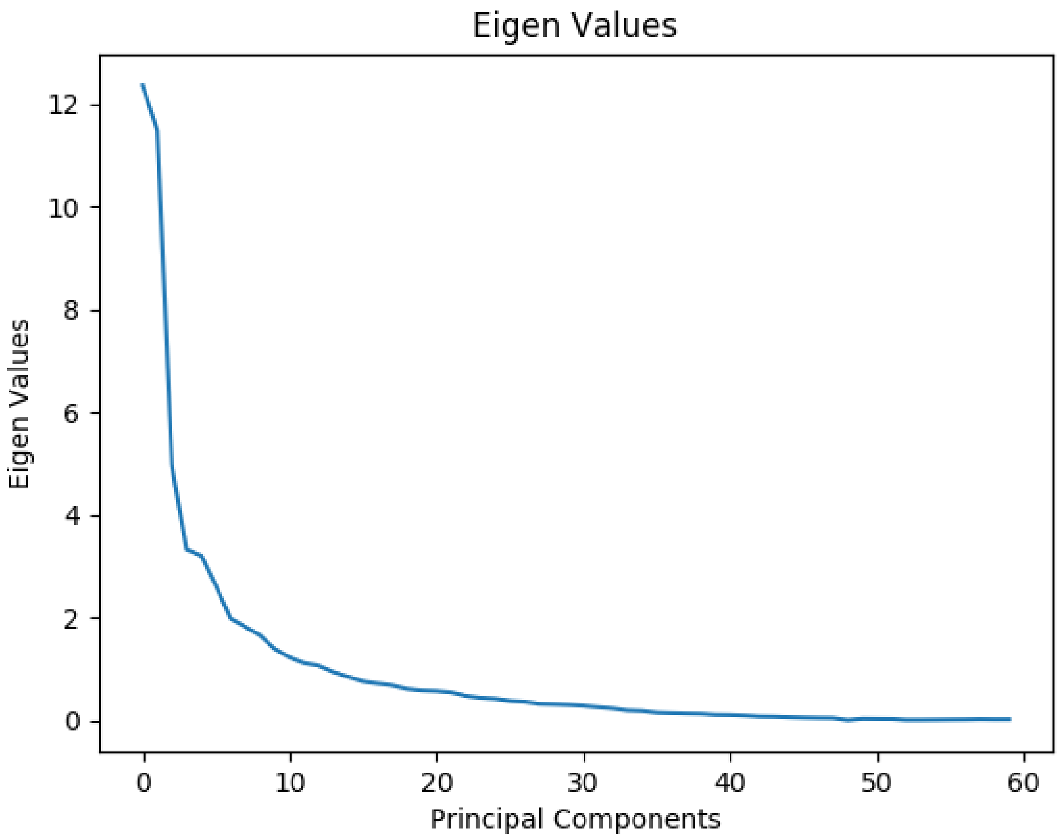
Size of covariance matrix of NEW - standardized training data 60

The top 5 eigen values are :

[12.35 11.484 4.947 3.324 3.205]

part b) iii (Standardized data)

Graph for eigen values- Standardized data



The optimum number of eigen values to choose is 10.

Since from that point onwards, the slope of the curve is decreasing.

Although we can choose this in many ways, this would be optimal since we want to extract the components on the steep slope. The components on the shallow slope contribute little to the solution.

Although we can consider number of eigen vectors in the range 9 to 15 because the slope starts getting shallow from 9th principal vector and continues to get shallower till 15th principal and from there onwards, it's dropping by very little.

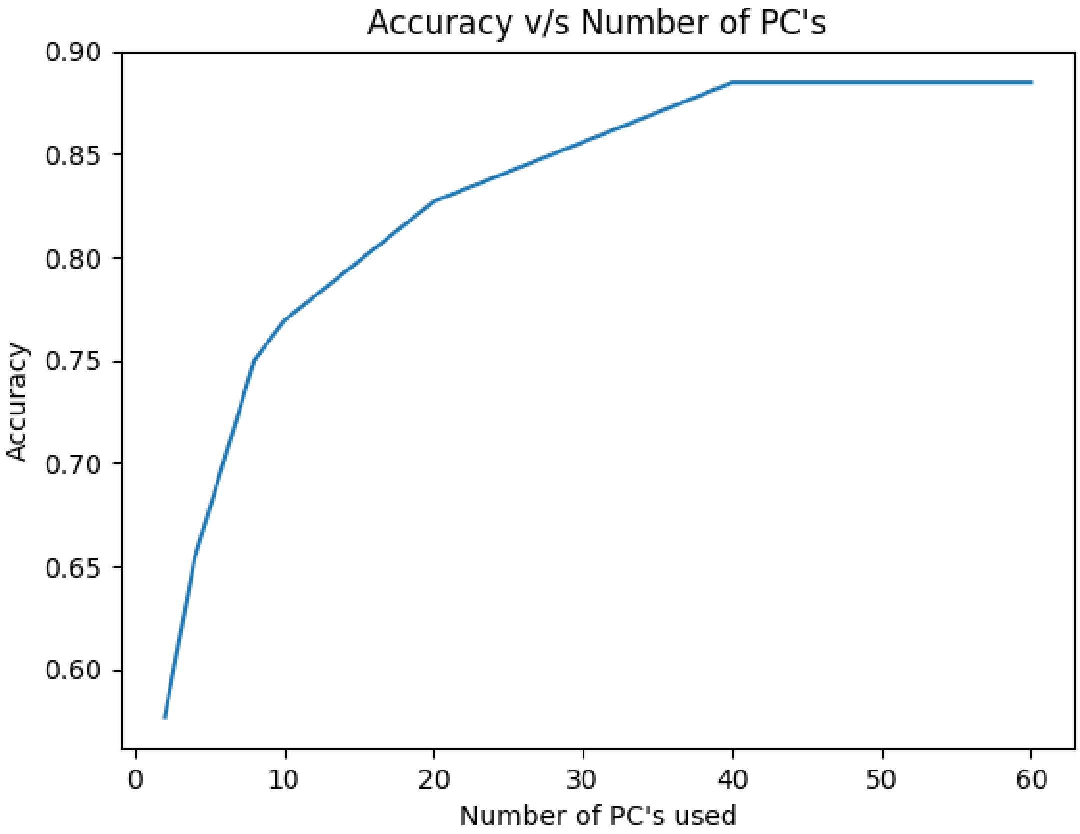
part b) iv (Standardized data)

part 1 in b)iv ---> p=10 (Standardized data)

The csv file of 12 columns for Standardized data is saved in p10_data_stand.csv file

The accuracy when p=10 for Standardized datset is 0.7692307692307693

part 2 in b) iv - all p's (Standardized data)



part 3 in b)iv (Standardized data)

Looking at the results, I would choose p as 40 because the accuracy for it is high

----------------Standardized over - all operations done ---------------------------------------

Accuracy of Normalized data [0.56 0.71 0.83 0.92 0.9 0.87 0.85]

Average : 0.8049450549450549

Accuracy of Standardized data [0.58 0.65 0.75 0.77 0.83 0.88 0.88]

Average : 0.7637362637362638

(d) (2 points) Comparing the results from (b) and (c), which of the two data-processing procedures, normalization or standardization, would you prefer for the given datasets? And why? (Answer without any justification will get zero point.)

- Since, the accuracy for Normalized dataset is high, while using pre-processing techniques, NORMALIZATION of data is a better idea.
- And in Normalization, the data points are scaled to a very small interval and even if there are outliers, they would be taken care of.
- And scaling would produce all points in the interval 0 to 1.
- And in Normalization, the accuracy is good for p=10 which means with minimum use of variables, we are getting a good accuracy.
- This will lead us to a less complicated model which is very important for predictions.

2. (20 points) [Decision Tree][Song Ju] In the given "hw2q2.csv", all of the input features are nominal except for the first column, which is a ratio and continuous. The output label has two class values: T or F. Complete the following tasks using the decision tree algorithm discussed in the lecture. In the case of ties, break ties in favor of the leftmost feature. (You can hand-draw all of your trees on paper and scan your results into the final pdf.)

(a) (10 points) Construct the tree manually using ID3/entropy computations, write down the computation process and show your tree step by step. (No partial credit)

To find out the root attribute, we need to calculate which attribute has highest gain.

Since attribute A has continuous values, we will find the gain at every possible split and then compare it with the other 4 attributes.

As there are 16 continuous values, there are 15 possible splits and the entropy for each one is give below.

NOTE :

All the calculation are performed in excel and we are attaching the excel workbook by name : hw2_q2_solution for further reference.

The notation split(x) says that records are splitted in the manner : the records < x into 1 category and >=x in another category

Attribute A :

The entropy at the following splits are :

Split(2) : 0.9863

Split(3) – 0.934

Split(6) – 0.875

Split(7) -0.981

Split(10) -0.953

Split(11) -0.987

Split(16) – 0.969

Split(17) -0.935

Split(25) – 0.833

Split(27) – 0.948

Split(29) - 0.982

Split(33) – 0.987

Split(34) – 0.985

Split(36) -0.981

Split(45) – 0.875

Split(50) – 0.934

Attribute B : 0.9772

Attribute C : 0.9772

Attribute D : 0.6774

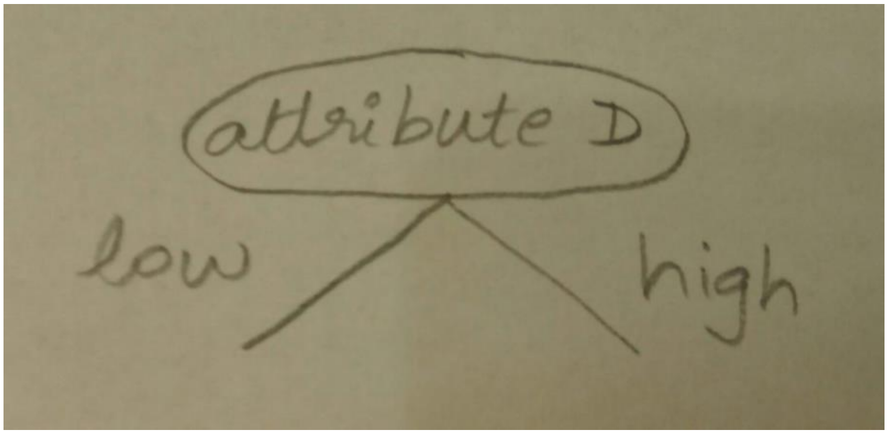
Attribute D : 0.6774

Attribute E : 0.8828

Since the entropy is low for attribute D among others, we choose it as root.

As the entropy is low, the gain automatically will be high.

So far, our decision tree is



Taking the data points which have the attribute : LOW are total 8 :

3	BLUE	SMALL	LOW	COOL	F
16	BLUE	LARGE	LOW	COOL	F
33	BLUE	SMALL	LOW	HOT	F
2	RED	LARGE	LOW	COOL	F
6	RED	SMALL	LOW	COOL	T
11	RED	SMALL	LOW	COOL	F
45	RED	SMALL	LOW	HOT	F
50	RED	LARGE	LOW	HOT	F

Now again to further drill down, we should repeat the procedure as above.

After arranging the continuous variables , there are 11 possible splits :

Feature A :

Split(2) – 0.541

Split(3) – 0.518

Split(6) – 0.488

Split(11) – 0.344

Split(16) – 0.406

Split(33) – 0.451

Split(45) – 0.488

Split(50) – 0.518

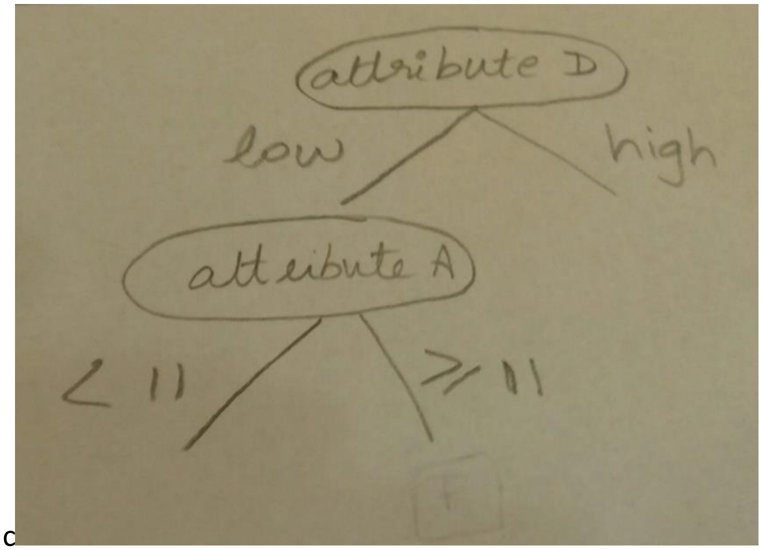
Feature B : 0.451

Feature C : 0.451

Feature E : 0.451

As we can see, attribute A will be selected further for the records which have attribute D as LOW .

The tree at this point is :



c c

Now, continuing the left subtree, if we proceed with records whose Attribute D is LOW and whose attribute A is split at less than 11 :

The records are :

2	RED	LARGE	LOW	COOL	F
3	BLUE	SMALL	LOW	COOL	F
6	RED	SMALL	LOW	COOL	T

Eliminating Attribute D and A, we have :

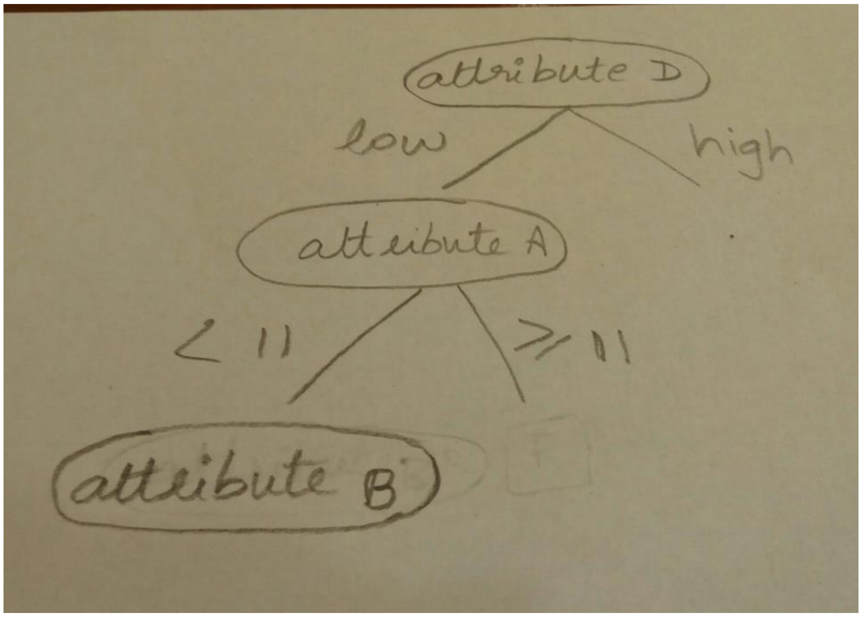
RED	LARGE	COOL	F
BLUE	SMALL	COOL	F
RED	SMALL	COOL	T

Bu visual analyzation, the attribute 5 would have a high entropy since all values are 'Cool'.

Now if we see attribute B and attribute C would have similar entropy.

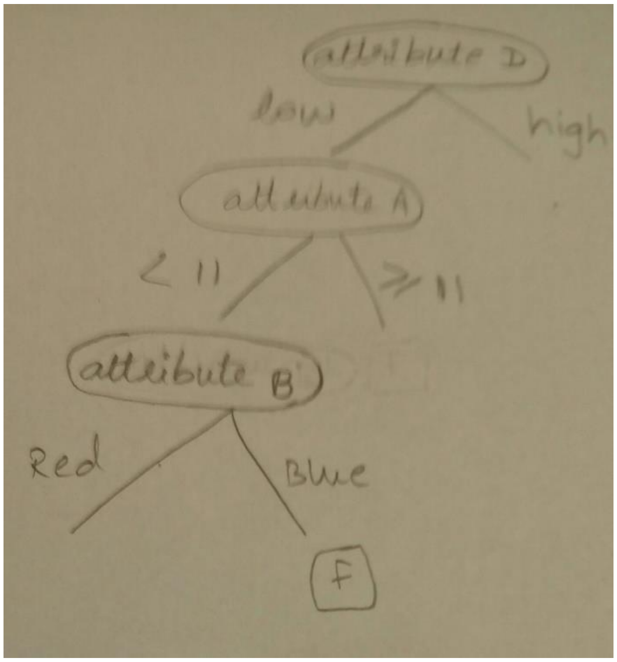
So considering a tie, we break it taking lest most attribute, which is B in our case.

The tree at this point would be :



Since there is only with 1 record with attribute B as Blue and it's label is 'F', we can directly classify them without further splitting.

The tree at this point would look like :

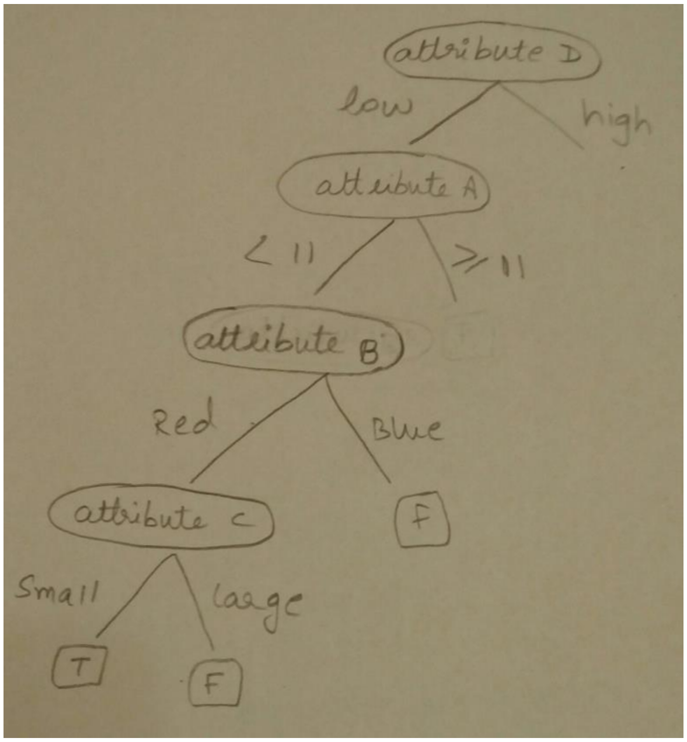


Coming to the records with Attribute B as Red, we have only 2 which can be split in the following manner :

RED	LARGE	COOL	F
RED	SMALL	COOL	T

As Attribute E is 'Cool' for both records, we can split using Attribute C – Small : true and Large : false

The tree now would look like :



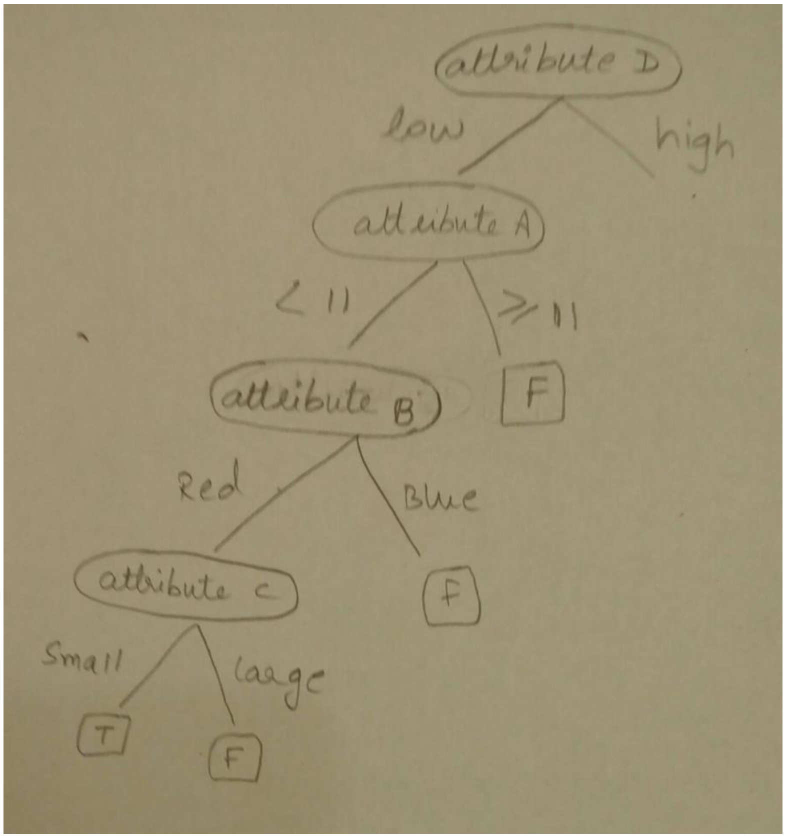
Now, continuing the left subtree, if we proceed with records whose Attribute D is LOW and whose attribute A is split at greater than 11 :

The records are :

11	RED	SMALL	LOW	COOL	F
16	BLUE	LARGE	LOW	COOL	F
33	BLUE	SMALL	LOW	HOT	F
45	RED	SMALL	LOW	HOT	F
50	RED	LARGE	LOW	HOT	F

Since all records label is ‘F’, no further split is required.

So the tree at this point is :



Now coming to the right subtree :

The records whose attribute D is “High” :

7	BLUE	LARGE	HIGH	COOL	F
17	BLUE	LARGE	HIGH	COOL	F
27	BLUE	LARGE	HIGH	HOT	T
29	BLUE	SMALL	HIGH	HOT	T
34	BLUE	LARGE	HIGH	HOT	T
10	RED	SMALL	HIGH	COOL	T
25	RED	SMALL	HIGH	HOT	T
36	RED	LARGE	HIGH	HOT	T

Repeating the process of calculating entropy :

For Attribute A:

Split(10) – 0.518

Split(17) – 0.738

Split(25) – 0.344

Split(27) – 0.5

Split(29) - 0.607

Split(34) - 0.689

Split(36) – 0.755

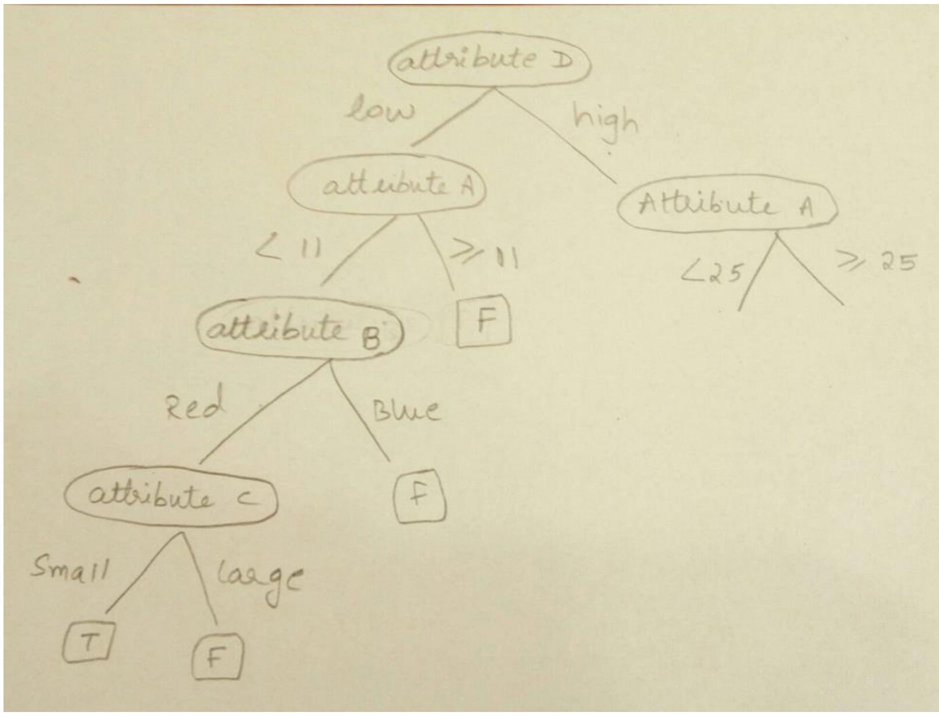
For attribute B : 0.607

For attribute C : 0.607

For attribute E : 0.344

As ties favor left most attribute, we would select Attribute A splitting them as <25 and >=25

The tree at this step is :



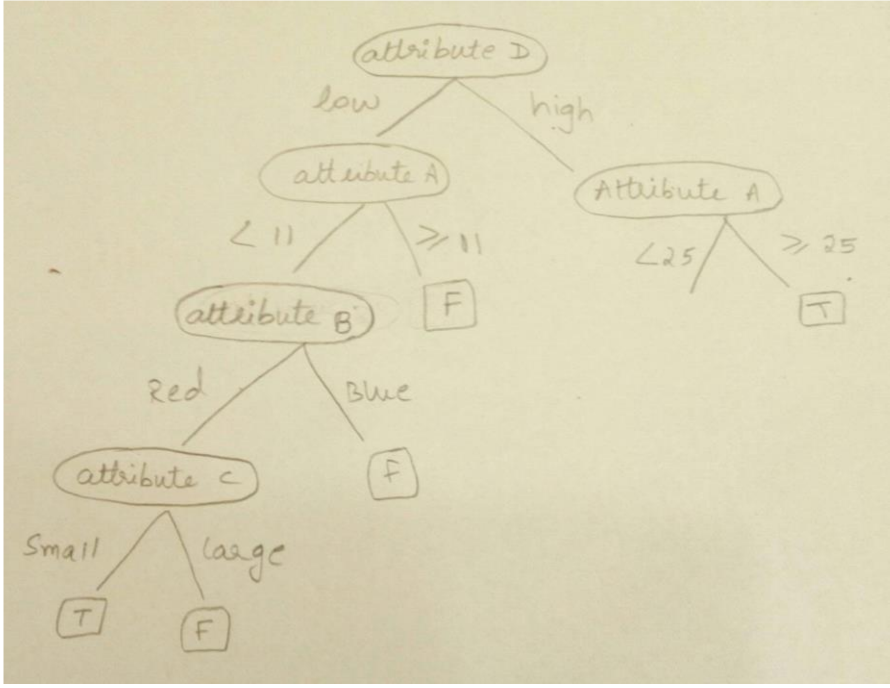
Further, splitting the data, the records with greater than or equal to25 as their Attribute A are :

25	RED	SMALL	HIGH	HOT	T
27	BLUE	LARGE	HIGH	HOT	T
29	BLUE	SMALL	HIGH	HOT	T

34	BLUE	LARGE	HIGH	HOT	T
36	RED	LARGE	HIGH	HOT	T

Since all the output labels are ‘T’, no further splitting is required and we can categorize all of them as True.

The tree at this point is :



Now, coming to the records which have value less than 25 are :

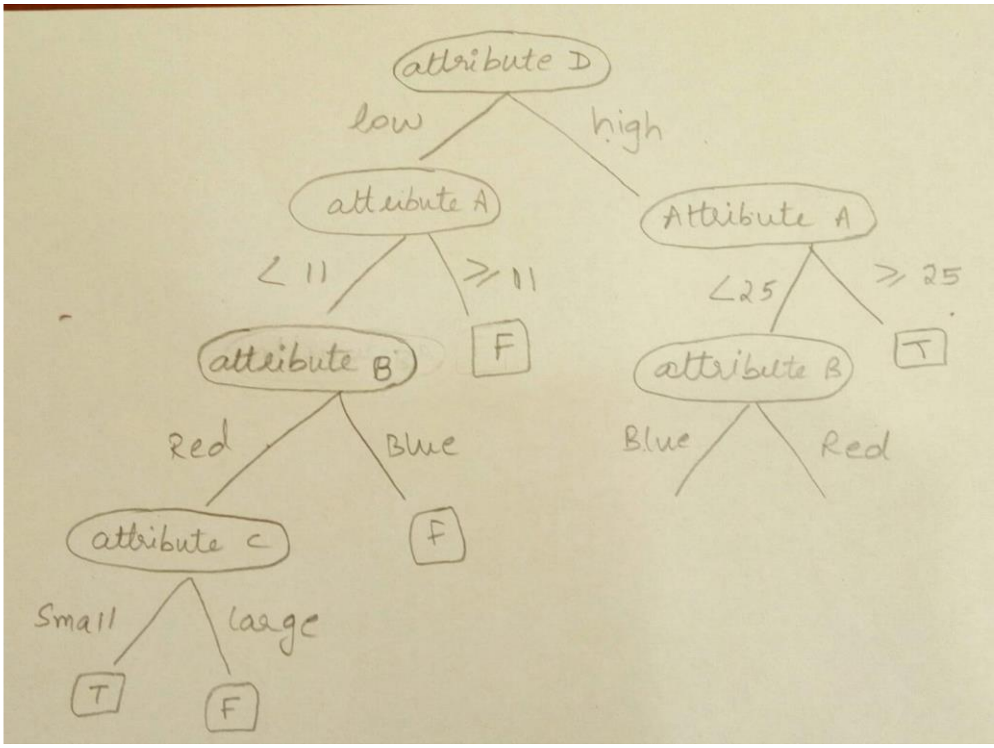
7	BLUE	LARGE	HIGH	COOL	F
10	RED	SMALL	HIGH	COOL	T
17	BLUE	LARGE	HIGH	COOL	F

Here, the attribute E is ‘Cool’ for all the records, so it will have high entropy.

Thus consider Attribute B and C.

Visually observing, they both would have equal entropy, so breaking the tie with selecting left most attribute, the next node would be Attribute B.

The tree at this point of time is :



The node would be divided like : BLUE :

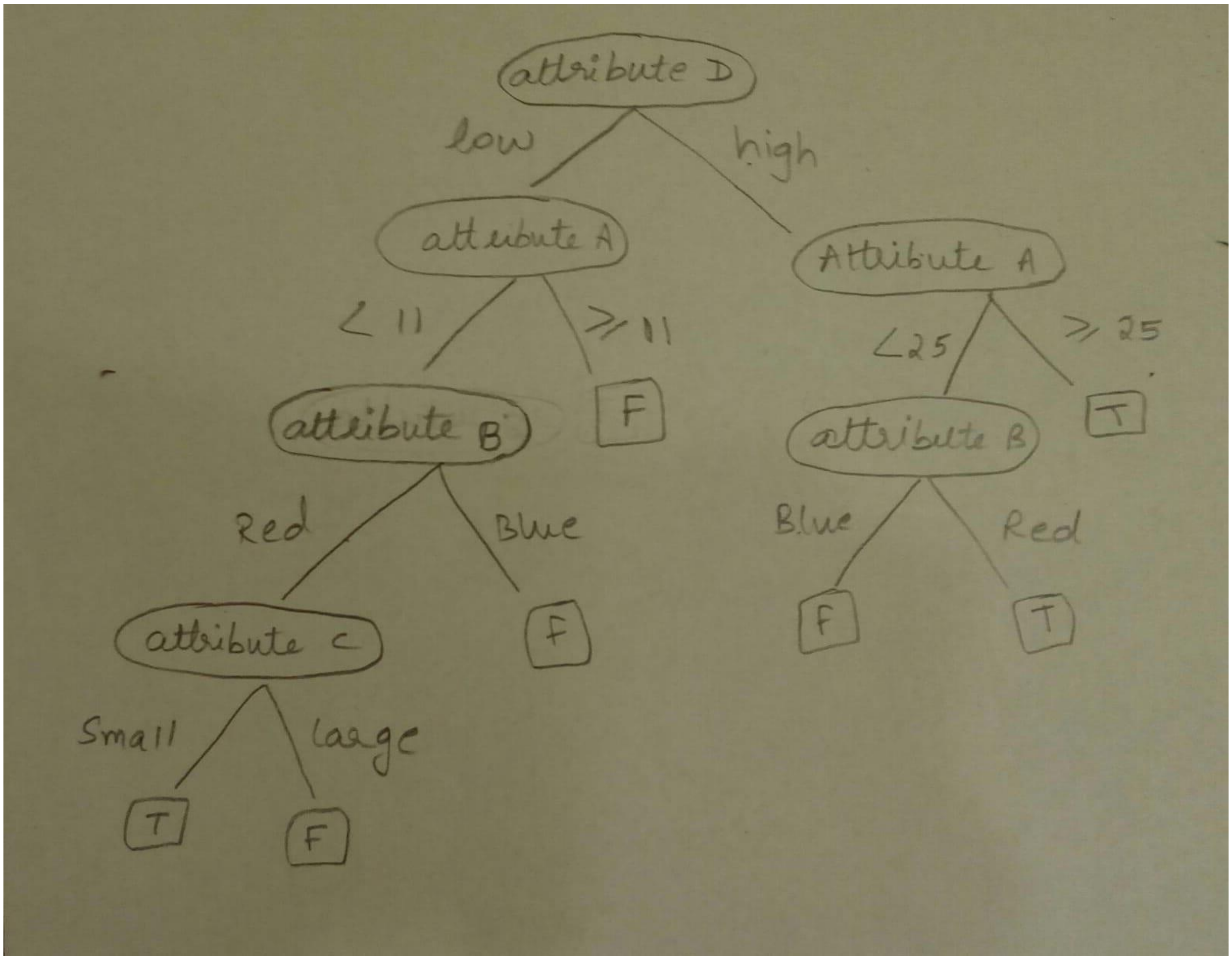
7	BLUE	LARGE	HIGH	COOL	F
17	BLUE	LARGE	HIGH	COOL	F

RED :

10	RED	SMALL	HIGH	COOL	T

As we could see, there is no need to further classify the data.

So the final tree is would look like :



(b) (10 points) Construct the tree manually using the Gini index, write down the computation process and show your tree step by step. (No partial credit)

NOTE :

All the calculation are performed in excel and we are attaching the excel workbook by name : hw2_q2_solution for further reference.

The notation split(x) says that records are splitted in the manner : the records < x into 1 category and >=x in another category

The gain of the Truth value (output variabe)of the dataset is : 0.494

To select a root node using Gini Index, here are the following steps :

All the possible attributes are tested and the one with lowest entropy is picked so that gain is high for it.

The gini indexes are given as following :

For attribute A :

Split(2) – 0.494

Split(3) – 0.467

Split(6) – 0.438

Split(7) – 0.487

Split(10) – 0.469

Split(11) - 0.491

Split(16) – 0.497

Split(17) – 0.456

Split(25) – 0.422

Split(27) – 0.464

Split(29) – 0.488

Split(33) – 0.491

Split(34) – 0.49

Split(36) -0.487

Split(45) – 0.429

Split(50) – 0.467

Attribute B : 0.4845

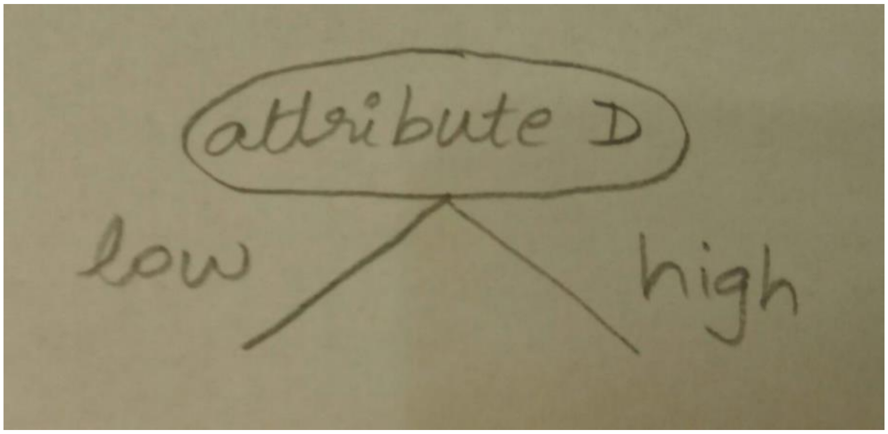
Attribute C : 0.4845

Attribute D : 0.2968

Attribute E : 0.4219

So we select Attribute D as root :

The tree as of now is :



Now we split the records as either low or high

Considering the left child of root for all records that are low :

LOW :

3	BLUE	SMALL	LOW	COOL	F
16	BLUE	LARGE	LOW	COOL	F
33	BLUE	SMALL	LOW	HOT	F
2	RED	LARGE	LOW	COOL	F
6	RED	SMALL	LOW	COOL	T
11	RED	SMALL	LOW	COOL	F
45	RED	SMALL	LOW	HOT	F
50	RED	LARGE	LOW	HOT	F

Now eliminating attribute D since we just already used and arranging the Feature A in ascending order :

2	RED	LARGE	LOW	COOL	F
3	BLUE	SMALL	LOW	COOL	F
6	RED	SMALL	LOW	COOL	T
11	RED	SMALL	LOW	COOL	F
16	BLUE	LARGE	LOW	COOL	F
33	BLUE	SMALL	LOW	HOT	F
45	RED	SMALL	LOW	HOT	F
50	RED	LARGE	LOW	HOT	F

Now, the gini Indexes are :

Split(3) – 0.214

Split(6) – 0.208

Split(11) – 0.167

Split(16) – 0.188

Split(33) – 0.2

Split(45) – 0.208

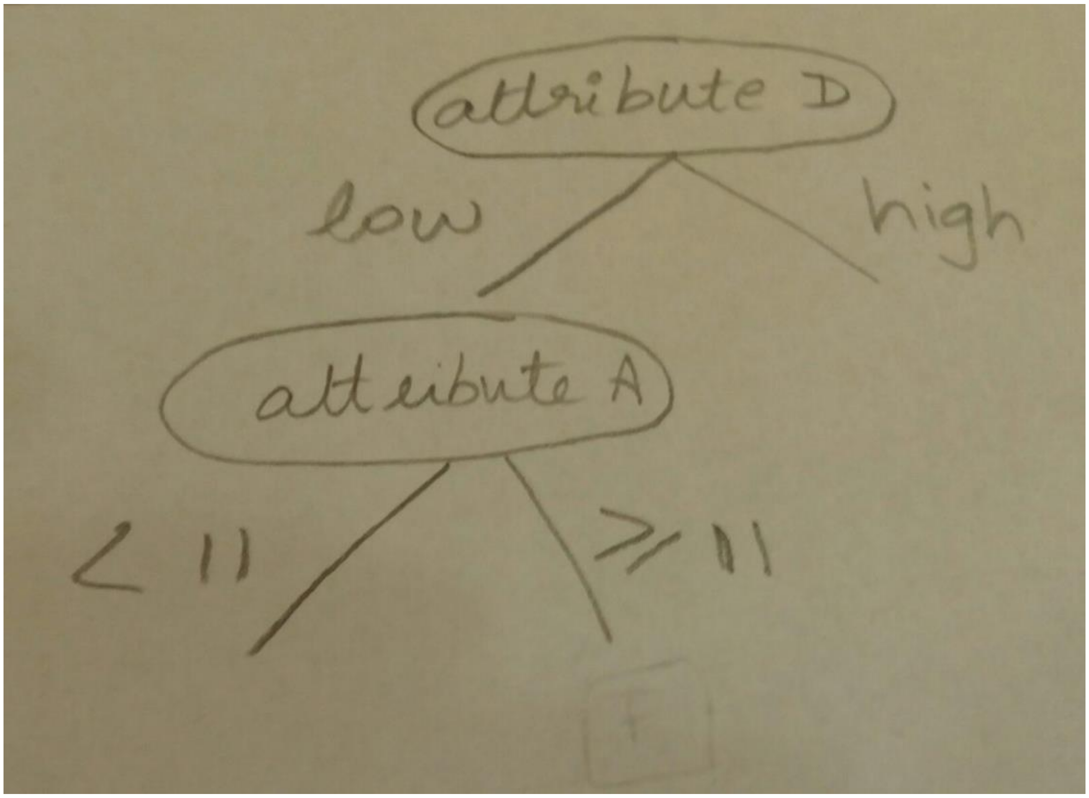
Split(50) – 0.214

Feature B - 0.2

Feature C -0.2

Feature E – 0.2

As the entropy is low for Feature A – by splitting at 11 :
The tree at this point is



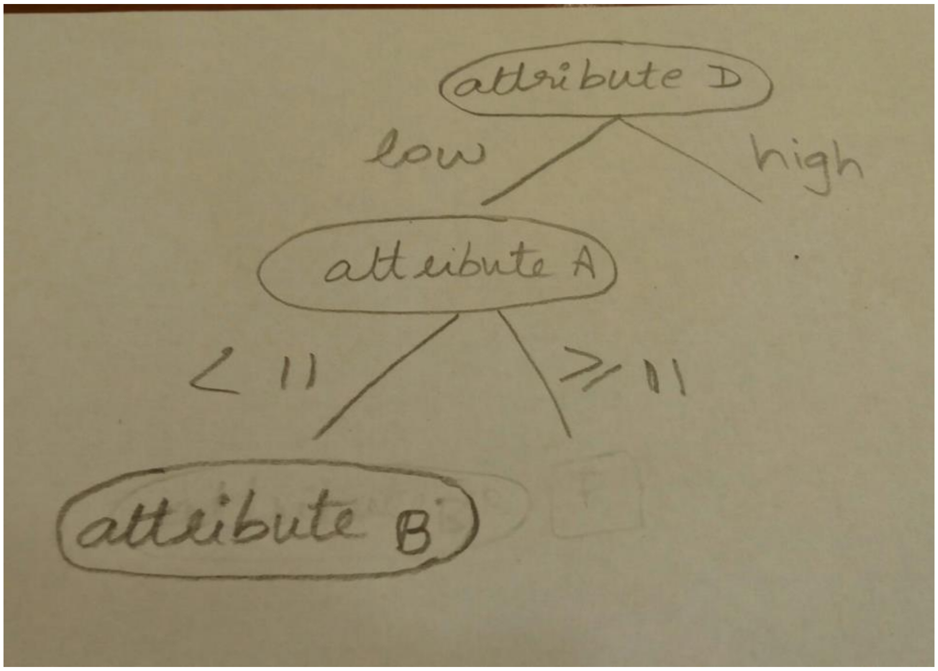
The records less than 11 are :

2	RED	LARGE	LOW	COOL	F
3	BLUE	SMALL	LOW	COOL	F
6	RED	SMALL	LOW	COOL	T

As we can see the attribute E has all values as Cool, so the only remaining attributes to consider are attribute B and attribute C.

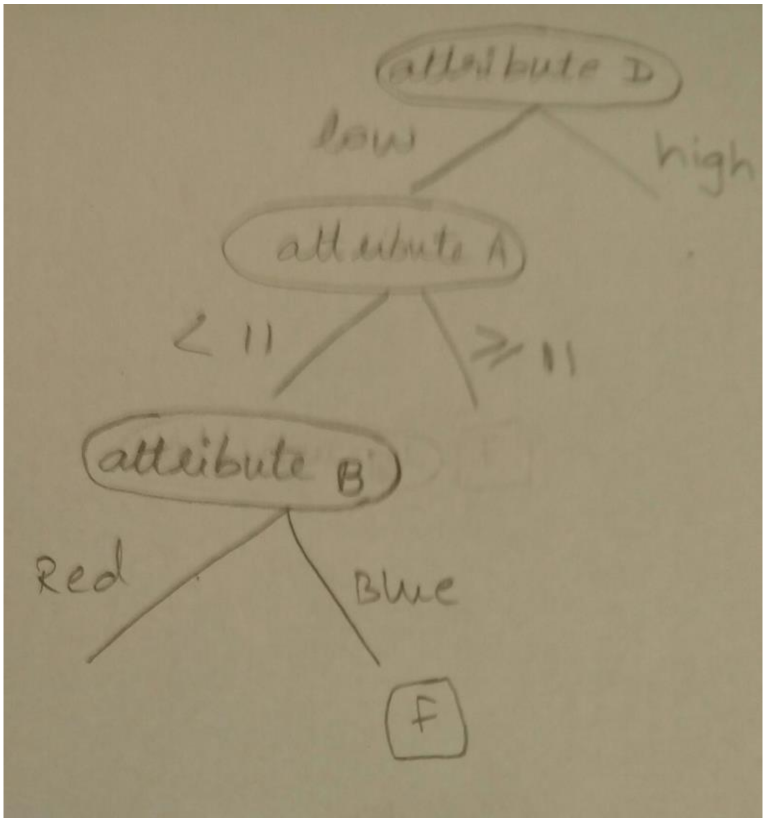
As ties favours left side, we choose attribute B.

The tree at this point is :



Now, attribute B values which are Blue are :

3	BLUE	SMALL	LOW	COOL	F

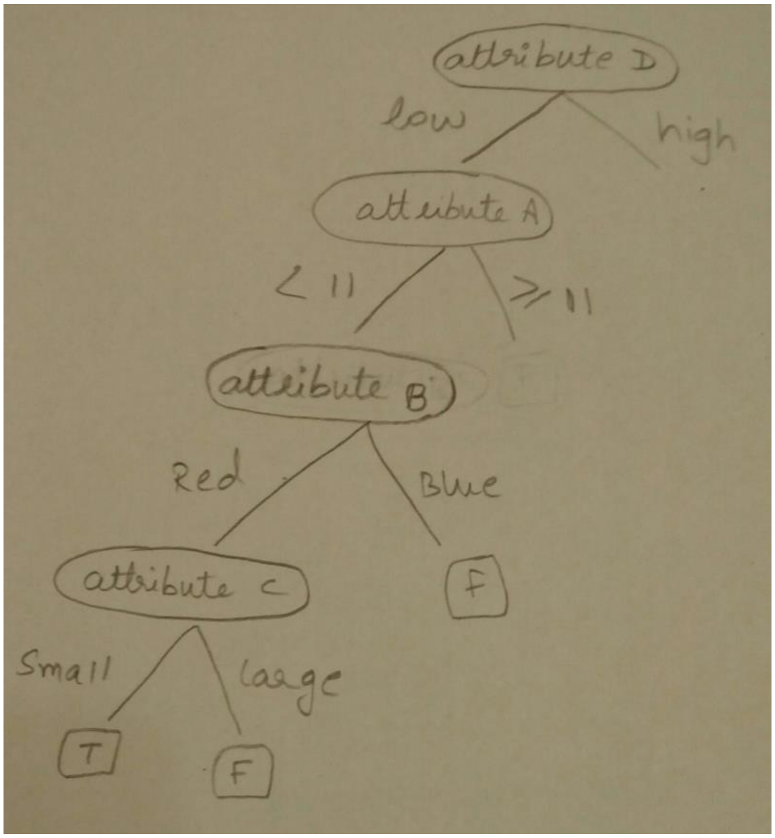


Now, attribute B values which are Red are :

2	RED	LARGE	LOW	COOL	F
6	RED	SMALL	LOW	COOL	T

As we can see they can be classified using attribute C :

The tree at this point is :

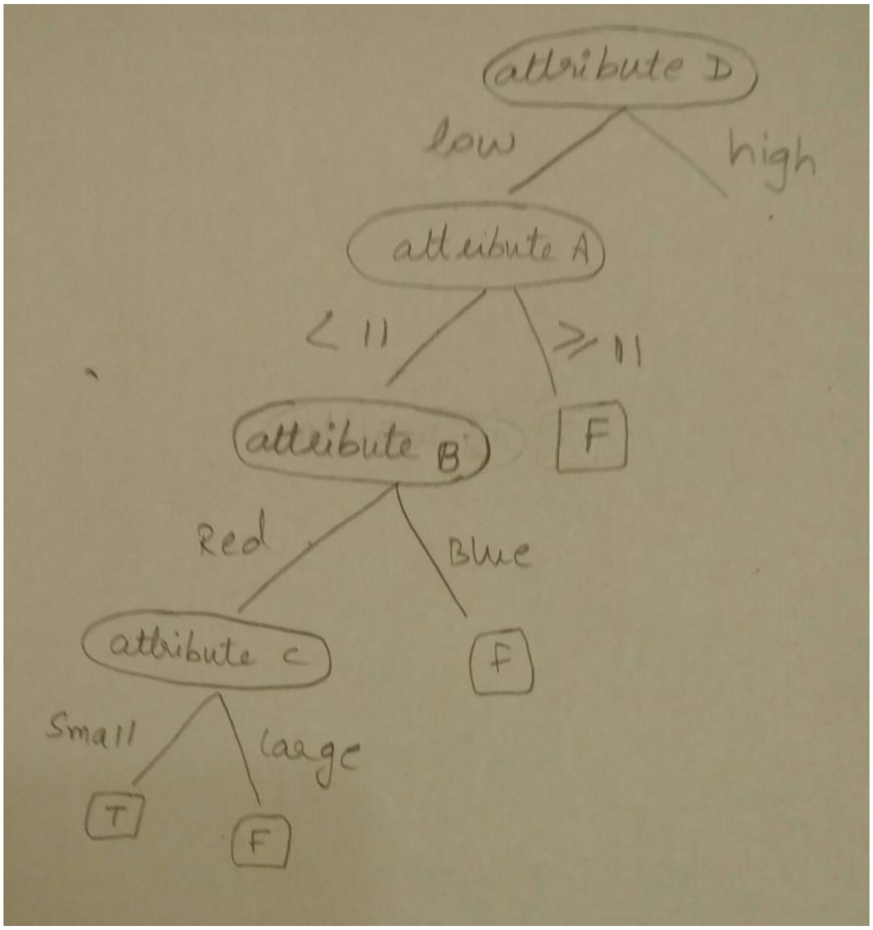


The records greater than or equal to 11 are :

11	RED	SMALL	LOW	COOL	F
16	BLUE	LARGE	LOW	COOL	F
33	BLUE	SMALL	LOW	HOT	F
45	RED	SMALL	LOW	HOT	F
50	RED	LARGE	LOW	HOT	F

As we can see, all are False, so no further classification is required.

The tree at this points is :



Considering the right branch of root for the records which are 'High' :

HIGH :

7	BLUE	LARGE	HIGH	COOL	F
10	RED	SMALL	HIGH	COOL	T
17	BLUE	LARGE	HIGH	COOL	F
25	RED	SMALL	HIGH	HOT	T
27	BLUE	LARGE	HIGH	HOT	T
29	BLUE	SMALL	HIGH	HOT	T
34	BLUE	LARGE	HIGH	HOT	T
36	RED	LARGE	HIGH	HOT	T

The gini indexes now for the attributes are :

Attribute A :

Split(10) – 0.214

Split(17) – 0.333

Split(25) – 0.167

Split(27) – 0.25

Split(29) – 0.3

Split(34) – 0.333

Split(26) – 0.357

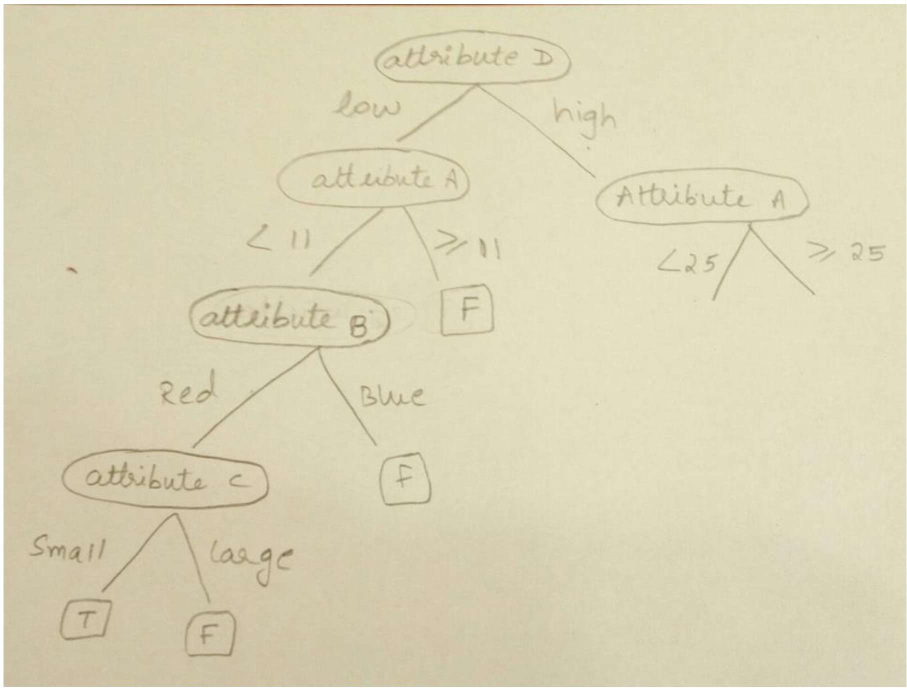
Feature B : 0.3

Feature C : 0.3

Feature E : 0.167

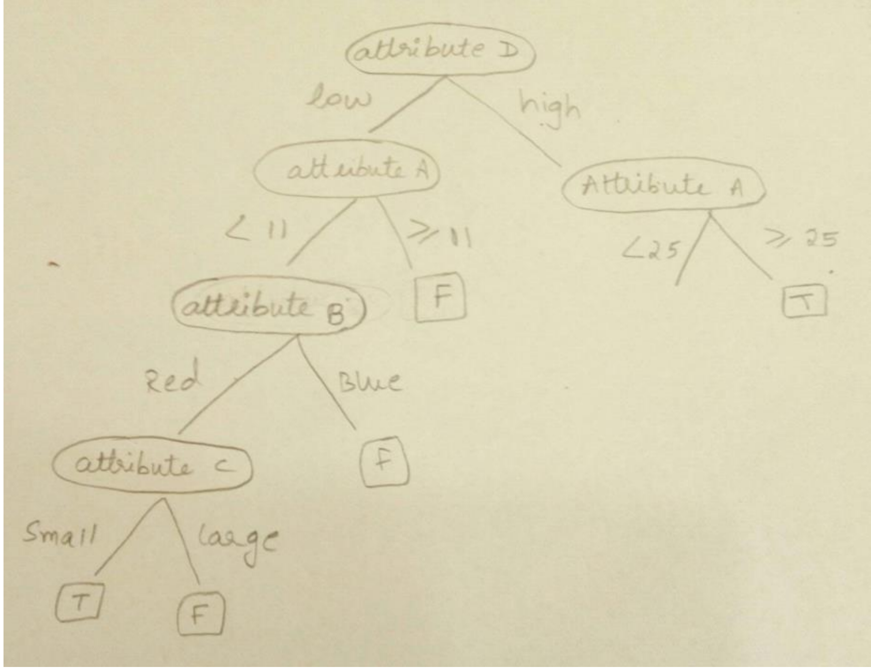
Since ties favour left side, we get attribute A as next node, splitting at 25

The tree at this point is :



The records which have attribute A greater than or equal to 25 are :

25	RED	SMALL	HIGH	HOT	T
27	BLUE	LARGE	HIGH	HOT	T
29	BLUE	SMALL	HIGH	HOT	T
34	BLUE	LARGE	HIGH	HOT	T
36	RED	LARGE	HIGH	HOT	T



Checking for values which are less than 25 are :

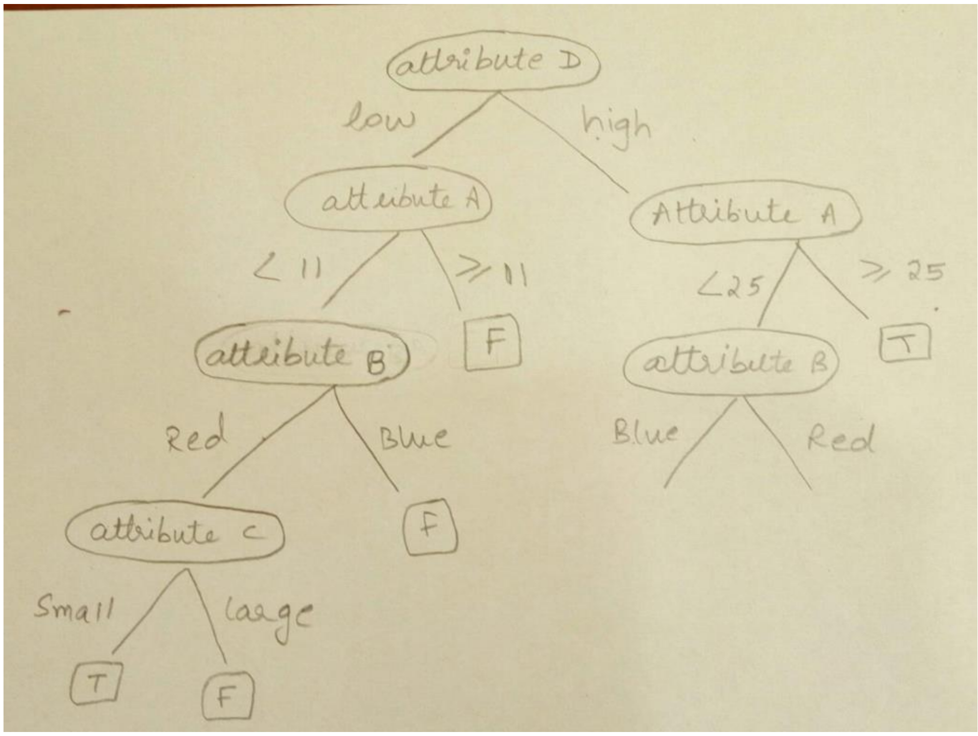
7	BLUE	LARGE	HIGH	COOL	F
10	RED	SMALL	HIGH	COOL	T
17	BLUE	LARGE	HIGH	COOL	F

Now, since attribute E is Cool for everything, we can select attribute B or C

As ties favour attribute B :

We select Attribute B as next node :

Tree at this point is :



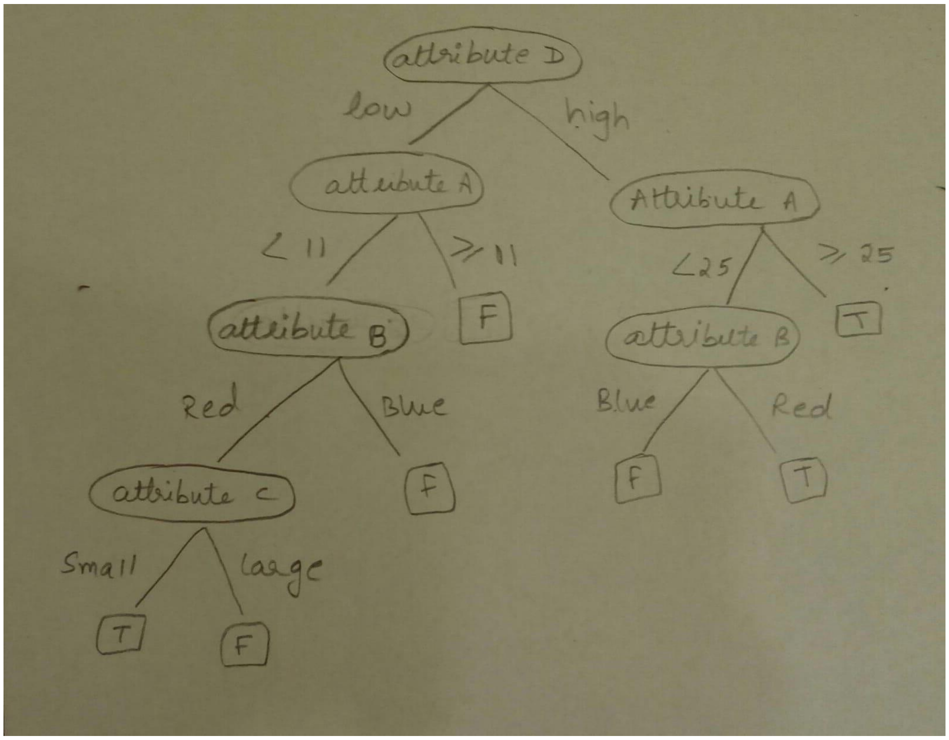
The records which are Blue for attribute B are :

7	BLUE	LARGE	HIGH	COOL	F
17	BLUE	LARGE	HIGH	COOL	F

The records which are Red for attribute B are :

10	RED	SMALL	HIGH	COOL	T

As there is no further classification is required, we can build the final tree :



If we observe, the tree is same for both Gini Index and Entropy as the algorithm used.

3. (30 points) [Evaluate Classifier][Song Ju] Sepsis is the leading cause of mortality in the United States. Septic shock, the most advanced complication of sepsis due to severe abnormalities of circulation and/or cellular metabolism, reaches a mortality rate as high as 50%. It is estimated that as many as 80% of sepsis deaths could be prevented with early diagnosis and intervention. To predict whether or not a patient has septic shock (Yes/No), consider using the decision tree shown in Figure 1 which involves Systolic Blood Pressure (SBP), Mean Arterial Pressure (MAP), and vasopressor (Vaso). We will focus on the sub-tree which splits on the attribute ”SBP” as shown in the red dashed region of Figure 1. Answer the following questions and show your work.

(a) (13 points) Post-pruning based on optimistic errors.

i. (4 points) Calculate the optimistic errors before splitting and after splitting using SBP respectively.

ii. (3 points) Based upon the optimistic errors, would the subtree be pruned or retained? If it is pruned, draw the resulting decision tree and use it for the next question; otherwise, use the original decision tree shown in Figure 1 for the next question.

iii. (6 points) Use the decision tree from (a)-(ii) above to classify the provided testing dataset ("hw2q3 test.csv"). Report the Accuracy, Recall, Precision, Specificity, Sensitivity, and F1 Measure.

(b) (13 points) Post-pruning based on pessimistic errors. When calculating pessimistic errors, each leaf node will add a factor of 0.5 to the error.

i. (4 points) Calculate the pessimistic errors before splitting and after splitting using SBP respectively.

ii. (3 points) Based on the pessimistic errors, would the subtree be pruned or retained? If it is pruned, draw the resulting decision tree and use it for the next question; otherwise, use the original decision tree shown in Figure 1 for the next question.

iii. (6 points) Use the decision tree from (b)-(ii) above to classify the provided testing dataset ("hw2q3 test.csv"). Report the Accuracy, Recall, Precision, Specificity, Sensitivity, and F1 Measure.

3.

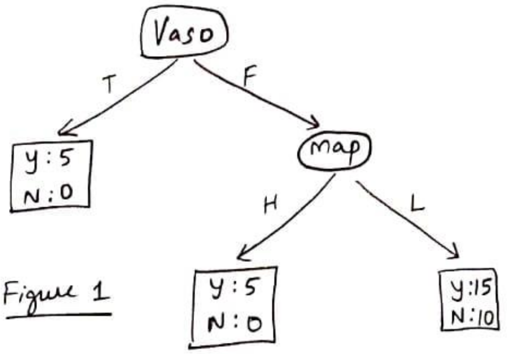


Figure 1

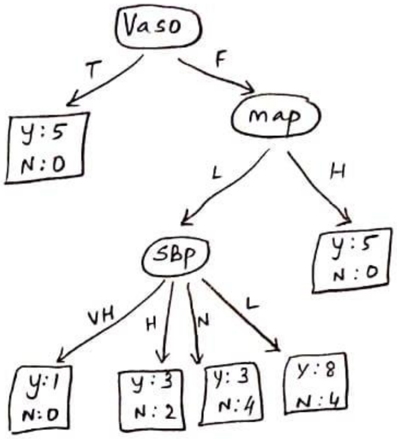


Figure -2

a) i) optimistic errors

before splitting = 10/35

after splitting = 9/35

ii) sub tree should not be pruned.

Because Opterror before split > Opt error after split

iii) Using figure 2 for prediction:

Vaso	MAP	SBP	sepsis shock	predicted	error
F	L	H	y	y	0
F	L	H	y	y	0
F	L	H	y	y	0
F	L	H	N	y	1
T	H	H	y	y	0
F	L	L	y	y	0
F	L	L	y	y	0
F	L	L	y	y	0
F	L	L	y	y	0
T	L	L	y	y	0
F	H	L	y	y	0
F	L	N	y	N	1
F	L	N	y	N	1
F	L	N	N	N	0
F	L	N	N	N	0
F	L	N	N	N	0
F	L	N	N	N	0
T	H	N	y	y	0
F	L	VH	y	y	0
T	L	VH	y	y	0

		Predicted class	
		yes	No
Actual class	yes	13 a TP	2 b FN
	No	1 c FP	4 d TN

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{17}{20}$$

$$\text{Precision}, (p) = \frac{a}{a+c} = \frac{13}{14}$$

$$\text{Recall}, (r) = \frac{a}{a+b} = \frac{13}{15}$$

$$\text{F1 measure } (F) = \frac{2rp}{r+p} = \frac{2a}{2a+b+c} = \frac{26}{29}$$

$$\text{sensitivity} = \frac{TP}{TP+FN} = \frac{13}{15}$$

$$\text{specificity} = \frac{TN}{TN+FP} = \frac{4}{5}$$

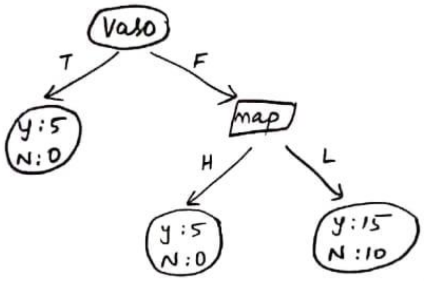
b) i) Pessimistic errors

before splitting $= [10 + 3(0.5)] / 35 = 0.328$

after splitting $= [9 + 6(0.5)] /35 = 0.343$

ii) sub tree should be pruned. Because error before split < error after split

Decision tree : Figure-3



iii)

VASO	MAP	SBP	SS	prediction	error
F	L	H	y	y	0
F	L	H	y	y	0
F	L	H	y	y	0
F	L	H	N	y	1
T	H	H	y	y	0
F	L	L	y	y	0
F	L	L	y	y	0
F	L	L	y	y	0
F	L	L	y	y	0
T	L	L	y	y	0
F	H	L	y	y	0
F	L	N	y	y	0
F	L	N	y	y	0
F	L	N	N	y	1
F	L	N	N	y	1
F	L	N	N	y	1
F	L	N	N	y	1
T	H	N	y	y	0
F	L	VH	y	y	0
T	L	VH	y	y	0

Using Figure - 3 : The following table predicts the error.

		Predicted class yes	Predicted class No
Actual class	yes	15 a TP	0 b FN
	No	5 c FP	0 d TN

Accuracy $= \frac{15}{20} = 3/4$

Precision, $p = \frac{15}{20} = \frac{3}{4}$

recall, $r = \frac{15}{15} = 1$

F_1 measure $= \frac{30}{35} = \frac{6}{7}$

Sensitivity = 1

specificity = 0

c.

	Optimistic DT (A)	Pessimistic DT (B)
Accuracy	85 %	75 %
Recall	86.67 %	100 %
Precision	92.86 %	75 %

Model B: the decision tree for pessimistic errors is better suitable for predicting septic shock disease. The decision making is highly conservative

TN = Correct Rejection
TP = Hit
FP = False Alarm
FN = Miss

TP : $15/20 = P(\text{Hit})$

FP : $5/20 = P(\text{False Alarm})$

Type II : β is critical in disease prediction.

The recall % is 100% Hence No patient is allowed to miss the septic shock prediction even if a normal patient is considered to be septic shock illness. Type II error is highly expensive and hence decision tree used in pessimistic decision making is better suitable for prediction.

4. (15 points) [Adaboost][Xi Yang] Consider the labeled data points in Figure 2, where ‘ ’ and ‘ ’ indicate class labels. We will use AdaBoost with decision stumps to train a classifier for the ‘ ’ and ‘ ’ labels. Each boosting iteration will select the stump that minimizes the weighted training error. Breaking ties by choosing ‘ ’. All of the data points start with uniform weights.

a) (4 points) In Figure 2, draw a decision boundary corresponding to the first decision stump that the algorithm would choose (the decision boundary should be either a vertical or horizontal straight line). Label the decision boundary as (1), also indicate the / sides of this boundary.

See the figure below

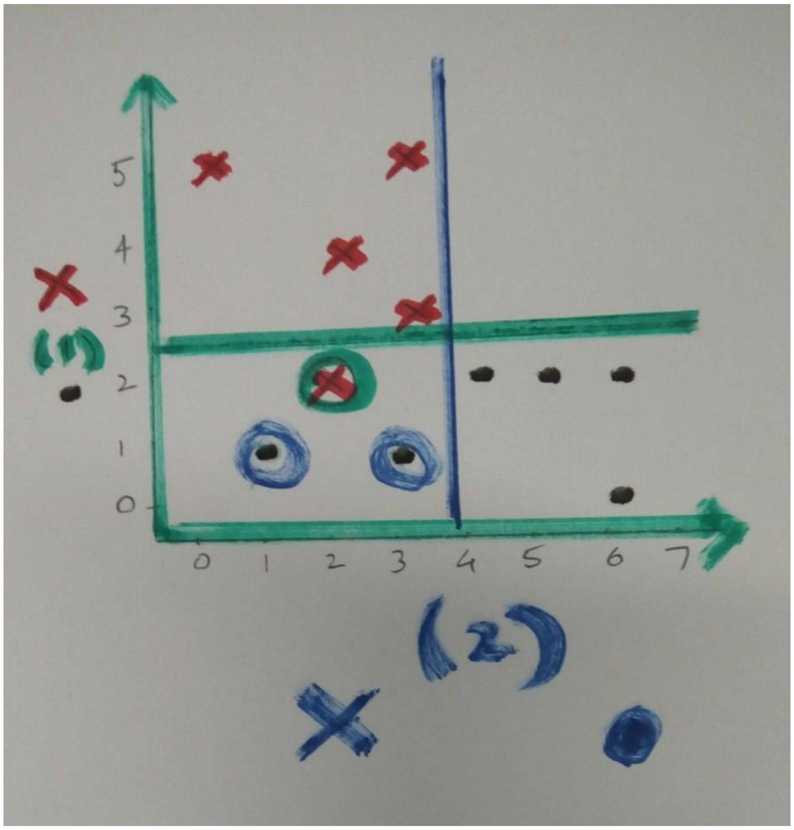
(b) (2 points) Circle the point(s) that have the highest weight after the first boosting iteration.

Refer to the figure below.

The green line (horizontal) represents the first decision boundary.

Above this line are ‘x’ points and below are the points ‘solid circles’

The ‘x’ which is circled with green color is the point that will have highest weight after the first boosting iteration.



(c) (5 points) After the labels have been reweighted in the first boosting iteration, what is the weighted error of the decision boundary (1)?

The initial weights are :

w_n (round 1) $= \frac{1}{N} = \frac{1}{11}$ (since there are 11 points)

- All points have equal weights
- No. of points misclassified : 1
- Weighted error =

$$\varepsilon = \frac{1}{11}\left[10 \times 0 + 1 \times 1\right] = \frac{1}{11} \times 1 = 0.09$$

- $\alpha = \frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right) = \frac{1}{2}\ln\left(\frac{1-0.09}{0.09}\right)$

4th contd:

$$\alpha = \frac{1}{2}\ln\left(\frac{0.91}{0.09}\right)$$

$$\alpha = \frac{1}{2}\ln(10.11) = \frac{1}{2} \times 2.31 = \underline{1.15}$$

classified correcty:

$$e^{-\alpha} = e^{-1.15} = 0.316$$

$$\text{weight} = 0.09 \times 0.316$$
$$= 0.02$$

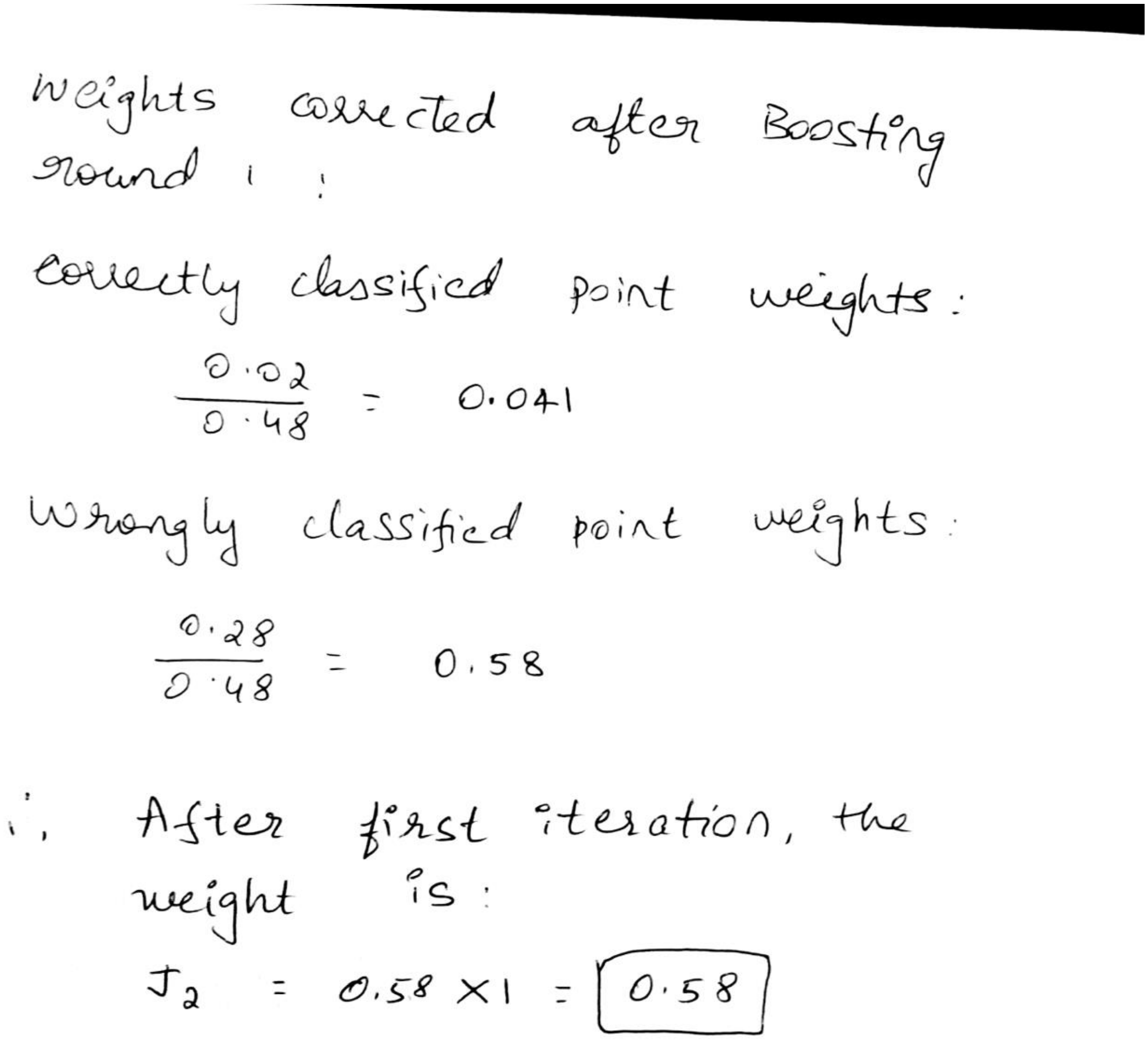
wrongly classified:

$$e^{\alpha} = e^{\Phi.15} = 3.15$$

$$\text{weight} = 0.09 \times 3.15$$
$$= 0.28$$

Normalized Weights:

$$10 \times 0.02 + 1 \times 0.28$$
$$= 0.2 + 0.28$$
$$= 0.48$$



Weights corrected after Boosting round 1 :

Correctly classified point weights:

$$\frac{0.02}{0.48} = 0.041$$

Wrongly classified point weights:

$$\frac{0.28}{0.48} = 0.58$$

$\therefore$ After first iteration, the weight is:

$$J_2 = 0.58 \times 1 = \boxed{0.58}$$

(d) (4 points) Draw the decision boundary corresponding to the second decision stump that the algorithm would choose (the decision boundary should be either a vertical or horizontal straight line). Label the decision boundary as (2), also indicate the / sides of this boundary. (Please display your answers for (a), (b) and (d) in a single figure.)

Refer to the figure above.

The second decision boundary is in blue color (vertical) straight line.

The left side of this line represents ‘x’ points and right side represent ‘solid circle’.

Although the question didn't ask, the 2 points (which are represented by solid circles) are circled in blue color

5. (20 points) [Na¨ıve Bayes + Decision Tree] [Ruth Okoilu] For this exercise, use the provided 'hw2q5.csv' which contains 24 data points. It has six attributes: each data point will be referred to using the first column "Id" and we will use columns 2-5 to predict the final column "Class" (whether or not a patient should have contact lens).

(a) (15 points) Compare the performance of two classifiers: Na¨ıve Bayes (NB) vs. Decision Tree (DT) using 5-fold cross-validation (CV) and report their 5-fold CV accuracy. For the ith fold, the testing dataset is composed of all the data points whose (Id mod 5 = i−1). Follow the lecture's code to build your decision trees except that multiple-way splitting is allowed and use Information Gain (IG) to select the best attribute. In the case of ties, break ties in favor of the leftmost feature. For each fold, show the induced Na¨ıve Bayes and DT models.

The testing data in this fold : 1 is :

```
   Id    patient age spectacle prescription astigmatic tear production rate
0   1        young              myope       no          reduced
5   6        young       hypermetrope       no           normal
10  11  pre-presbyopic              myope      yes          reduced
15  16  pre-presbyopic       hypermetrope      yes           normal
20  21     presbyopic       hypermetrope       no          reduced


  Class
0    No
5   Yes
```

```
10    No
15    No
20    No
Decision Tree
Predicted Values :
['No' 'Yes' 'No' 'Yes' 'No']
Actual values :
0      No
5     Yes
10     No
15     No
20     No
Name: Class, dtype: object
Accuracy
0.8
Classification report
           precision    recall  f1-score   support


        No       1.00      0.75      0.86         4
       Yes       0.50      1.00      0.67         1


avg / total       0.90      0.80      0.82         5


Confusion Matrix :
```

```
[[3 1]
 [0 1]]
```

The Decision Tree is available in the form of pdf : DT 1

Naive Bayes

```
Predictions
['No' 'Yes' 'No' 'Yes' 'No']
Actual values
0     No
5    Yes
10    No
15    No
20    No
Name: Class, dtype: object
Accuracy
0.8
Classification report
             precision    recall  f1-score   support

         No       1.00      0.75      0.86         4
        Yes       0.50      1.00      0.67         1

avg / total       0.90      0.80      0.82         5

Confusion Matrix :
```

```
[[3 1]

 [0 1]]

Post Probabilities for No and Yes for test data:

[[1.00000e+00 0.00000e+00]

 [1.20000e-05 9.99988e-01]

 [1.00000e+00 0.00000e+00]

 [1.40000e-05 9.99986e-01]

 [1.00000e+00 0.00000e+00]]

The testing data in this fold :  2 is :

   Id     patient age spectacle prescription astigmatic tear production rate  \

1    2          young                myope        no             normal

6    7          young          hypermetrope       yes            reduced

11  12  pre-presbyopic                myope       yes            normal

16  17      presbyopic                myope        no            reduced

21  22      presbyopic          hypermetrope        no            normal


   Class

1    Yes

6     No

11   Yes

16    No

21   Yes

Decision Tree

Predicted Values :
```

```
['Yes' 'No' 'No' 'No' 'No']

Actual values :

1     Yes

6      No

11    Yes

16     No

21    Yes

Name: Class, dtype: object

Accuracy

0.6

Classification report

             precision    recall  f1-score   support



         No       0.50      1.00      0.67         2

        Yes       1.00      0.33      0.50         3


avg / total       0.80      0.60      0.57         5


Confusion Matrix :

[[2 0]

 [2 1]]
```

The Decision Tree is available in the form of pdf : DT 2

Naive Bayes

Predictions

```
['Yes' 'No' 'Yes' 'No' 'Yes']
Actual values
1     Yes
6      No
11    Yes
16     No
21    Yes
Name: Class, dtype: object
Accuracy
1.0
Classification report
             precision    recall  f1-score   support


         No       1.00      1.00      1.00         2
        Yes       1.00      1.00      1.00         3


avg / total       1.00      1.00      1.00         5


Confusion Matrix :
[[2 0]
 [0 3]]
Post Probabilities for No and Yes for test data:
[[1.30000e-05 9.99987e-01]
 [1.00000e+00 0.00000e+00]
 [3.70000e-05 9.99963e-01]

 [1.00000e+00 0.00000e+00]

 [2.90000e-05 9.99971e-01]]

The testing data in this fold :  3 is :

    Id     patient age spectacle prescription astigmatic tear production rate  \

2    3          young                 myope       yes            reduced

7    8          young          hypermetrope       yes             normal

12  13  pre-presbyopic          hypermetrope        no            reduced

17  18      presbyopic                 myope        no             normal

22  23      presbyopic          hypermetrope       yes            reduced


   Class

2     No

7    Yes

12    No

17    No

22    No

Decision Tree

Predicted Values :

['No' 'No' 'No' 'Yes' 'No']

Actual values :

2      No

7     Yes

12     No
```

```
17    No
22    No
Name: Class, dtype: object
Accuracy
0.6
Classification report
             precision    recall  f1-score   support


         No       0.75      0.75      0.75         4
        Yes       0.00      0.00      0.00         1


avg / total       0.60      0.60      0.60         5


Confusion Matrix :
[[3 1]
 [1 0]]
```

The Decision Tree is available in the form of pdf : DT 3

Naive Bayes

```
Predictions
['No' 'Yes' 'No' 'Yes' 'No']
Actual values
2     No
7    Yes
12    No
```

```
17    No
22    No
Name: Class, dtype: object
Accuracy
0.8
Classification report
             precision    recall  f1-score   support

         No       1.00      0.75      0.86         4
        Yes       0.50      1.00      0.67         1

avg / total       0.90      0.80      0.82         5

Confusion Matrix :
[[3 1]
 [0 1]]
Post Probabilities for No and Yes for test data:
[[1.00000e+00 0.00000e+00]
 [1.80000e-05 9.99982e-01]
 [1.00000e+00 0.00000e+00]
 [6.00000e-06 9.99994e-01]
 [1.00000e+00 0.00000e+00]]
The testing data in this fold :  4 is :
    Id    patient age spectacle prescription astigmatic tear production rate  \
3    4          young                myope       yes             normal

8    9  pre-presbyopic                myope        no            reduced

13  14  pre-presbyopic          hypermetrope        no            normal

18  19      presbyopic                myope       yes           reduced

23  24      presbyopic          hypermetrope       yes            normal


   Class

3    Yes

8     No

13   Yes

18    No

23    No

Decision Tree

Predicted Values :

['Yes' 'No' 'No' 'No' 'Yes']

Actual values :

3     Yes

8      No

13    Yes

18     No

23     No

Name: Class, dtype: object

Accuracy

0.6
```

```
Classification report

             precision    recall  f1-score   support


         No       0.67      0.67      0.67         3

        Yes       0.50      0.50      0.50         2


avg / total       0.60      0.60      0.60         5


Confusion Matrix :

[[2 1]

 [1 1]]

The Decision Tree is available in the form of pdf : DT 4

Naive Bayes

Predictions

['Yes' 'No' 'Yes' 'No' 'Yes']

Actual values

3     Yes

8      No

13    Yes

18     No

23     No

Name: Class, dtype: object

Accuracy

0.8
```

```
Classification report
             precision    recall  f1-score   support


         No       1.00      0.67      0.80         3
        Yes       0.67      1.00      0.80         2


avg / total       0.87      0.80      0.80         5


Confusion Matrix :
[[2 1]
 [0 2]]
Post Probabilities for No and Yes for test data:
[[7.00000e-06 9.99993e-01]
 [1.00000e+00 0.00000e+00]
 [1.50000e-05 9.99985e-01]
 [1.00000e+00 0.00000e+00]
 [1.60000e-05 9.99984e-01]]
The testing data in this fold :  5 is :
    Id     patient age spectacle prescription astigmatic tear production rate  \
4    5           young         hypermetrope         no              reduced
9   10  pre-presbyopic                myope         no               normal
14  15  pre-presbyopic         hypermetrope        yes              reduced
19  20      presbyopic                myope        yes               normal
```

```
   Class
4     No
9    Yes
14    No
19   Yes
Decision Tree
Predicted Values :
['No' 'No' 'No' 'Yes']
Actual values :
4      No
9     Yes
14     No
19    Yes
Name: Class, dtype: object
Accuracy
0.75
Classification report
             precision    recall  f1-score   support


         No       0.67      1.00      0.80         2
        Yes       1.00      0.50      0.67         2


avg / total       0.83      0.75      0.73         4
```

```
Confusion Matrix :
[[2 0]
 [1 1]]
```

The Decision Tree is available in the form of pdf : DT 5

Naive Bayes

```
Predictions
['No' 'Yes' 'No' 'Yes']
Actual values
4      No
9     Yes
14     No
19    Yes
Name: Class, dtype: object
Accuracy
1.0
Classification report
             precision    recall  f1-score   support


         No       1.00      1.00      1.00         2
        Yes       1.00      1.00      1.00         2


avg / total       1.00      1.00      1.00         4


Confusion Matrix :
```

```
[[2 0]

 [0 2]]

Post Probabilities for No and Yes for test data:

[[1.00000e+00 0.00000e+00]

 [3.40000e-05 9.99966e-01]

 [1.00000e+00 0.00000e+00]

 [4.30000e-05 9.99957e-01]]

The accuacies for the 5 folds of Decision Tree is :

[0.8, 0.6, 0.6, 0.6, 0.75]

Average for Decision Tree is

0.67

The accuracies in the 5 folds of Naive bayes are

[0.8, 1.0, 0.8, 0.8, 1.0]

Average for Naive Bayes is

0.8800000000000001
```

(b) (5 points) Based on the 5-fold CV accuracy from (a), which classifier, NB or DT, would you choose? Report your final model for the selected classifier. Show your work. No Partial Credit.

The accuracies of Decision tree and Naïve Bayes are respectively :

```
[ 0.8   0.6   0.6   1.    0.75]

[ 0.8  1.   0.8  0.8  1. ]
```

The average for them is:
DT 0.79
NB 0.88

As per the results of accuracies, Naïve Bayes shows to be a better classifier considering accuracies.
Since we are using same data for training and testing in both the models in each fold, the model accuracies are comparable.

Only in 4th fold, the accuracy for Decision tree is higher. For the rest of the folds, the accuracy of Naïve Bayes is good.
The final model – naïve Bayes is trained using the entire dataset.
Here's the model :

Final Model : Naïve Bayes

The values here represent No and Yes probabilities for the given record.

Totally there are 24 records.

```
[[1.00000000e+00 0.00000000e+00]

 [7.49178219e-06 9.99992508e-01]

 [1.00000000e+00 0.00000000e+00]

 [1.07266580e-05 9.99989273e-01]

 [1.00000000e+00 0.00000000e+00]

 [1.07266580e-05 9.99989273e-01]

 [1.00000000e+00 0.00000000e+00]

 [1.53582991e-05 9.99984642e-01]

 [1.00000000e+00 0.00000000e+00]

 [1.25284850e-05 9.99987472e-01]

 [1.00000000e+00 0.00000000e+00]

 [1.79381192e-05 9.99982062e-01]

 [1.00000000e+00 0.00000000e+00]

 [1.79381192e-05 9.99982062e-01]

 [1.00000000e+00 0.00000000e+00]

 [2.56835020e-05 9.99974316e-01]
 [1.00000000e+00 0.00000000e+00]

 [1.16724405e-05 9.99988328e-01]

 [1.00000000e+00 0.00000000e+00]

 [1.67124522e-05 9.99983288e-01]

 [1.00000000e+00 0.00000000e+00]

 [1.67124522e-05 9.99983288e-01]

 [1.00000000e+00 0.00000000e+00]

 [2.39286249e-05 9.99976071e-01]]
```

Accuracy for final model using training data

0.875

6. (15 points) [KNN + CV] [Ruth Okoilu] Consider the following dataset with two real-valued inputs x1 and x2 and a binary output class y shown in Table 1. Each data point will be referred to using the first column ”ID” in the following. Use KNN with unweighted Euclidean distance to predict the class y.

(a) (2 points) What are the 3 nearest neighbors for data points 5 and 10 respectively. (No partial credit).

The nearest 3 neighbors of 5th and 10th point are respectively :

```
 [[2 10 3]]

 [[3 2 5]]
```

(b) (5 points) What is the leave-one-out cross-validation error of 1-NN on this dataset? (No partial credit).

LOOCV : 1

Predicted ['P']

```
Actual Class    N
Name: 0, dtype: object
LOOCV :  2
Predicted  ['N']
Actual Class    N
Name: 1, dtype: object
LOOCV :  3
Predicted  ['N']
Actual Class    P
Name: 2, dtype: object
LOOCV :  4
Predicted  ['N']
Actual Class    P
LOOCV :  5
Predicted  ['N']
Actual Class    N
Name: 4, dtype: object
LOOCV :  6
Predicted  ['P']
Actual Class    P
Name: 5, dtype: object
LOOCV :  7
Predicted  ['P']
Actual Class    N
```

```
Name: 6, dtype: object

LOOCV :  8

Predicted  ['N']

Actual Class    P

Name: 7, dtype: object

LOOCV :  9

Predicted  ['N']

Actual Class    P

Name: 8, dtype: object

LOOCV :  10

Predicted  ['P']

Actual Class    N

Name: 9, dtype: object

The leave-one-out cross-validation error of 1-NN on this dataset is  70.0
```

(c) (5 points) What is the 5-fold cross-validation error of 3-NN on this dataset? For the ith fold where i = 1,2,3,4,5, the testing dataset is composed of all the data points whose (ID mod 5 = i−1). (No partial credit).

```
Fold 1

Predicted Values ['P' 'P']

Actual Values   Class

0     N

5     P

Fold 2

Predicted Values ['N' 'P']

Actual Values   Class
```

```
1    N
6    N
Fold 3
Predicted Values ['N' 'N']
Actual Values   Class
2    P
7    P
Fold 4
Predicted Values ['P' 'N']
Actual Values   Class
3    P
8    P
Fold 5
Predicted Values ['P' 'P']
Actual Values   Class
4    N
9    N
```

As we can see, there are 7 points whose predicted value is not equal to actual value.

Therefore the error is 7/10 = 0.7 or 70%

(d) (3 points) Based on the results of (b) and (c), can we determine which is a better classifier, 1-NN or 3-NN? Why? (Answers without a correct justification will get zero points.)

Since the two classifiers : 1-NN and 3-NN are not tested with the same training and test set, we cannot compare them.

Although the data set used is same, at every iteration in LOOCV (10 iterations), the training and testing set used are different.

While in 3-NN, we used 5 fold cross validation. This mean that in every fold the training and test set used are not equivalent to those used in 1-NN model's LOOCV .

Thus, we cannot compare which classifier is better although their accuracies are same.

In case, comparison is apt, because training data is same, the 2 models are good enough as their accuracies are same.