Assignment 2: Report
Pankil Kalra, 2018061

QUESTION 1

a)

PCA or Principal Component Analysis is a method of dimensionality Reduction(feature extraction). It works by projecting the data into a smaller space(into principal comonences) such that the variance of the projected data is maximum.

Standardising the dataset before performing PCA is important.

PCA works by computing the covariance matrix of the input dataset, then computing the eigenvector and eigenvalue of the covariance vector. These eigenvectors and eigenvalues are used to identify the principal components.

Sklearn's implementation uses SVD.

b)

SVD or Singular Value Decomposition is another dimensionality reduction method (feature selection).

It works by dividing the input matrix into $A = USV^{T.}$

Where U and V are orthoganl matrixes which orthonormal eigenvectors chosen from AA^T and A^TA respectively.

S is a diagonal matrix with r elements. These r elements are equal to the root of the positive eigenvalues of AA^T or A^TA.

c)

TSNE or t-distributed stochastic neighbor embedding is an algorithm for visualisation of data.

It is utilised to visualise data upto a few dimentions. A probability distribution is constructed such that similar objects are assigned higher probability while dissimiar points are given lower probability. At the end, points which are close to each other are close in space and they are far away from points that are different from them.

d)

Training frequencies:
320
395
314
339
333
318
353

345
328
315

Testing frequencies:
80
99
79
85
83
80
88
86
82
78

Training percentages:
0.09523809523809523
0.11755952380952381
0.09345238095238095
0.10089285714285715
0.09910714285714285
0.09464285714285714
0.10505952380952381
0.10267857142857142
0.09761904761904762
0.09375

Testing percentages:
0.09523809523809523
0.11785714285714285
0.09404761904761905
0.10119047619047619
0.0988095238095238
0.09523809523809523
0.10476190476190476
0.10238095238095238
0.09761904761904762
0.09285714285714286

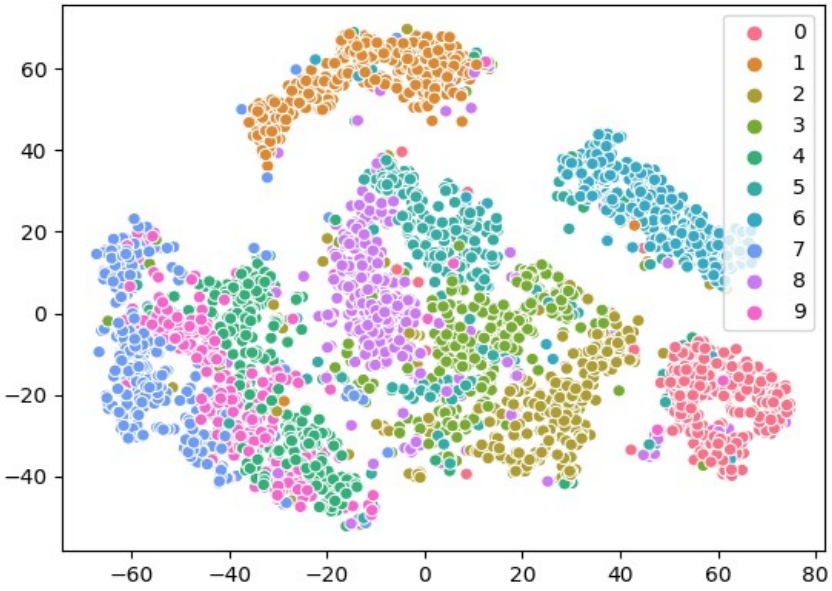
Clearly as seen from the above values, the **percentage** of a class variable is **same** in training and testing data. This is due to stratified sampling.

e)

Number of dimensions obtained after performing PCA is 25.

Applying PCA
Testing accuracy: 0.8714285714285714

TSNE Plot:

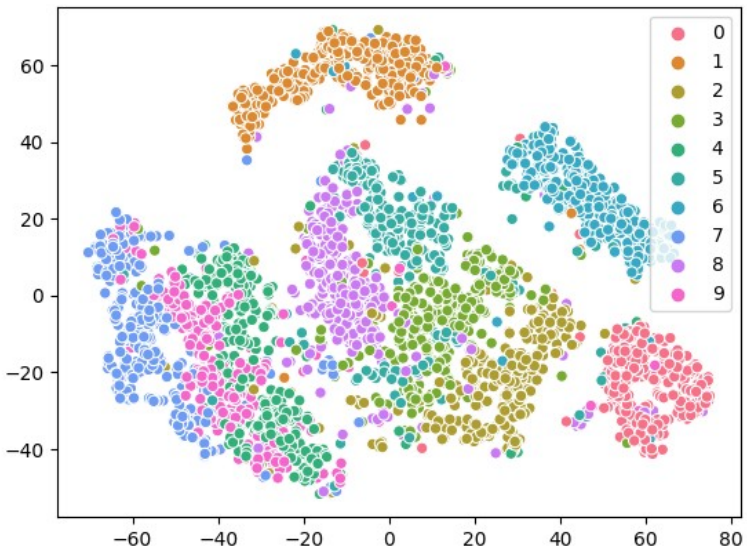


We can clearly see the separated clusters of different classes.

f)
Number of features left after performing SVD is 25.

Applying SVD
Testing accuracy: 0.8785714285714286

TSNE PLOT:
We can clearly see the separated clusters of different classes.



g)

Accuracy obtained after SVD is a little more than after applying PCA. There is not much difference in the accuracies.

Sklearn's implementation of PCA actually uses SVD in its working. Applying SVD was faster than appplying PCA but the results obtained are similar. For this dataset SVD should be preferred as it is faster than PCA and gives similar results.

QUESTION 2

```
pankil@pankil-ThinkPad-X390:~/Desktop/sem5/ML/assignments/assignment2$ python3 Q
2.py
Bias: -0.04092601745830393
Variance: 0.018504714004347295
MSE: 0.020181329524864802
MSE - Bias^2 - Variance: 1.6766155201096422e-06
```

Number of bootstrap samples taken: 1000
Size of each bootstrap sample: 8000 rows

a)

Bias: -0.04092601745830393
Variance: 0.018504714004347295

b)

MSE: 0.020181329524864802
MSE - Bias^2 - Variance: 1.6766155201096422e-06

Mse – Bias^2 – Variance is equal to Noise^2. Observing the value we can say that noise in the recorded data is close to zero (of the order 10^-6).

QUESTION 3

DATASET A

a)

```
Grid Search gives optimal max depth: 11
Applying KfoldCV using DT classifier and max_depth 11 (on 4 number of folds
DT: Training accuracy: 0.9835317460317461 , Validation accuracy 0.7342261904761
905

Applying KfoldCV using GNB classifier(on 4 number of folds
GNB: Training accuracy: 0.5987103174603174 , Validation accuracy 0.554166666666
6667
```

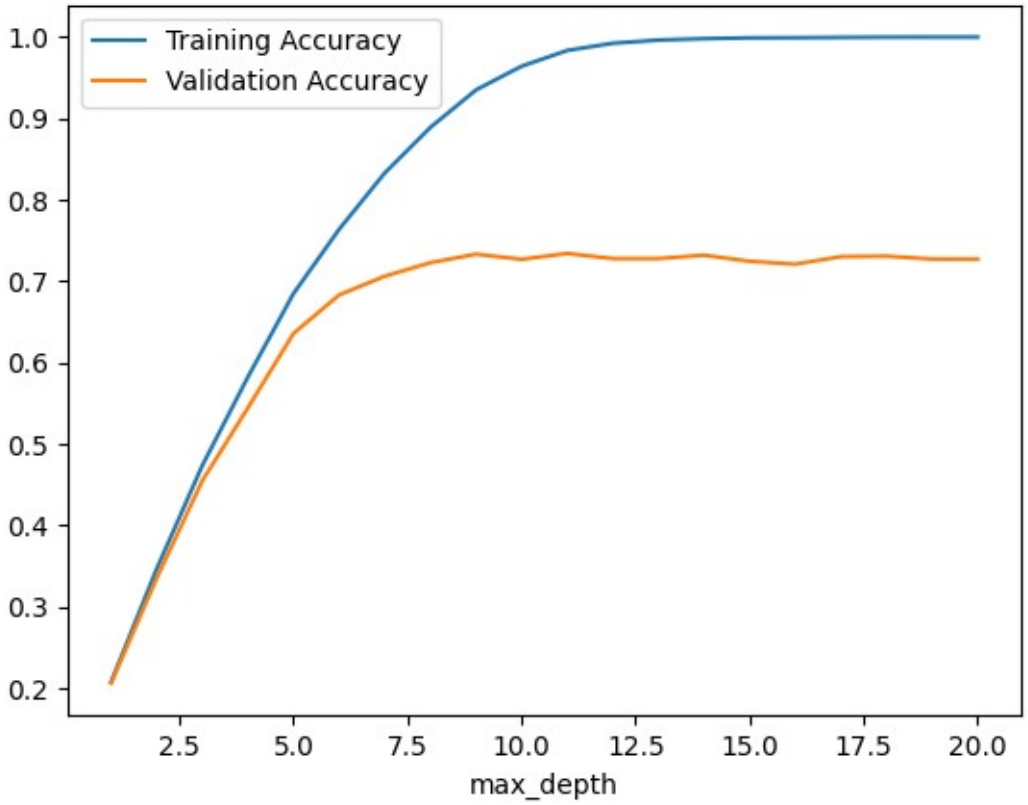
Grid search was performed for values of depth from 1 to 20.

Optimal depth of decision tree for grid search came out to be 11.

After applying K Fold Cross validation on depth 11, the validation accuracy of DT was found to be 0.7342261904761905.

After applying K Fold Cross validation, the validation accuracy of GNB was found out to be 0.5541666666666667.

b)



After the value of 10 for max_depth, the validation accuracy of DT on the dataset is more or less constant. The model improved as the max_depth increased from 1 to 10 but after that the performance of the model stagnated. The model after max_depth 10 is overfitting on training data as training accuracy is still increasing.

c)
DT performed significantly better than GNB.

The best fold of DT with max_depth = 11 was saved and then loaded.

```
Clearly best bodel is DT on depth 11
Running KfoldCV again and saving the best model
Loading best model

Using DT,Test accurancy: 0.7380952380952381
```

Accuracy on test data was found to be: 0.7380952380952381.

The best fold of GNB was saved and then loaded.

```
Using GNB,Test accurancy: 0.6
```

Accuracy on test data was found to be 0.6.

d)

For Decision Tree best fold model:

Confusion Matrix:

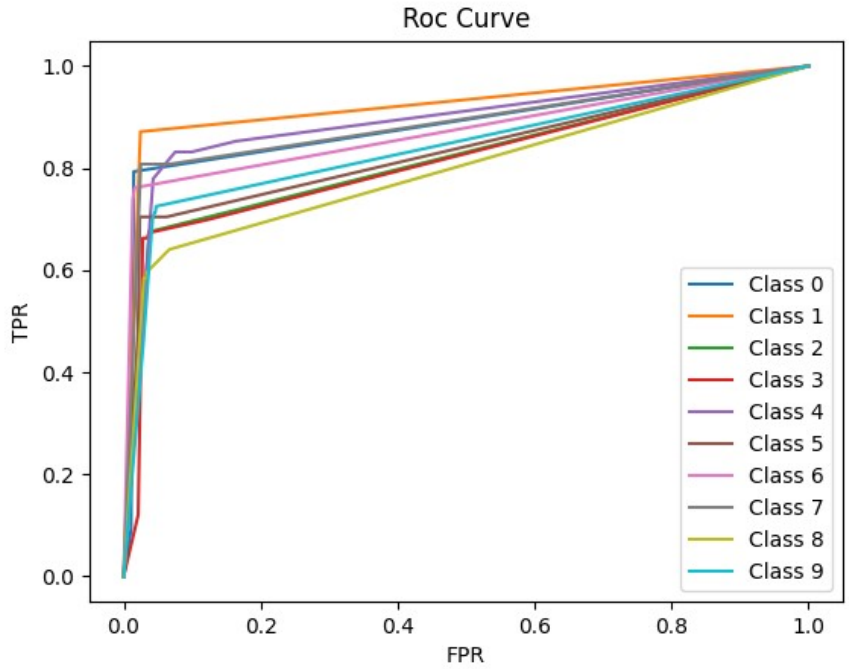
```
[[69  0  3  2  3  4  0  2  3  1]
 [ 0 88  3  1  1  1  2  5  0  0]
 [ 1  1 46  5  2  1  2  4  4  2]
 [ 3  0  5 55  3  4  0  0  4  9]
 [ 0  2  3  4 74  0  1  3  0  8]
 [ 2  1  4  3  1 50  2  0  4  4]
 [ 1  2  3  1  7  2 67  0  4  1]
 [ 0  0  5  1  4  3  0 63  0  2]
 [ 3  7  4  5  6  2  4  1 52  5]
 [ 1  5  0  4  5  1  1  4  3 56]]
```

Accuracy: 0.7380952380952381

Macro Precision: 0.7365831795829215
Macro Recall: 0.7340010095144687
Macro F1: 0.7352898275576883

Micro Precision: 0.7380952380952381
Micro Recall: 0.7380952380952381
Micro F1: 0.7380952380952381

ROC Curve:



For Gaussian Naive Bayes Classifier best fold model:

Confusion Matrix:

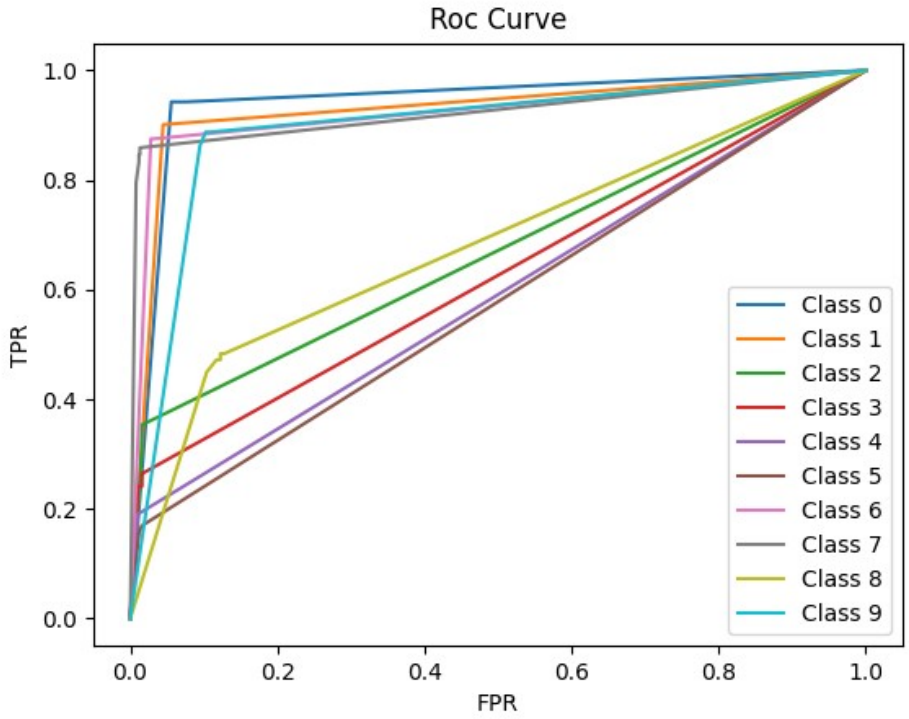
[[82 0 1 1 0 0 1 0 0 2]
[0 91 0 1 0 4 0 0 4 1]
[12 2 24 6 0 3 11 1 9 0]
[16 5 5 21 1 0 3 3 23 6]
[8 2 0 0 18 0 4 1 20 42]
[12 2 1 1 1 12 3 0 32 7]
[0 5 3 0 0 1 77 0 2 0]
[0 0 1 1 1 0 0 66 0 9]
[5 16 1 2 3 4 1 0 42 15]
[2 1 0 0 1 0 0 4 1 71]]

Accuracy: 0.6

Macro Precision: 0.6285283178023551
Macro Recall: 0.5888523786536409
Macro F1: 0.6080438043194594

Micro Precision: 0.6
Micro Recall: 0.6
Micro F1: 0.6

Roc Curve:



DATASET B

a)

```
Grid Search gives optimal max depth: 11
Applying KfoldCV using DT classifier and max_depth 11 (on 4 number of folds
DT: Training accuracy: 0.951686507936508 , Validation accuracy 0.587797619047619

Applying KfoldCV using GNB classifier(on 4 number of folds
GNB: Training accuracy: 0.5753968253968254 , Validation accuracy 0.5702380952380952
```

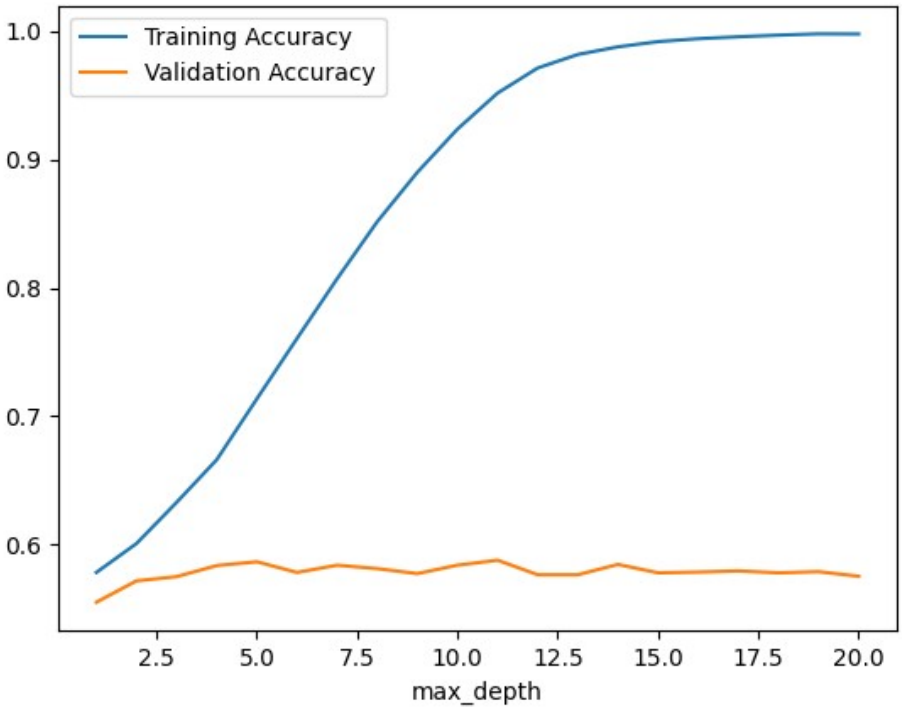
Grid search was performed for values of depth from 1 to 20.

Optimal depth of decision tree for grid search came out to be 11.

After applying K Fold Cross validation on depth 11, the validation accuracy of DT was found to be 0.587797619047619.

After applying K Fold Cross validation, the validation accuracy of GNB was found out to be 0.5702380952380952.

b)



After the value of 5 for max_depth, the validation accuracy of DT on the dataset is more or less constant. The model improved as the max_depth increased from 1 to 5 but after that the performance of the model stagnated. The model after max_depth 5 is overfitting on training data as training accuracy is still increasing.
c)

Both DT and GNB performed similarly(accuracy wise).
DT had better best validation accuracy intially but GNB gave better test accuracy .

The best fold of DT with max_depth = 11 was saved and then loaded.

```
Clearly best bodel is DT on depth 11
Loading best model

Using DT,Test accurancy: 0.5845238095238096
```

Accuracy on test data was found to be: 0.5845238095238096.

The best fold of GNB was saved and then loaded.

```
Using GNB,Test accurancy: 0.6095238095238096
```

Accuracy on test data was found to be 0.6095238095238096.

d)

For Decision Tree best fold model:

Confusion Matrix:

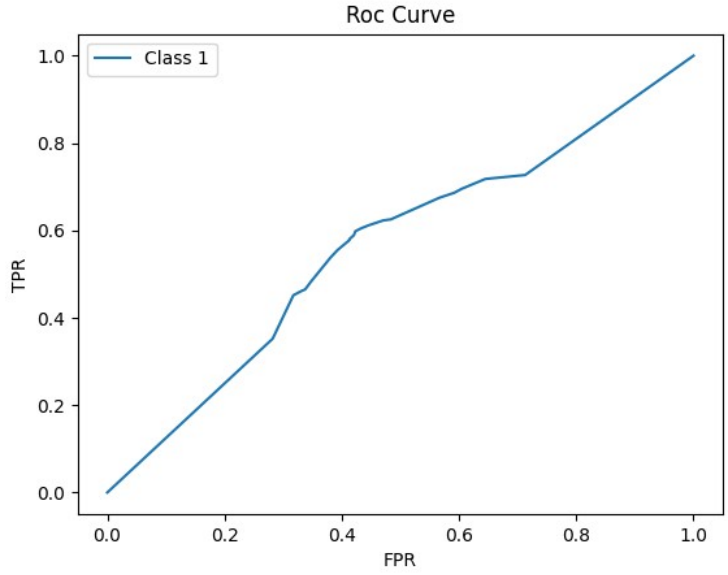
[[230 167]
 [182 261]]

Accuracy: 0.5845238095238096

Macro Precision: 0.5840327556483078
Macro Recall: 0.5842549368571283
Macro F1: 0.584143825125863

Micro Precision: 0.5845238095238096
Micro Recall: 0.5845238095238096
Micro F1: 0.5845238095238096

Roc Curve:



For Gaussian Naive Bayes Classifier best fold model:

Confusion Matrix:

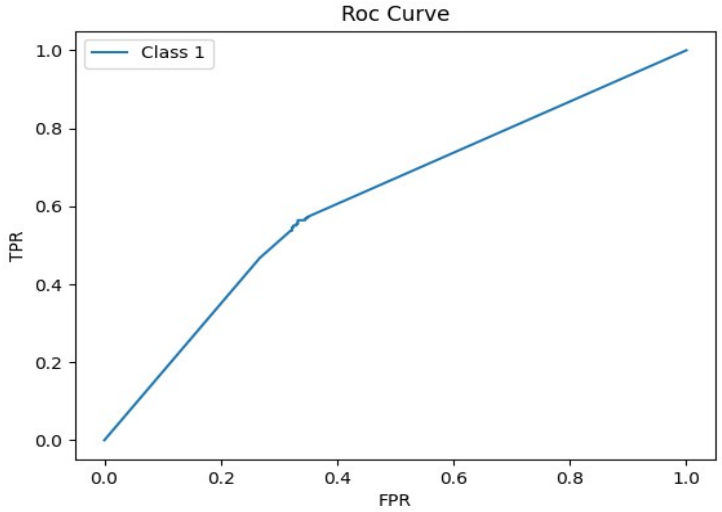
[[268 129]
 [199 244]]

Accuracy: 0.6095238095238096

Macro Precision: 0.6140156494882054
Macro Recall: 0.6129265200061409
Macro F1: 0.6134706013490725

Micro Precision: 0.6095238095238096
Micro Recall: 0.6095238095238096
Micro F1: 0.6095238095238096

Roc Curve:



QUESTION 4

Gaussian Naive Bayes was implemented from scratch.

```
pankil@pankil-ThinkPad-X390:~/Desktop/sem5/ML/assignments/assignment2$ python3
-W ignore Q4.py
My implementation score on dataset A: 0.5511904761904762
Sklearn score on dataset A: 0.55
My implementation score on dataset B: 0.5642857142857143
Sklearn score on dataset B: 0.5642857142857143
```

Dataset A:

My implementation score: 0.5511
Sklearn implementation score: 0.55

Dataset B:

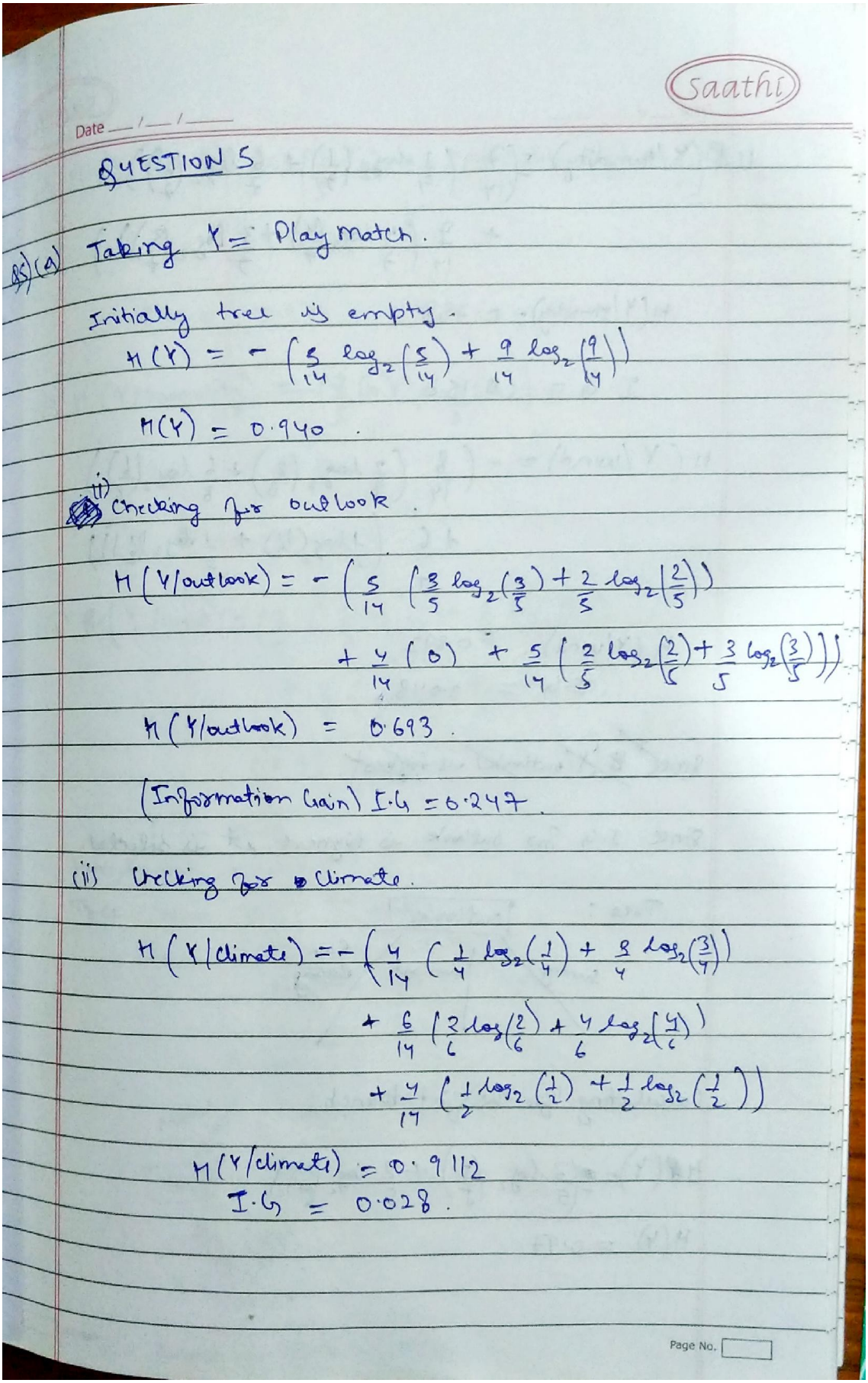
My implementation score: 0.5642

Sklearn implementation score: 0.5642

Clear my implementation and Sklearn’s implementation of Gaussian Naive Bayes Classifier give nearly the same accuracy.

P.T.O

QUESTION 5



QUESTION 5

(a) Taking Y = Play match.

Initially tree is empty.

$$H(Y) = -\left(\frac{5}{14}\log_2\left(\frac{5}{14}\right) + \frac{9}{14}\log_2\left(\frac{9}{14}\right)\right)$$

$$H(Y) = 0.940$$

(i) Checking for outlook.

$$H(Y/\text{outlook}) = -\left(\frac{5}{14}\left(\frac{3}{5}\log_2\left(\frac{3}{5}\right) + \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right) + \frac{4}{14}(0) + \frac{5}{14}\left(\frac{2}{5}\log_2\left(\frac{2}{5}\right) + \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right)\right)$$

$$H(Y/\text{outlook}) = 0.693$$

(Information Gain) I.G = 0.247

(ii) Checking for climate.

$$H(Y/\text{climate}) = -\left(\frac{4}{14}\left(\frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{3}{4}\log_2\left(\frac{3}{4}\right)\right) + \frac{6}{14}\left(\frac{2}{6}\log_2\left(\frac{2}{6}\right) + \frac{4}{6}\log_2\left(\frac{4}{6}\right)\right) + \frac{4}{14}\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right)\right)$$

$$H(Y/\text{climate}) = 0.9112$$

$$I.G = 0.028.$$

Saathi

Date ___/___/_____

$$H(Y/\text{Humidity}) = -\left(\frac{7}{14}\left(\frac{1}{7}\log_2\left(\frac{1}{7}\right) + \frac{6}{7}\log_2\left(\frac{6}{7}\right)\right) + \frac{7}{14}\left(\frac{4}{7}\log_2\left(\frac{4}{7}\right) + \frac{3}{7}\log_2\left(\frac{3}{7}\right)\right)\right)$$

$H(Y/\text{Humidity}) = 0.788$.

$I.G = 0.152$

$$H(Y/\text{wind}) = -\left(\frac{8}{14}\left(\frac{2}{8}\log_2\left(\frac{2}{8}\right) + \frac{6}{8}\log_2\left(\frac{6}{8}\right)\right) + \frac{6}{14}\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right)\right)$$

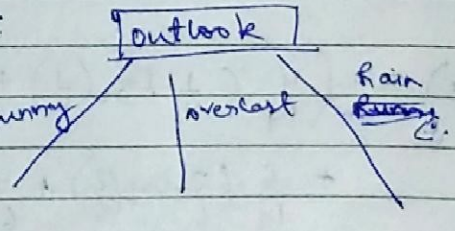
$H(Y/\text{wind}) = 0.892$

$I.G = 0.048$.

~~Since H(Y/outlook) is highest~~

Since I.G for outlook is highest, it is selected.

Tree :



Calculating for the first branch.

$$H(Y) = -\left(\frac{3}{5}\log_2\left(\frac{3}{5}\right) + \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right)$$

$H(Y) = 0.97$

Saathi

Date ___/___/_____

(i) $H(Y/\text{climate}) = -\left(\frac{1}{5}(0) + \frac{2}{5}\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right)\right)$

$= 0.40$.

$I.G = 0.57$.

(ii) $H(Y/\text{Humidity}) = -\left(\frac{2}{5}(0) + \frac{3}{5}(0)\right)$

$= 0$.

$I.G = 0.97$.

(iii) $H(Y/\text{wind}) = -\left(\frac{3}{5}\left(\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right)\right.$

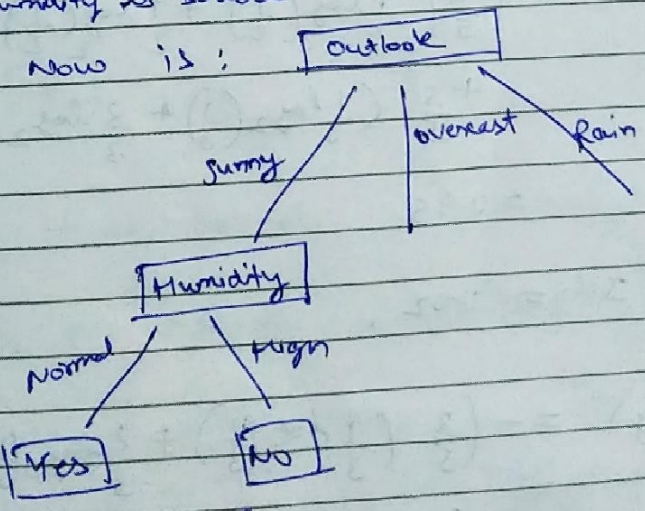
$\left. + \frac{2}{5}\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right)\right)$

$= 0.95$.

$I.G = 0.02$.

Hence, Humidity is selected

∴ Tree now is:



By Because

turned out to be leaf nodes.

Page No.

Date ___/___/_____

Saath

Considering second branch

Outlook

overcast

Yes → As this is a leaf node with all Y's as Yes.

Considering Third Branch

Outlook

Rain.

$$H(Y) = -\left(\frac{2}{5}\log_2\left(\frac{2}{5}\right) + \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right)$$

$$H(Y) = 0.97$$

(i) $$H(Y/\text{climate}) = -\left(\frac{2}{5}\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) + \frac{3}{5}\left(\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right)\right)$$

$$= 0.95$$

$$I.G = 0.02$$

(ii) $$H(Y/\text{Humidity}) = -\left(\frac{3}{5}\left(\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right) + \frac{2}{5}\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right)\right)$$

$$= 0.95.$$

Page No.

Saathi

Date ___/___/_____

$I.G = 0.02$.

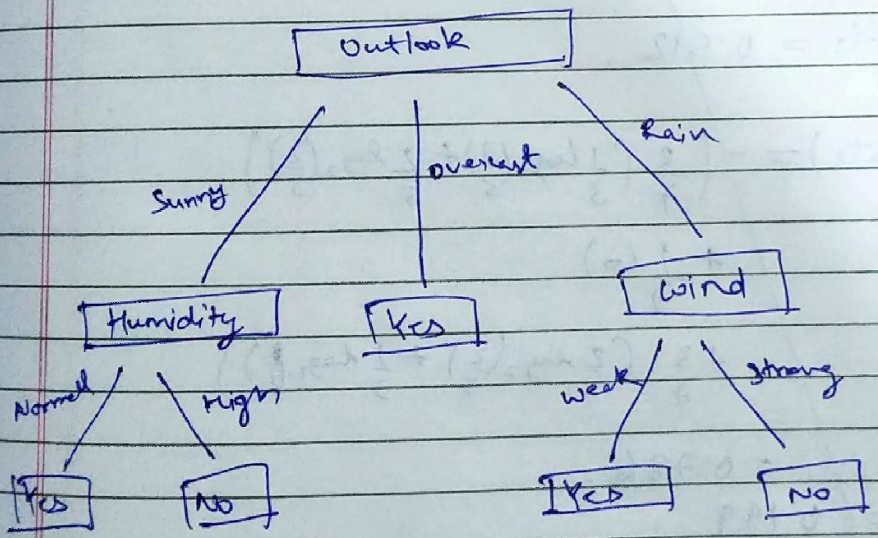
$H(Y/wind) = \left(\frac{3}{5}(0) + \frac{2}{5}(0)\right)$

$= 0$.

$I.G = 0.97$.

Hence, wind is selected as a decision attribute here.

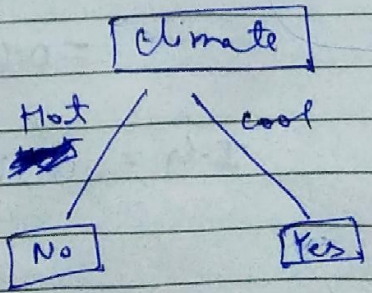
∴ Final tree:



Q5)(b) Yes, it is possible.

training
Consider the set of $\{D_1, D_2, D_5\}$

Resultant Decision Tree:



Page No.

Saathi

Date ___/___/_____

Q5) (a) Constructing Tree from Training Data:

$$H(Y) = -\left(\frac{4}{7}\log_2\left(\frac{4}{7}\right) + \frac{3}{7}\log_2\left(\frac{3}{7}\right)\right)$$

$$H(Y) = 0.985.$$

(i) $$H(Y/\text{outlook}) = -\left(\frac{2}{7}(0) + \frac{2}{7}(0) + \frac{3}{7}\left(\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right)\right)$$

$$= 0.393.$$

$$I.G = 0.592.$$

(ii) $$H(Y/\text{climate}) = -\left(\frac{3}{7}\left(\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right) + \frac{1}{7}(0) + \frac{3}{7}\left(\frac{2}{3}\log_2\left(\frac{2}{3}\right) + \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right)\right)$$

$$= 0.786.$$

$$I.G = 0.199.$$

(iii) $$H(Y/\text{Humidity}) = -\left(\frac{3}{7}\left(\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right) + \frac{4}{7}\left(\frac{2}{4}\log_2\left(\frac{2}{4}\right) + \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right)\right)$$

$$= 0.964.$$

$$I.G = 0.021.$$

Page No.

Saathi

Date __/__/____

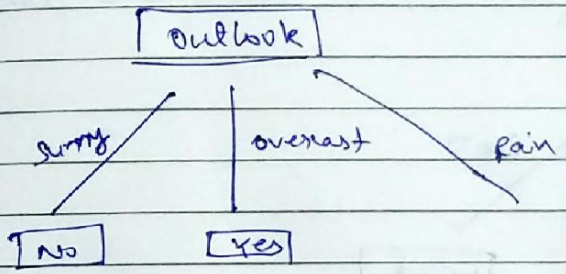
(iv) $H(Y/\text{wind}) = -\left(\frac{4}{7}\left(\frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{3}{4}\log_2\left(\frac{3}{4}\right)\right)\right.$

$\left. + \frac{3}{7}\left(\frac{2}{3}\log_2\left(\frac{2}{3}\right) + \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right)\right]$

$= 0.857$

$I.G = 0.128$.

Selecting Outlook as decision attribute.



Considering 3rd branch:

$H(Y) = -\left(\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right)$

$H(Y) = 0.918$

(i) $H(Y/\text{climate}) = -\left(\frac{2}{3}\left[\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right]\right.$

$\left. + \frac{1}{3}(0)\right)$

$= 0.667$.

$I.G = 0.251$.

(ii) $H(Y/\text{Humidity}) = -\left(\frac{3}{3}\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right)\right.$

$\left. + \frac{1}{3}(0)\right)$

$= 0.667 \Rightarrow I.G = 0.251$.

Page No.

Date ___/___/_____

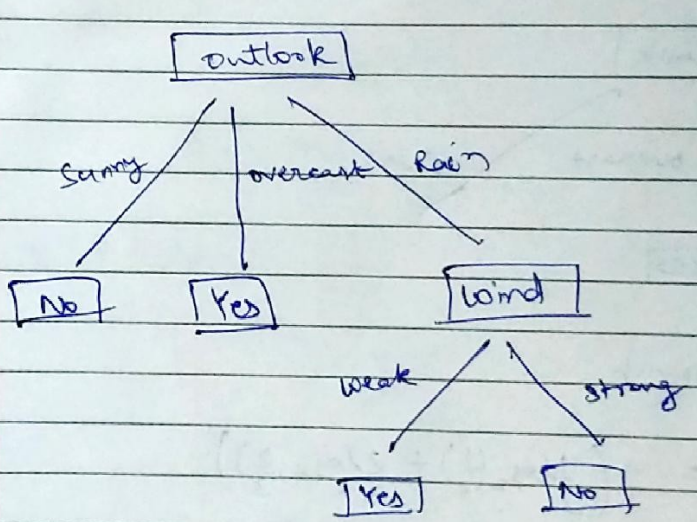
(iii) $H(Y/wind) = -\left(\frac{2}{3}(0) + \frac{1}{3}(0)\right)$

$= 0.$

$I.G = 0.918.$

∴, Selecting wind as decision attribute.

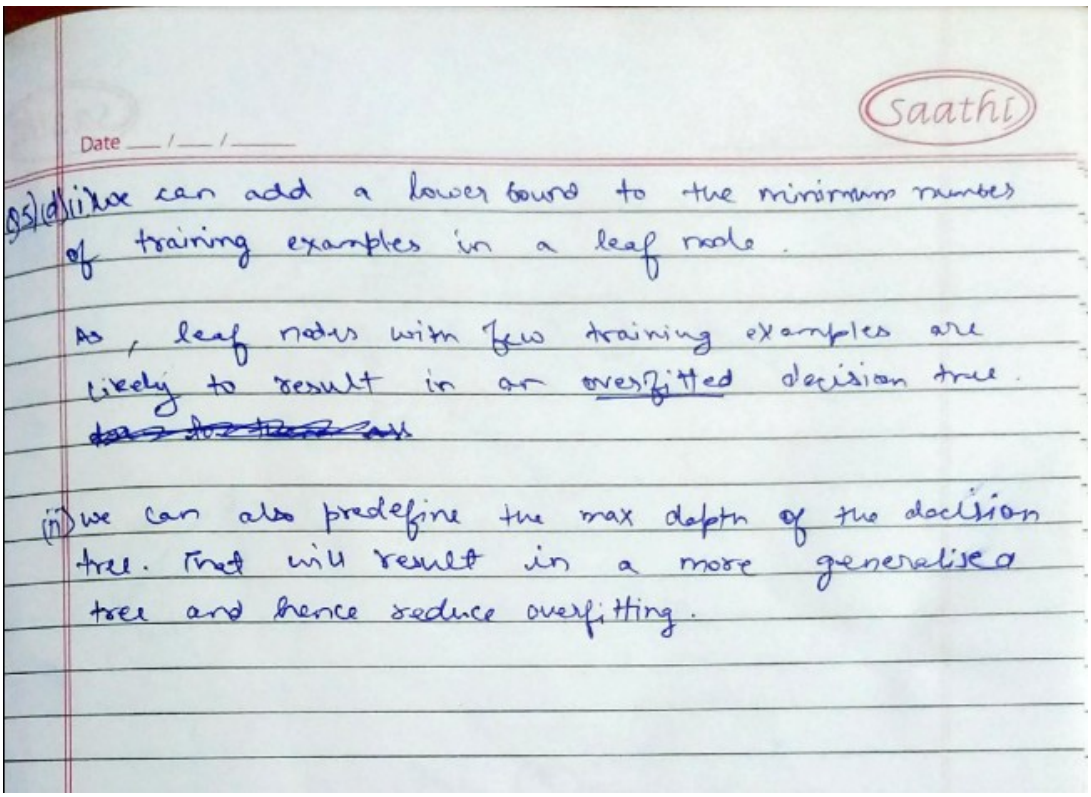
∴, Final tree :



	Actual Values (Y)	Predicted Values (Y)
D_8	No	No
D_9	Yes	No
D_{10}	Yes	Yes
D_{11}	Yes	No
D_{12}	Yes	Yes
D_{13}	Yes	Yes
D_{14}	No	No.

Accuracy $= \frac{5}{7}$ or 71.4 %

Page No.



Date ___/___/_____

Saathi

Q5(d)(i) We can add a lower bound to the minimum number of training examples in a leaf node.

As, leaf nodes with few training examples are likely to result in an overfitted decision tree.

(ii) We can also predefine the max depth of the decision tree. That will result in a more generalised tree and hence reduce overfitting.

P.T.O

QUESTION 6

Date ___/___/_____

Saathi

QUESTION 6.

w_1 = Tough, w_2 = Course, w_3 = ?, w_4 = Course

We need to find out $P\left(\frac{w_3}{w_2, w_4}\right)$.

→ Not considering w_1 as w_2 and w_4 are given so we are using markov's assumption

We know, using formula of Naive Bayes for 3 events :

$$P\left(\frac{A}{B,C}\right) = \frac{P\left(\frac{B}{A,C}\right) \cdot P\left(\frac{A}{C}\right)}{P\left(\frac{B}{C}\right)}$$

taking, $A = w_3$, $B = w_4$, $C = w_2$

$$P\left(\frac{w_3}{w_4, w_2}\right) = \frac{P\left(\frac{w_4}{w_3, w_2}\right) \cdot P\left(\frac{w_3}{w_2}\right)}{P\left(\frac{w_4}{w_2}\right)}$$

$$P\left(\frac{w_3}{w_4, w_2}\right) = \frac{P\left(\frac{w_4}{w_3}\right) \cdot P\left(\frac{w_3}{w_2}\right)}{P\left(\frac{w_4}{w_2}\right)} \quad \{\text{using markov's assumption}\}$$

Case (i):

when w_3 = tough

$$P\left(\frac{w_4}{w_3}\right) = 0.3 \qquad P\left(\frac{w_3}{w_2}\right) = 0.5.$$

$$P\left(\frac{w_4}{w_2}\right) = 0.5 \cdot 0.3 + 0.5 \cdot 0.5 = 0.40$$

Page No.

Date ___/___/_____

$$\Rightarrow P\left(\frac{w_3}{w_4, w_2}\right) = \frac{(0.3)(0.5)}{(0.4)} = 0.375.$$

∴, posterior probability that missing word is tough is 0.375.

Case (ii)

w_3 = course

Probability that third word is course = 1 − probability third word is tough = 1 − 0.375

= 0.625

∴, posterior probability that missing word is course is 0.625.

P.T.O

QUESTION 7

Saathi

Date ___/___/_____

QUESTION 7

Q7 (a) Decision trees do not assume independence of the input features whereas logistic regression treats each feature independently.

This property enables decision trees to approximate complicated functions better than logistic regression classifiers.

(b) The biggest weakness of decision tree classifiers is that they are more likely to overfit on the training data as compared to logistic regression classifiers, as they can perform multiple splits for different attributes but in logistic regression there is one parameter for each feature.

(c) NA

(d) Yes. The upper bound on depth is $O(\log n)$.

Explanation:

There are n values of x_1, and since the vectors are linearly separable, for each x_1 we can get a value of x_2 above which the vector is in class 1.

We can separate the n x_1 values using a decision tree (e.g. self balancing binary search tree or AVL). For the x_2 cutoff we can add another node below each leaf.

So, Total depth = $\log n + 1 = O(\log n)$.

Page No.

Saathi

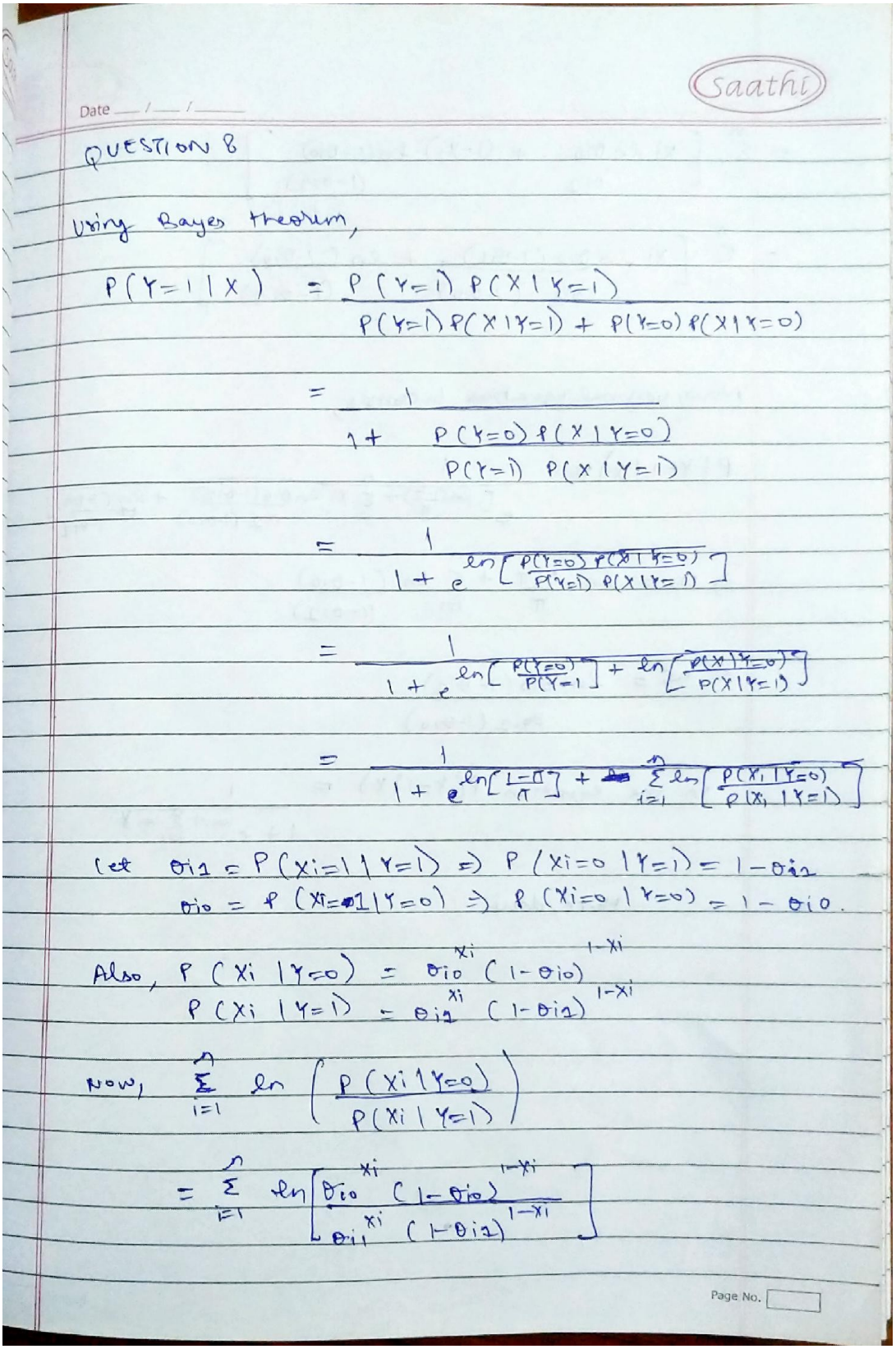
Date ___/___/_____

(e) Yes, a decision tree can still classify these input vectors.

We can build a decision tree (as done in previous part) to separate the n x_i values. The depth of this tree would be $\log(n)$.

For separating the x_i values, as we cannot get a cutoff in this case, we can extend to another tree of $\log n$ depth below each leaf. So, total depth of tree is $2\log(n)$ or $O(\log n)$

QUESTION 8



QUESTION 8

Using Bayes theorem,

$$P(Y=1|X) = \frac{P(Y=1)\,P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)\,P(X|Y=0)}{P(Y=1)\,P(X|Y=1)}}$$

$$= \frac{1}{1 + e^{\ln\left[\frac{P(Y=0)\,P(X|Y=0)}{P(Y=1)\,P(X|Y=1)}\right]}}$$

$$= \frac{1}{1 + e^{\ln\left[\frac{P(Y=0)}{P(Y=1)}\right] + \ln\left[\frac{P(X|Y=0)}{P(X|Y=1)}\right]}}$$

$$= \frac{1}{1 + e^{\ln\left[\frac{1-\pi}{\pi}\right] + \sum_{i=1}^{n} \ln\left[\frac{P(X_i|Y=0)}{P(X_i|Y=1)}\right]}}$$

(let $\theta_{i1} = P(X_i=1|Y=1) \Rightarrow P(X_i=0|Y=1) = 1-\theta_{i1}$

$\theta_{i0} = P(X_i=1|Y=0) \Rightarrow P(X_i=0|Y=0) = 1-\theta_{i0}$.

Also, $P(X_i|Y=0) = \theta_{i0}^{X_i}(1-\theta_{i0})^{1-X_i}$

$P(X_i|Y=1) = \theta_{i1}^{X_i}(1-\theta_{i1})^{1-X_i}$

Now, $\sum_{i=1}^{n} \ln\left(\frac{P(X_i|Y=0)}{P(X_i|Y=1)}\right)$

$$= \sum_{i=1}^{n} \ln\left[\frac{\theta_{i0}^{X_i}(1-\theta_{i0})^{1-X_i}}{\theta_{i1}^{X_i}(1-\theta_{i1})^{1-X_i}}\right]$$

Date ___/___/_____

$$= \sum_{i=1}^{n} \left[x_i \ln \frac{\theta_{i0}}{\theta_{i1}} + (1-x_i) \ln \frac{(1-\theta_{i0})}{(1-\theta_{i1})} \right]$$

$$= \sum_{i=1}^{n} \left[x_i \ln \frac{\theta_{i0}(1-\theta_{i1})}{\theta_{i1}(1-\theta_{i0})} + \ln \frac{(1-\theta_{i0})}{(1-\theta_{i1})} \right]$$

Now, original equation becomes,

$$P(Y=1|X) = \frac{1}{e^{\left[\ln \frac{(1-\pi)}{\pi} + \sum_{i=1}^{n} x_i \ln \frac{\theta_{i0}(1-\theta_{i1})}{\theta_{i1}(1-\theta_{i0})} + \ln \frac{(1-\theta_{i0})}{(1-\theta_{i1})} \right]}}$$

$$\Rightarrow w_0 = \ln \frac{1-\pi}{\pi} + \sum_{i=1}^{n} \ln \frac{(1-\theta_{i0})}{(1-\theta_{i1})}$$

$$\Rightarrow w_i = \ln \frac{\theta_{i0}(1-\theta_{i1})}{\theta_{i1}(1-\theta_{i0})}$$

in the equation $P(Y=1|X) = \frac{1}{1+e^{w_0 + \sum_{i=1}^{n} w_i x_i}}$

Hence, done.

Page No.