Machine Learning
Assignment 2
Report

Question 1:

a) PCA called the Principal Component Analysis, is an unsupervised technique used for dimensionality reduction, feature extraction, and data visualization.

   Poor visualization, with non-linear manifold data. It aims to find the directions along which the variance is maximum in high dimensional data and maps it onto a subspace with equal or fewer dimensions than the original one. The principal components or the eigenvectors of the subspace formed are along the directions of maximum variance. The largest eigenvalue corresponds to maximum variance along that eigenvector.

   From the wiki,

   "*Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components.*"

b) SVD or Singular Value Decomposition is the most popular technique for Dimensionality Reduction. It allows an exact representation of the data and makes it easier to eliminate the less important parts of the representation to represent the data approximately into less number of dimensions. It works well on the sparse data. Lesser the number of dimensions, the lesser the accuracy. Poor visualization with non-linear manifold data.

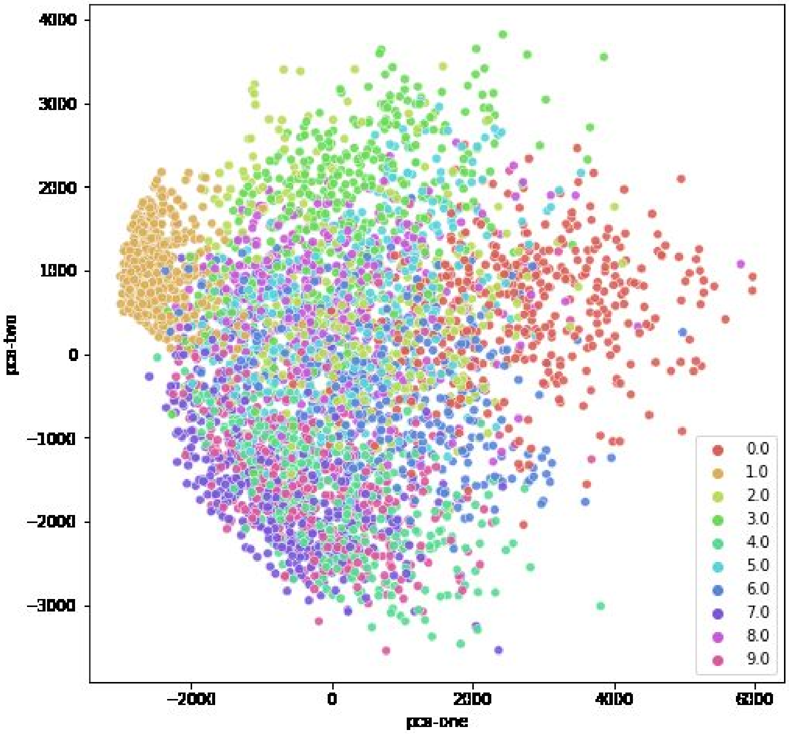
   PCA uses SVD after subtracting the mean value from each data point.

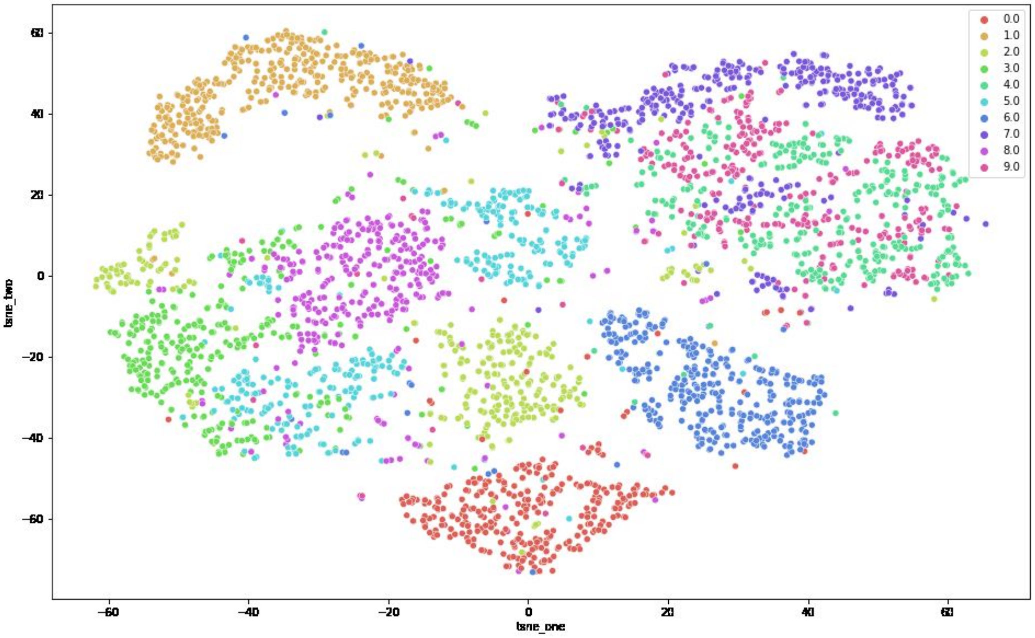
   It is right to say that PCA produces a subspace that spans the deviations from the mean, while SVD produces a subspace that spans itself or spans the deviation from zero. (the linear combination of the columns produces the entire subspace)

c) t-Distributed Stochastic Neighbour Embedding or t-SNE is an unsupervised, non-linear technique that is used for visualizing high dimensional data. It gives an intuition of how our data looks in a high-dimensional space. It calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space. At every data point, a gaussian distribution is centered and it manipulates the perplexity or the number of nearest neighbors by varying the variance. However, it is not a clustering approach since it does not preserve the inputs like PCA and SVD. It is meant only for exploration.

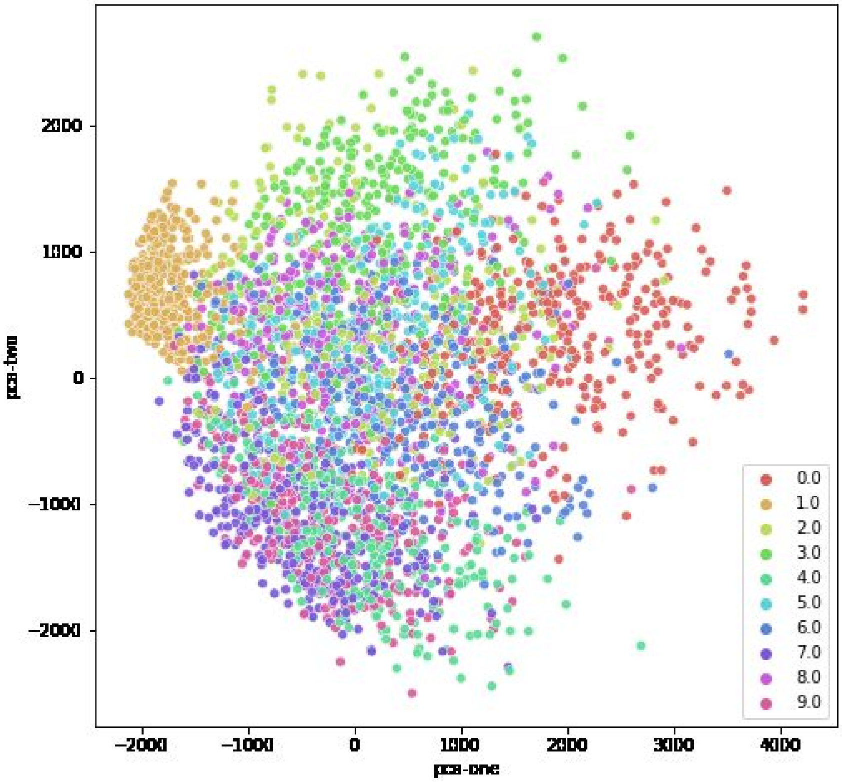
d) Suppose I want to split the data that I have in the ratio of 80:20. The stratified split splits the data in the following way:
   i) Suppose there are n classes in which we want our data to be classified. While performing a stratified split, it'll count the number of instances of each of the classes and split them into the same ratio as mentioned above. This is done for each of the classes.
   ii) Hence, the class frequency in each of the classes in train and test set are in the ratio of 80:20.

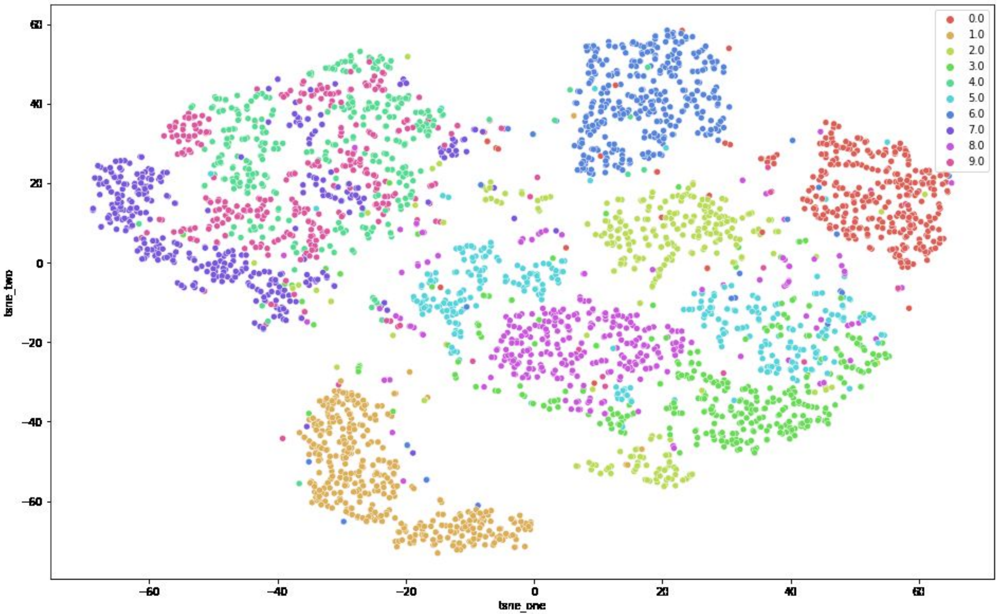
e) Using SVD, Accuracy Reported = 87.14%





f) Using PCA, Accuracy Reported=87.38%





g)

The cluster obtained from both SVD and PCA is the same since PCA uses SVD to find the orthogonal principal components. PCA could be performed in multiple ways but using SVD is one of the best approaches since it gives an exact representation of high dimensional data. Apart from that, before performing SVD, the mean value is

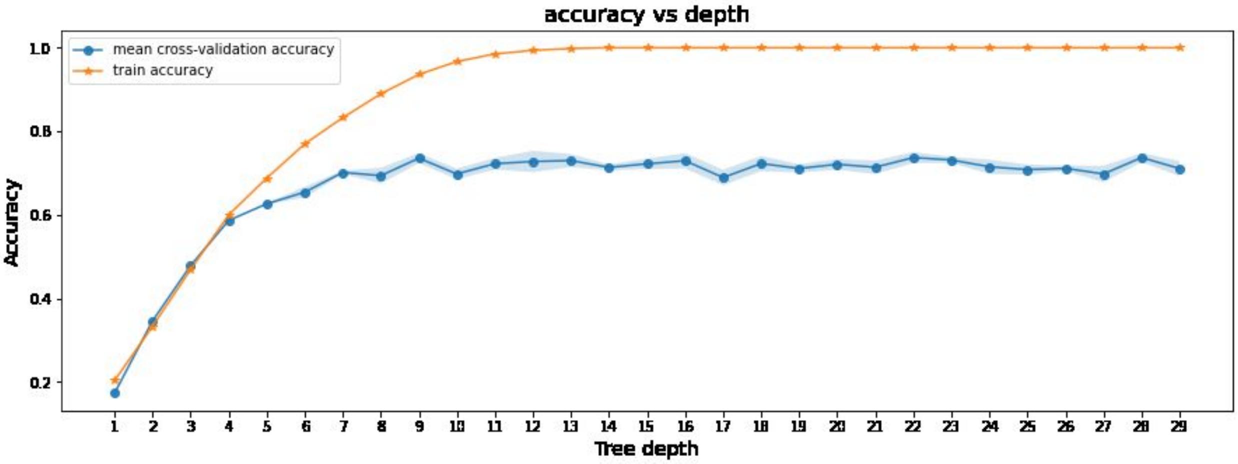
subtracted from each data point. On visual analysis using TSNE, we see better results for PCA. The interpretation that can be made is that the features in our data are less sensitive towards the mean, therefore subtracting the mean would lead to better results as observed.
Top 50 features PCA with Logistic Regression = 87.38% accuracy
Top 50 features SVD with Logistic Regression = 87.14% accuracy

Question 2:
The value of `0.004001074077663171,` obtained in this case is very close to zero since we've assumed the noise to be zero.
The expression **MSE - BIAS^2 - VARIANCE** measures the noise in the data. Hence, the results corroborate with our assumption.

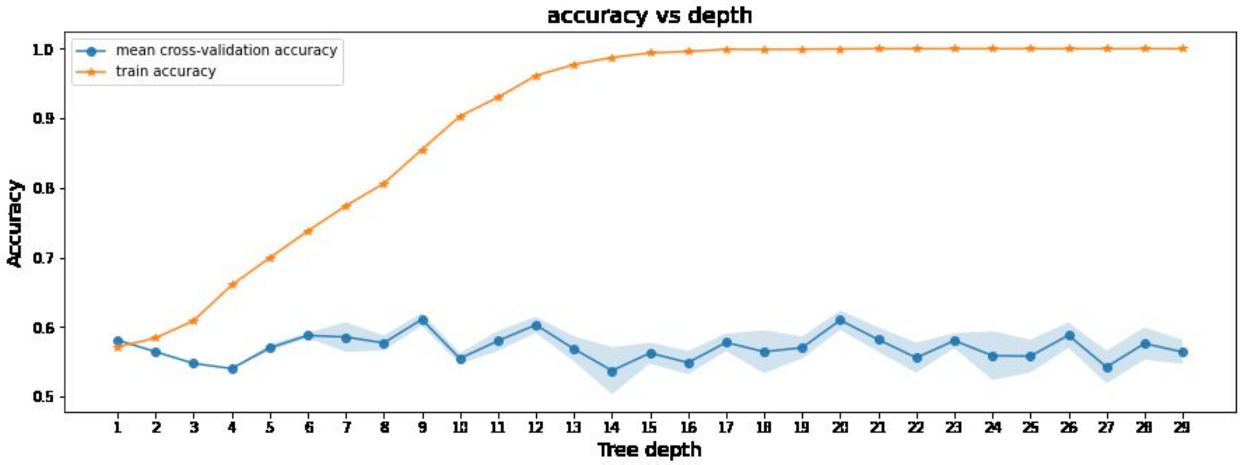
Question 3:

a) The optimal Depth of Decision Tree for Dataset A is 109 with an accuracy of 75.86%. I ran for depth 1 to 30 and 100 to 150
The optimal Depth of Decision Tree for Dataset B is 9 with an accuracy of 61.56%. I ran for depth 1 to 30 and 100 to 150
(Since larger depths were taking v longer to execute)

b) For Dataset A



For Dataset B



The point to be observed here that the curves for both train and validation accuracy begin at the same threshold but as the depth increases, the training accuracy shoots up and reaches 1, while the validation accuracy keeps fluctuating showing sharp peaks at some points. This clearly indicates that the model is overfitting the data and hence producing less accuracy on the validation data.

c)

Best model for DataSet B on testdata = 0.5952380952380952

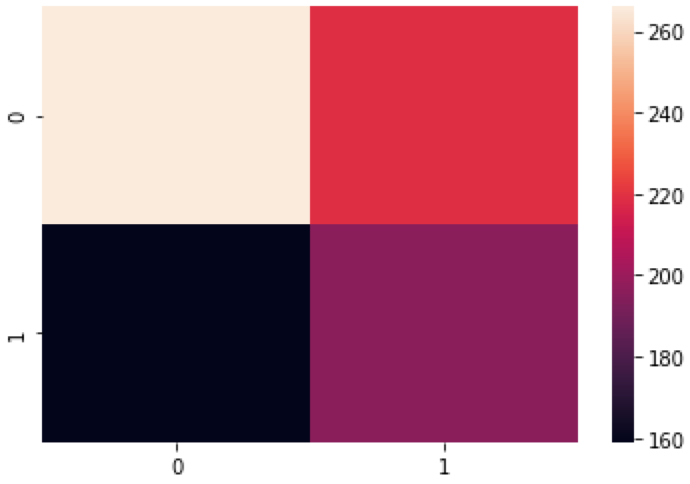
Best model for DataSet A on test data = 0.7357142857142858

d) Dataset B

GNB:

Recall, precision, f1-score, accuracy = ((0.4782608695652174, 0.5284450063211126, 0.5903954802259888, 0.555952380952381))

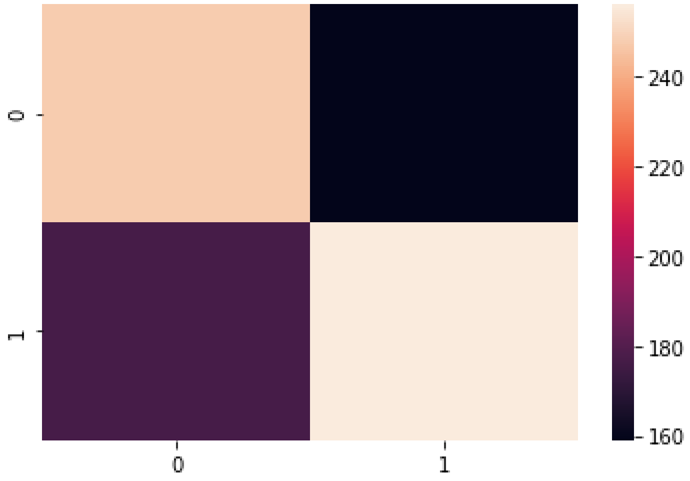
Confusion Matrix: {'tp': 209, 'tn': 258, 'fp': 145, 'fn': 228}



Decision Tree:

Recall, precision, f1-score, accuracy = ((0.5583524027459954, 0.580952380952381, 0.6054590570719603, 0.580952380952381))

Confusion Matrix :[248., 159.], [177., 256.]

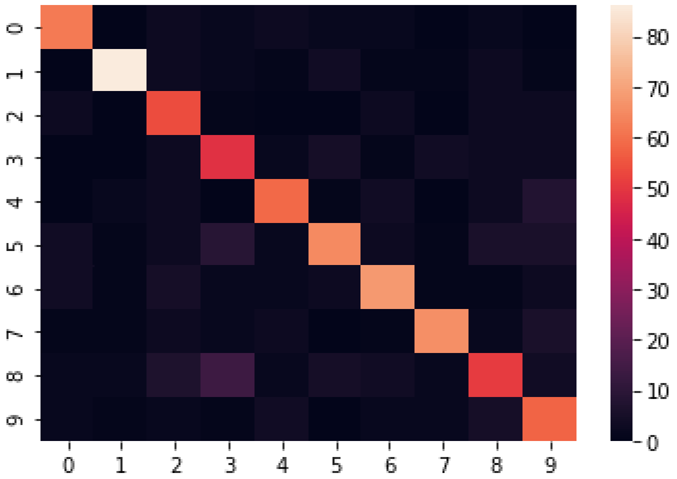


DataSet A

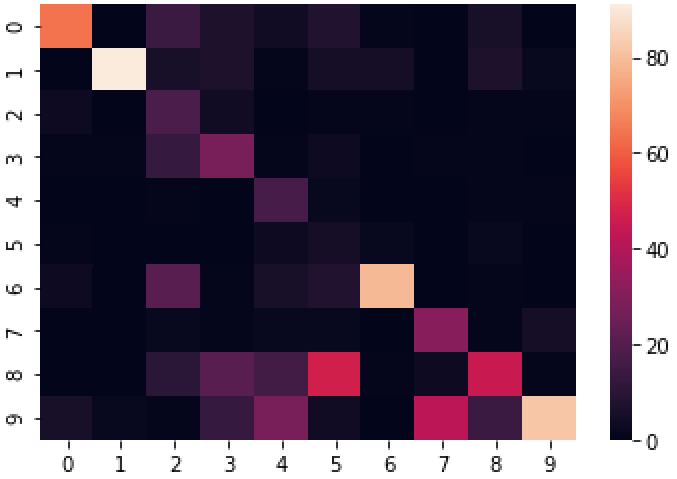
Decision Tree:

micro_precision, macro_precision, micro_recall, macro_recall, micro_f1, macro_f1 = (0.7166666666666667, 0.7105631902680207, 0.7166666666666667, 0.7112385371210171, 0.7166666666666667, 0.7109007033017711)

Confusion Matrix : {0.0: {'tp': 67, 'tn': 746, 'fn': 19, 'fp': 8}, 1.0: {'tp': 88, 'tn': 731, 'fn': 5, 'fp': 16}, 2.0: {'tp': 36, 'tn': 752, 'fn': 25, 'fp': 27}, 3.0: {'tp': 61, 'tn': 714, 'fn': 30, 'fp': 35}, 4.0: {'tp': 57, 'tn': 738, 'fn': 23, 'fp': 22}, 5.0: {'tp': 46, 'tn': 742, 'fn': 29, 'fp': 23}, 6.0: {'tp': 71, 'tn': 719, 'fn': 25, 'fp': 25}, 7.0: {'tp': 73, 'tn': 731, 'fn': 13, 'fp': 23}, 8.0: {'tp': 48, 'tn': 719, 'fn': 45, 'fp': 28}, 9.0: {'tp': 55, 'tn': 730, 'fn': 24, 'fp': 31}})



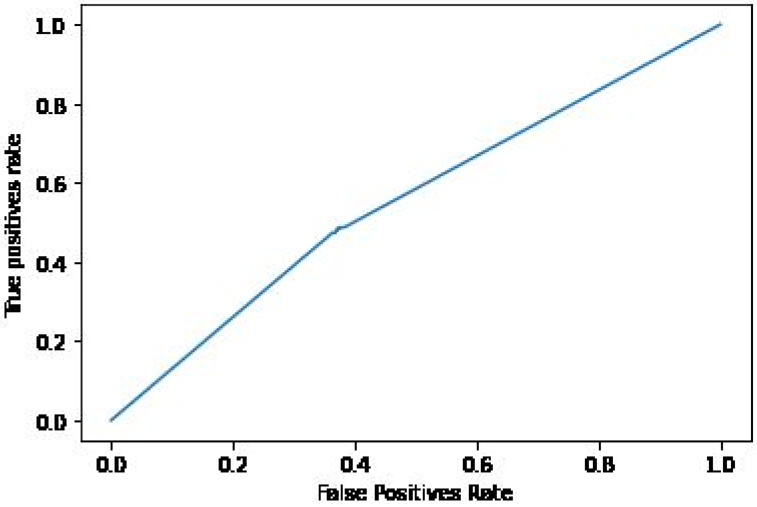
```
GNB:
micro_precision, macro_precision, micro_recall, macro_recall,
micro_f1, macro_f1 = (0.6047619047619047, 0.6313270798996296,
0.6047619047619047, 0.5827273045234753, 0.6047619047619047,
0.6060544441217973)

Confusion Matrix: {0.0: {'tp': 78, 'tn': 715, 'fn': 8, 'fp': 39}, 1.0: {'tp':
90, 'tn': 719, 'fn': 3, 'fp': 28}, 2.0: {'tp': 15, 'tn': 766, 'fn': 46, 'fp':
13}, 3.0: {'tp': 46, 'tn': 735, 'fn': 45, 'fp': 14}, 4.0: {'tp': 24, 'tn':
756, 'fn': 56, 'fp': 4}, 5.0: {'tp': 5, 'tn': 756, 'fn': 70, 'fp': 9}, 6.0:
{'tp': 87, 'tn': 724, 'fn': 9, 'fp': 20}, 7.0: {'tp': 46, 'tn': 742, 'fn': 40,
'fp': 12}, 8.0: {'tp': 46, 'tn': 676, 'fn': 47, 'fp': 71}, 9.0: {'tp': 71,
'tn': 639, 'fn': 8, 'fp': 122}})
```



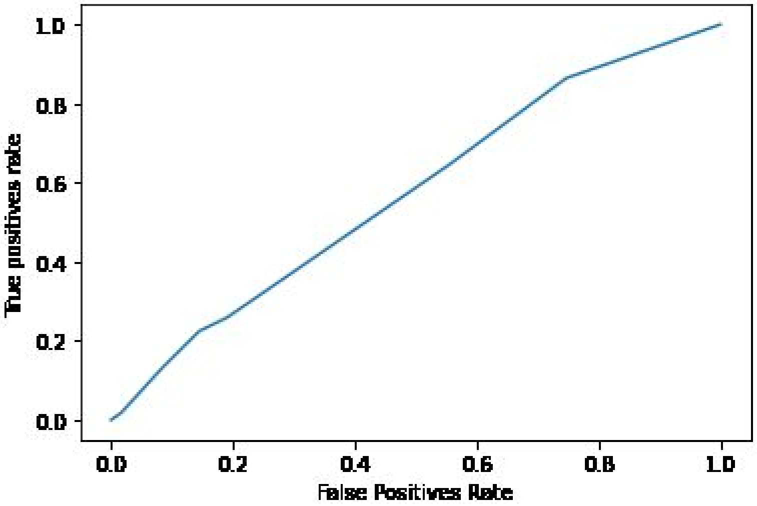
ROC Curves

DataSet B

For GNB:

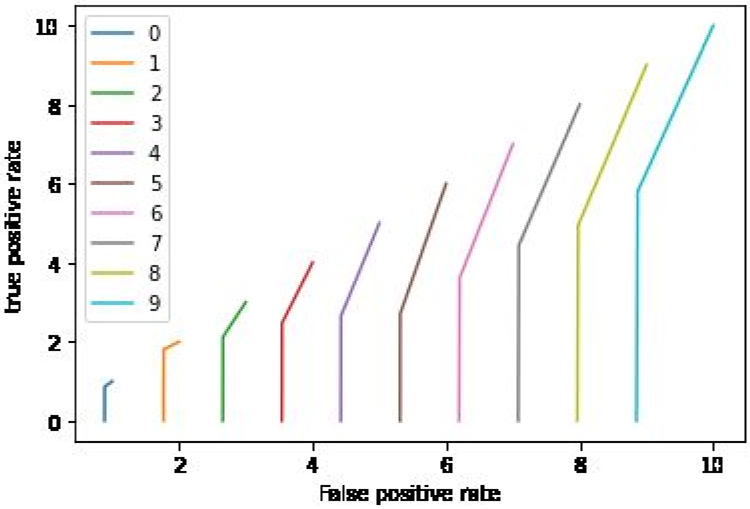


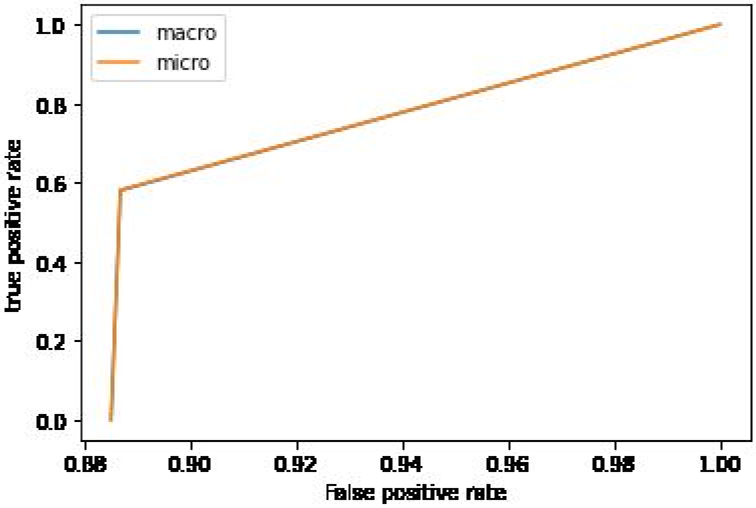
For Decision Tree



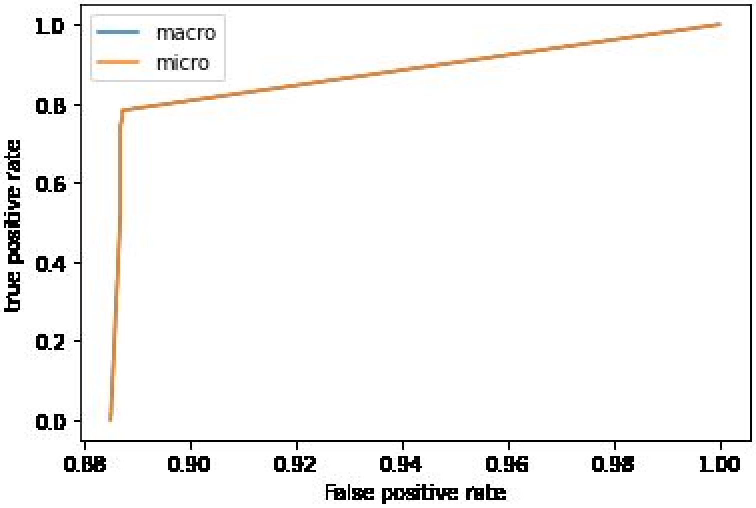
Dataset A

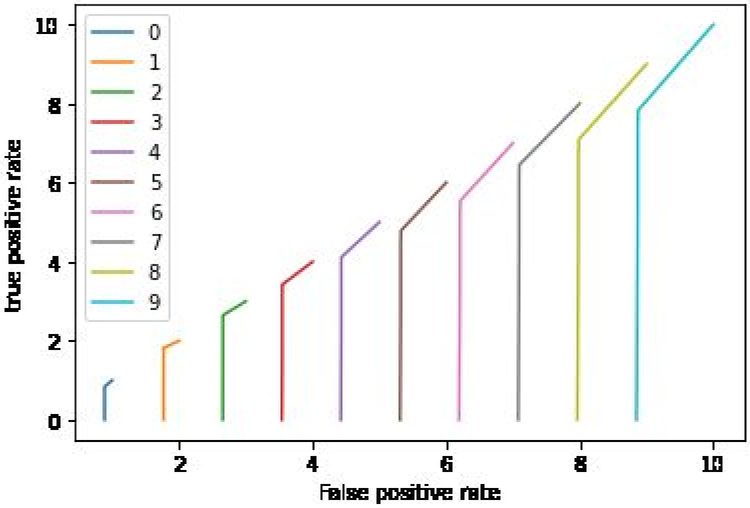
GNB:





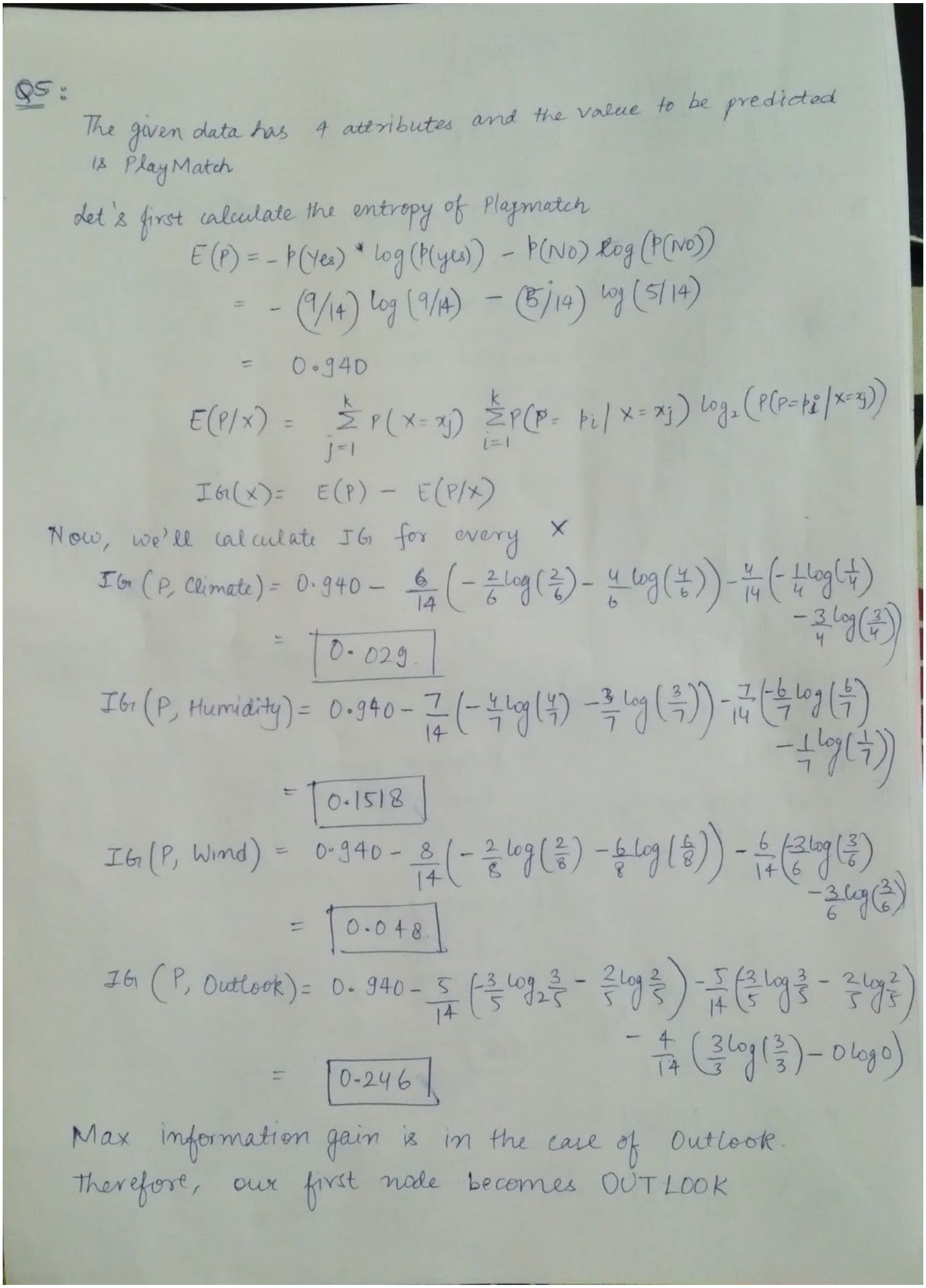
Decision Tree





Question 4:
Accuracy scores
Dataset A:
Self GNB: 0.5216666666666666
Sklearn GNB: 0.5952380952380952
Dataset B:
Self GNB: 0.4992857142857143
Sklearn GNB: 0.5721428571428572

Question 5:



Q5:

The given data has 4 attributes and the value to be predicted is PlayMatch

let's first calculate the entropy of Playmatch

$$E(P) = -p(\text{Yes}) * \log(p(\text{yes})) - p(\text{No}) \log(P(\text{NO}))$$

$$= -(9/14) \log(9/14) - (5/14) \log(5/14)$$

$$= 0.940$$

$$E(P/x) = \sum_{j=1}^{k} P(x = x_j) \sum_{i=1}^{k} P(P = p_i / x = x_j) \log_2 (P(P = p_i / x = x_j))$$

$$IG(x) = E(P) - E(P/x)$$

Now, we'll calculate IG for every x

$$IG(P, \text{Climate}) = 0.940 - \frac{6}{14}\left(-\frac{2}{6}\log\left(\frac{2}{6}\right) - \frac{4}{6}\log\left(\frac{4}{6}\right)\right) - \frac{4}{14}\left(-\frac{1}{4}\log\left(\frac{1}{4}\right) - \frac{3}{4}\log\left(\frac{3}{4}\right)\right)$$

$$= \boxed{0.029}$$

$$IG(P, \text{Humidity}) = 0.940 - \frac{7}{14}\left(-\frac{4}{7}\log\left(\frac{4}{7}\right) - \frac{3}{7}\log\left(\frac{3}{7}\right)\right) - \frac{7}{14}\left(\frac{-6}{7}\log\left(\frac{6}{7}\right) - \frac{1}{7}\log\left(\frac{1}{7}\right)\right)$$

$$= \boxed{0.1518}$$

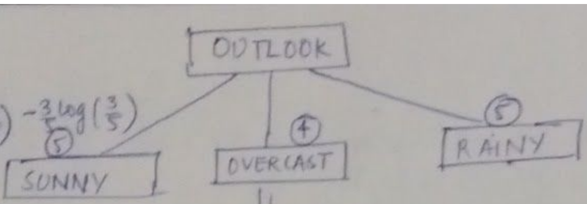
$$IG(P, \text{Wind}) = 0.940 - \frac{8}{14}\left(-\frac{2}{8}\log\left(\frac{2}{8}\right) - \frac{6}{8}\log\left(\frac{6}{8}\right)\right) - \frac{6}{14}\left(\frac{-3}{6}\log\left(\frac{3}{6}\right) - \frac{3}{6}\log\left(\frac{3}{6}\right)\right)$$

$$= \boxed{0.048}$$

$$IG(P, \text{Outlook}) = 0.940 - \frac{5}{14}\left(\frac{-3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log\frac{2}{5}\right) - \frac{5}{14}\left(\frac{-3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5}\right) - \frac{4}{14}\left(\frac{3}{3}\log\left(\frac{3}{3}\right) - 0\log 0\right)$$

$$= \boxed{0.246}$$

Max information gain is in the case of Outlook.
therefore, our first node becomes OUTLOOK



$E(P) = -\frac{2}{5}\log\left(\frac{2}{5}\right) - \frac{3}{5}\log\left(\frac{3}{5}\right)$

$= 0.971$

$IG(P, \text{wind}) =$

$0.971 - \frac{3}{5}\left(-\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3}\right)$

$- \frac{2}{5}\left(-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}\right)$

$IG(P,W) = \boxed{0.02}$

$IG(P, \text{Humidity}) =$

$0.971 - \frac{3}{5}\left(-\frac{3}{3}\log\left(\frac{3}{3}\right)\right)$

$- \frac{2}{5}\left(-\frac{2}{2}\log\left(\frac{2}{2}\right)\right)$

$= \boxed{0.971}$

$IG(P, \text{Climate}) =$

$0.971 - \frac{2}{5}\left(-\frac{0}{2}\log\frac{0}{2} - \frac{2}{2}\log\frac{2}{2}\right)$

$- \frac{1}{5}(0) - \frac{2}{5}\left(-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}\right)$

$= \boxed{0.571}$

So, next node on the left will be Humidity

$\Downarrow$

$E(P) = -\frac{4}{4}\log\left(\frac{4}{4}\right) - \frac{0}{4}\log\left(\frac{0}{4}\right)$

$E(P) = 0$

Therefore, whenever outlook is overcast, we're free to play

$E(P) = -\frac{3}{5}\log\left(\frac{3}{5}\right) - \frac{2}{5}\log\left(\frac{2}{5}\right)$

$= 0.971$

$IG(P, \text{wind}) = 0.971 - \frac{3}{5}(0) - \frac{2}{5}(0)$

$= \boxed{0.971}$

$IG(P, \text{Humidity}) =$

$0.971 - \frac{2}{5}(1) - \frac{3}{5}\left(\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3}\right)$

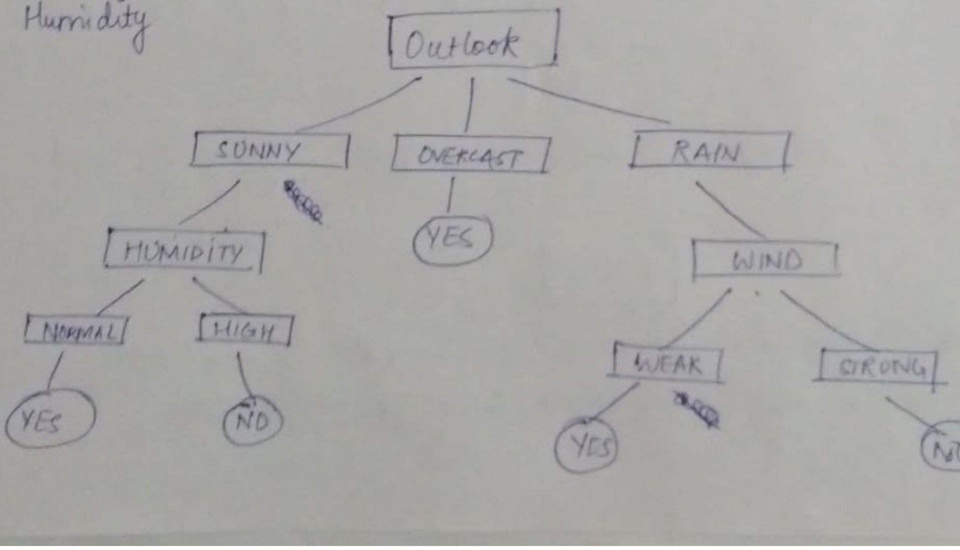
$= \boxed{0.02}$

Similarly

$IG(P, \text{Climate}) = \boxed{0.02}$

Since max IG is for wind, the next node will be Wind.

Therefore, our final tree looks like this,



(b) Yes, it is possible.

Let's consider the days = {D1, D2, D4, D10, D11, D12}

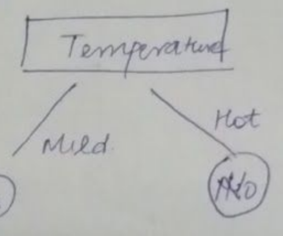
for this particular set, let's calculate the entropy.

Climate ~ Hot, mild

$E(Y, \text{Climate}) = \frac{2}{6}[0] - \frac{4}{6}[0] = 0$

Since, entropy for this set on the attribute Temperature is zero, the temperature attribute (alone) itself predicts the output.

The corresponding tree for this would look like this



(c) D1 - D7 - Train, D8 - D14 - Test

$E(Y) = -\frac{3}{7}\log\left(\frac{3}{7}\right) - \frac{4}{7}\log\left(\frac{4}{7}\right) = 0.985$

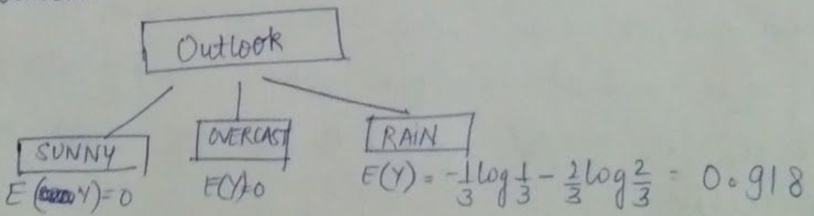
$IG(Y, \text{Climate}) = 0.985 - \frac{3}{7}\left(-\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3}\right) - \frac{3}{7}\left(\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3}\right) - \frac{1}{7}(0)$

$IG(Y, \text{outlook}) = 0.985 - \frac{2}{7}(0) - \frac{3}{7}\left(\frac{-1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3}\right) - \frac{2}{7}(0)$

$IG(Y, \text{Wind}) = 0.985 - \frac{4}{7}\left(\frac{-3}{4}\log\left(\frac{3}{4}\right) - \frac{1}{4}\log\left(\frac{3}{4}\right)\right) - \frac{3}{7}\left(\frac{-2}{3}\log\left(\frac{2}{3}\right) - \frac{1}{3}\log\left(\frac{1}{3}\right)\right)$

$IG(Y, \text{Humidity}) = 0.985 - \frac{4}{7}\left[\frac{-1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}\right] - \frac{3}{7}\left[\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3}\right]$

Since, outlook has max-info gain, therefore our first node becomes, outlook.



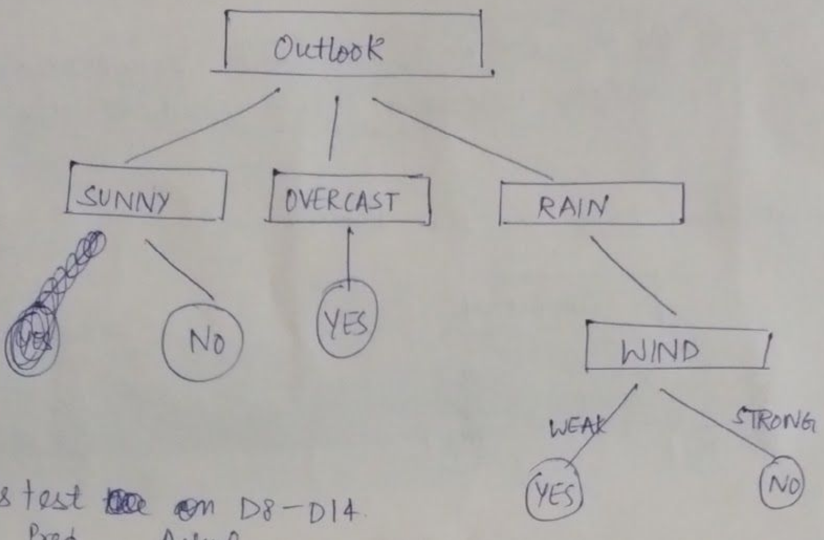
$IG(Y, \text{Climate}) = 0.918 - \frac{1}{3}(0) - \frac{2}{3}(1) = 0.248$

$IG(Y, \text{Wind}) = 0.918 - \frac{2}{3}(0) - \frac{1}{3}(0) = \boxed{0.918}$

$IG(Y, \text{Humid}) = 0.918 - \frac{1}{3}(0) - \frac{2}{3}(1) = 0.248$

Since, wind has max. ~~outlook~~ I G.,
let's choose wind,

Therefore, final tree becomes

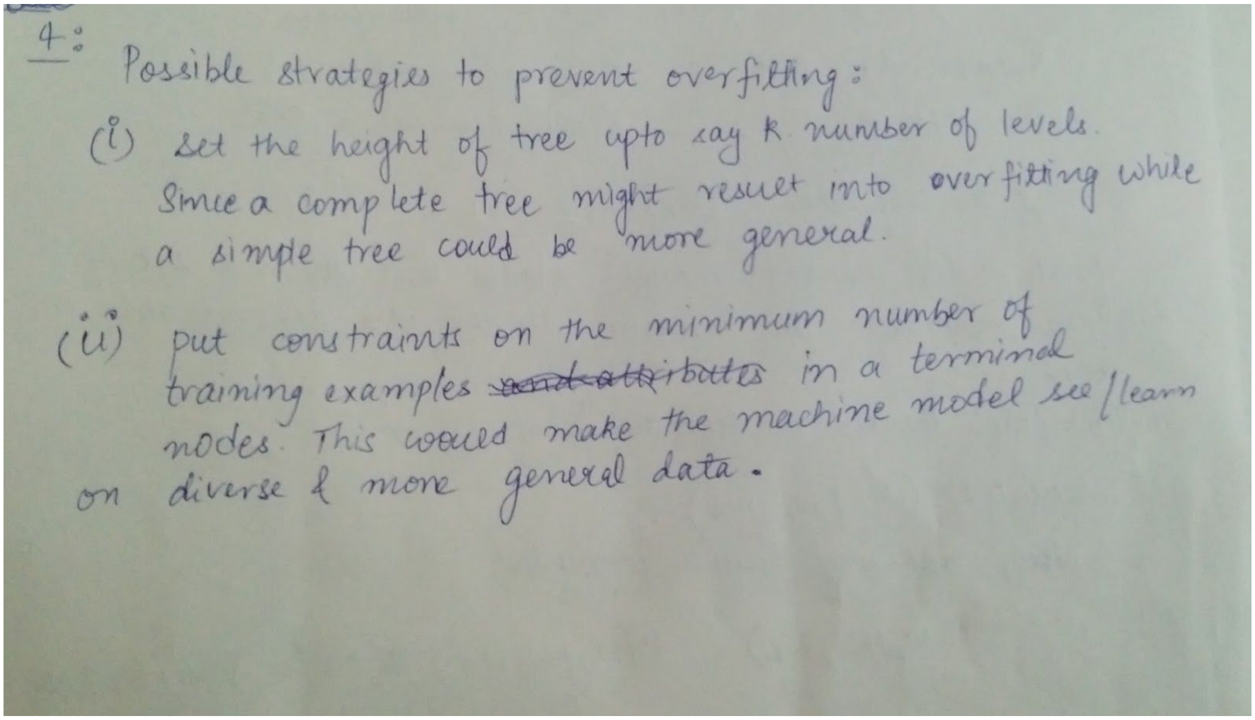


Let's test on D8−D14.

	Pred	Actual
D8 :	No	No ✓
D9 :	No	Yes ✗
D10 :	Yes	Yes ✓
D11 :	No	Yes ✗
D12 :	Yes	Yes ✓
D13 :	Yes	Yes ✓
D14 :	No	No. ✓

$\text{Accuracy} = \frac{5}{7} \approx 0.7142$

The thing to observe here was the incorrect input was only when outlook was sunny. This is because, the training set did not include examples where sunny attribute produced yes. Hence, the model failed to classify those instances correctly.

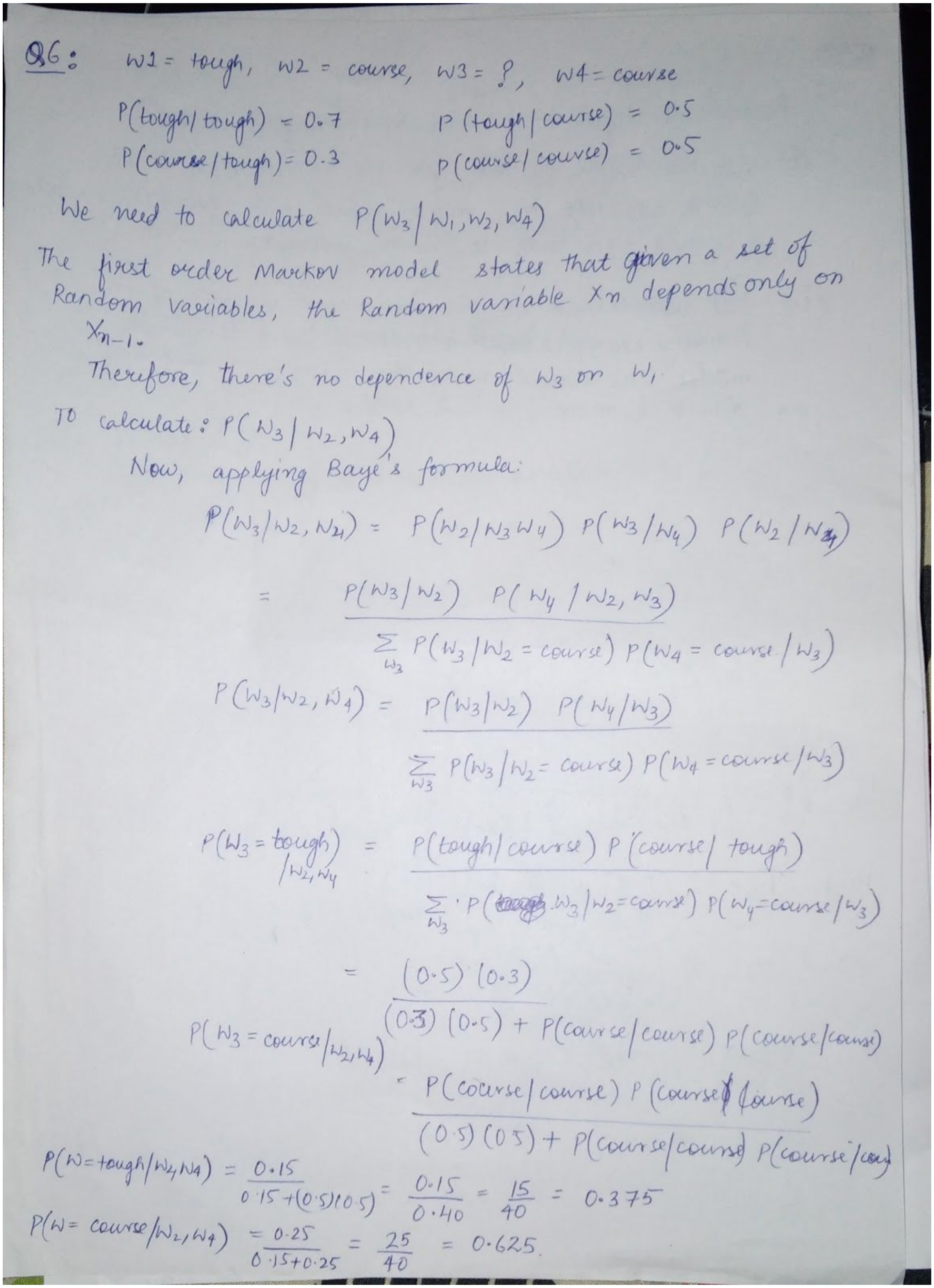


4: Possible strategies to prevent overfitting:

(i) set the height of tree upto say k number of levels. Since a complete tree might resuet into over fitting while a simple tree could be more general.

(ii) put constraints on the minimum number of training examples ~~and attributes~~ in a terminal nodes. This woued make the machine model see/learn on diverse & more general data.

Question 6:



Q6: w1 = tough, w2 = course, w3 = ?, w4 = course

P(tough/ tough) = 0.7 P (tough | course) = 0.5

P(course / tough) = 0.3 P(course/ course) = 0.5

We need to calculate $P(W_3 / W_1, W_2, W_4)$

The first order Markov model states that given a set of Random variables, the Random variable X_n depends only on X_{n-1}.

Therefore, there's no dependence of W_3 on W_1

TO calculate: $P(W_3 | W_2, W_4)$

Now, applying Baye's formula:

$$P(W_3/W_2, W_4) = P(W_2/W_3 W_4)\, P(W_3/W_4)\, P(W_2/W_4)$$

$$= \frac{P(W_3/W_2)\ P(W_4 / W_2, W_3)}{\sum_{W_3} P(W_3/W_2 = \text{course})\, P(W_4 = \text{course}/W_3)}$$

$$P(W_3/W_2, W_4) = \frac{P(W_3/W_2)\ P(W_4/W_3)}{\sum_{W_3} P(W_3/W_2 = \text{course})\, P(W_4 = \text{course}/W_3)}$$

$$P(W_3 = \text{tough} / W_2, W_4) = \frac{P(\text{tough}/\text{course})\ P(\text{course}/\text{tough})}{\sum_{W_3} \cdot P(W_3/W_2 = \text{course})\, P(W_4 = \text{course}/W_3)}$$

$$= \frac{(0.5)(0.3)}{(0.3)(0.5) + P(\text{course}/\text{course})\, P(\text{course}/\text{course})}$$

$$P(W_3 = \text{course}/W_2, W_4) = \frac{P(\text{course}/\text{course})\ P(\text{course}/\text{course})}{(0.5)(0.5) + P(\text{course}/\text{course})\, P(\text{course}/\text{cou}}$$

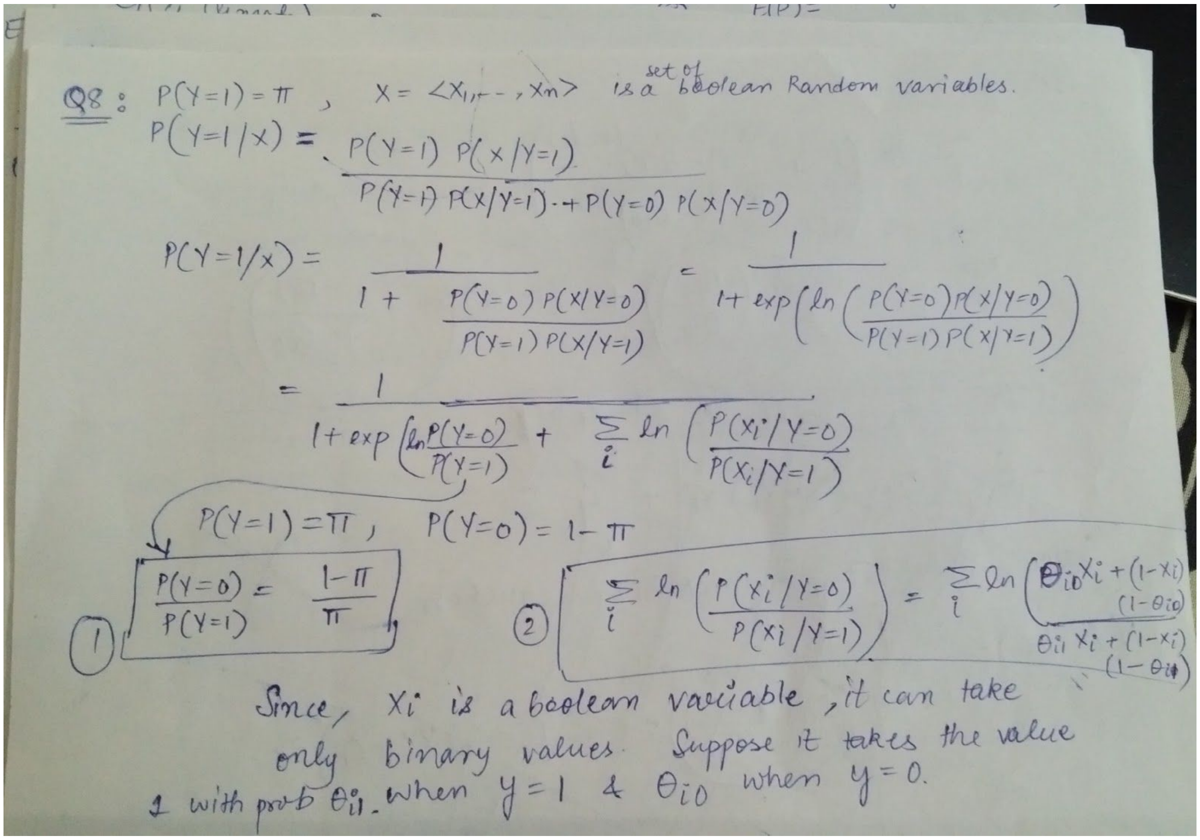
$$P(W = \text{tough}/W_2, W_4) = \frac{0.15}{0.15 + (0.5)(0.5)} = \frac{0.15}{0.40} = \frac{15}{40} = 0.375$$

$$P(W = \text{course}/W_2, W_4) = \frac{0.25}{0.15 + 0.25} = \frac{25}{40} = 0.625.$$

Question 7:

a) Since logistic regression treats the features independently and decision trees do not, decision trees have the power to comprehend complicated relationships among the features.
b) Decision trees split data at each level and form complicated relationships among features for classification and therefore stand a high chance to overfit the data. On the other hand, logistic regression having only one parameter is less likely to overfit.
c) Yes, decision trees can classify these vectors. The data, being linearly separable, for each x1 there is a threshold for x2. Therefore, splitting the data on x1 could be one strategy. Using this strategy, the depth of the tree will be log(n). The number of nodes will be >=1. The depth will be O(log n)
d) Yes, decision trees will still be able to classify these points. Since the data is not linearly separable, we cannot assume the threshold on x2 now. Hence, to classify we'll have to assume different values for x2. In this case, the depth will still be O(log n)

Question 8:



Q8 : $P(Y=1)=\pi$, $X=\langle X_1,--,X_n\rangle$ is a set of boolean Random variables.

$$P(Y=1/x) = \frac{P(Y=1)\,P(x/Y=1)}{P(Y=1)\,P(x/Y=1) + P(Y=0)\,P(x/Y=0)}$$

$$P(Y=1/x) = \frac{1}{1+\frac{P(Y=0)\,P(X/Y=0)}{P(Y=1)\,P(X/Y=1)}} = \frac{1}{1+\exp\left(\ln\left(\frac{P(Y=0)P(x/Y=0)}{P(Y=1)P(x/Y=1)}\right)\right)}$$

$$= \frac{1}{1+\exp\left(\ln\frac{P(Y=0)}{P(Y=1)} + \sum_i \ln\left(\frac{P(X_i/Y=0)}{P(X_i/Y=1)}\right)\right)}$$

$P(Y=1)=\pi$, $P(Y=0)=1-\pi$

$$① \boxed{\frac{P(Y=0)}{P(Y=1)} = \frac{1-\pi}{\pi}}$$

$$② \boxed{\sum_i \ln\left(\frac{P(X_i/Y=0)}{P(X_i/Y=1)}\right) = \sum_i \ln\left(\frac{\theta_{i0}^{X_i} + (1-\theta_{i0})^{(1-X_i)}}{\theta_{i1}^{X_i} + (1-\theta_{i1})^{(1-X_i)}}\right)}$$

Since, X_i is a boolean variable, it can take only binary values. Suppose it takes the value 1 with prob θ_{i1}. when $y=1$ & θ_{i0} when $y=0$.

Solving 2 :

$$\sum_i \ln\left(\frac{(\theta_{i1})^{x_i}(1-\theta_{i1})^{(1-x_i)}}{(\theta_{i0})^{x_i}(1-\theta_{i0})^{1-x_i}}\right)$$

$$\Rightarrow \sum_i x_i \overbrace{\ln\left(\frac{\theta_{i1}}{\theta_{i0}}\right)}^{C_2} + (1-x_i)\overbrace{\ln\left(\frac{1-\theta_{i1}}{1-\theta_{i0}}\right)}^{C_1}$$

~~$\Rightarrow \sum_i C_2 x_i + (1-x_i) C_1$~~

~~$\Rightarrow \sum_i (C_2 - C_1) x_i + C_1$~~

~~Now, substituting it back in our equation~~

~~$P(Y=1/X) = \frac{1}{1+\exp\left(\overbrace{\ln\left(\frac{1-\pi}{\pi}\right)}^{C_3} + \sum_i (C_2-C_1)x_i + C_1\right)}$~~

~~$= \frac{1}{1+\exp\left(C_3 + \sum_{i=}(C_2-C_1)x_i + \sum_i C_1\right)}$~~

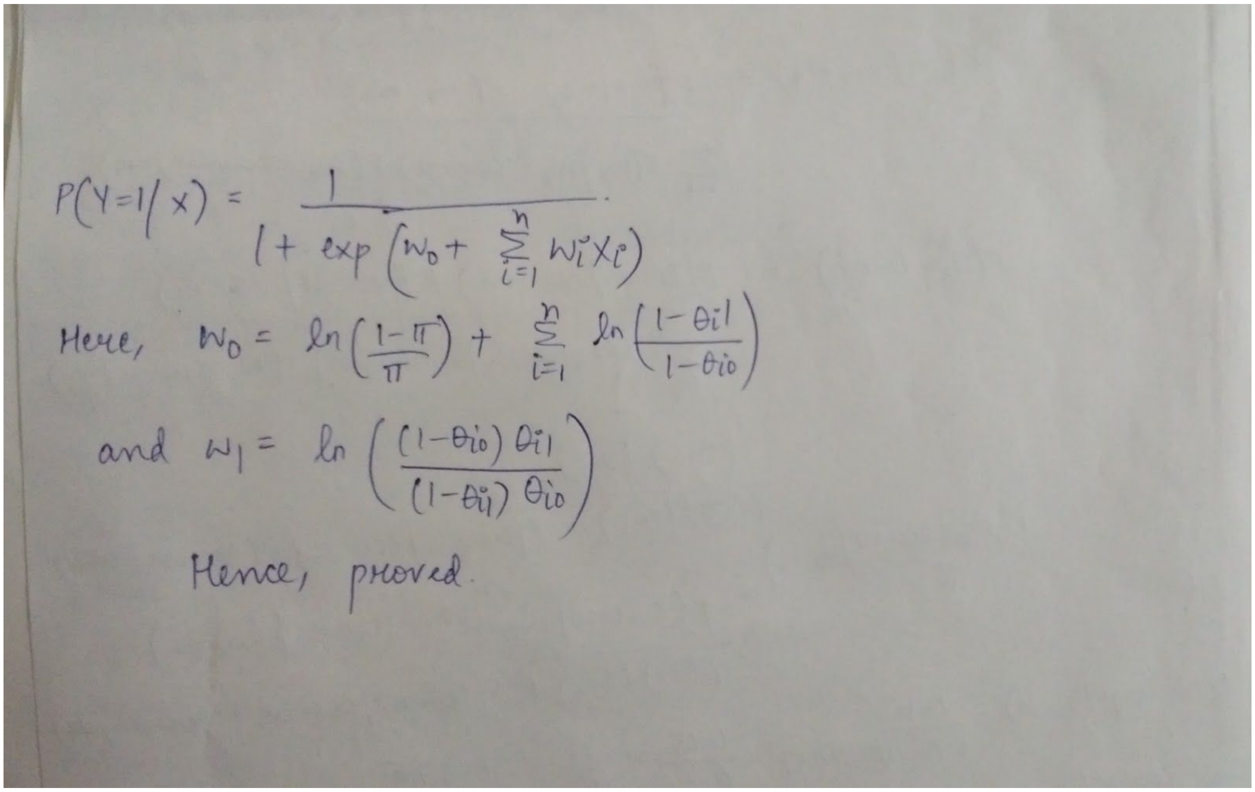
~~$= \frac{1}{1+\exp\left(w_0 + \sum_{i=1}^{n} w_i x_i\right)}$ where~~

$$\Rightarrow \sum_i x_i \ln\left(\frac{\theta_{i1}}{\theta_{i0}}\right) + (1-x_i)\ln\left(\frac{1-\theta_{i1}}{1-\theta_{i0}}\right)$$

$$\Rightarrow \sum_i x_i \ln\left(\frac{(1-\theta_{i0})(\theta_{i1})}{(1-\theta_{i1})(\theta_{i0})}\right) + \ln\left(\frac{1-\theta_{i1}}{1-\theta_{i0}}\right)$$

Substituting back,

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



$$P(Y=1/x) = \frac{1}{1+\exp\left(w_0 + \sum_{i=1}^{n} w_i x_i\right)}.$$

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

and $w_1 = \ln\left(\frac{(1-\theta_{i0})\,\theta_{i1}}{(1-\theta_{i1})\,\theta_{i0}}\right)$

Hence, proved.

$$P(X_1 = x_{1j} \mid Y = y_k) = \theta_{1k}^{x_{ij}} (1-\theta_{1K})^{1-x_{ij}}$$

Likelihood:

$$L(\theta_{1k}) = \prod_{j=1}^{M} P(X_{ij} / \theta_{1k})$$

$$I(Y^j = y_k) = 1 \quad \text{if} \quad Y^j = y_k \text{, otherwise } 0.$$

Here j is the training sample.

$$L(\theta_{1k}) = \ln \sum_{j=1}^{M} P(X_{1j} / \theta_{1k})^{I(Y^j = y_k)}$$

$$= \sum_{j=1}^{M} I(y^j = y_k) \ln (X_{ij} \mid \theta_{1k})$$

$$= \sum_{j=1}^{M} I(Y^j - y_k) \left[x_{ij} \ln \theta_{1k} + (1-x_{ij}) \ln (1-\theta_{1k}) \right]$$

$$\frac{\partial L(\theta_{1k})}{\partial \theta_{1k}} = \frac{1}{\theta_{1k}} \boxed{\sum_{j=1}^{M} I(Y^j = y_k) x_{ij}} + \frac{1}{1-\theta_{1k}} \boxed{\sum_{j=1}^{M} I(Y^j = y_k)(1-x_{ij})}$$

Setting the derivate to zero,

$$\boxed{\theta_{1k_0} = \frac{\sum_{j=1}^{M} I(Y^j = y_k) x_{ij}}{\sum_{j=1}^{M} I(Y^j = y_k)}}$$

Hence, the parameter can be represented through the expression written above