Data Mining Assignment #4

Data Mining Assignment 4

TEAM MEMBERS

1. Nithyadharshni Sampathkumar

2. Saai Krishnan Udayakumar

3. Sibi Senthur Muthusamy

Problem 1:

We try to normalize age → Age: (70-20)/(80-20) → **0.83333**

Similarly, Income → (50-10)/(110-10) → **0.4**

Calculating the Output from the First Hidden Layer,

O1= 4*0.83333+2*0.4-3 → 1.13332

Calculating the Output from the Second Hidden Layer,

O2= -2*0.83333+2*0.4+1 → 0.13334

So, the Output Function would be, **y=1-3*f(O1) + 3*f(O2)**

Each neuron in the hidden and output layer is a neuron with transfer function, 1/(1+e^-y)

So, f(1.13332) → 1/(1+e^(-1.13332)) → **0.75645** and , f(0.13334) → 1/(1+e^(-0.13334)) → **0.53332**

Y=1-3*0.75645+3*0.53332 → 0.33061 -→ 1/(1+e^-y) -→ 1/(1+e^-(0.33061)) → 0.5819

Hence, we can conclude that the Output is "Maybe" (Since we are given that the output range in between 0.25 – 0.7499 is Maybe)

Problem 2:

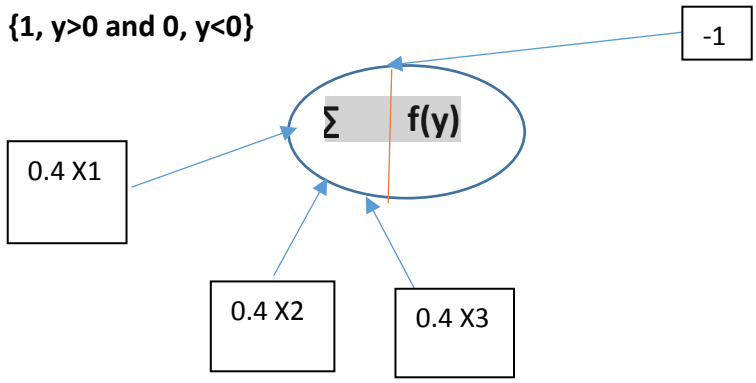
a) (x1 AND x2) AND x3 --→

(Logic)

For the AND function we use equal weights to the three inputs such that sum of any two weights is still less the Initial absolute value of weight (or the Intercept).

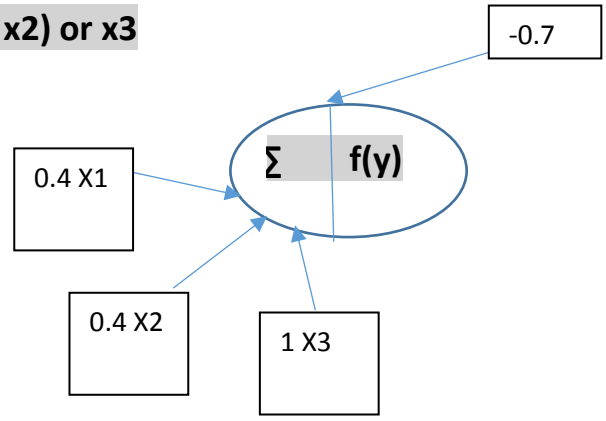
Let us Consider that the Output Layer is a neuron with Step Function,

f(y) = **{1, y>0 and 0, y<0}**



Hence the value of y= 0.4 (x1) + 0.4 (x2) + 0.4 (x3) -1

b) (x1 AND x2) or x3

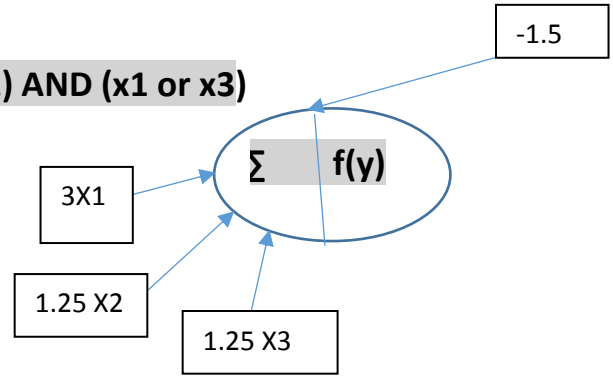


Hence the value of y= 0.4(x1) + 0.4(x2) + (x3) -0.7

Logic:

We construct the function, such that x3 is carriers larger weight than the x1 and x2, since x3 is the OR function and flag set at x3 is likely to turn the step function on. Also, the individual weights of x1 and x2 must be less than the absolute initial weight (Wo) since both the variables are required for the AND function to be turned on and turning off x1 or x2 must yield 0 with our step function.

c) (x1 or x2) AND (x1 or x3)



Hence the value of y= 3(x1) + 1.25(x2) + 1.25 (x3) -1.5

(Logic)

We construct the function such the weight of x1 (Since it is involved in both the functions) is greater than the other individual absolute weights. Also, the combined weight of x2 and x3 must be greater than the absolute initial weight since if both x2 and x3 are turned on, the function must yield a positive response. Also, the individual weight of x2 or x3 must be less than the initial intercept value (or initial weight)

Problem 3:

We've been given that there are a total of 1000 pieces of fruit. Out of which we know that 500 are Bananas, 300 are Orange and 200 are Other fruits.

Let's apply the Naïve Bayes Model,

Naïve Bayes assumes that the features are independent of each other and all the features independently contribute to the fact the fruit is a Banana or an Orange or other fruit,

Let Consider Case (1) for Banana:

Probability (Fruit=Banana | {Type=Long, Sweet, Green}) = (Probability ({Type=Long, Sweet, Green} | Fruit=Banana) * Probability (Fruit=Banana)) / (Probability {Type=Long, Sweet, Green})

Probability (Fruit=Banana | {Type=Long, Sweet, Green}) =

(Probability ({Type=Long}/{Fruit=Banana}) * Probability ({Type=Sweet}/{Fruit=Banana}) * Probability ({Type=Green}/{Fruit=Banana}) * Probability (Fruit=Banana)) / (Probability {Type=Long, Sweet, Green})

Based on our Naïve assumption of Independence among features

Given Data,

Probability ({Type=Long}/{Fruit=Banana}) = 401/503 = 0.7972

Probability ({Type=Sweet}/{Fruit=Banana}) = 351/503 = 0.69781

Probability ({Type=Green}/{Fruit=Banana}) = 1/503 = 0.001988 (Using Laplace-1 Estimator)

(We don't want the probability to be zero) → (We have built the Green probability based on our assumption that the classes in the Features (Colors attribute) are Yellow or not (which does not include green) and hence the frequency of Type → Green/Banana is zero which we handle it using Laplace-1 Estimator) I am using the same rationale for other type of fruits as well.

Probability (Fruit=Banana)) = 500/1000 = 0.5 (We only apply Laplace Estimator on Conditional Probabilities) (Also, based on Professor's remark)

(Probability {Type=Long, Sweet, Green}) = P1 (Let assume it as P1 --→ we don't need to compute this probability since we have been asked to classify the fruit alone, hence the numerator value(s) alone is enough to gauge the type of fruit, however we calculate P1 at last step to get a better sense of the individual probability values.

Probability (Fruit=Banana | {Type=Long, Sweet, Green}) = [(0.7972)*(0.69781)*(0.001988)*(0.5)]/(P1) - → **0.000553/(P1)**

Let Consider Case (2) for Orange:

Probability (Fruit=Orange | {Type=Long, Sweet, Green}) = (Probability ({Type=Long, Sweet, Green} | Fruit=Orange) * Probability (Fruit=Orange)) / (Probability {Type=Long, Sweet, Green})

Given Data,

Probability ({Type=Long}/{Fruit=Orange}) = 0/300 = 0 ~ (We don't want the probability to be zero)

Probability ({Type=Sweet}/{Fruit=Orange}) = 150/300 = 0.5

Probability ({Type=Green}/{Fruit=Orange}) = 0/300 = 0 ~ (We don't want the probability to be zero)

Adjusting the probability,

Probability ({Type=Long}/{Fruit=Orange}) = 0/300 = 0 ~ 1/303 ~ 0.0033

Probability ({Type=Sweet}/{Fruit=Orange}) = 150/300 ~ 151/303 ~ 0.4983

Probability ({Type=Green}/{Fruit=Orange}) = 0/300 = 0 ~ 1/303 ~ 0.0033

Probability (Fruit=Orange)) = 300/1000 = 0.3

Probability (Fruit=Orange | {Type=Long, Sweet, Green}) = [(0.0033)*(0.4983)*(0.0033)*(0.3)]/(P1) → **0.00000163/(P1)**

Let Consider Case (3) for Other Fruits:

Probability (Fruit=Other | {Type=Long, Sweet, Green}) = (Probability ({Type=Long, Sweet, Green} | Fruit=Other) * Probability (Fruit=Other)) / (Probability {Type=Long, Sweet, Green})

Given Data,

Probability ({Type=Long}/{Fruit=Other}) = 101/203 = 0.4975

Probability ({Type=Sweet}/{Fruit=Other}) = 151/203 = 0.7438

Probability ({Type=Green}/{Fruit=Other}) = 1/203 = 0.00493

Probability (Fruit=Other) = 200/1000 = 0.2

Probability (Fruit=Other | {Type=Long, Sweet, Green}) = [(0.4975)*(0.7438)*(0.00493)*(0.2)]/(P1)
→ **0.000364/(P1)**

P1 is nothing but sum of numerators of all three cases, (0.000364+0.00000163+0.000553)
→ 0.00091864

Final Prediction/Probability

Fruit	Probability	Prediction
Banana	0.6019	Yes (Fruit is Banana, highest Probability among the three)
Orange	0.00177	No
Other Fruits	0.39623	No

So the fruit that is Long, Sweet and Green is Banana

Problem 4:

From the data we see that there is a total of 285 patient records.

We observe that there are **201** instances of "non-recurrences" and **84** instances of "recurrences" of cancer based on the 9 attributes.

Using the four attributes "menopause", "node-caps", "deg-malig" and "irradiat", we construct the probability table as follows:

Probability (Recurrence | {menopause=premenopausal, node-caps=yes, deg_malig=high, irradiat=yes}) = (Probability ({menopause=premenopausal}/{Recurrence}) * Probability ({node-caps=yes}/{Recurrence}) * Probability ({deg-malig=high}/{Recurrence}) * Probability ({irradiant=yes}/{Recurrence}) * Probability (Recurrence))/ (Probability {menopause=premenopausal, node-caps=yes, deg_malig=high, irradiat=yes})

Based on our Naïve assumption of Independence among features

Frequency Table (Menopause)

Attribute	Recurrence	Non-Recurrence	Total
Premenopausal	48	102	150
Ge40	34	94	128
It40	02	05	07
Total	84	201	285

Frequency Table (node-caps)

Attribute	Recurrence	Non-Recurrence	Total
Yes	31	25	56
No	50	171	221
Total	81	196	277

→ Seems like we are having missing values with respect to the node-caps, based on Professor's reading – Missing values in Naïve Bayes could be handled the following way

"If a value is missing in a training instance, it is simply not included in the frequency counts, and the probability ratios are based on the number of values that actually occur rather than on the total number of instances."

Frequency Table (Irradiation)

Attribute	Recurrence	Non-Recurrence	Total
Yes	31	37	68
No	53	164	217
Total	84	201	285

Frequency Table (degree of malignancy)

Attribute	Recurrence	Non-Recurrence	Total
Low	12	59	71
High	44	40	84
Medium	28	102	130
Total	84	201	285

Given Data,

Probability ({menopause=premenopausal}/ Recurrence) = 48/84 = 0.5714

Probability ({node-caps=yes}/ Recurrence) = 31/81 = 0.3827

Probability ({deg-malig=high}/ Recurrence) = 44/84 = 0.5238

Probability ({irradiant=yes}/ Recurrence) = 31/84 = 0.3690

Probability (Recurrence)) = 84/285 = 0.2947

(Probability {menopause=premenopausal, node-caps=yes, deg_malig=high, irradiat=yes}) = P1 (Let us assume it as P1 → we don't need to compute this probability since we have been asked to only classify the person, hence the numerator value(s) alone is enough. We however calculate P1 at last step to get a better sense of the individual probability values.

Probability(Recurrence|{menopause=premenopausal, node-caps=yes, deg_malig=high, irradiat=yes})= [(0.5714)*(0.3827)*(0.5238)*(0.3690)*(0.2947)]/(P1) → **0.0124/(P1)**

Given Data,

Probability ({menopause=premenopausal}/ Non-Recurrence) = 102/201 = 0.5074

Probability ({node-caps=yes}/ Non-Recurrence) = 25/196 = 0.1275

Probability ({deg-malig=high}/ Non-Recurrence) = 40/201 = 0.1990

Probability ({irradiant=yes}/ Non-Recurrence) = 37/201 = 0.1840

Probability (Non-Recurrence)) = 201/285 = 0.7052

(Probability {menopause=premenopausal, node-caps=yes, deg_malig=high, irradiat=yes}) = P1

Probability(Non-Recurrence|{menopause=premenopausal, node-caps=yes, deg_malig=high, irradiat=yes})= [(0.5074)*(0.1275)*(0.1990)*(0.1840)*(0.7052)]/(P1) → **0.00167/(P1)**

P1 is nothing but sum of numerators of the two cases, (0.0124+0.00167) → 0.01407

Final Prediction/Probability

Class	Probability	Prediction
Recurrence	0.8813	Yes (Class=recurrence, highest Probability among the two)
Non-Recurrence	**0.1187**	**No**

Hence, a woman who is premenopausal, with node-caps, high degree of malignancy and who has had irradiation is likely to have a recurrence of cancer.

Problem 6:

We know that **w= ∑xyα** for all i's =0 (after applying the Lagrangian constraints and deriving the optimum conditions)

W= 0.414 [4 2.9] + 0 + 0 – 0.018 [2.5 1] + 0 + 0 + 0.018 [3.5 4] + 0 - 0.414 [2 2.1] + 0

2*1 matrix of (xi1, xi2 values)

W= [0.846 0.3852]

W1 **W2**

We perform the normal matrix operation to arrive at the value of W, the first entry signifies the value of w1 and the second the value of w2.

Hence, the Equation of the Hyperplane is y=b+0.846(x1) + 0.3852(x2)

Now, Computing the value of b,

The equation of b is -→ W(transpose)x+ b = 1 , we must essentially look upon support vectors that are on the positive side of the plane to compute the value of b,

Hence, the value of b=1- [0.846 0.3852][3.5 4]

b= 1 – [2.9295+1.528]

b=-3.4575

Equation of the hyperplane y=-3.4575+0.8456(X1) + 0.3852(X2)

b) Distance of the point (x6) from the hyper-plane (line) would be,

Distance from a point to a plane is |a+b(x1)+c(x2)| / (sqrt(b^2+c^2))

Distance (d) = |-3.4575+0.8456(1.9)+0.3852(1.9)|/sqrt(0.8456^2+0.3852^2)

Distance (d) = 1.29601

We know that the point (1.9,1.9) lies on the negative side of the hyper plane. So, we are essentially looking to compute the margin between the hyper plane and the support vectors on the negative side of the hyper plan.

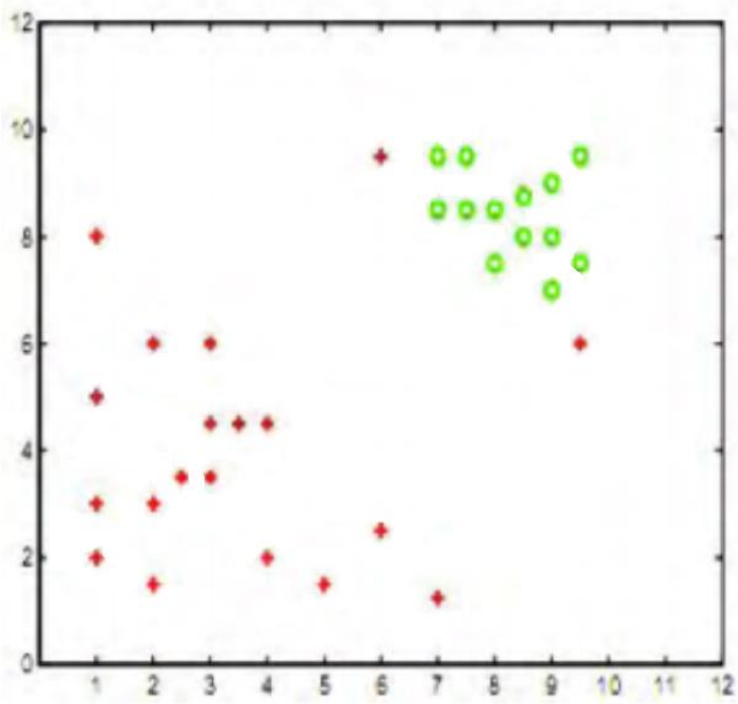
We know that the Margin is --→ 2/|W| --→ For, one side of the hyper plane, margin reduces to 1/|W| -→ 1/|0.8456^2+0.3852^2| -→ 1/|0.8634| -→ 1.15821

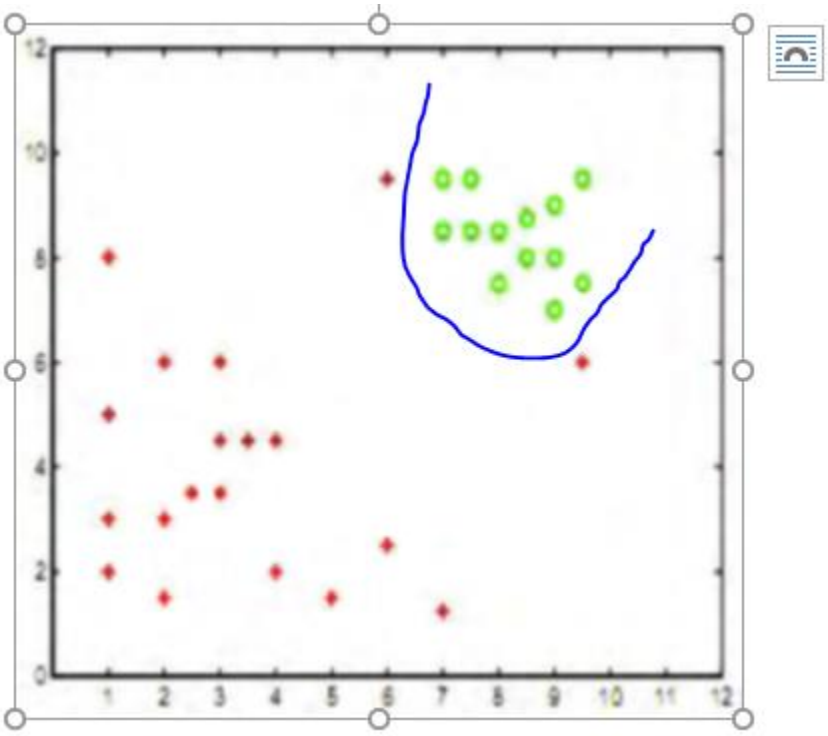
Margin (M) = 1.15821

From our results we can infer that Margin M is less than the Distance of the point from the hyperplane, hence we can conclude the point **(x6) -→ (1.9,1.9) lies outside the margin.**

(c) Substituting the point (3,3) in the hyper-plane equation we get y=-3.4575+0.846(3) +0.3852(3) -→ -3.4575+2.538+1.1556 → 0.2361 which is greater than 0, hence the point lies on the positive side of the hyper plane. **(Belongs to class Positive)**

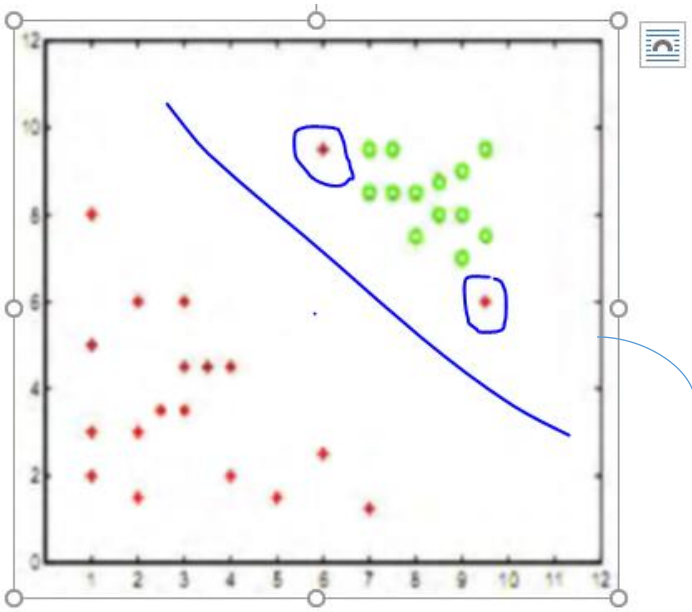
Problem 5





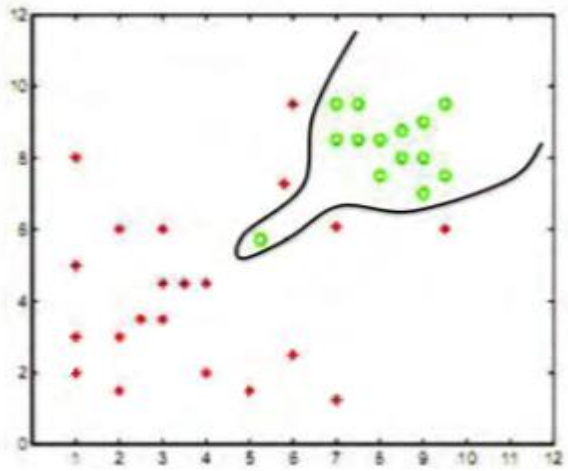
(Question #5 (a))

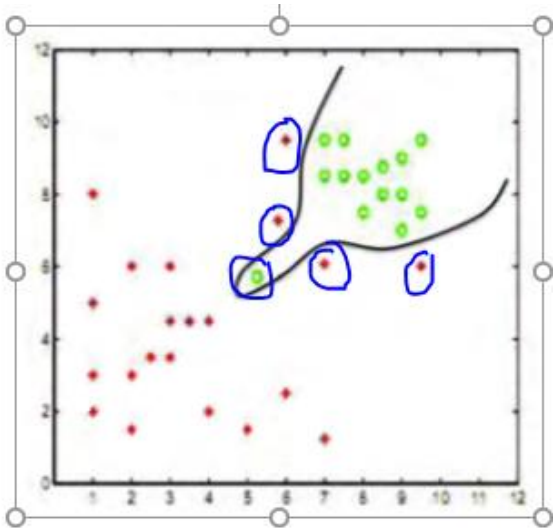
a) **For larger values of Cost Parameter C, the penalty for misclassification is larger** and hence we would be having stricter (or smaller) margins (in our case we will be having stricter margins) so as to allow fewer or no misclassification errors (less error). A wider margin is likely to bring in few misclassifications which we need to avoid for larger C values as the cost for penalty with respect to misclassification instances is larger. Essentially we will be having a parabola classifying the green points and red points perfectly. Overall, the emphasis is on making less misclassification errors.



We allow a few misclassification points and our hyperplane tends to be linear.

b) For c=0, we try to maximize the margin between the points while we can afford to misclassify a few points. So, essentially the penalty for misclassification is minimum and we will have a linear classifier classifying the points (shown above) while misclassifying a few points on either side.





POSSIBLE PROBLEM → Overfitting

c) The model's separating hyperplane is too generic to the training data as shown above and is highly susceptible to few of the training points which are closer to the hyper plane. Hence, the model is highly complex and vulnerable to **Overfitting problem and hence we might be getting low accuracy in the (unknown) test data**.