HOMEWORK - 2 EE 559 ML -> 1

Sudesh Kumar
Santhosh Kumar
4166249920

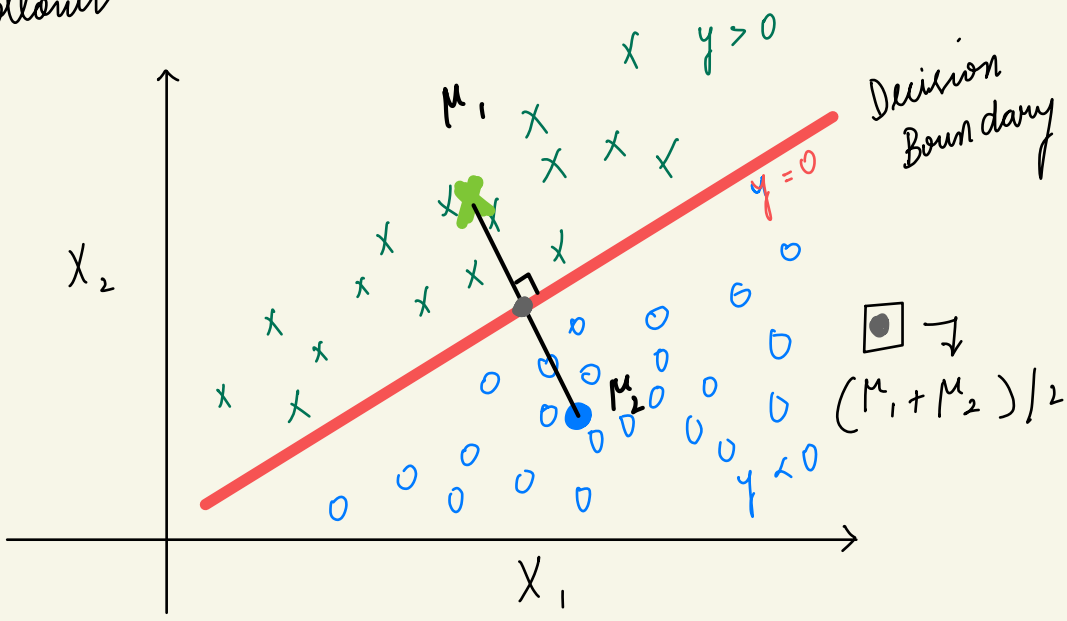
1. For a 2-class nearest-means classifier (NMC) based on D features, and for given mean vectors μ_1 and μ_2

We know the dimensions of the sample means if we have D features and 2 classes in,

sample_means.shape = (2 x D)

$$\mu_1 = [x_1, x_2, \ldots x_D]$$
$$\mu_2 = [x_1, x_2, \ldots x_D]$$

If we have 2 classes and 2 features, the decision boundary can be plotted as follows



sample_means = [[x_1, x_2], → μ_1
[x_1, x_2] → μ_2
]

It's clearly seen that the decision boundary is perpendicular to the line joining the sample means μ_1 and μ_2 and passes through the mid-point of this line.

The mid-point is given by,

$$m = (\mu_1 + \mu_2)/2$$

$$m = (x_1^{(1)} + x_1^{(2)})/2, \quad (x_2^{(1)} + x_2^{(2)})/2$$

Nearest Means Classifier for 2 classes and D dimensional input features:

So, the decision boundary is a hyperplane with (D-1) dimensions.

- The weights vector, w determines the direction of the decision boundary because it is the normal vector to this hyperplane.
- The bias or offset term, w_0 determines the location of the hyperplane.

Hence,

The decision boundary is represented by,

$$g(x) = y$$

$$\Rightarrow y = w^T x + w_0 = 0$$

(w: normal vector; w_0: bias term)

Dimensions:

- w vector → $(1 \times D)$ [row vector]
- w_0 → $(1,)$ [Scalar]
- μ_1 → $(1, D)$
- μ_2 → $(1, D)$

We know that,

If we have two vectors x and y, the difference between them $x - y$ points in the direction $\perp^n$ to the line connecting x and y.

Now, w vector is the normal to y.

So,

$$\boxed{w = \mu_1 - \mu_2}$$

We know, y passes thro the mid-point $x = (\mu_1 + \mu_2)/2$. Hence, the bias term w_0 is given by,

$$w_0 = -w(\mu_1 + \mu_2)/2$$

$$\Rightarrow \boxed{w_0 = -\frac{1}{2}\left[\|\mu_1\|^2 - \|\mu_2\|^2\right]}$$

a] The decision boundary is given by,

$$g(x) = y = w^T x + w_0 = 0$$

Decision Rule :-

A test data point x_test is assigned to C_1 if,

$$y > 0 \quad (or) \quad w^T x\_test > 0$$

$$\Rightarrow \| x\_test - \mu_1 \|_2 > \| x\_test - \mu_2 \|_2$$

A test data point x_test is assigned to C_2 if,

$$y < 0 \quad (or) \quad w^T x\_test < 0$$

$\| x \|_2 \Rightarrow L_2$ norm of x

$$\Rightarrow \| x\_test - \mu_1 \|_2 < \| x\_test - \mu_2 \|_2$$

If x_test lies on the decision boundary, $y = 0$, it belongs to neither C_1 nor C_2

In practice, x_test is assigned to C_1 if

$$y \geq 0 \quad (or) \quad w^T x\_test \geq 0 \quad \| x\_test - \mu_1 \|_2 \leq \| x\_test - \mu_2 \|_2$$

(b) If the classifier is linear, starting from the general expression for a 2-class discriminant function g(x) for a linear classifier, find an expression for the weights for the NMC in terms of the mean vectors μ_1, μ_2

The expression for the weights for the NMC is given by,

$$w = (\mu_1 - \mu_2)$$

The expression for the bias term for the NMC is given by,

$$w_0 = -\frac{1}{2}\left[\|\mu_1\|^2 - \|\mu_2\|^2\right]$$

d] Given $x \in \Gamma_k$ iff $k = \text{argmax}_m (g_m(x))$ $m = 1, 2, \ldots C$

So, basically for $C > 2$ and D features, there will be C discriminant functions which are passed to argmin() in the classical Nearest Means Classifier.

Here,

$$k = \text{argmax}(g_1(x), g_2(x), \ldots g_C(x))$$

Each $g_m(x)$ where $m \in [1, 2, \ldots, C]$ represents the linear function for class m.

The decision boundary between class C_i and C_j is given by,

$$g_i(x) = g_j(x)$$

It corresponds to a $(D-1)$ dimensional hyperplane defined by, ↗ bias of $g_i(x)$

$$(w_i - w_j)^T x + (w_{0i} - w_{0j}) = 0$$

↗ bias of $g_j(x)$

Here,

$g_i(x)$ is given by,

$$g_i(x) = w_i^T x + w_{0i}$$

From our findings,

$$w_i = \mu_i - \bar{\mu}_i$$

$$w_{0i} = -\frac{1}{2}\left(\|\mu_i\|^2 - \|\bar{\mu}_i\|^2\right)$$

$$g_i(x) = g_j(x)$$

$$w_i^T x + w_{0i} = w_j^T x + w_{0j}$$

$$(\mu_i - \bar{\mu}_i)^T x - \frac{1}{2}\left(\|\mu_i\|^2 - \|\bar{\mu}_i\|^2\right) = (\mu_j - \bar{\mu}_j)^T x - \frac{1}{2}\left(\|\mu_j\|^2 - \|\bar{\mu}_j\|^2\right)$$

$$-\frac{1}{2}\|\mu_i\|^2 + \mu_i^T x = -\frac{1}{2}\|\mu_j\|^2 + \mu_j^T x$$

$$(\mu_i - \mu_j)^T x - \frac{1}{2}\left(\|\mu_i\|^2 - \|\mu_j\|^2\right) = 0$$

We have considered the terms depending only on class i and class j.

Decision boundary between class C_i and C_j is given by,

$$g_i(x) = g_j(x)$$

$$(w_i - w_j)^T x + (w_{0i} - w_{0j}) = 0$$

$$\boxed{(\mu_i - \mu_j)^T x - \frac{1}{2}\left(\|\mu_i\|^2 - \|\mu_j\|^2\right) = 0} \rightarrow (1)$$

Since, we must be considering the $\text{argmax}_m\,(g_m(x))$ we must negate our expression in order to get minimum distance

$$g_m(x) = -\left(\mu_m^T x - \frac{1}{2}\|\mu_m\|^2\right)$$

$$g_m(x) = \frac{1}{2}\|\mu_m\|^2 - \mu_m^T x$$

Decision Rule :-

$g_i(x) > g_j(x) \rightarrow x$ belongs to class i

$g_i(x) < g_j(x) \rightarrow x$ belongs to class j

e]

Yes, $g_m(x)$ is linear

Since, $g_m(x)$ is a linear function of x.

Expressions for weights and biases.

$$w_{0m} = \frac{1}{2} \|\mu_m\|^2$$

$$w_m = -\mu_m$$

f] Is multiclass NMC an example of the MVM method?

Yes, multiclass NMC is an example of the MVM method.

Reference of MVM from Bishop :-

We can avoid these difficulties by considering a single K-class discriminant comprising K linear functions of the form

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0} \tag{4.9}$$

and then assigning a point $\mathbf{x}$ to class $\mathcal{C}_k$ if $y_k(\mathbf{x}) > y_j(\mathbf{x})$ for all $j \neq k$. The decision boundary between class $\mathcal{C}_k$ and class $\mathcal{C}_j$ is therefore given by $y_k(\mathbf{x}) = y_j(\mathbf{x})$ and hence corresponds to a $(D-1)$-dimensional hyperplane defined by

$$(\mathbf{w}_k - \mathbf{w}_j)^{\mathrm{T}}\mathbf{x} + (w_{k0} - w_{j0}) = 0. \tag{4.10}$$

The above is the decision rule for MVM.

Since in multiclass NMC defined by $k = \text{argmax}_m(g_m(x))$, we basically choose the class for which discriminant function gives the maximum value.

Both decision rules are the same. Hence, NMC is an example of MVM method.

2. C-class Nearest Mean Classifier (NMC) for C classes and D features.

(a) i] Report the classification accuracy on the Training set.

```
[5]  ✓ 0.1s
...  Accuracy for the Training set is: 85.23809523809524%
     Classification Error Rate for the Training Set is: 14.761904761904763%
```

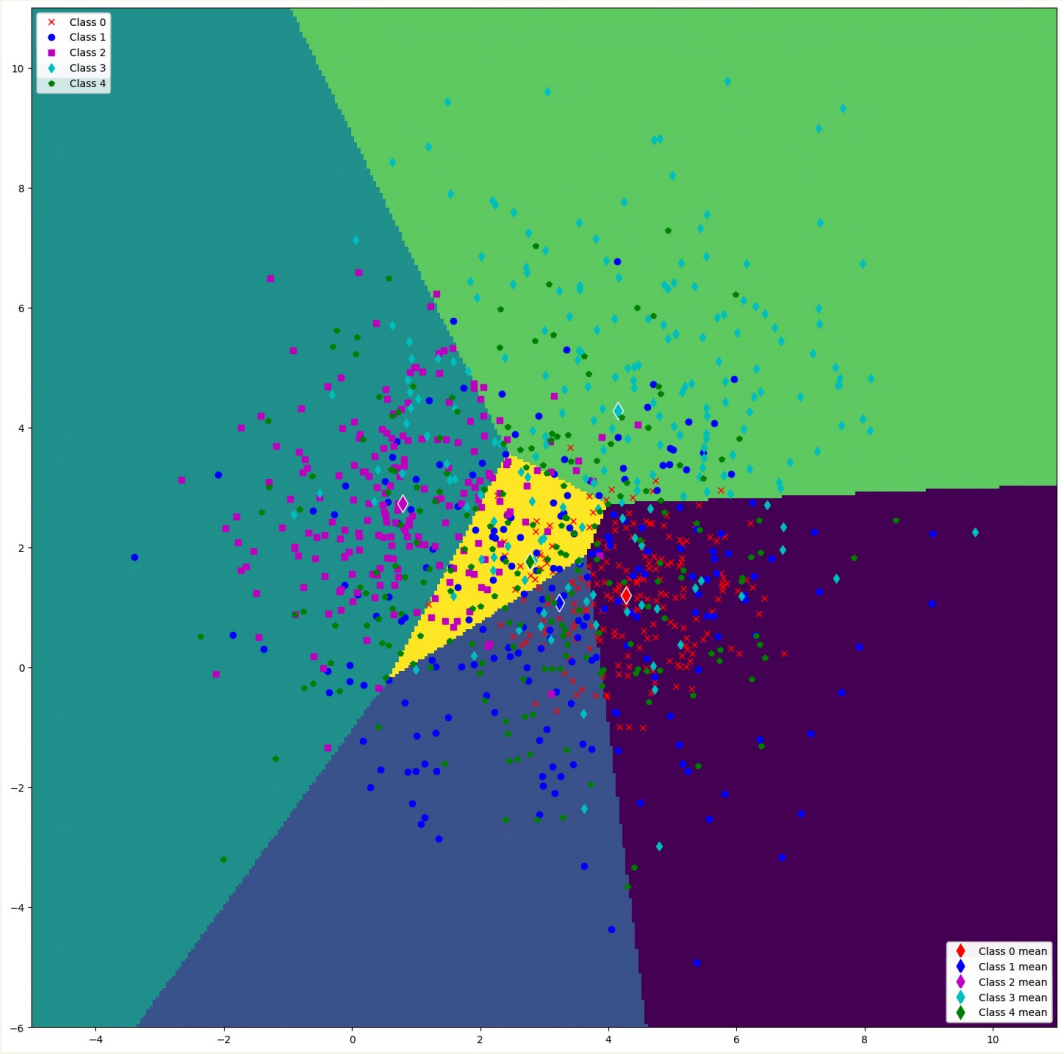
(a) ii] Report the classification accuracy on the Test set.

```
[7]  ✓ 0.1s
...  Accuracy for the Test set is: 82.44444444444444%
     Classification Error Rate for the Test Set is: 17.555555555555554%
```

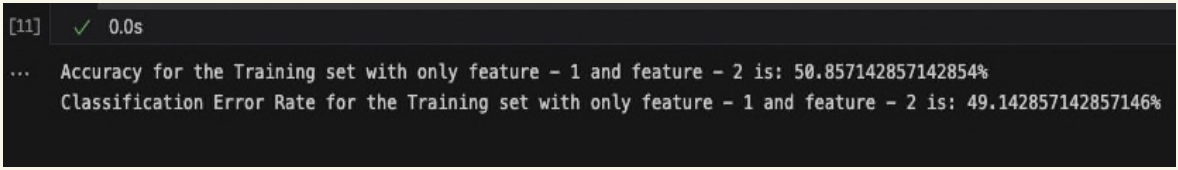
(b) For visualization, run it again using only the following 2 features: X1 and X2.

(b) ii] Plotting Decision Boundaries for the X_train_2features_1 (X1 and X2) in 2D Space.

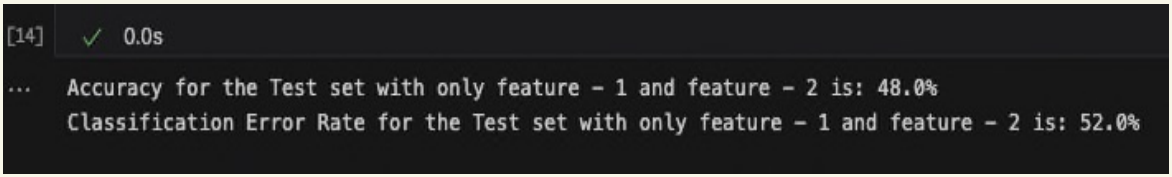
Training Data, Decision Boundaries and Decision Regions for 2 features X_1 and X_2



(b) iii] Report the classification accuracy on the Training set using only the 2 features X1 and X2



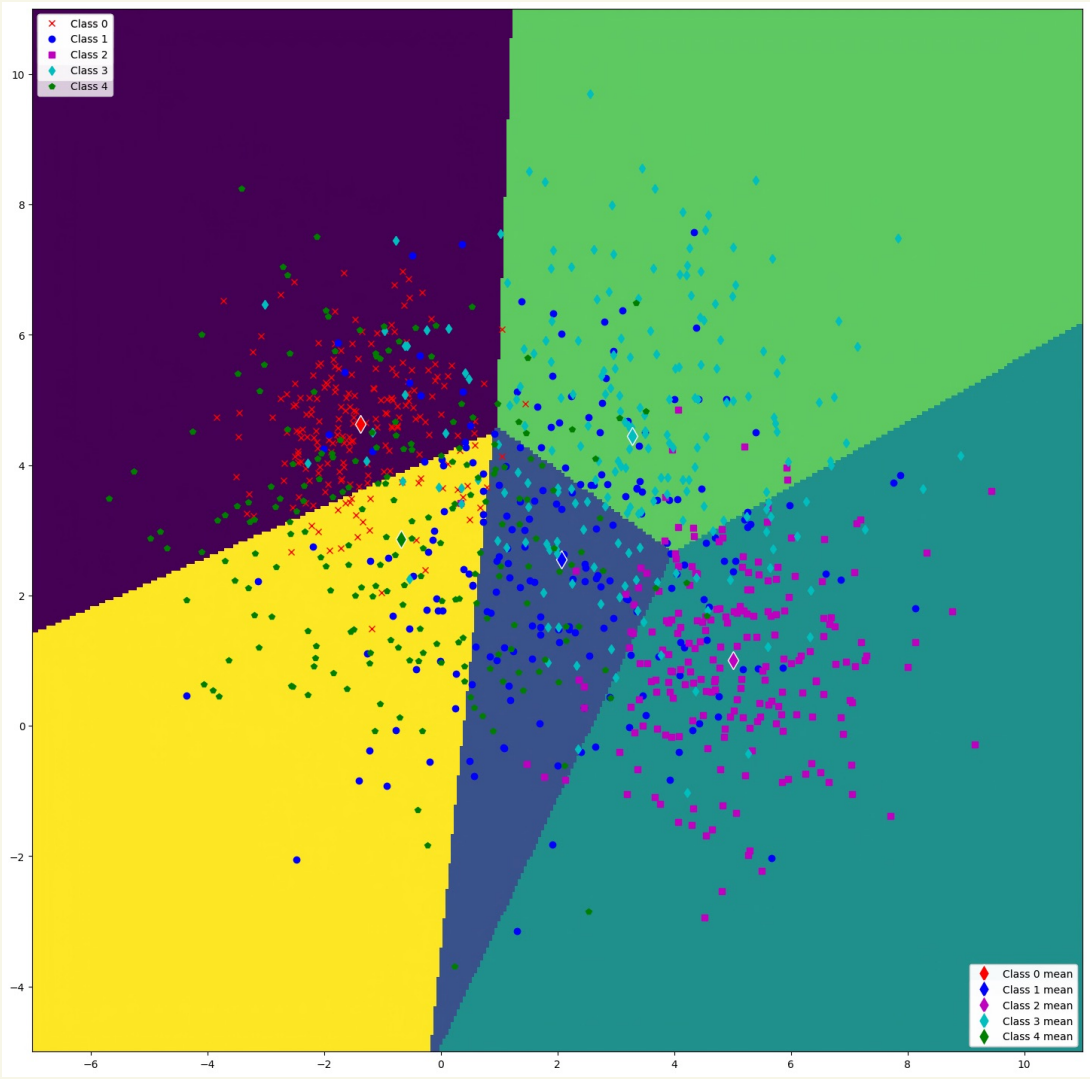
(b) vi] Report the classification accuracy on the Test set using only the 2 features X1 and X2.



(c) Repeat (b) using only the following features: X3 and X4.

(c) ii] Plotting Decision Boundaries for the X_train_2features_2 (X3 and X4) in 2D Space.

Training Data, Decision Boundaries and Decision Regions features X3 and X4



(c) iii] Report the classification accuracy on the Training set using only the 2 features X3 and X4.

```
[18]  ✓ 0.1s

...   Accuracy for the Training set with only feature - 3 and feature - 4 is: 60.952380952380956%
      Classification Error Rate for the Training set with only feature - 3 and feature - 4 is: 39.04761904761905%
```

(c) vi] Report the classification accuracy on the Test set using only the 2 features.

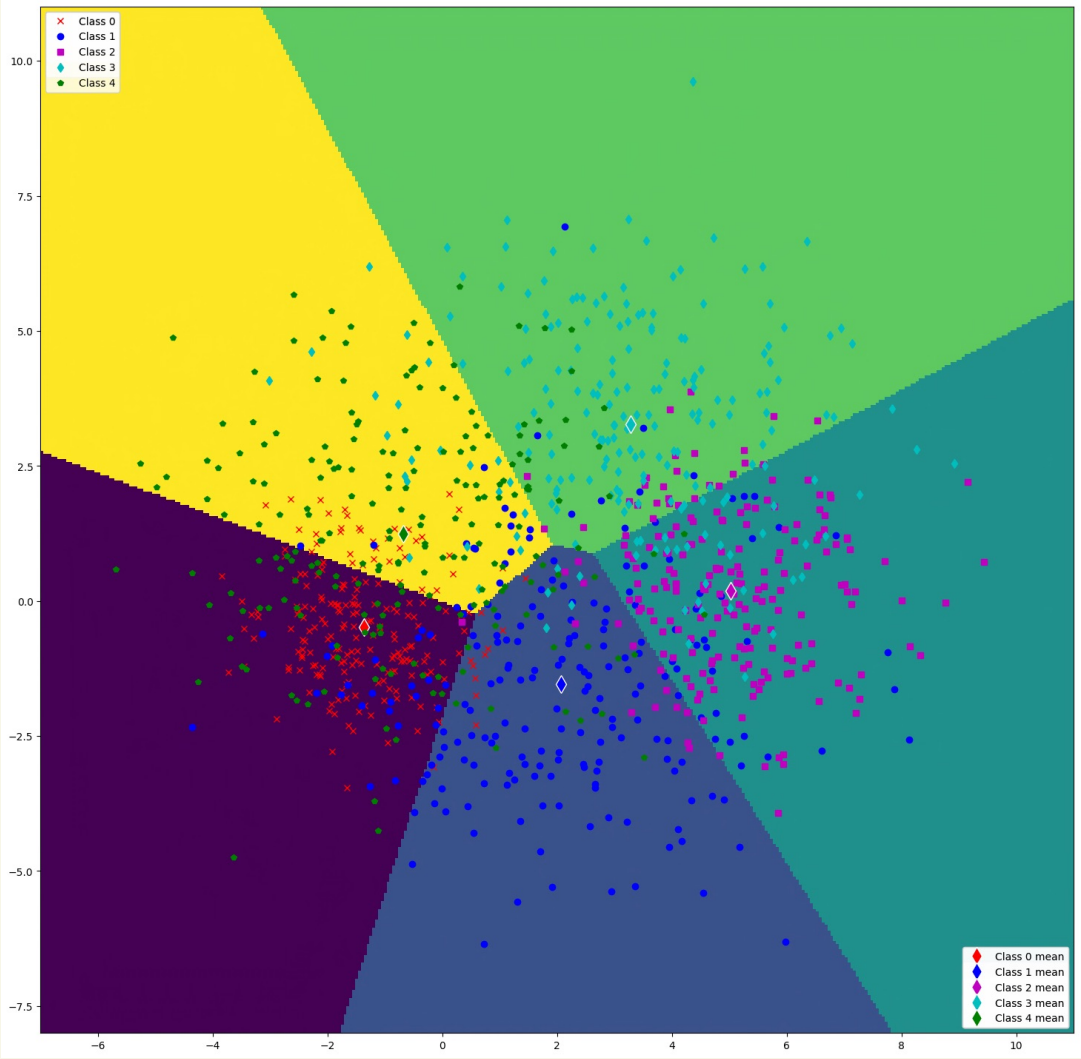
```
[21]  ✓ 0.0s

...   Accuracy for the Test set with only feature - 3 and feature - 4 is: 60.44444444444444%
      Classification Error Rate for the Test set with only feature - 3 and feature - 4 is: 39.55555555555556%
```

(d) Repeat (b) using only the following features: X3 and X7.

(d) ii] Plotting Decision Boundaries for the X_train_2features_3 (X3 and X7) in 2D Space.

Training Data, Decision Boundaries and Decision Regions for features X3 and X7



(d) iii] Report the classification accuracy on the Training set using only the 2 features X3 and X7

```
[25]  ✓ 0.1s

...   Accuracy for the Training set with only feature - 3 and feature - 7 is: 67.61904761904762%
      Classification Error Rate for the Training set with only feature - 3 and feature - 7 is: 32.38095238095238%
```

(d) vi] Report the classification accuracy on the Test set using only the 2 features X3 and X7.

```
[28]  ✓ 0.0s

...   Accuracy for the Test set with only feature - 3 and feature - 7 is: 63.77777777777778%
      Classification Error Rate for the Test set with only feature - 3 and feature - 7 is: 36.22222222222222%
```

(e) Of (b), (c), (d): which gives the best training accuracy? the best test accuracy?

(e) i] Of (b), (c), (d): which gives the best Training accuracy?

Answer : (d)

(e) ii] Of (b), (c), (d): which gives the best Test accuracy?

Answer : (d)

(e) iii] Does the use of all 7 features perform better than the pairs tried in (b), (c), (d)?

Yes , the use of all 7 features perform better!

2 f]

No, I don't see any indeterminate regions

All the decision regions are convex.

3.

This problem uses the notation we used in Lecture 5, and m, $m0$ and m1 are positive integers. For the following computational complexity:

$$p(m) = 10m - 50$$

$p(m) = 10m - 50$

Big O notation :-

This notation provides an upper bound on the growth of an algorithm's running time or space usage. For example, an algorithm with a running time of O(n) grows linearly with the size of the input.

Big Ω notation :-

This notation provides a lower bound on the growth of an algorithm's running time or space usage. For example, an algorithm with a running time of Omega(n) grows at least linearly with the size of the input.

Big θ notation :-

This notation provides a tight bound on the growth of an algorithm's running time or space usage, i.e., it gives both an upper and a lower bound. For example, an algorithm with a running time of Theta(n) grows exactly linearly with the size of the input.

3] a)

Is $p(m) = O(m)$? Yes

$p(m) = O(q(m))$, if there exist positive constants a and m_0 such that $0 \leq p(m) \leq a\,q(m)$ $\forall\ m \geq m_0$

$q(m) = m$

The function $p(m)$ grows at most as fast as m. Hence, its growth rate is no worse than m

Yes, $p(m) = O(m)$

If $a = 1$,

$0 \leq p(m) \leq 1\,q(m)$

$\Rightarrow 0 \leq 10m - 50 \leq 1(m)$

$10m - 50 \leq m \rightarrow (1)$

For larger values of m, $\forall\ m \geq m_0$ (1) does not hold true.

We need a larger a to hold the inequality true.

If we choose $a = 10$,

$\Rightarrow 0 \leq 10m - 50 \leq 10m$

The smallest positive value of m_0 is given by,

$0 \leq 10m - 50$

At $m = m_0$

$50 \leq 10 m_0$

$\Rightarrow 5 \leq m_0$

The smallest positive value of m_0 is 5.

$a = 10$

$m_0 = 5$

37] b)

Is $p(m) = \Omega(m)$? Yes

$p(m) = \Omega(q(m))$ if there exist positive constants b and m_1 such that $0 \leq b\,q(m) \leq p(m)$ $\forall\ m \geq m_1$

$q(m) = m$

The function $p(m)$ grows no slower than $b\,q(m)$. Yes, $p(m) = \Omega(m)$

$$0 \leq 1(m) \leq 10m - 50 \quad \rightarrow (2)$$

This holds true for $b = 1$

The smallest positive value of m_1 is given by,

$$m \leq 10m - 50$$

$$\Rightarrow 50 \leq 9m_1 \quad (\text{At } m_1)$$

$$\frac{50}{9} \leq m_1$$

$\Rightarrow m_1 \geq 5.5$ Since, m is an integer,

The smallest positive value of m_1 is 6

$b = 1$

$m_1 = 6$

3] c)

Is $p(m) = \Theta(m)$?

Yes

$p(m) = \Theta(q(m))$ if there exists positive constants a, b and m_2 such that

$$0 \leq b\,q(m) \leq p(m) \leq a\,q(m) \quad \forall\, m \geq m_2$$

The function $p(m)$ grows at exactly the same rate as m. Hence its growth rate is exactly equal to m.

Since, $p(m)$ is both $O(m)$ and $\Omega(m)$, it is also $\Theta(m)$. It's growth rate is bounded both above and below by m.

a] Yes

b] Yes

c] Yes

4. This problem also uses the notation of Lecture 5, and here also all m are positive integers.

(a) Suppose we have a function $p(m)$ that can be expressed as:

$$p(m) = p_1(m) + p_2(m) + p_3(m)$$

and we have:

$$p_k(m) = O(q_k(m)),\ k=1,2,3 \quad \longrightarrow \text{i]}$$

Prove that:

$$p(m) = O(q_1(m) + q_2(m) + q_3(m)) \quad \longrightarrow \text{ii]}$$

$p(m) = O(q(m))$, if there exist positive constants a and m_0 such that $0 \le p(m) \le a\,q(m)$ $\forall\ m \ge m_0$

$\Rightarrow$ Asymptotic upper Bound

From i]

$$p_k(m) = O(q_k(m)) \quad k = 1, 2, 3$$

$p_1(m) = O(q_1(m))$ such that

$$0 \le p_1(m) \le a_1 q_1(m) \quad \forall\ m \ge m_1 \longrightarrow (1)$$

$p_2(m) = O(q_2(m))$ such that,

$$0 \le p_2(m) \le a_2 q_2(m) \quad \forall\ m \ge m_2 \longrightarrow (2)$$

$p_3(m) = O(q_3(m))$ such that,

$0 \leq p_3(m) \leq a_3 q_3(m) \quad \forall \; m \geq m_3 \rightarrow (3)$

Adding (1), (2) and (3) we get

$$0 \leq p_1(m) + p_2(m) + p_3(m) \leq a_1 q_1(m) + a_2 q_2(m) + a_3 q_3(m)$$

$$\forall \quad m \geq \max(m_1, m_2, m_3) \downarrow (4)$$

$p(m) = p_1(m) + p_2(m) + p_3(m)$

Let's choose a value 'a' = $\max(a_1, a_2, a_3)$

Substituting in (4),

$$0 \leq p(m) \leq a(q_1(m) + q_2(m) + q_3(m))$$

$$\forall \; m \geq \max(m_1, m_2, m_3)$$

Hence,

$$p(m) = O(q_1(m) + q_2(m) + q_3(m))$$

Hence Proved!

(b) Is a similar statement to (a) true for Ω(..) ? (That is, if you replace each O(..) in part (a) with Ω(..), would the last equation be true?) Justify your answer.

From i],

$$p_k(n) = \Omega(q_k(n)) \quad k = 1, 2, 3$$

$p_1(n) = \Omega(q_1(n))$ such that,

$$0 \le b_1 q_1(n) \le p_1(n) \quad \forall \quad n \ge n_1 \rightarrow (3)$$

$p_2(n) = \Omega(q_2(n))$ such that,

$$0 \le b_2 q_2(n) \le p_2(n) \quad \forall \quad n \ge n_2 \rightarrow (4)$$

$p_3(n) = \Omega(q_3(n))$ such that,

$$0 \le b_3 q_3(n) \le p_3(n) \quad \forall \quad n \ge n_3 \rightarrow (5)$$

Adding (3), (4) and (5), we get

$$0 \le b_1 q_1(n) + b_2 q_2(n) + b_3 q_3(n) \le p_1(n) + p_2(n) + p_3(n)$$

Let,

$$p(m) = p_1(m) + p_2(m) + p_3(m)$$

Let $b = \min(b_1, b_2, b_3)$

$$\Rightarrow 0 \leq b\left(q_1(m) + q_2(m) + q_3(m)\right) \leq p(m)$$

$$\forall \; m \geq \max(m_1, m_2, m_3)$$

By definition of asymptotic lower bound,

$$p(m) = \Omega\left(q_1(m) + q_2(m) + q_3(m)\right)$$

Hence Proved !

Yes, it is True.

5]

a]

(a) Find the asymptotic upper bound for *p(m)*, in simplest form (no unnecessary constants) but no looser than necessary.

$$p(m) = m^2 \log_2 m + 10\left(\frac{2^m}{\log_2 m}\right) + 0.1(2^{(m-5)})$$

We can split $p(m)$ into 3 individual terms

$$p(m) = p_1(m) + p_2(m) + p_3(m)$$

The asymptotic upper bound of $p_1(m)$ is given by,

$$O(m^2 \log_2 m)$$

The asymptotic upper bound of $p_2(m)$ is given by,

$$O\left(\frac{2^m}{\log_2 m}\right)$$

The asymptotic upper bound of $p_3(m)$ is given by,

$$O(2^m)$$

By using the result from 4],

$$p(m) = O(q_1(m) + q_2(m) + q_3(m))$$

The asymptotic upper bound of $p(m)$ is given by,

$$p(m) = O\left(m^2 \log_2 m + \frac{2^m}{\log_2 m} + 2^m\right)$$

$$= O(2^m)$$

The asymptotic upper bound of $p(m)$ in simplest form is $O(2^m)$

(b) Find the asymptotic lower bound for p(m), in simplest form (no unnecessary constants) but no looser than necessary.

From 5 a],

$$p(m) = p_1(m) + p_2(m) + p_3(m)$$

The asymptotic lower bound of $p(m)$ is given by,

$$p(m) = \Omega\left(m^2 \log_2 m + \frac{2^m}{\log_2 m} + 2^m\right)$$

The tightest lower bound is given by, $\Omega(2^m)$