Module 4
Machine Learning-based Predictive Analytics

Dr. Natarajan Meghanathan
Professor of Computer Science
Jackson State University
Email: natarajan.meghanathan@jsums.edu

4.1 Naïve Bayes Classifier

Conditional Probability Fundamentals

- Probability of an event: is the likelihood of the event to happen (in a scale of 0 to 1: 0 – not at all; 1 – will definitely happen)
- P(rain today) = 0.40; in a scale of 0 to 100, there is a 40% (0.40 in a scale of 0 to 1) chance for a rain today.
- P(A | B) is the probability for an event A to happen given that B has happened
  - It is different from P(A) as well as P(B).

$$P(A \mid B) = \frac{P(B \mid A) * P(A)}{P(B)}$$

- If not directly given, P(B) can be computed as follows where ~A represents the case of A not happening.

$$P(B) = P(B \mid A) * P(A) + P(B \mid \sim A) * P(\sim A)$$

Conditional Probability: Example

- A lab has been testing patients for Covid with one of the two possible results: positive or negative.
- The lab guarantees that their results are 99% accurate :
  - i.e., if you have the disease, the result will be positive in 99 of 100 tests done;
  - likewise, if you do not have the disease, the result will be negative in 99 of the 100 tests done).
- Let 3% of the population actually have Covid.
- If the test taken for a person gives a positive result, what is the probability that the person has Covid?

- Given:
- P(Covid among the people) = P(a person has Covid) = 0.03
- P(a person does not have Covid) = 1-0.03 =0.97
- P(test result is positive | the person has Covid) = 0.99
- P(test result is negative | the person does not have Covid) = 0.99
- P(test result is positive | the person does not have Covid) = 0.01
- To find: P(the person has Covid | test result is positive).

Conditional Probability: Example (continued)

- P(the person has Covid | test result is +ve)

  = P(test result is +ve | the person has Covid) * P(the person has Covid)

  P(test result is +ve)

- P(test result is +ve)

  = P(test result is +ve | the person has Covid) * P(the person has Covid)

  + P(test result is +ve | the person does not have Covid) * P(the person does not have Covid)

  = 0.99 * 0.03 + 0.01 * 0.97 = 0.0394

P(the person has Covid | test result is +ve) = 0.99 *0.03 / 0.0394

P(the person has Covid | test result is +ve) = 0.7538 (~ 75%)

Classification

- Supervised machine learning
- The "training" dataset comprises of data (a collection of features) and its classification to a particular class (binary: 0 or 1, sometimes multi-class: 0, 1, 2, 3, …)
- The goal of a classification algorithm is to build a model based on the training dataset and use it to predict the class for test data.

outlook	temp	humidity	windy	play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

Sepal Length	Sepal Breadth	Petal Length	Petal Breadth	Species
6.8	3.2	5.9	2.3	Iris-virginica
6.9	3.1	5.1	2.3	Iris-virginica
4.9	3	1.4	0.2	Iris-setosa
5.6	3	4.5	1.5	Iris-versicolor
4.8	3.1	1.6	0.2	Iris-setosa
5.8	2.8	5.1	2.4	Iris-virginica
7.2	3.6	6.1	2.5	Iris-virginica
5.1	3.5	1.4	0.3	Iris-setosa
4.7	3.2	1.6	0.2	Iris-setosa
6.6	3	4.4	1.4	Iris-versicolor

Naïve Bayes Classifier

- Suitable for both binary and multi-class classification.
- Scalable and faster
- Can be easily trained on small dataset

- Assumes the features are independent
- More suitable if the features take categorical values rather than numerical values

Notations and Terminologies

- Let there be a dataset X with ‘m’ features (a.k.a. dimensions) {denoted as $X_1, X_2, \ldots, X_m$} and ‘n’ records (indicated as $X^{(1)}, X^{(2)}, \ldots, X^{(n)}$) each belonging to a class in the set {0, 1} denoted by Y.

For the set Y:

let the **positive class** (the class which we represent the not-normal class or behavior: say, having covid in the covid dataset) be denoted by ‘1’
the **negative class** (the class that represents normality or a normal behavio say, not having covid in the covid dataset) be denoted by ‘0’

	X_1	X_2	X_3	X_4	Y
	Sore throat and Cough	Runny nose	Fever	Difficulty in Breathing	COVID-19?
$X^{(1)}$	Yes	No	Yes	Mild	Yes
$X^{(2)}$	Yes	Yes	No	No	No
$X^{(3)}$	Yes	No	Yes	Strong	Yes
	No	Yes	Yes	Mild	Yes
:	No	No	No	No	No
:	No	Yes	Yes	Strong	Yes
$X^{(7)}$	No	Yes	No	No	No
$X^{(8)}$	Yes	Yes	Yes	Strong	Yes

Naïve Bayes Classifier

$$P(Y = k \mid X^{(t)} = [X^{(t)}_1, X^{(t)}_2, ..., X^{(t)}_m])$$

$$= \frac{P(X^{(t)} = [X^{(t)}_1, X^{(t)}_2, ..., X^{(t)}_m] \mid Y = k) * P(Y = k)}{P(X^{(t)} = [X^{(t)}_1, X^{(t)}_2, ..., X^{(t)}_m])}$$

$$P(Y = k \mid X^{(t)} = [X^{(t)}_1, X^{(t)}_2, ..., X^{(t)}_m])$$

$$= \frac{P(X_1 = X^{(t)}_1 \mid Y = k) * P(X_2 = X^{(t)}_2 \mid Y = k) * ..... * P(X_m = X^{(t)}_m \mid Y = k) * P(Y = k)}{P(X_1 = X^{(t)}_1) * P(X_2 = X^{(t)}_2) * ..... * P(X_m = X^{(t)}_m)}$$

Naïve Bayes Classifier (Example 1)

X1	X2	X3	X4	
Sore throat and Cough	Runny nose	Fever	Difficulty in Breathing	COVID-19?
Yes	No	Yes	Mild	Yes
Yes	Yes	No	No	No
Yes	No	Yes	Strong	Yes
No	Yes	Yes	Mild	Yes
No	No	No	No	No
No	Yes	Yes	Strong	Yes
No	Yes	No	No	No
Yes	Yes	Yes	Strong	Yes

Let the test case be: X1 = No; X2 = No; X3 = Yes; X4 = Strong

P(Covid-19 = Yes | X1 = No, X2 = No, X3 = Yes, X4 = Strong)

$$= \frac{P(X1 = No, X2 = No, X3 = Yes, X4 = Strong \mid Yes) * P(Yes)}{P(X1 = No, X2 = No, X3 = Yes, X4 = Strong)}$$

Naïve Bayes Classifier (Example 1)

X1	X2	X3	X4	
Sore throat and Cough	Runny nose	Fever	Difficulty in Breathing	COVID-19?
Yes	No	Yes	Mild	Yes
Yes	Yes	No	No	No
Yes	No	Yes	Strong	Yes
No	Yes	Yes	Mild	Yes
No	No	No	No	No
No	Yes	Yes	Strong	Yes
No	Yes	No	No	No
Yes	Yes	Yes	Strong	Yes

Let the test case be: X1 = No; X2 = No; X3 = Yes; X4 = Strong

P(Covid-19 = Yes | X1 = No, X2 = No, X3 = Yes, X4 = Strong)

= P(X1 = No | Yes) * P(X2 = No | Yes) * P(X3 = Yes | Yes) * P(X4 = Strong | Yes) * P(Yes)

P(X1 = No) * P(X2 = No) * P(X3 = Yes) * P(X4 = Strong)

Naïve Bayes Classifier (Example 1)

X1	X2	X3	X4	
Sore throat and Cough	Runny nose	Fever	Difficulty in Breathing	COVID-19?
Yes	No	Yes	Mild	Yes
Yes	Yes	No	No	No
Yes	No	Yes	Strong	Yes
No	Yes	Yes	Mild	Yes
No	No	No	No	No
No	Yes	Yes	Strong	Yes
No	Yes	No	No	No
Yes	Yes	Yes	Strong	Yes

P(Covid-19 = Yes | X1 = No, X2 = No, X3 = Yes, X4 = Strong)

$$= \frac{2/5 * 2/5 * 5/5 * 3/5 * 5/8}{4/8 * 3/8 * 5/8 * 3/8} = 1.3653$$

Naïve Bayes Classifier (Example 1)

X1	X2	X3	X4	
Sore throat and Cough	Runny nose	Fever	Difficulty in Breathing	COVID-19?
Yes	No	Yes	Mild	Yes
Yes	Yes	No	No	No
Yes	No	Yes	Strong	Yes
No	Yes	Yes	Mild	Yes
No	No	No	No	No
No	Yes	Yes	Strong	Yes
No	Yes	No	No	No
Yes	Yes	Yes	Strong	Yes

P(Covid-19 = No | X1 = No, X2 = No, X3 = Yes, X4 = Strong)

$$= \frac{P(X1 = No \mid No) * P(X2 = No \mid No) * P(X3 = Yes \mid No) * P(X4 = Strong \mid No) * P(No)}{P(X1 = No) * P(X2 = No) * P(X3 = Yes) * P(X4 = Strong)}$$

Naïve Bayes Classifier (Example 1)

X1 Sore throat and Cough	X2 Runny nose	X3 Fever	X4 Difficulty in Breathing	COVID-19?
Yes	No	Yes	Mild	Yes
Yes	Yes	No	No	No
Yes	No	Yes	Strong	Yes
No	Yes	Yes	Mild	Yes
No	No	No	No	No
No	Yes	Yes	Strong	Yes
No	Yes	No	No	No
Yes	Yes	Yes	Strong	Yes

P(Covid-19 = No | X1 = No, X2 = No, X3 = Yes, X4 = Strong)

$$= \frac{2/3 * 1/3 * 0/3 * 0/3 * 3/8}{4/8 * 3/8 * 5/8 * 3/8} = 0$$

Naïve Bayes Example 1: Conclusions

P(Covid-19 = Yes | X1 = No, X2 = No, X3 = Yes, X4 = Strong) = 1.3653

P(Covid-19 = No | X1 = No, X2 = No, X3 = Yes, X4 = Strong) = 0

Hence, the person with features X1 = No, X2 = No, X3 = Yes, X4 = Strong is predicted to have Covid

Naïve Bayes Classifier (Example 2)

X1	X2	X3	
headache	runny nose	fever	flu
N	Y	Y	N
Y	N	N	N
N	N	N	N
Y	Y	Y	Y
Y	Y	N	Y
N	N	Y	Y

X1	X2	X3	
headache	runny nose	fever	flu
Y	N	Y	?

P(Flu = Y | X1 = Y, X2 = N, X3 = Y)

$$= \frac{[P(X1 = Y \mid Flu = Y) * P(X2 = N \mid Flu = Y) * P(X3 = Y \mid Flu = Y)] * P(Flu = Y)}{P(X1 = Y) * P(X2 = N) * P(X3 = Y)}$$

P(Flu = Y | X1 = Y, X2 = N, X3 = Y)

$$= \frac{[2/3 * 1/3 * 2/3] * 3/6}{3/6 * 3/6 * 3/6} = 0.5926$$

Naïve Bayes Classifier (Example 2)

X1	X2	X3	
headache	runny nose	fever	flu
N	Y	Y	N
Y	N	N	N
N	N	N	N
Y	Y	Y	Y
Y	Y	N	Y
N	N	Y	Y

X1	X2	X3	
headache	runny nose	fever	flu
Y	N	Y	?

$$P(Flu = N \mid X1 = Y, X2 = N, X3 = Y) = \frac{[P(X1 = Y \mid Flu = N) * P(X2 = N \mid Flu = N) * P(X3 = Y \mid Flu = N)] * P(Flu = N)}{P(X1 = Y) * P(X2 = N) * P(X3 = Y)}$$

$$P(Flu = N \mid X1 = Y, X2 = N, X3 = Y) = \frac{[1/3 * 2/3 * 1/3] * 3/6}{3/6 * 3/6 * 3/6} = 0.2963 < 0.5926: P(Flu = Y)$$

Hence, given X1 = Y, X2 = N, X3 = Y, the person is predicted to have a Flu

Naïve Bayes Classifier (Example 3)

X1	X2	X3	
headache	runny nose	fever	flu
N	Y	Y	N
Y	N	N	N
N	N	N	N
Y	Y	Y	Y
Y	Y	N	Y
N	N	Y	Y

X1	X2	X3	
headache	runny nose	fever	flu
N	Y	N	?

$$P(Flu = Y \mid X1 = N, X2 = Y, X3 = N) = \frac{[P(X1 = N \mid Flu = Y) * P(X2 = Y \mid Flu = Y) * P(X3 = N \mid Flu = Y)] * P(Flu = Y)}{P(X1 = N) * P(X2 = Y) * P(X3 = N)}$$

$$P(Flu = Y \mid X1 = N, X2 = Y, X3 = N) = \frac{[1/3 * 2/3 * 1/3] * 3/6}{3/6 * 3/6 * 3/6} = 0.2963$$

Naïve Bayes Classifier (Example 3)

X1	X2	X3	
headache	runny nose	fever	flu
N	Y	Y	N
Y	N	N	N
N	N	N	N
Y	Y	Y	Y
Y	Y	N	Y
N	N	Y	Y

X1	X2	X3	
headache	runny nose	fever	flu
N	Y	N	?

$$P(Flu = N \mid X1 = N, X2 = Y, X3 = N) = \frac{[P(X1 = N \mid Flu = N) * P(X2 = Y \mid Flu = N) * P(X3 = N \mid Flu = N)] * P(Flu = N)}{P(X1 = N) * P(X2 = Y) * P(X3 = N)}$$

$$P(Flu = N \mid X1 = N, X2 = Y, X3 = N) = \frac{[2/3 * 1/3 * 2/3] * 3/6}{3/6 * 3/6 * 3/6} = 0.5926 > P(Flu = Y \mid X1 = N, X2 = Y, X3 = N)$$

Hence, given X1 = N, X2 = Y, X3 = N, the person is predicted to NOT have a Flu

Naïve Bayes Classifier: Example 4

- We have data on 1000 patients. They were diagnosed to be Flu, Allergy or other Disease using the three symptoms as shown below. We will use this to predict the type of any new patient we encounter.

Type	Fever		Sneezing		Runny Nose		Total
	Yes	No	Yes	No	Yes	No	
Flu	400	100	350	150	450	50	500
Allergy	0	300	150	150	300	0	300
Other	100	100	150	50	50	150	200
Total	500	500	650	350	800	200	1000

Test Case: Fever – Yes, Sneezing - No, and Runny Nose - Yes

P(Flu) = 500/1000 = 0.5 P(Allergy) = 300/1000 = 0.3 P(Other) = 200/1000 = 0.2

P(Fever: Yes) = 500/1000 = 0.5 P(Fever: No) = 500/1000 = 0.5
P(Sneezing: Yes) = 650/1000 = 0.65 P(Sneezing: No) = 350/1000 = 0.35
P(Runny Nose: Yes) = 800/1000 = 0.8 P(Runny nose: No) = 200/1000 = 0.2

Naïve Bayes Classifier: Example 4

Type	Fever		Sneezing		Runny Nose		Total
	Yes	No	Yes	No	Yes	No	
Flu	400	100	350	150	450	50	500
Allergy	0	300	150	150	300	0	300
Other	100	100	150	50	50	150	200
Total	500	500	650	350	800	200	1000

Test Case: Fever – Yes, Sneezing - No, and Runny Nose - Yes

P(Flu | Fever – Yes, Sneezing – No and Runny Nose – Yes)
= P(Fever – Yes| Flu) * P(Sneezing – No | Fu) * P(Runny Nose – Yes|Flu)* P(Flu)

P(Fever – Yes) * P(Sneezing – No) * P(Runny Nose - Yes)

P(Flu | Fever – Yes, Sneezing – No and Runny Nose – Yes)
= (400/500) * (150/500) * (450/500) * (500/1000)
---------- = 0.7714 (Flu)
(500/1000) * (350/1000) * (800/1000)

Naïve Bayes Classifier: Example 4

Type	Fever		Sneezing		Runny Nose		Total
	Yes	No	Yes	No	Yes	No	
Flu	400	100	350	150	450	50	500
Allergy	0	300	150	150	300	0	300
Other	100	100	150	50	50	150	200
Total	500	500	650	350	800	200	1000

Test Case: Fever – Yes, Sneezing - No, and Runny Nose - Yes

P(Allergy | Fever – Yes, Sneezing – No and Runny Nose – Yes)
= P(Fever – Yes| Allergy) * P(Sneezing – No | Allergy) * P(Runny Nose – Yes| Allergy)* P(Allergy)

P(Fever – Yes) * P(Sneezing – No) * P(Runny Nose - Yes)

P(Allergy | Fever – Yes, Sneezing – No and Runny Nose – Yes)
= (0/300) * (150/300) * (300/300) * (300/1000)
---------- = 0.0 (Allergy)
(500/1000) * (350/1000) * (800/1000)

Naïve Bayes Classifier: Example 4

Type	Fever		Sneezing		Runny Nose		Total
	Yes	No	Yes	No	Yes	No	
Flu	400	100	350	150	450	50	500
Allergy	0	300	150	150	300	0	300
Other	100	100	150	50	50	150	200
Total	500	500	650	350	800	200	1000

Test Case: Fever – Yes, Sneezing - No, and Runny Nose - Yes

P(Other | Fever – Yes, Sneezing – No and Runny Nose – Yes)
= P(Fever – Yes| Other) * P(Sneezing – No | Other) * P(Runny Nose – Yes| Other)* P(Other)

P(Fever – Yes) * P(Sneezing – No) * P(Runny Nose - Yes)

P(Other | Fever – Yes, Sneezing – No and Runny Nose – Yes)
= (100/200) * (50/200) * (50/200) * (200/1000)
---------- = 0.0446 (Other)
(500/1000) * (350/1000) * (800/1000)

Naïve Bayes Classifier: Example 4 Conclusions

Type	Fever		Sneezing		Runny Nose		Total
	Yes	No	Yes	No	Yes	No	
Flu	400	100	350	150	450	50	500
Allergy	0	300	150	150	300	0	300
Other	100	100	150	50	50	150	200
Total	500	500	650	350	800	200	1000

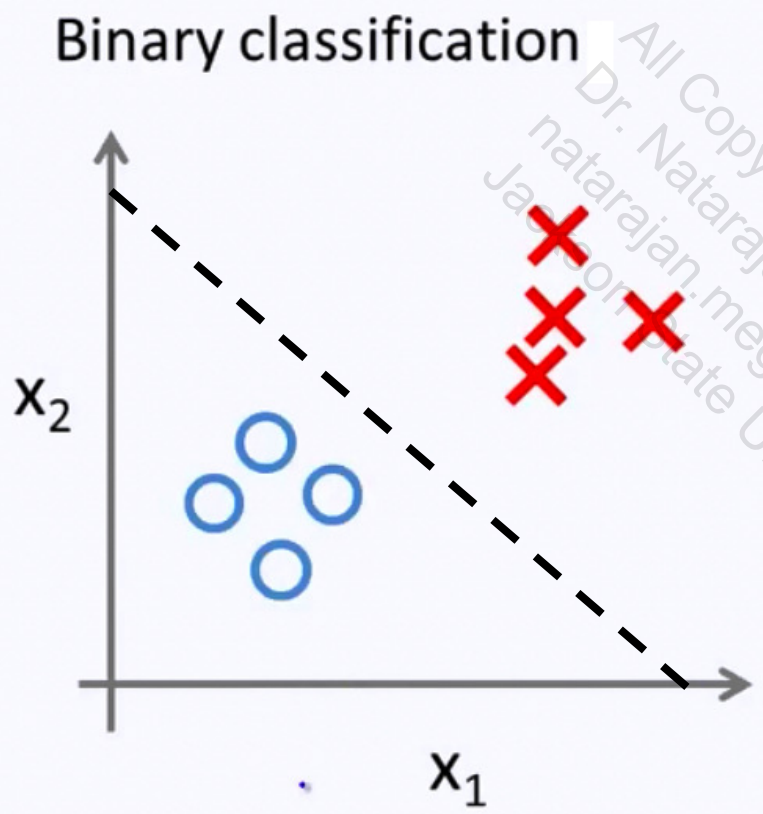
Test Case: Fever – Yes, Sneezing - No, and Runny Nose - Yes

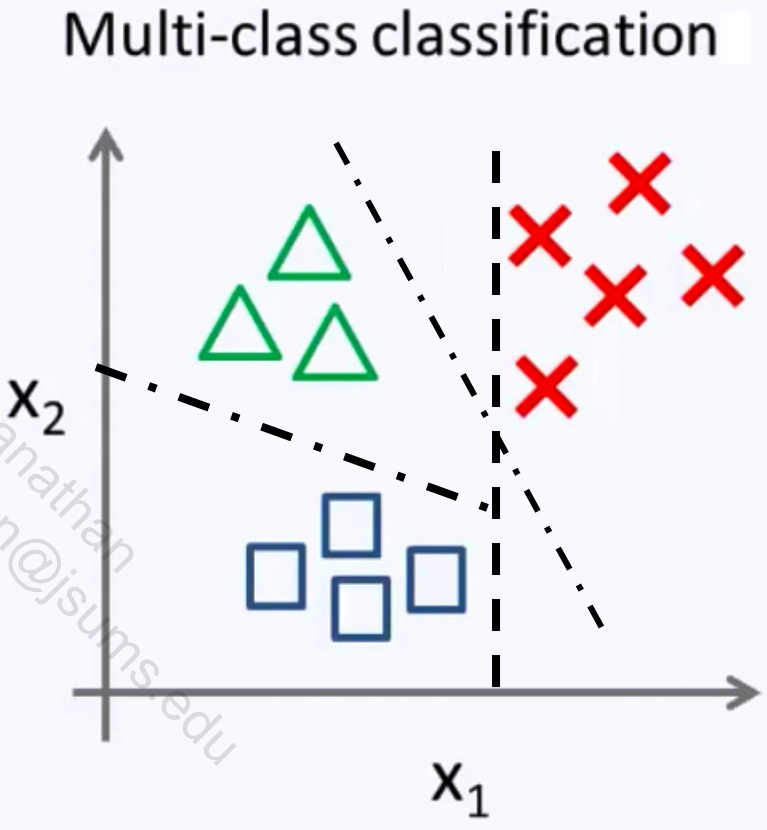
P(Flu | Fever – Yes, Sneezing - No, and Runny Nose - Yes) = 0.7714
P(Allergy | Fever – Yes, Sneezing - No, and Runny Nose - Yes) = 0.0
P(Other | Fever – Yes, Sneezing - No, and Runny Nose - Yes) = 0.0446

Hence, the person is predicted to have a Flu

4.2 Logistic Regression

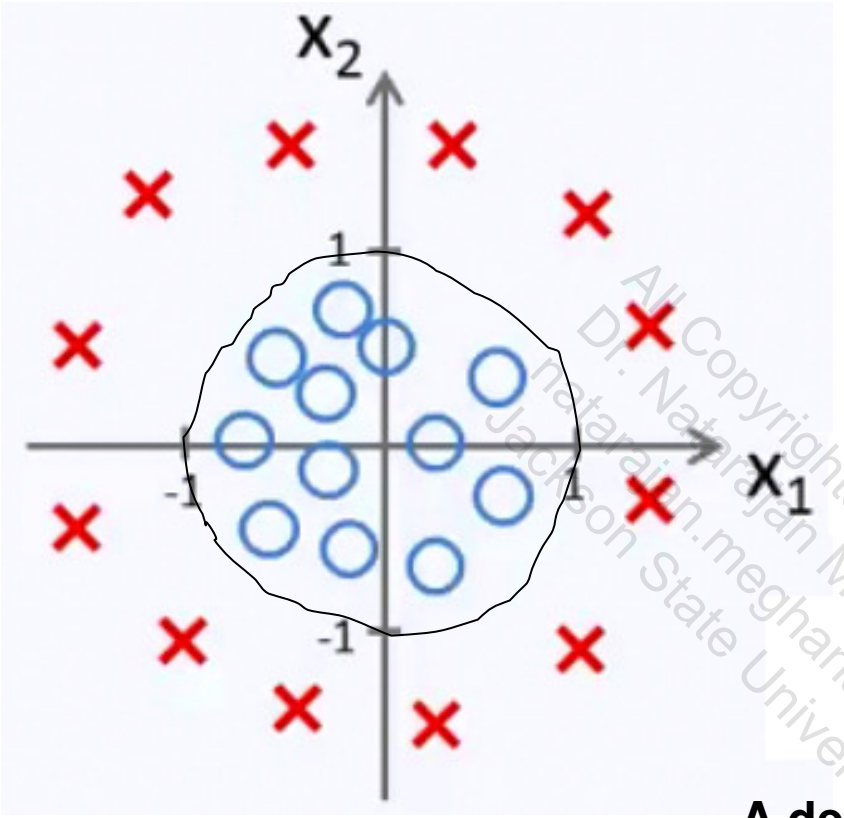
Decision Boundary (Linear)





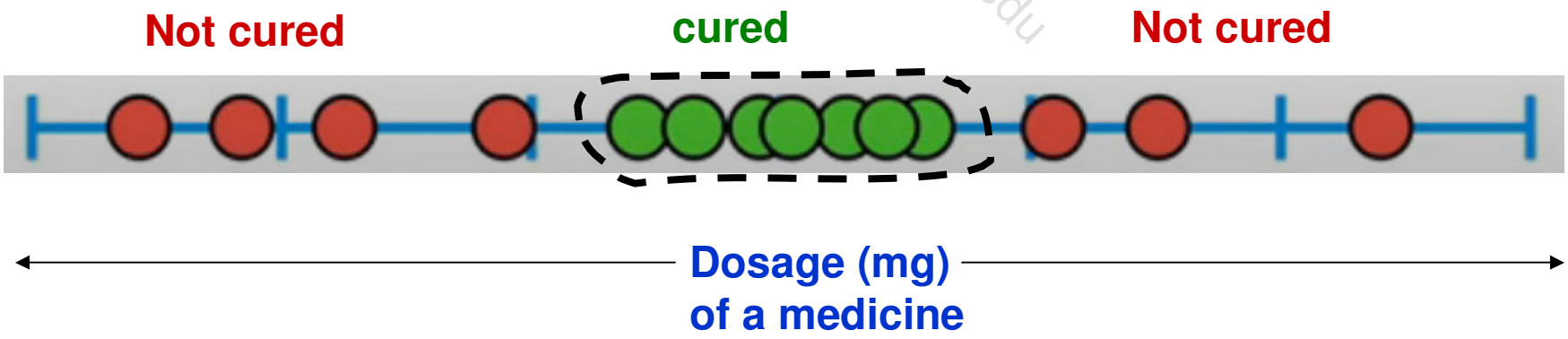
The classifiers vary on the basis of how the decision boundary is identified/ constructed based on the training dataset and how it is used to classify test data

Other Decision Boundaries



Non-linear decision boundary

A decision boundary that can be modeled only if we were to scale-up the dimensions



Online Calculator for Logistic Regression

https://stats.blue/Stats_Suite/logistic_regression_calculator.html

C-Reactive Protein (CRP) Dataset

	X_1	X_2	Y
	CRP	Temp, C	Class
$x^{(1)}$	40	36	0
$x^{(2)}$	11.1	37.2	0
$x^{(3)}$	30	36.5	0
:	21.4	39.4	0
:	10.7	39.6	0
:	3.4	40.7	0
:	42	37.6	1
:	31.1	42.2	1
	50	38.5	1
	60.4	39.4	1
$x^{(11)}$	45.7	38.6	1
$x^{(12)}$	17.3	42.7	1

C-Reactive Protein (CRP) is a protein whose concentration in the blood increases when there is inflammation in the body in response to infections due to bacteria, virus, etc.
The body temperature is also high in case of infections.

The CRP levels and raise in body temperature are more in case of bacterial infections (class 1) compared to virus infections (class 0)

Hypothesis Representation

- Given a dataset X(n, m) with ‘n’ records and ‘m’ features, we will train the dataset to fit a model

  $z = \Theta_0 + \Theta_1 X_1 + \Theta_2 X_2 + \ldots + \Theta_m X_m$

  where $\Theta = [\Theta_0, \Theta_1, \Theta_2, \ldots, \Theta_m]$ is the parameter vector.
- We will then map z to a scale of 0…1 using the following sigmoid function that represents the hypothesis $h_\Theta(X)$

$$h_\Theta(X) = g(z) = \frac{1}{1 + e^{-z}}$$

- For mathematical simplicity and ease of representation, we will represent the feature vector as $X = [X_0 = 1, X_1, X_2, \ldots, X_m]$ and represent the computation of z as the dot product of the vectors Θ and X.
- i.e., $z = \Theta X = [\Theta_0, \Theta_1, \Theta_2, \ldots, \Theta_m] \bullet [1, X_1, X_2, \ldots, X_m]$
- i.e., $z = \Theta_0 + \Theta_1 X_1 + \Theta_2 X_2 + \ldots + \Theta_m X_m$

Hypothesis and Example

- We can thus compute the hypothesis as:

$$h_\Theta(X) = g(\Theta * X) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_m X_m)}}$$

- For the ith record $X^{(i)}$, we will then compute

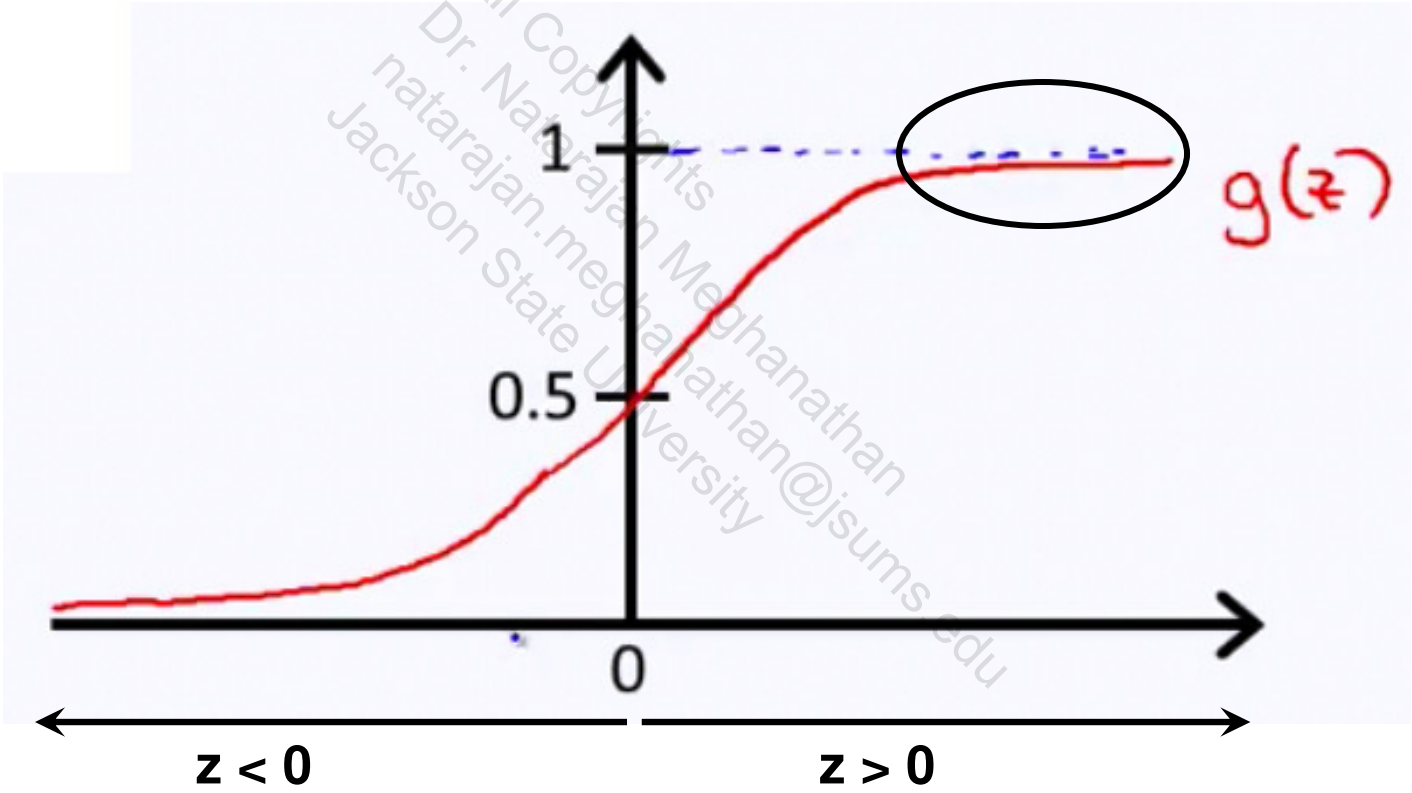
$$h_\Theta(X^{(i)}) = g(\Theta * X^{(i)}) = \frac{1}{1 + e^{-\left(\theta_0 + \theta_1 X_1^{(i)} + \theta_2 X_2^{(i)} + \ldots + \theta_m X_m^{(i)}\right)}}$$

For example, if Θ_0 = -867.5885, Θ_1 = 3.6152 and Θ_2 = 19.5431 for the CRP dataset (to classify bacteria vs. virus infection), for the 10th record with feature vector $X^{(10)} = [X_0^{(10)} = 1, X_1^{(10)} = 60.4, X_2^{(10)} = 39.4]$ and class $Y^{(10)} = 1$,

$$h_\Theta\left(X^{(10)}\right) = \frac{1}{1 + e^{-(-867.5885 + 3.6152 * 60.4 + 19.5431 * 39.4)}} = \frac{1}{1 + e^{-120.7677}} = 1$$

Sigmoid Function

$$g(z) = \frac{1}{1 + e^{-z}}$$



We can thus interpret the nature of the sigmoid function as more the value of z > 0, the more closer is the sigmoid function value to 1 and likewise lower the value of z < 0, the more closer is the value of the function to 0.

Interpretation of the Hypothesis

- The value of the hypothesis function $h_\Theta(X)$ is the estimated probability that Y = 1 for the input X.
- In other words, for a record $X^{(i)}$, $h_\Theta(X^{(i)})$ is the estimated probability that $Y^{(i)} = 1$ (irrespective of the actual value of $Y^{(i)}$) for the input $X^{(i)}$.
- In our previous example, for $X^{(10)}$ in the CRP dataset, $h_\Theta(X^{(10)}) = 1$ is the estimated probability that $Y^{(10)} = 1$ and it aligns well with the actual class $Y^{(10)} = 1$.
- Consider the fourth record $X^{(4)} = [1, 21.4, 39.4]$ in the CRP dataset.

X_0	X_1	X_2	Y	ΘX	h(ΘX)
1	40	36	0	-19.43	3.6E-09
1	11.1	37.2	0	-100.5	2.4E-44
1	30	36.5	0	-45.81	1.3E-20
1	21.4	39.4	0	-20.23	1.6E-09
1	10.7	39.6	0	-55	1.3E-24
1	3.4	40.7	0	-59.89	9.7E-27
1	42	37.6	1	19.07	1
1	31.1	42.2	1	69.563	1
1	50	38.5	1	65.581	1
1	60.4	39.4	1	120.77	1
1	45.7	38.6	1	51.99	1
1	17.3	42.7	1	29.445	1

$$h_\Theta\left(X^{(4)}\right) = \frac{1}{1+e^{-(-867.5885+3.6152*21.4+19.5431*39.4)}} = \frac{1}{1+e^{-(-20.2333)}} = 1.6357*10^{-9}$$

$1.6357*10^{-9}$ is the estimated probability that $Y^{(4)} = 1$
i.e., $1 - 1.6357*10^{-9} \sim 1$ is the estimated probability that $Y^{(4)} = 0$ and it aligns well with the actual class $Y^{(4)} = 0$

Example run on the CRP dataset

- $\Theta_0 = -867.5885$, $\Theta_1 = 3.6152$ and $\Theta_2 = 19.5431$

	CRP (X1)	Temp (X2)	Actual Class	z = ΘX	g(z) = hΘ(X)	Pred. class
1	40	36	0	-19.4289	3.64868E-09	0
2	11.1	37.2	0	-100.456	2.35675E-44	0
3	30	36.5	0	-45.8094	1.27424E-20	0
4	21.4	39.4	0	-20.2251	1.64573E-09	0
5	10.7	39.6	0	-54.9991	1.30075E-24	0
6	3.4	40.7	0	-59.8926	9.74883E-27	0
7	42	37.6	1	19.07046	0.999999995	1
8	31.1	42.2	1	69.56304	1	1
9	50	38.5	1	65.58085	1	1
10	60.4	39.4	1	120.7677	1	1
11	45.7	38.6	1	51.9898	1	1
12	17.3	42.7	1	29.44483	1	1

If g(z) > 0.5, the predicted class is 1
If g(z) < 0.5, the predicted class is 0

Interpreting the Decision Boundary
(for the CRP dataset)

- Through Gradient Descent, we found:
  ΘX = -867.5885 + 3.6152*X1 + 19.5431*X2
- If ΘX = 0, we will have: $h_\Theta(X) = \frac{1}{1+e^{-0}} = \frac{1}{1+1} = 0.5$

ΘX = -867.5885 + 3.6152*X1 + 19.5431*X2 = 0

3.6152*X1 + 19.5431*X2 = 867.5885

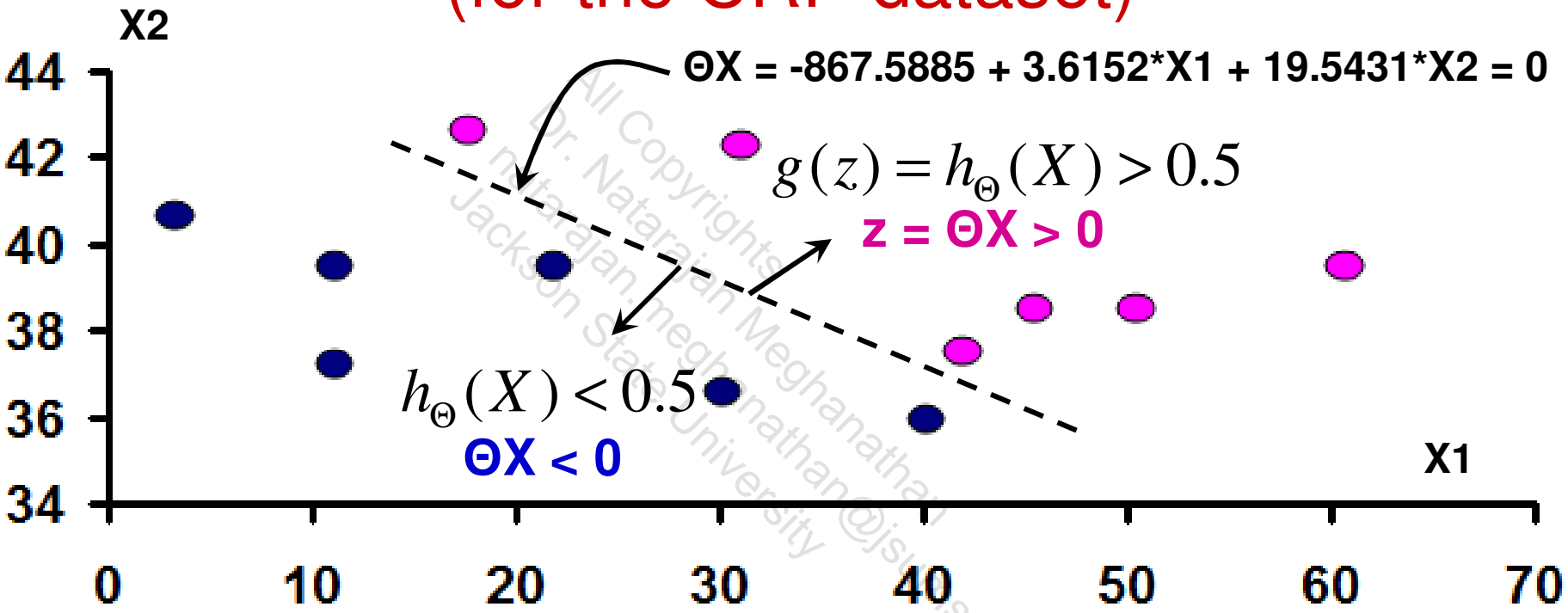
Let X1 = 0 ➔ 19.5431*X2 = 867.5885 ➔ X2 = 44.39

Let X2 = 0 ➔ 3.6152*X1 = 867.5885 ➔ X1 = 239.98

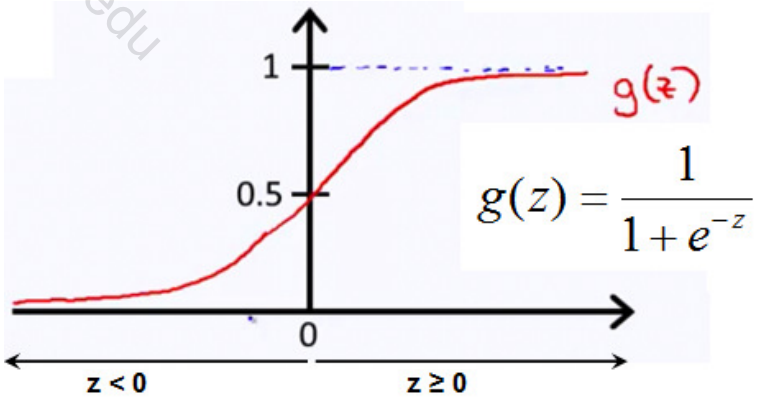
In the (X1, X2) coordinate system, the decision boundary -
-867.5885 + 3.6152*X1 + 19.5431*X2 = 0

intersects the X1 axis at (239.98, 0) & the X2 axis at (0, 44.39)

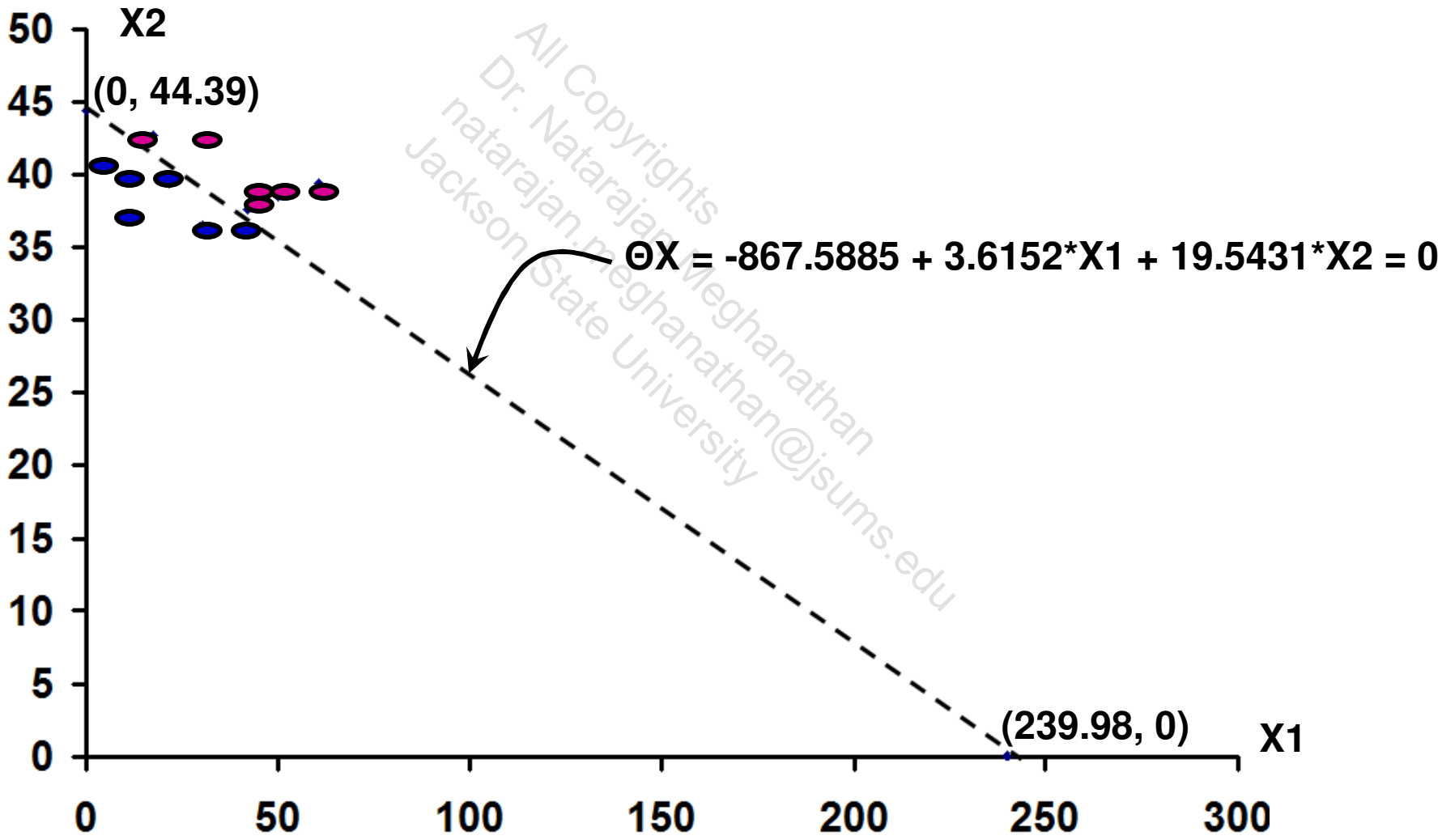
Interpreting the Decision Boundary
(for the CRP dataset)



The further a data point (X1, X2) from the decision boundary, the more likely that it is going to be correctly classified.



Interpreting the Decision Boundary
(for the CRP dataset)



Ex-2: Prostate Cancer Antigen (PSA) Dataset

PSA Level	Prostate Cancer (1-Yes; 0 – No) Actual	Pred
3.8	1	1
3.4	1	1
2.9	1	1
2.8	1	1
2.7	1	1
2.1	1	1
1.6	1	0
2.5	0	1
2	0	0
1.7	0	0
1.4	0	0
1.2	0	0
0.9	0	0
0.8	0	0

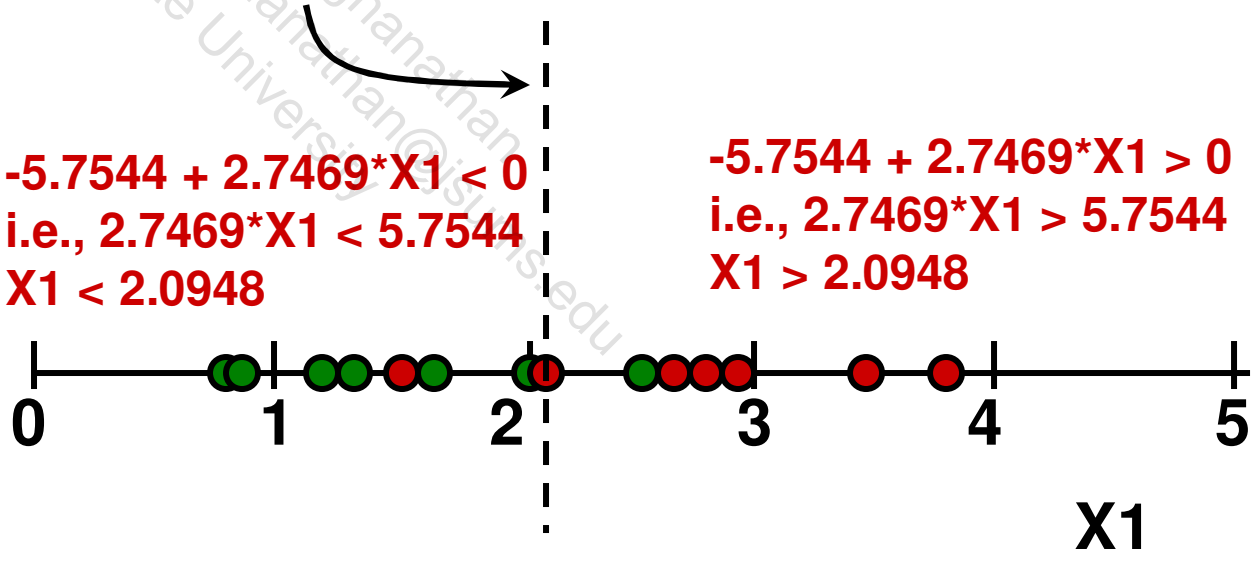
From the online calculator,

$\Theta X = -5.7544 + 2.7469 * X1$

To find the decision boundary, set $\Theta X = 0$

$-5.7544 + 2.7469 * X1 = 0$

$X1 = 5.7544/2.7469 = 2.0948$

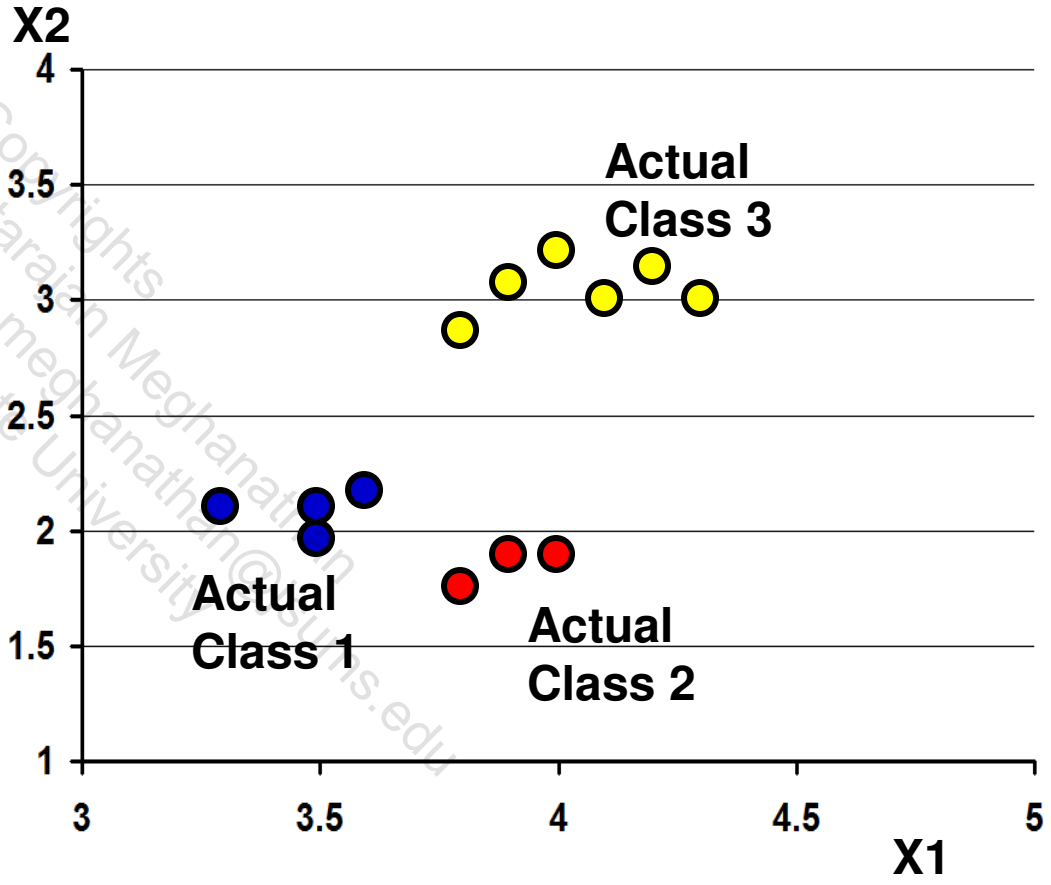


Logistic Regression for Multi-class Classification

We find the parameters for dataset versions featuring a particular class vs. rest

Ex-3: Multi-class Dataset

	X1	X2	Actual Class
1	3.9	1.9	2
2	3.3	2.1	1
3	3.5	2	1
4	4	1.9	2
5	3.8	1.8	2
6	4.2	3.1	3
7	3.8	2.9	3
8	4.1	3	3
9	4.3	3	3
10	3.5	2.1	1
11	3.6	2.2	1
12	3.9	3.1	3
13	4	3.2	3



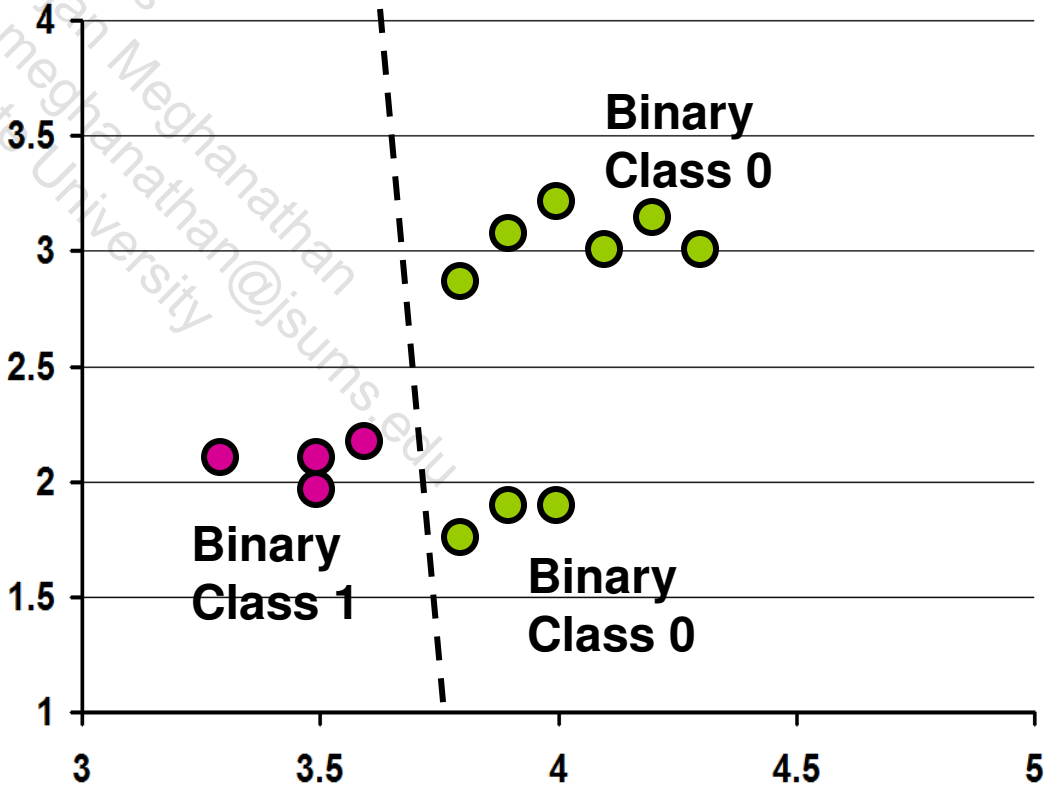
Ex-3 (1): Binary Classification

Actual Class 1 vs. rest: Replace
Actual Class 1 with Binary Class 1
and the other classes with Binary Class 0

Parameter Values for Decision Boundary

Θ0 = 724.9637
Θ1 = -196.163
Θ2 = 0.2306

	X1	X2	Binary Class
1	3.9	1.9	0
2	3.3	2.1	1
3	3.5	2	1
4	4	1.9	0
5	3.8	1.8	0
6	4.2	3.1	0
7	3.8	2.9	0
8	4.1	3	0
9	4.3	3	0
10	3.5	2.1	1
11	3.6	2.2	1
12	3.9	3.1	0
13	4	3.2	0



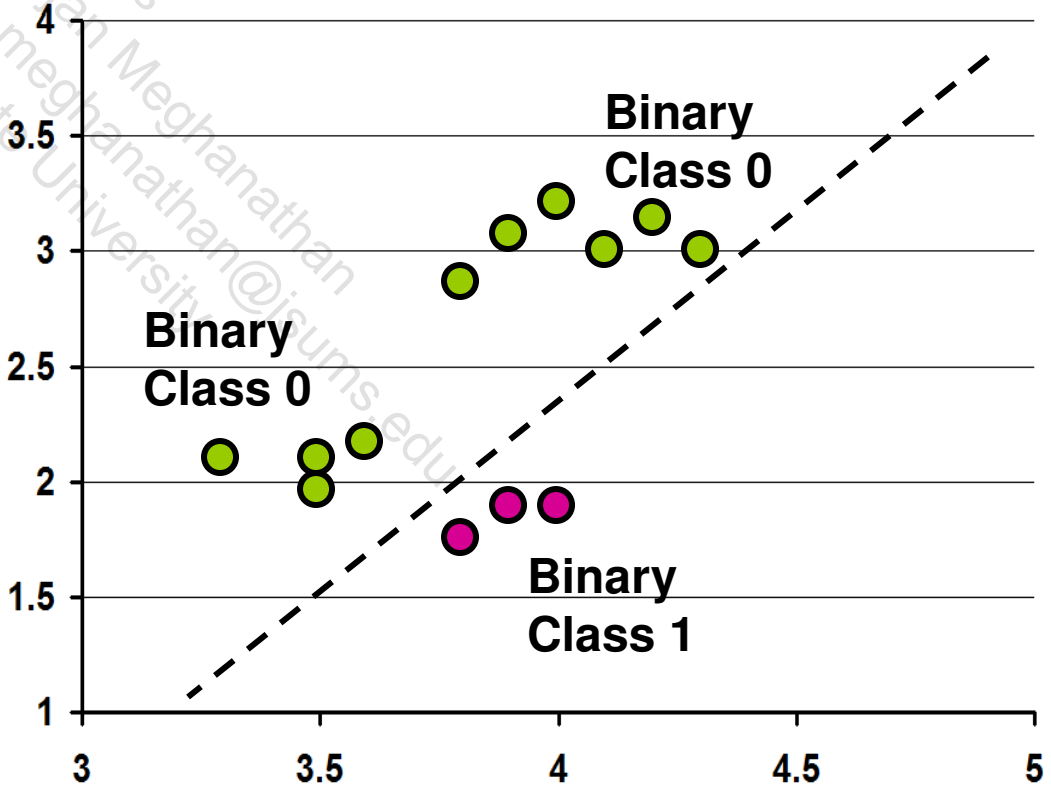
Ex-3 (2): Binary Classification

Actual Class 2 vs. rest: Replace
Actual Class 2 with Binary Class 1
and the other classes with Binary Class 0

	X1	X2	Binary Class
1	3.9	1.9	1
2	3.3	2.1	0
3	3.5	2	0
4	4	1.9	1
5	3.8	1.8	1
6	4.2	3.1	0
7	3.8	2.9	0
8	4.1	3	0
9	4.3	3	0
10	3.5	2.1	0
11	3.6	2.2	0
12	3.9	3.1	0
13	4	3.2	0

Parameter Values for Decision Boundary

Θ0 = -178.489
Θ1 = 84.3611
Θ2 = -68.1207



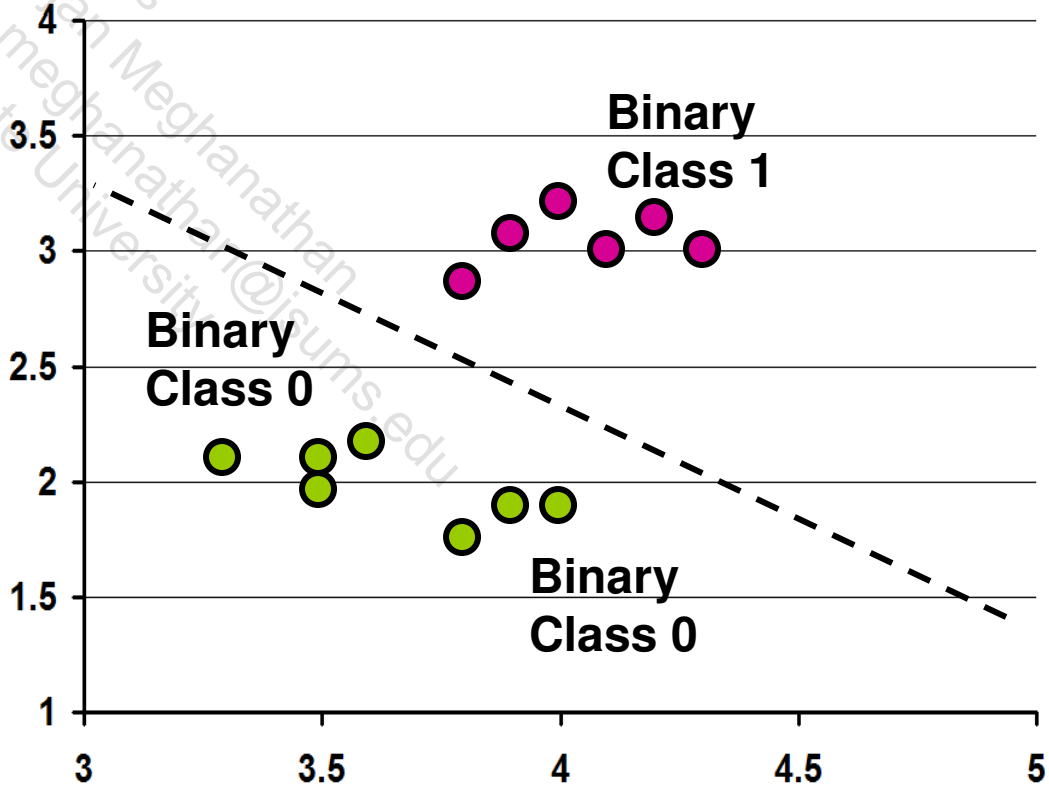
Ex-3 (3): Binary Classification

Actual Class 3 vs. rest: Replace Actual Class 3 with Binary Class 1 and the other classes with Binary Class 0

	X1	X2	Binary Class
1	3.9	1.9	0
2	3.3	2.1	0
3	3.5	2	0
4	4	1.9	0
5	3.8	1.8	0
6	4.2	3.1	1
7	3.8	2.9	1
8	4.1	3	1
9	4.3	3	1
10	3.5	2.1	0
11	3.6	2.2	0
12	3.9	3.1	1
13	4	3.2	1

Parameter Values for Decision Boundary

Θ0 = -125.585
Θ1 = -6.6405
Θ2 = 58.9069

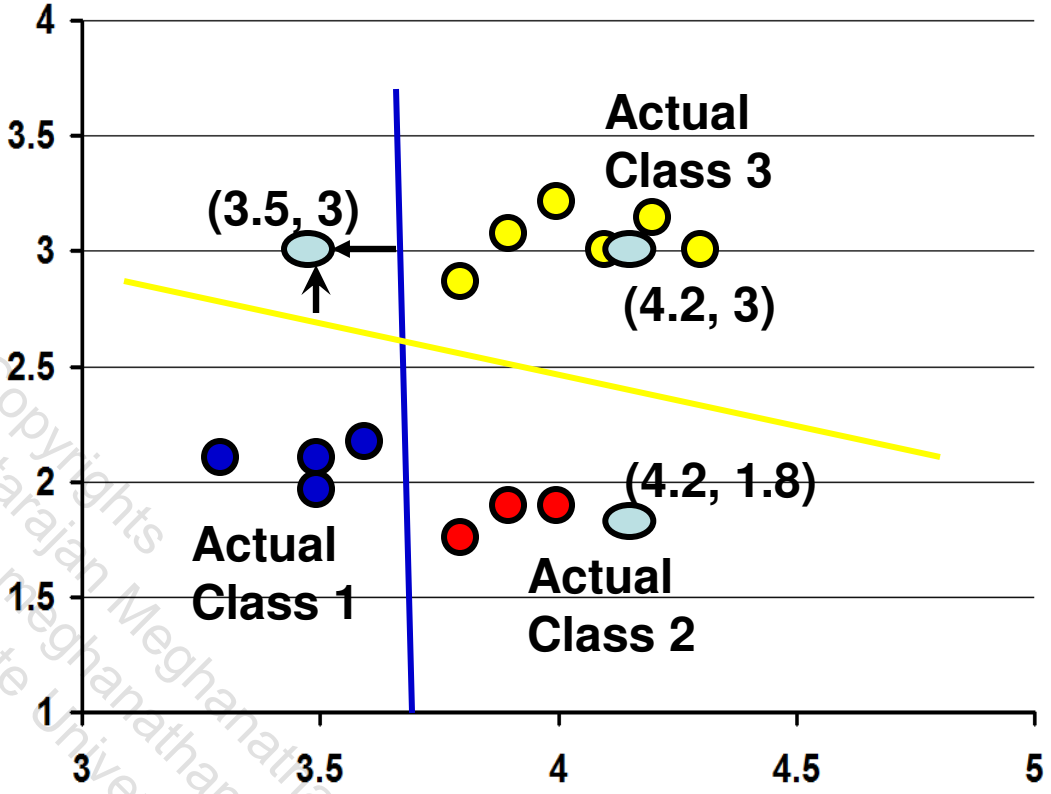


Ex-3 (4): Testing

$z = \Theta X = \Theta 0 + \Theta 1 * X1 + \Theta 2 * X2$

$$g(z) = \frac{1}{1+e^{-z}}$$

$$h(\Theta X) = \frac{1}{1+e^{-(\Theta X)}}$$



Parameter Decision Boundaries

Class 1	Class 2	Class 3
Θ0 = 724.9637	Θ0 = -178.489	Θ0 = -125.585
Θ1 = -196.163	Θ1 = 84.3611	Θ1 = -6.6405
Θ2 = 0.2306	Θ2 = -68.1207	Θ2 = 58.9069

testing		ΘX for each class			h(Θ(X)) for each class			
X1	X2	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3	Final Predicted Class
3.5	3	39.085	-87.5873	27.89395	1	9.15E-39	1	Class 1
4.2	3	-98.2291	-28.5345	23.2456	2.19E-43	4.05E-13	1	Class 3
4.2	1.8	-98.50582	53.21036	-47.4427	1.66E-43	1	2.49E-21	Class 2

4.3 DBSCAN Clustering

Clustering

- Unsupervised machine learning

- Given a set of data points, we want to identify clusters (of at least certain minimum number of data points in each cluster) such that a data point is more closer to other data points within its home cluster compared to data points in other clusters.

Density-based Spatial Clustering of Applications with Noise (DBSCAN)

- DBSCAN detects clusters based on the density of the data points.
  - Given a set of points in some space, it groups together points that are closely packed.
  - Detects outliers that are farther away from the rest of the data points.
- The algorithm operates based on two parameters: eps (ε) and minPts (k).
  - minPts is the minimum number of data points within a cluster
  - eps (ε) is the radius of the neighborhood used for cluster expansion
- The value of minPts (k) is typically twice the number of dimensions in the dataset.
- Given a value for minPts (k), the value for eps (ε) is determined using the k-distance graph (explained in the next few slides)

DBSCAN: Preliminary Steps

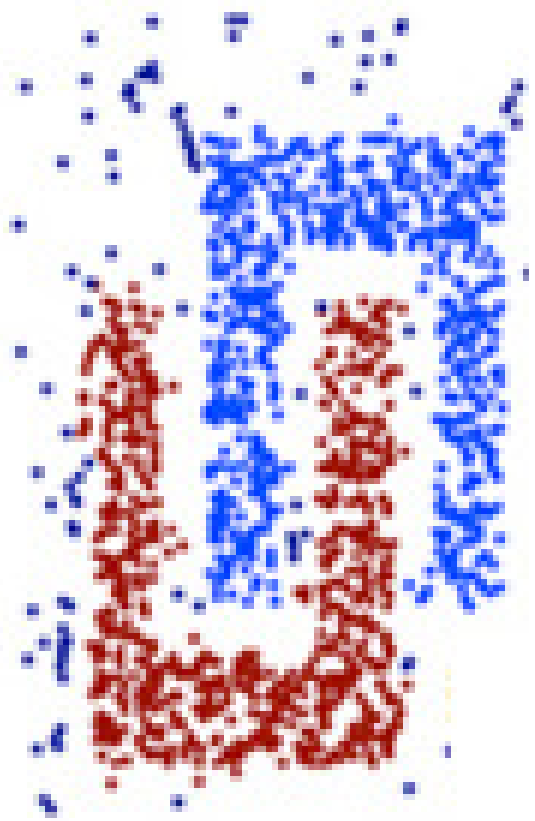
- Normalize the feature values of the dataset, each in a scale of 0 to 1,
  - Preferable to use the square root of the sum of the squares approach for normalization.
- Determine the pair wise distance between the data points in the normalized space.
  - Use the Euclidean distance measure.
- For a given value of minPts (k), determine for each data point (say, i): the average distance (kavg-dist(i)) of the first k-nearest neighbors of i.
- Sort the kavg-dist values of the data points and plot the sorted values (to obtain the k-distance graph).
- The eps (ε) value to be used for clustering is the kavg-dist value at which there is a steep increase in the slope of the k-distance graph.
- For each data point, find the neighbor data points that are at a distance less than or equal to eps (ε), referred to as ε-neighborhood.
- A data point is a *core* (c) point if there are at least minPts neighbors within the eps (ε) distance, including itself.
- A data point is a *non-core* (nc) point if there are less than minPts neighbors within the eps (ε) distance, including itself.

DBSCAN: Clustering Steps

- Start from a randomly chosen core point (that has been not yet chosen for clustering) to be part of a new cluster and include its ε-neighborhood in the cluster.
  - Call the ε-neighborhood data points added to the cluster as the auxiliary points.
  - Go through the list of auxiliary data points, one at a time.
    - If an auxiliary data point is a core point whose ε-neighborhood is not yet included in any cluster, its ε-neighborhood is added to the growing cluster.
- Continue the above steps as long as there is at least one core point that has been not yet chosen for clustering.
- The algorithm stops when all the core points have been explored.
- The non-core data points that are not part of any cluster at the time of the termination of the algorithm are referred to as outliers.

Advantages of DBSCAN

- Can detect clusters of arbitrary shapes, including non-convex clusters
  - A cluster is a non-convex cluster if the line connecting two data points within the cluster goes through some other cluster.
- Can be used for outlier detection
- Can be used for contact tracing
- Can automatically detect the appropriate number of clusters in the dataset
  - Unlike, K-means clustering for which the number of clusters (K) needs to be specified a priori.

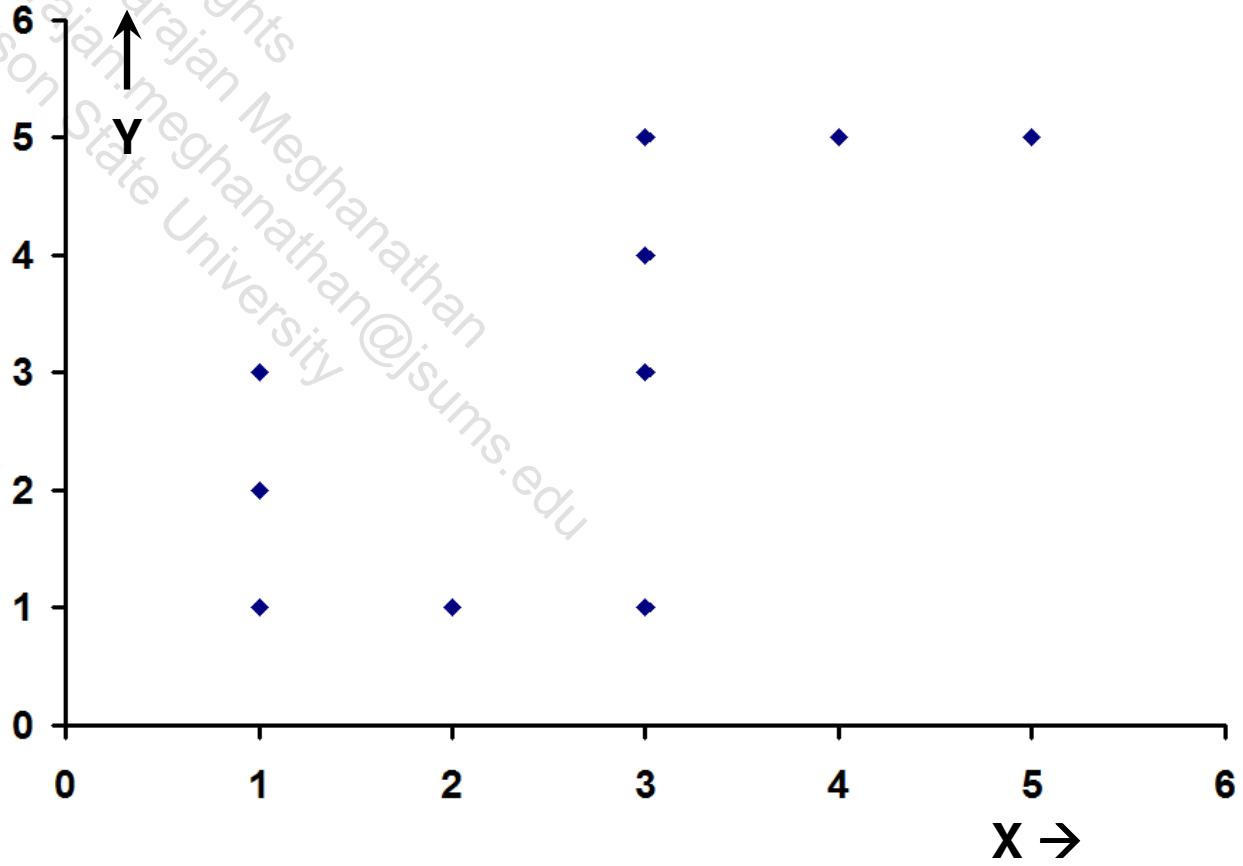


DBSCAN: Example 1

Consider the following dataset as the (X, Y) coordinates of the rooms occupied by flu-infected students in a dormitory. Determine the clusters among the co-ordinates.

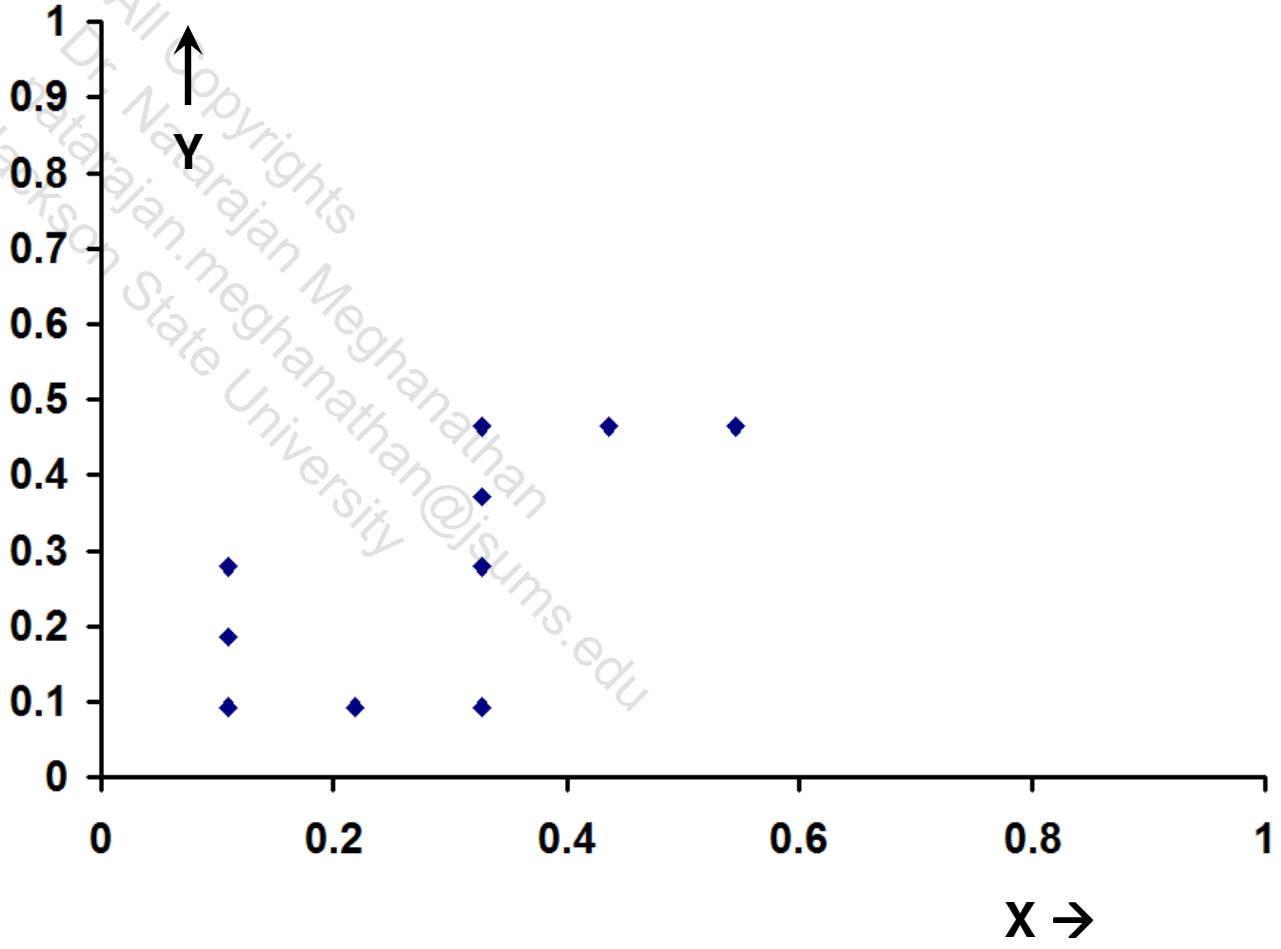
	X	Y
1	1	3
2	1	2
3	1	1
4	2	1
5	3	1
6	3	3
7	3	4
8	3	5
9	4	5
10	5	5

Visualization of the raw dataset



Ex-1 (1): Normalization Step

	X1	X2
1	0.109109	0.278543
2	0.109109	0.185695
3	0.109109	0.092848
4	0.218218	0.092848
5	0.327327	0.092848
6	0.327327	0.278543
7	0.327327	0.371391
8	0.327327	0.464238
9	0.436436	0.464238
10	0.545545	0.464238



Ex-1 (2): Pair wise Distances

	1	2	3	4	5	6	7	8	9	10	Kavg distances
1	0	0.09	0.19	0.22	0.29	0.22	0.24	0.29	0.38	0.47	0.18
2	0.09	0	0.09	0.14	0.24	0.24	0.29	0.35	0.43	0.52	0.14
3	0.19	0.09	0	0.11	0.22	0.29	0.35	0.43	0.5	0.57	0.1525
4	0.22	0.14	0.11	0	0.11	0.22	0.3	0.39	0.43	0.5	0.145
5	0.29	0.24	0.22	0.11	0	0.19	0.28	0.37	0.39	0.43	0.19
6	0.22	0.24	0.29	0.22	0.19	0	0.09	0.19	0.22	0.29	0.1725
7	0.24	0.29	0.35	0.3	0.28	0.09	0	0.09	0.14	0.24	0.14
8	0.29	0.35	0.43	0.39	0.37	0.19	0.09	0	0.11	0.22	0.1525
9	0.38	0.43	0.5	0.43	0.39	0.22	0.14	0.11	0	0.11	0.145
10	0.47	0.52	0.57	0.5	0.43	0.29	0.24	0.22	0.11	0	0.215

 # dimensions = 2
minPts = 2*# dimensions = 4
K = minPts = 4 (for determining the Kavg distances)

The closest K = 4 pair wise distances for each data point are highlighted and their average is determined

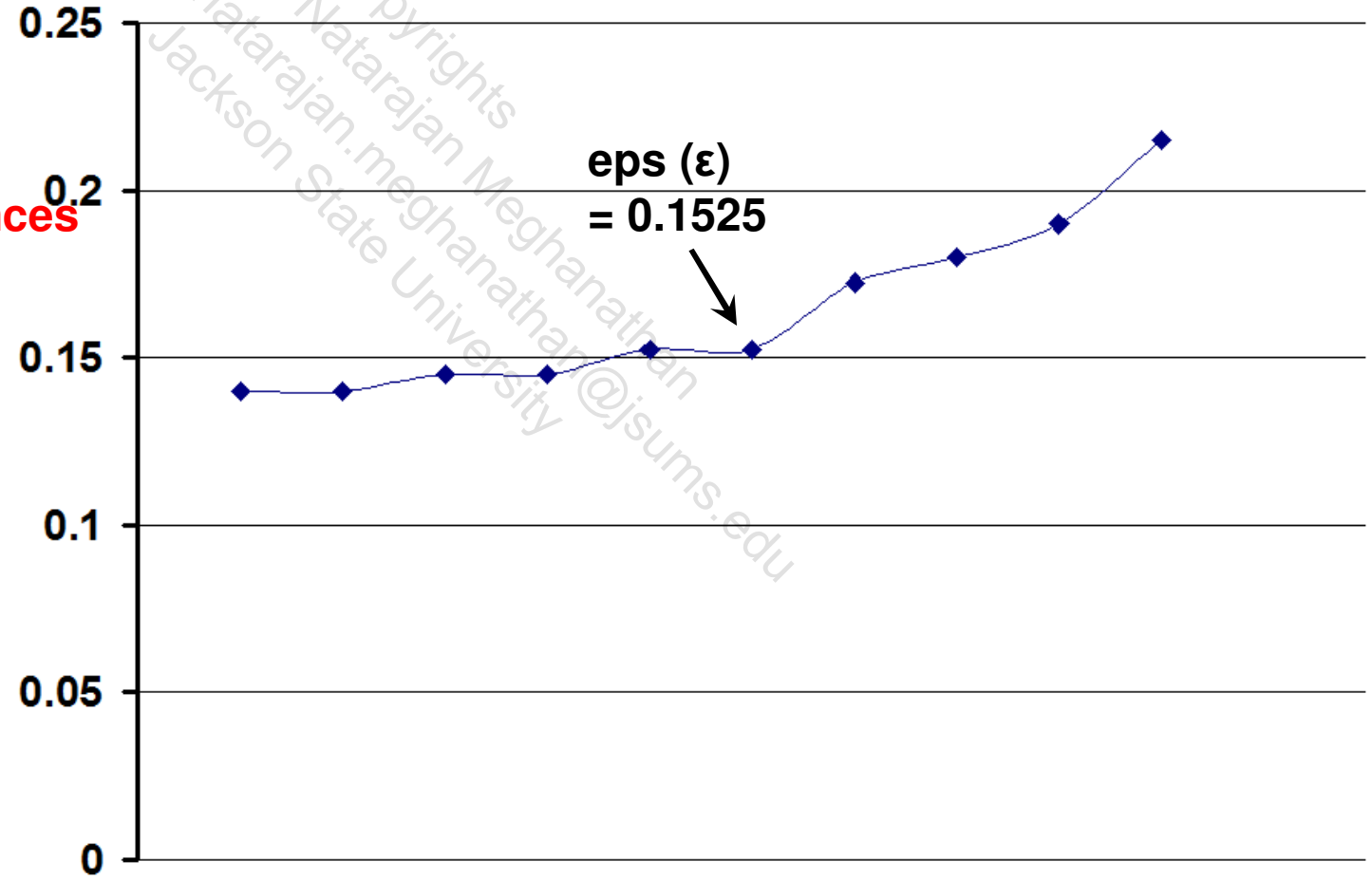
Note: You can use the PairwiseDistance.jar file (passed with the dataset) to get the pairwise distances.

Ex-1 (3): Elbow Method to Determine eps (ε)

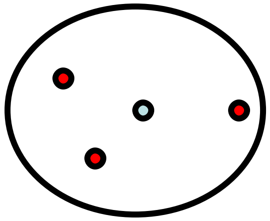
- Plot the sorted order of the kavg distances of the data points and identify the kavg value beyond which there is a steep increase in the slope.

Sorted order of the Kavg Pair wise distances

0.14
0.14
0.145
0.145
0.1525
0.1525
0.1725
0.18
0.19
0.215



Ex-1 (4): ε-Neighborhood and Core/Non-core Data Points



Pair wise Distances <= ε Value of 0.1525 are highlighted

	1	2	3	4	5	6	7	8	9	10
1	0	0.09	0.19	0.22	0.29	0.22	0.24	0.29	0.38	0.47
2	0.09	0	0.09	0.14	0.24	0.24	0.29	0.35	0.43	0.52
3	0.19	0.09	0	0.11	0.22	0.29	0.35	0.43	0.5	0.57
4	0.22	0.14	0.11	0	0.11	0.22	0.3	0.39	0.43	0.5
5	0.29	0.24	0.22	0.11	0	0.19	0.28	0.37	0.39	0.43
6	0.22	0.24	0.29	0.22	0.19	0	0.09	0.19	0.22	0.29
7	0.24	0.29	0.35	0.3	0.28	0.09	0	0.09	0.14	0.24
8	0.29	0.35	0.43	0.39	0.37	0.19	0.09	0	0.11	0.22
9	0.38	0.43	0.5	0.43	0.39	0.22	0.14	0.11	0	0.11
10	0.47	0.52	0.57	0.5	0.43	0.29	0.24	0.22	0.11	0

A data point is a Core/C point if 1+|ε-Neighborhood| is >= minPts (4)

Otherwise, it is a Non-core/nc point

Data Point	ε-Neighborhood	1 + |ε-Neighborhood|	Core/C or Non-Core/nc
1	2	2	nc
2	1, 3, 4	4	c
3	2, 4	3	nc
4	2, 3, 5	4	c
5	4	2	nc
6	7	2	nc
7	6, 8, 9	4	c
8	7, 9	3	nc
9	7, 8, 10	3	c
10	9	2	nc

Ex-1 (5): Cluster 1

Data Point	ε-Neighborhood	1 + |ε-Neighborhood|	Core/C or Non-Core/nc
1	2	2	nc
2	1, 3, 4	4	c
3	2, 4	3	nc
4	2, 3, 5	4	c
5	4	2	nc
6	7	2	nc
7	6, 8, 9	4	c
8	7, 9	3	nc
9	7, 8, 10	3	c
10	9	2	nc

Start with a randomly chosen core point (say, data point # 4) and grow the cluster based on ε-Neighborhood by exploring the core points included

C1 = {4}
C1 = {4, 2, 3, 5}
C1 = {4, 2, 3, 5, 1}
The growth of cluster C1 stops here as 3, 5 and 1 are all non-core/nc points.

Ex-1 (5): Cluster 2

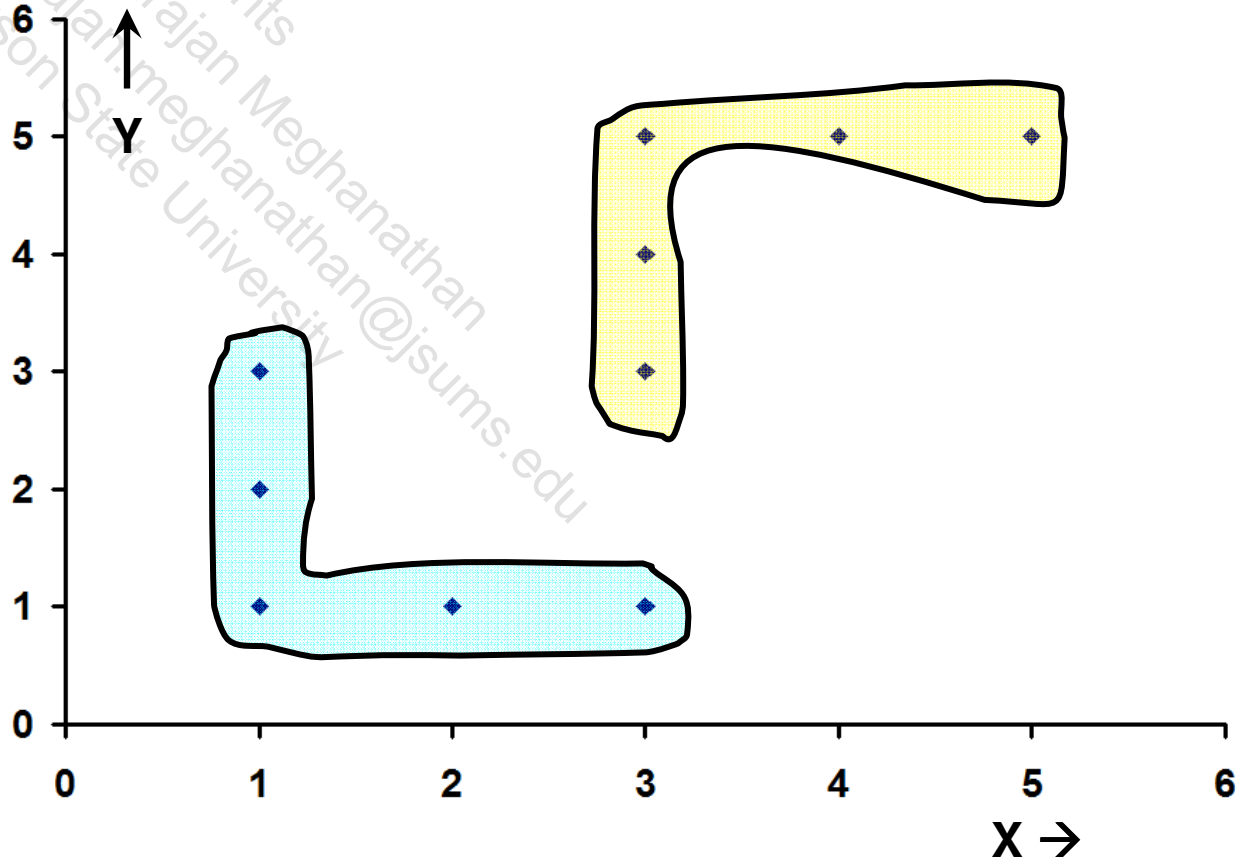
Data Point	ε-Neighborhood	1 + |ε-Neighborhood|	Core/C or Non-Core/nc
1	2	2	nc
2	1, 3, 4	4	c
3	2, 4	3	nc
4	2, 3, 5	4	c
5	4	2	nc
6	7	2	nc
7	6, 8, 9	4	c
8	7, 9	3	nc
9	7, 8, 10	3	c
10	9	2	nc

Start with a randomly chosen core point among the remaining core points.
In this case, there is only one core point (# 7) left. Pick 7 for inclusion in Cluster 2 and grow the cluster based on ε-Neighborhood by exploring the core points included

C2 = {7}
C2 = {7, 6, 8, 9}
C2 = {7, 6, 8, 9, 10}
The growth of cluster C2 stops here.

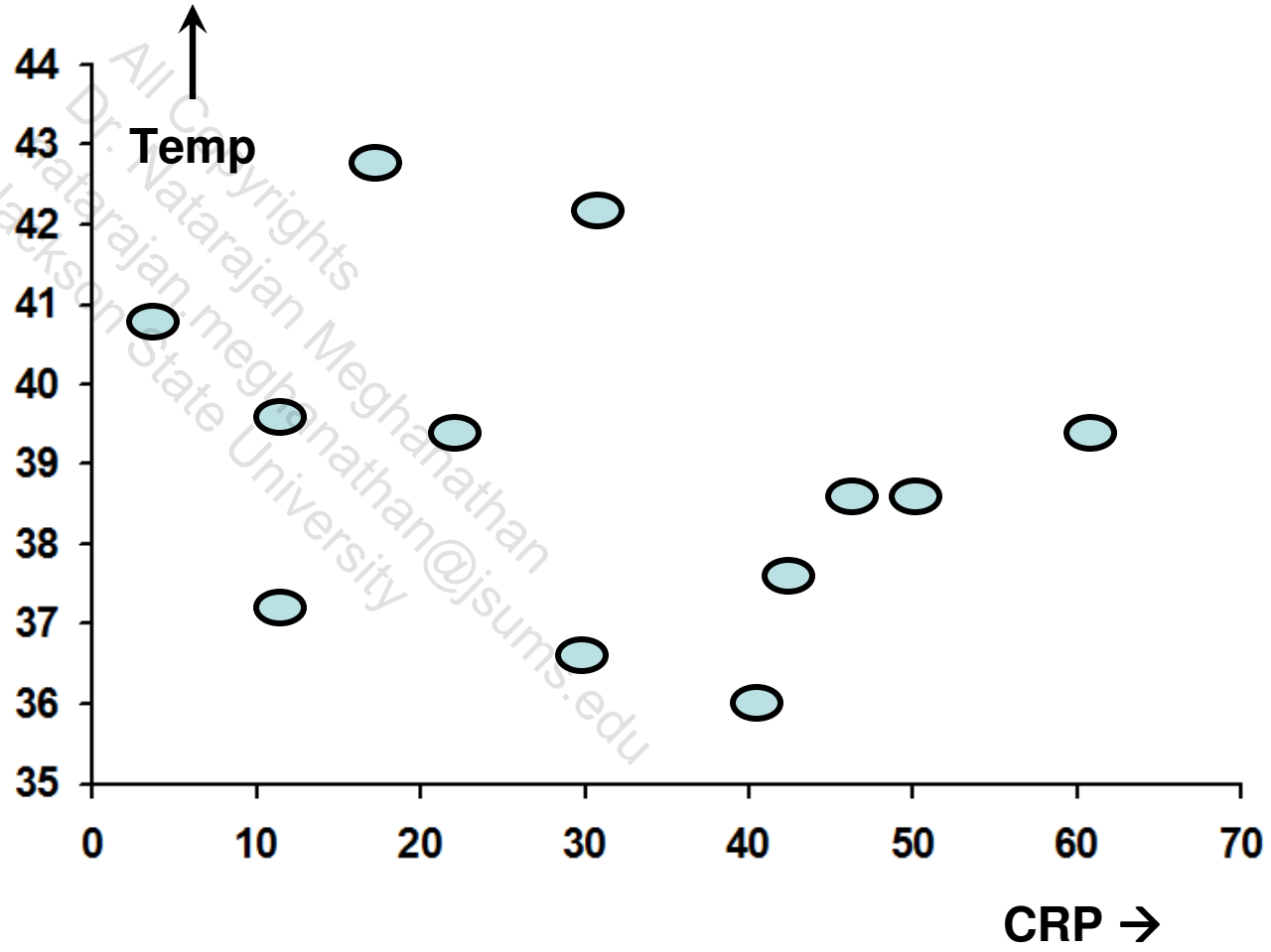
Ex-1: DBSCAN Clusters

Data Point	ε-Neighborhood	1 + |ε-Neighborhood|	Core/C or Non-Core/nc
1	2	2	nc
2	1, 3, 4	4	c
3	2, 4	3	nc
4	2, 3, 5	4	c
5	4	2	nc
6	7	2	nc
7	6, 8, 9	4	c
8	7, 9	3	nc
9	7, 8, 10	3	c
10	9	2	nc



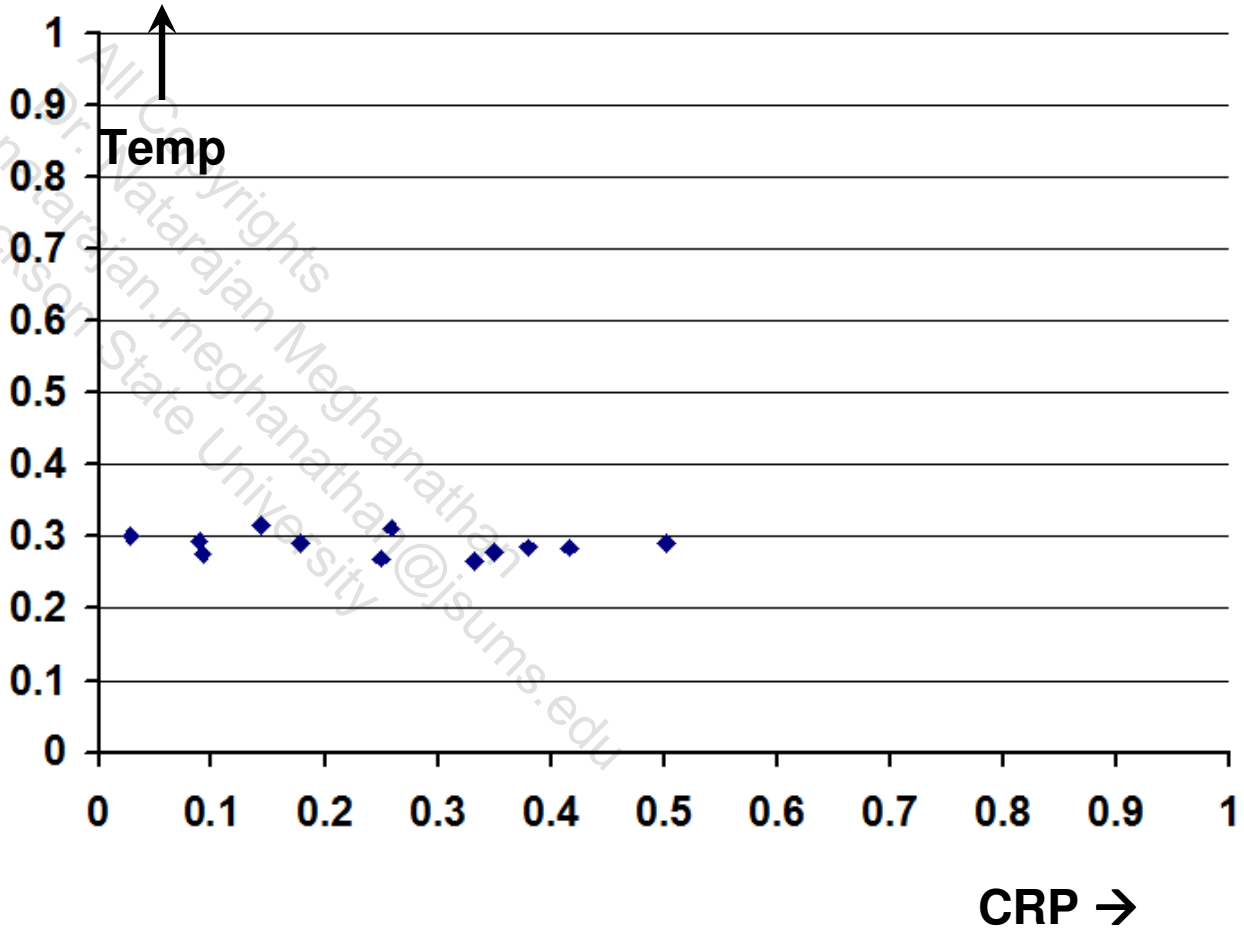
Ex-2: CRP Data Clustering

CRP	Temp, C
40	36
11.1	37.2
30	36.5
21.4	39.4
10.7	39.6
3.4	40.7
42	37.6
31.1	42.2
50	38.5
60.4	39.4
45.7	38.6
17.3	42.7



Ex-2 (1): Normalization Step

	CRP	Temp
1	0.332476	0.26589
2	0.092262	0.274753
3	0.249357	0.269583
4	0.177875	0.291002
5	0.088937	0.29248
6	0.02826	0.300604
7	0.3491	0.277708
8	0.2585	0.311683
9	0.415595	0.284355
10	0.502039	0.291002
11	0.379854	0.285094
12	0.143796	0.315376



Ex-2 (2): Pair wise Distances

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0.24	0.08	0.16	0.24	0.31	0.02	0.09	0.09	0.17	0.05	0.2
2	0.24	0	0.16	0.09	0.02	0.07	0.26	0.17	0.32	0.41	0.29	0.07
3	0.08	0.16	0	0.07	0.16	0.22	0.1	0.04	0.17	0.25	0.13	0.12
4	0.16	0.09	0.07	0	0.09	0.15	0.17	0.08	0.24	0.32	0.2	0.04
5	0.24	0.02	0.16	0.09	0	0.06	0.26	0.17	0.33	0.41	0.29	0.06
6	0.31	0.07	0.22	0.15	0.06	0	0.32	0.23	0.39	0.47	0.35	0.12
7	0.02	0.26	0.1	0.17	0.26	0.32	0	0.1	0.07	0.15	0.03	0.21
8	0.09	0.17	0.04	0.08	0.17	0.23	0.1	0	0.16	0.24	0.12	0.11
9	0.09	0.32	0.17	0.24	0.33	0.39	0.07	0.16	0	0.09	0.04	0.27
10	0.17	0.41	0.25	0.32	0.41	0.47	0.15	0.24	0.09	0	0.12	0.36
11	0.05	0.29	0.13	0.2	0.29	0.35	0.03	0.12	0.04	0.12	0	0.24
12	0.2	0.07	0.12	0.04	0.06	0.12	0.21	0.11	0.27	0.36	0.24	0

Kavg distances

0.06
0.0625
0.0725
0.07
0.0575
0.1
0.055
0.0775
0.0725
0.1325
0.06
0.07

 # dimensions = 2
minPts = 2*# dimensions = 4
K = minPts = 4 (for determining the Kavg distances)

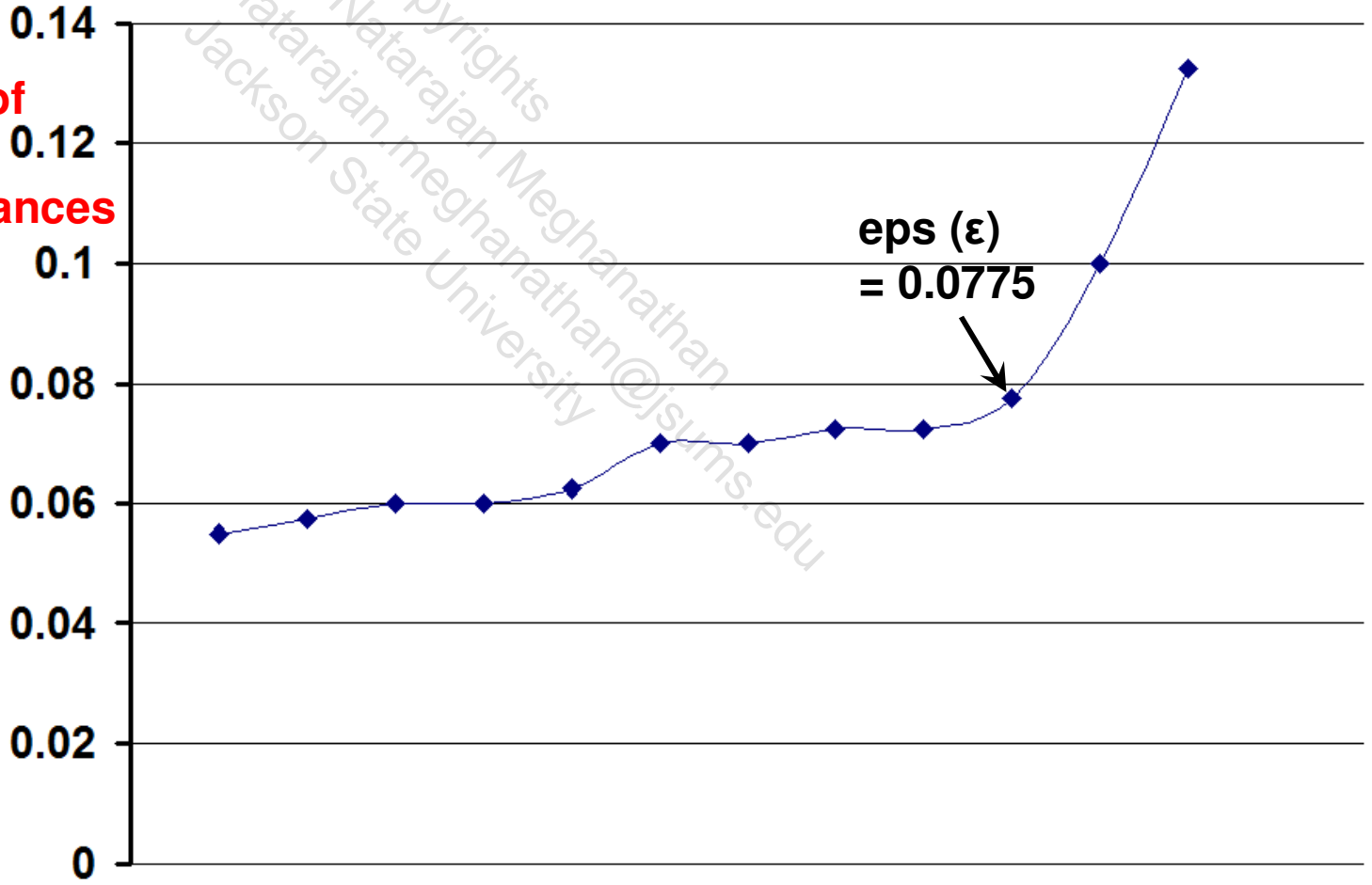
The closest K = 4 pair wise distances for each data point are highlighted and their average is determined

Ex-2 (3): Elbow Method to Determine eps (ε)

- Plot the sorted order of the kavg distances of the data points and identify the kavg value beyond which there is a steep increase in the slope.

Sorted order of the Kavg Pair wise distances

0.055
0.0575
0.06
0.06
0.0625
0.07
0.07
0.0725
0.0725
0.0775
0.1
0.1325



Ex-2 (4): ε-Neighborhood and Core/Non-core Data Points

Pair wise Distances <= ε Value of 0.0775 are highlighted

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0.24	0.08	0.16	0.24	0.31	0.02	0.09	0.09	0.17	0.05	0.2
2	0.24	0	0.16	0.09	0.02	0.07	0.26	0.17	0.32	0.41	0.29	0.07
3	0.08	0.16	0	0.07	0.16	0.22	0.1	0.04	0.17	0.25	0.13	0.12
4	0.16	0.09	0.07	0	0.09	0.15	0.17	0.08	0.24	0.32	0.2	0.04
5	0.24	0.02	0.16	0.09	0	0.06	0.26	0.17	0.33	0.41	0.29	0.06
6	0.31	0.07	0.22	0.15	0.06	0	0.32	0.23	0.39	0.47	0.35	0.12
7	0.02	0.26	0.1	0.17	0.26	0.32	0	0.1	0.07	0.15	0.03	0.21
8	0.09	0.17	0.04	0.08	0.17	0.23	0.1	0	0.16	0.24	0.12	0.11
9	0.09	0.32	0.17	0.24	0.33	0.39	0.07	0.16	0	0.09	0.04	0.27
10	0.17	0.41	0.25	0.32	0.41	0.47	0.15	0.24	0.09	0	0.12	0.36
11	0.05	0.29	0.13	0.2	0.29	0.35	0.03	0.12	0.04	0.12	0	0.24
12	0.2	0.07	0.12	0.04	0.06	0.12	0.21	0.11	0.27	0.36	0.24	0

A data point is a Core/C point if 1+|ε-Neighborhood| is >= minPts (4)

Otherwise, it is a Non-core/nc point

Data Point	ε-Neighborhood	1 + |ε-Neighborhood|	Core/C or Non-Core/nc
1	7, 11	3	nc
2	5, 6, 12	4	c
3	4, 8	3	nc
4	3, 12	3	nc
5	2, 6, 12	4	c
6	2, 5	3	nc
7	1, 9, 11	4	c
8	3	2	nc
9	7, 11	3	nc
10	-	1	nc
11	1, 7, 9	4	c
12	2, 4, 5	4	c

Ex-2 (5): Cluster 1

Data Point	ε-Neighborhood	1 + |ε-Neighborhood|	Core/C or Non-Core/nc
1	7, 11	3	nc
2	5, 6, 12	4	c
3	4, 8	3	nc
4	3, 12	3	nc
5	2, 6, 12	4	c
6	2, 5	3	nc
7	1, 9, 11	4	c
8	3	2	nc
9	7, 11	3	nc
10	-	1	nc
11	1, 7, 9	4	c
12	2, 4, 5	4	c

Start with a randomly chosen core point (say, data point # 7) and grow the cluster based on ε-Neighborhood by exploring the core points included

C1 = {7}
C1 = {7, 1, 9, 11}
C1 = {7, 1, 9, 11}
The growth of cluster C1 stops here as 1 and 9 are non-core/nc points.

Ex-2 (5): Cluster 2

Data Point	ε-Neighborhood	1 + |ε-Neighborhood|	Core/C or Non-Core/nc
1	7, 11	3	nc
2	5, 6, 12	4	c
3	4, 8	3	nc
4	3, 12	3	nc
5	2, 6, 12	4	c
6	2, 5	3	nc
7	1, 9, 11	4	c
8	3	2	nc
9	7, 11	3	nc
10	-	1	nc
11	1, 7, 9	4	c
12	2, 4, 5	4	c

Start with a randomly chosen core point among the remaining core points (2, 5 and 12). Lets say, we pick data point 5 and grow the cluster based on ε-Neighborhood by exploring the core points included

C2 = {5}
C2 = {5, 2, 6, 12}
C2 = {5, 2, 6, 12}
C2 = {5, 2, 6, 12, 4} STOP!

Ex-2 (5): Outliers

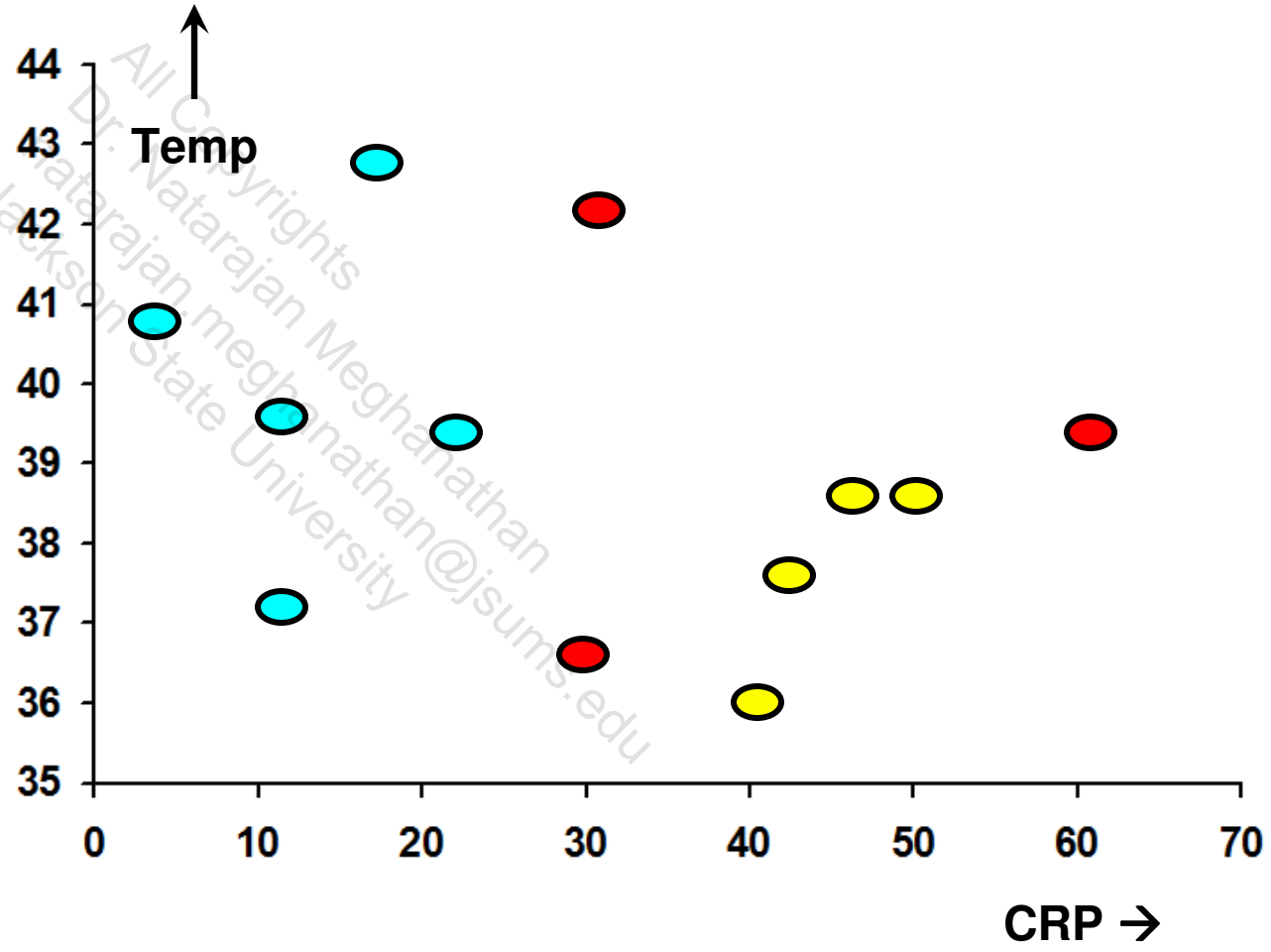
Data Point	ε-Neighborhood	1 + |ε-Neighborhood|	Core/C or Non-Core/nc
1	7, 11	3	nc
2	5, 6, 12	4	c
3	4, 8	3	nc
4	3, 12	3	nc
5	2, 6, 12	4	c
6	2, 5	3	nc
7	1, 9, 11	4	c
8	3	2	nc
9	7, 11	3	nc
10	-	1	nc
11	1, 7, 9	4	c
12	2, 4, 5	4	c

There are no more core points that can be considered for cluster formation.

All the remaining non-selected data points (3, 8 and 10) are non-core/nc points. They are considered as outliers.

Ex-2: Clusters 1 and 2, Outliers

	CRP	Temp, C
1	40	36
2	11.1	37.2
3	30	36.5
4	21.4	39.4
5	10.7	39.6
6	3.4	40.7
7	42	37.6
8	31.1	42.2
9	50	38.5
10	60.4	39.4
11	45.7	38.6
12	17.3	42.7



C1 = {7, 1, 9, 11}
C2 = {5, 2, 6, 12, 4}

Outliers = {3, 8, 10}