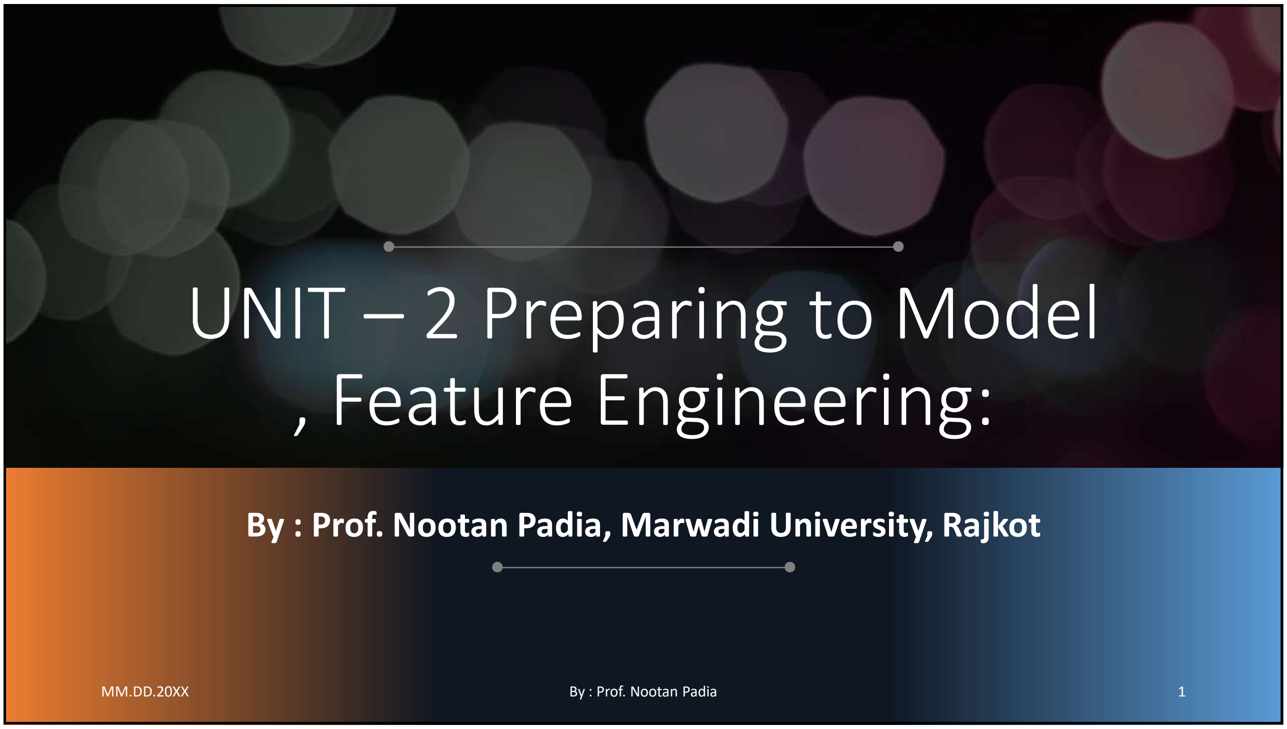
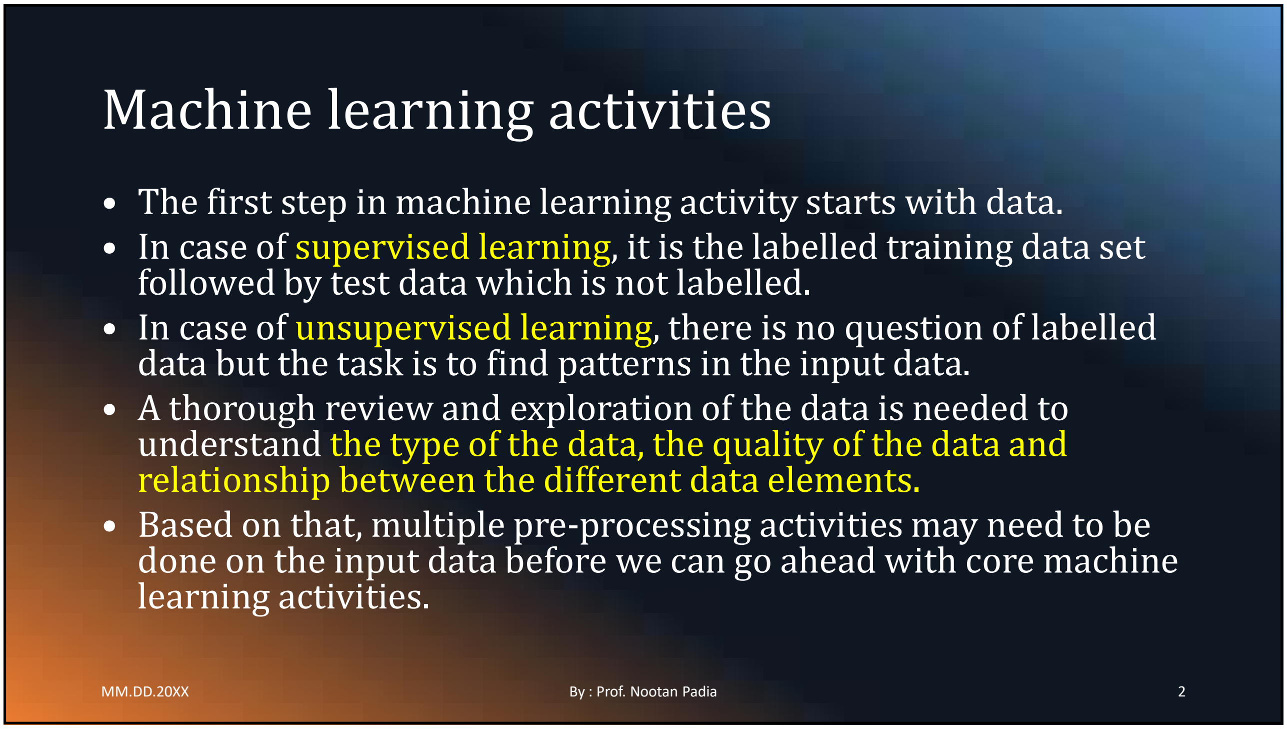


UNIT – 2 Preparing to Model , Feature Engineering:

By : Prof. Nootan Padia, Marwadi University, Rajkot

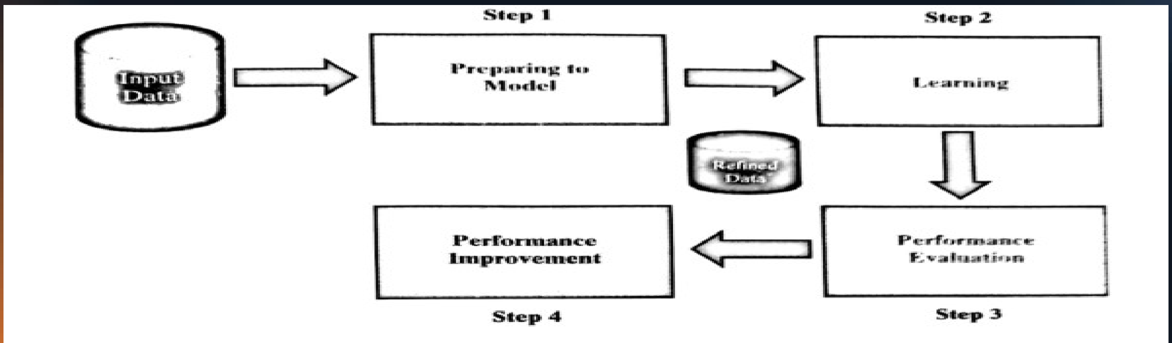


Machine learning activities

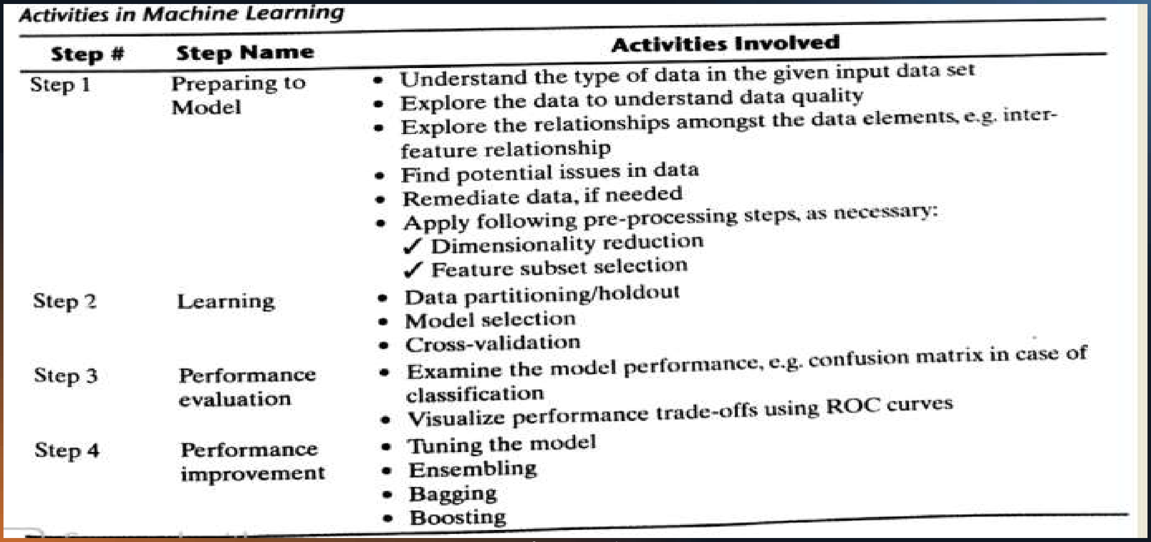
- The first step in machine learning activity starts with data.
- In case of supervised learning, it is the labelled training data set followed by test data which is not labelled.
- In case of unsupervised learning, there is no question of labelled data but the task is to find patterns in the input data.
- A thorough review and exploration of the data is needed to understand the type of the data, the quality of the data and relationship between the different data elements.
- Based on that, multiple pre-processing activities may need to be done on the input data before we can go ahead with core machine learning activities.

- Following are the typical preparation activities done once the input data comes into the machine learning system:
  - Understand the type of data in the given input data set.
  - Explore the data to understand the nature and quality.
  - Explore the relationships amongst the data elements, e.g. inter-feature relationship.
  - Find potential issues in data.
  - Do the necessary remediation, e.g. assign missing data values, etc., if needed.
  - Apply pre-processing steps, as necessary.
  - Once the data is prepared for modeling, then the learning tasks start off. As a part of it, do the following activities:
    - The input data is first divided into parts — the training data and the test data (called holdout). This step is applicable for supervised learning only.
    - Consider different models or learning algorithms for selection.
    - Train the model based on the training data for supervised learning problem and apply to unknown data. Directly apply the chosen unsupervised model on the input data for unsupervised learning problem.

- After the model is selected, trained (for supervised learning), and applied on input data, the performance of the model is evaluated.
- Based on options available, specific actions can be taken to improve the performance of the model, if possible.
- Four step machine learning process :



- Table shown below contains a summary of steps and activities involved.

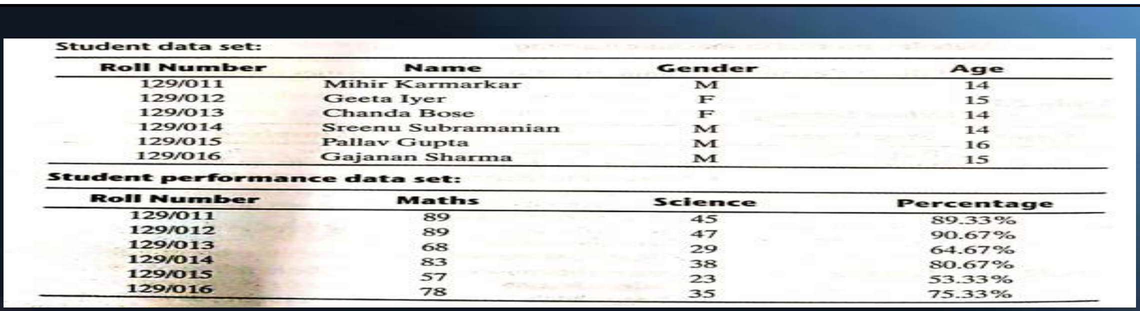


Activities in Machine Learning

Step #	Step Name	Activities Involved
Step 1	Preparing to Model	• Understand the type of data in the given input data set • Explore the data to understand data quality • Explore the relationships amongst the data elements, e.g. inter-feature relationship • Find potential issues in data • Remediate data, if needed • Apply following pre-processing steps, as necessary: ✓ Dimensionality reduction ✓ Feature subset selection
Step 2	Learning	• Data partitioning/holdout • Model selection • Cross-validation
Step 3	Performance evaluation	• Examine the model performance, e.g. confusion matrix in case of classification • Visualize performance trade-offs using ROC curves
Step 4	Performance improvement	• Tuning the model • Ensembling • Bagging • Boosting

Basic types of data in ML

- Before starting with types of data, let's first understand what a data set is and what are the elements of a data set.
- A data set is a collection of related information or records.
- The information may be on some entity or some subject area.
- For example, we may have a data set on students in which each record consists of information about a specific student.
- Again, we can have a data set on student performance which has records providing performance, i.e. marks on the individual subjects.
- Each row of a data set is called a record.
- Each data set also has multiple attributes, each of which gives information on a specific characteristic.
- Example dataset is shown in next slide:



Student data set:

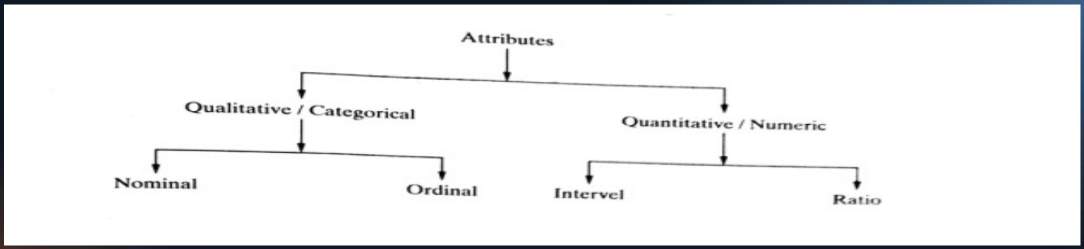
Roll Number	Name	Gender	Age
129/011	Mihir Karmarkar	M	14
129/012	Geeta Iyer	F	15
129/013	Chanda Bose	F	14
129/014	Sreenu Subramanian	M	14
129/015	Pallav Gupta	M	16
129/016	Gajanan Sharma	M	15

Student performance data set:

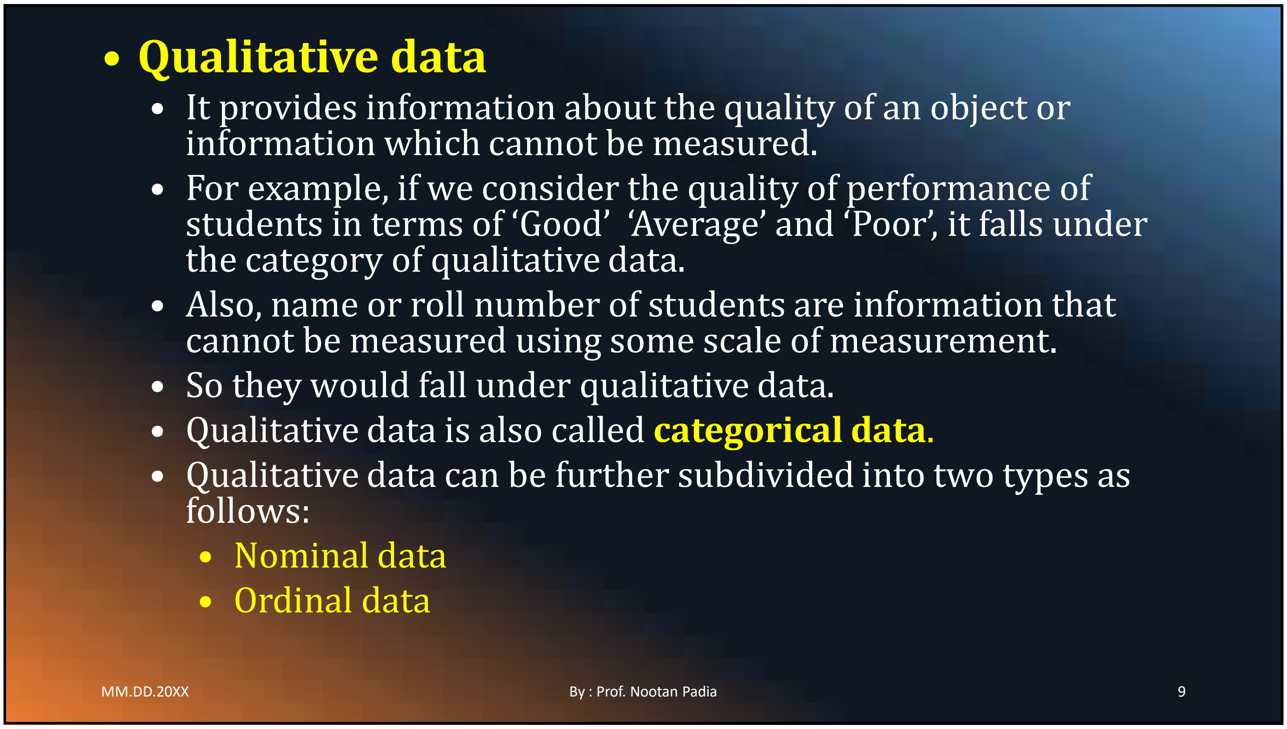
Roll Number	Maths	Science	Percentage
129/011	89	45	89.33%
129/012	89	47	90.67%
129/013	68	29	64.67%
129/014	83	38	80.67%
129/015	57	23	53.33%
129/016	78	35	75.33%

- For example, in the data set on students, there are four attributes namely Roll Number, Name, Gender, and Age, each of which understandably is a specific characteristic about the student entity.
- Attributes can also be termed as feature, variable, dimension or field.
- Both the data sets, Student and Student Performance, are having four features or dimensions; hence they are told to have four-dimensional data space.
- A row or Section record represents a point in the four-dimensional data space as each row has specific values for each of the four attributes or features.
- Value, of an attribute, quite understandably, may vary from record to record.
- For example, if we refer to the first two records in the Student data set, the value of attributes Name, Gender, and Age are different .

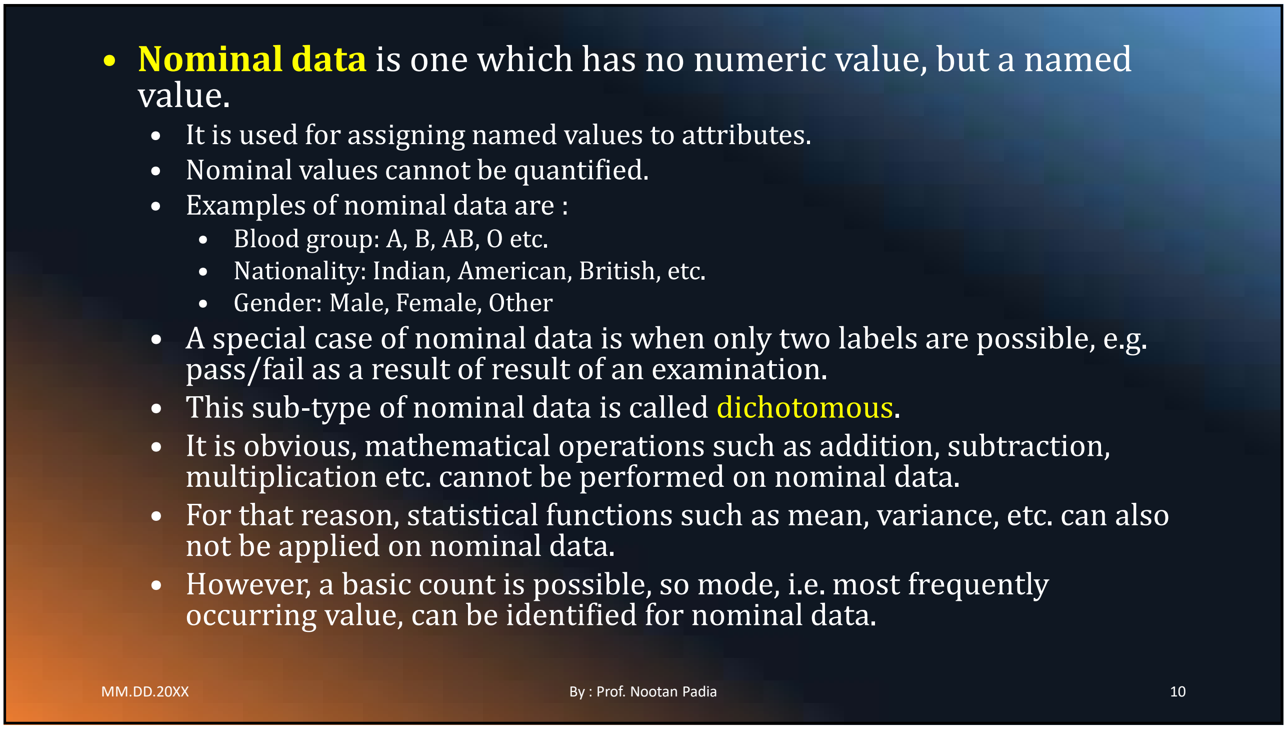
- Now that a context of data sets is given, let's try to understand the different types of data that we generally come across in machine learning problems.



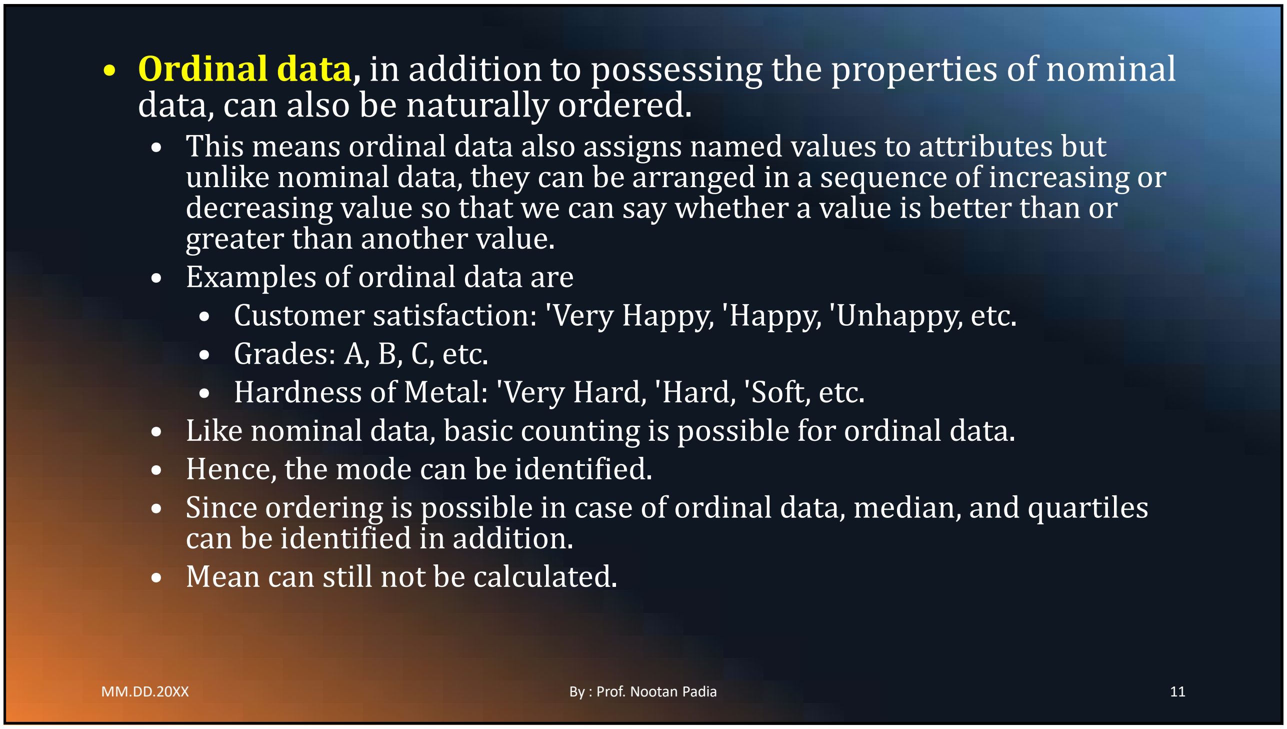
- Data can broadly be divided into following two types:
  - Qualitative data
  - Quantitative data



- **Qualitative data**
  - It provides information about the quality of an object or information which cannot be measured.
  - For example, if we consider the quality of performance of students in terms of 'Good' 'Average' and 'Poor', it falls under the category of qualitative data.
  - Also, name or roll number of students are information that cannot be measured using some scale of measurement.
  - So they would fall under qualitative data.
  - Qualitative data is also called **categorical data**.
  - Qualitative data can be further subdivided into two types as follows:
    - Nominal data
    - Ordinal data

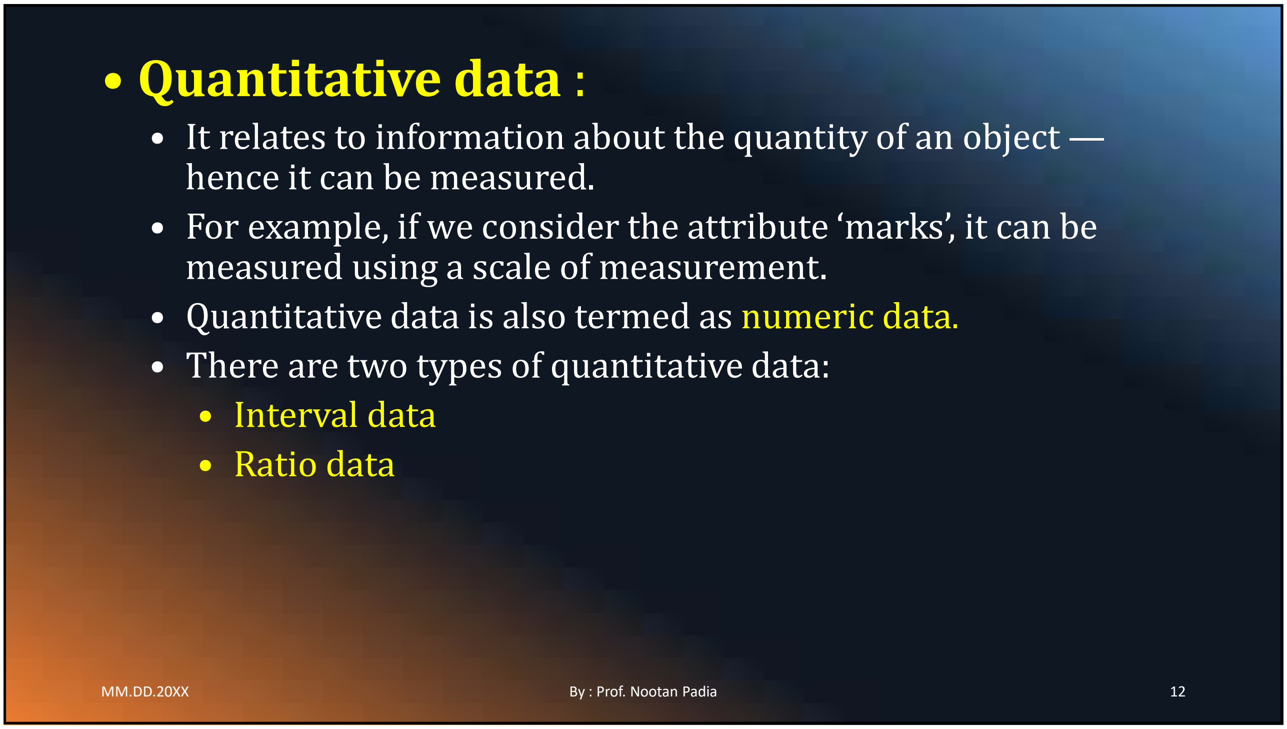


- **Nominal data** is one which has no numeric value, but a named value.
  - It is used for assigning named values to attributes.
  - Nominal values cannot be quantified.
  - Examples of nominal data are :
    - Blood group: A, B, AB, O etc.
    - Nationality: Indian, American, British, etc.
    - Gender: Male, Female, Other
  - A special case of nominal data is when only two labels are possible, e.g. pass/fail as a result of result of an examination.
  - This sub-type of nominal data is called dichotomous.
  - It is obvious, mathematical operations such as addition, subtraction, multiplication etc. cannot be performed on nominal data.
  - For that reason, statistical functions such as mean, variance, etc. can also not be applied on nominal data.
  - However, a basic count is possible, so mode, i.e. most frequently occurring value, can be identified for nominal data.



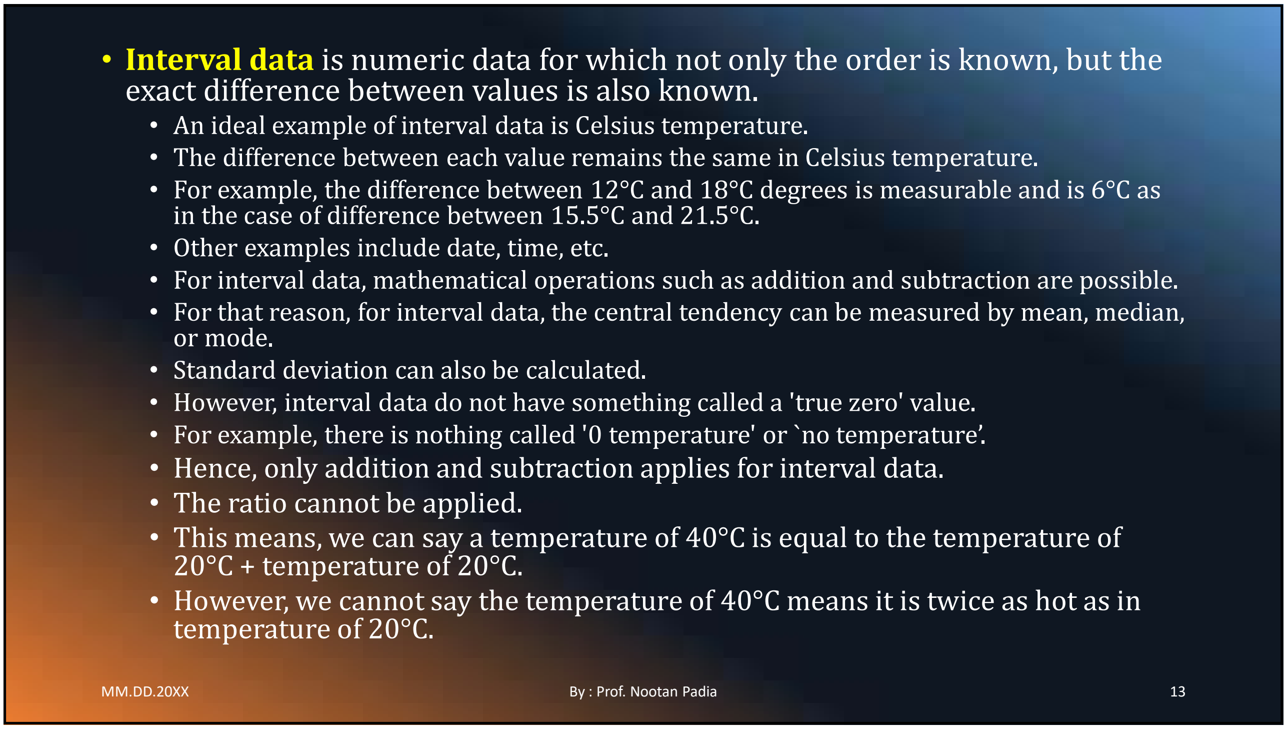
- **Ordinal data,** in addition to possessing the properties of nominal data, can also be naturally ordered.
  - This means ordinal data also assigns named values to attributes but unlike nominal data, they can be arranged in a sequence of increasing or decreasing value so that we can say whether a value is better than or greater than another value.
  - Examples of ordinal data are
    - Customer satisfaction: 'Very Happy, 'Happy, 'Unhappy, etc.
    - Grades: A, B, C, etc.
    - Hardness of Metal: 'Very Hard, 'Hard, 'Soft, etc.
  - Like nominal data, basic counting is possible for ordinal data.
  - Hence, the mode can be identified.
  - Since ordering is possible in case of ordinal data, median, and quartiles can be identified in addition.
  - Mean can still not be calculated.

MM.DD.20XX By : Prof. Nootan Padia 11

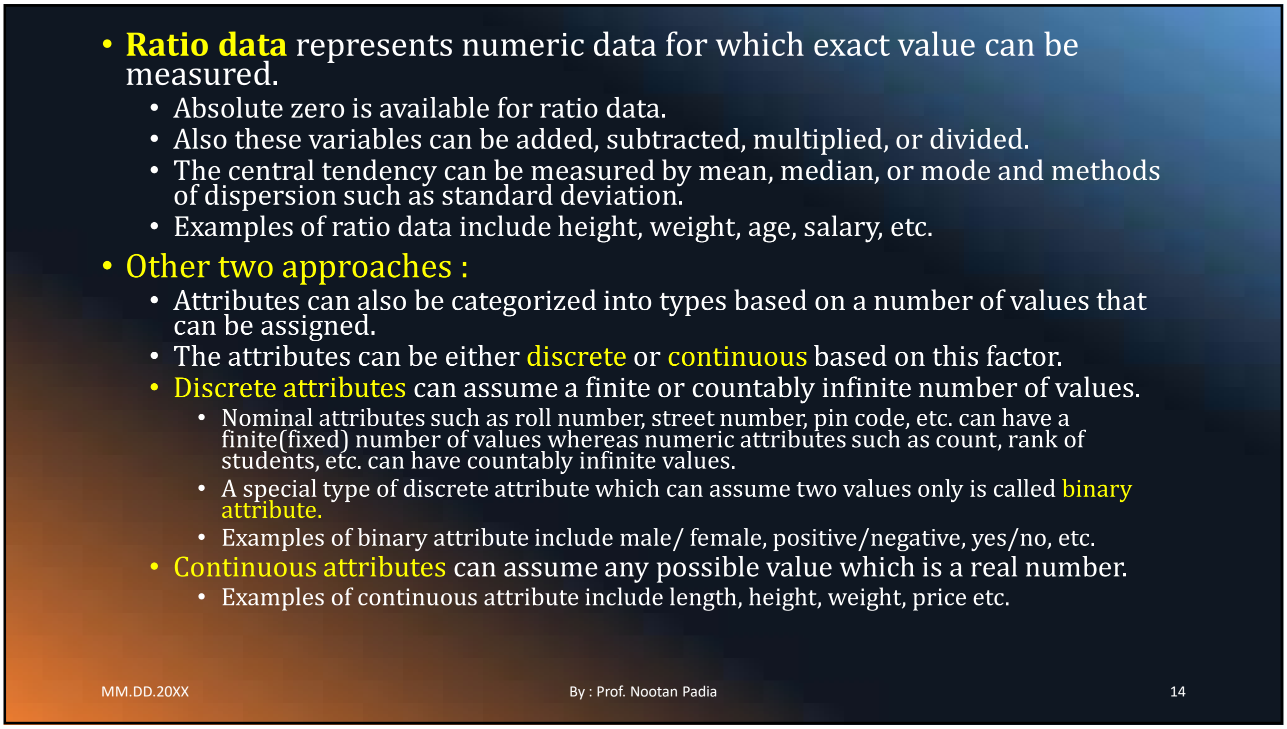


- **Quantitative data :**
  - It relates to information about the quantity of an object — hence it can be measured.
  - For example, if we consider the attribute ‘marks’, it can be measured using a scale of measurement.
  - Quantitative data is also termed as numeric data.
  - There are two types of quantitative data:
    - Interval data
    - Ratio data

MM.DD.20XX By : Prof. Nootan Padia 12



- **Interval data** is numeric data for which not only the order is known, but the exact difference between values is also known.
  - An ideal example of interval data is Celsius temperature.
  - The difference between each value remains the same in Celsius temperature.
  - For example, the difference between 12°C and 18°C degrees is measurable and is 6°C as in the case of difference between 15.5°C and 21.5°C.
  - Other examples include date, time, etc.
  - For interval data, mathematical operations such as addition and subtraction are possible.
  - For that reason, for interval data, the central tendency can be measured by mean, median, or mode.
  - Standard deviation can also be calculated.
  - However, interval data do not have something called a 'true zero' value.
  - For example, there is nothing called '0 temperature' or `no temperature'.
  - Hence, only addition and subtraction applies for interval data.
  - The ratio cannot be applied.
  - This means, we can say a temperature of 40°C is equal to the temperature of 20°C + temperature of 20°C.
  - However, we cannot say the temperature of 40°C means it is twice as hot as in temperature of 20°C.

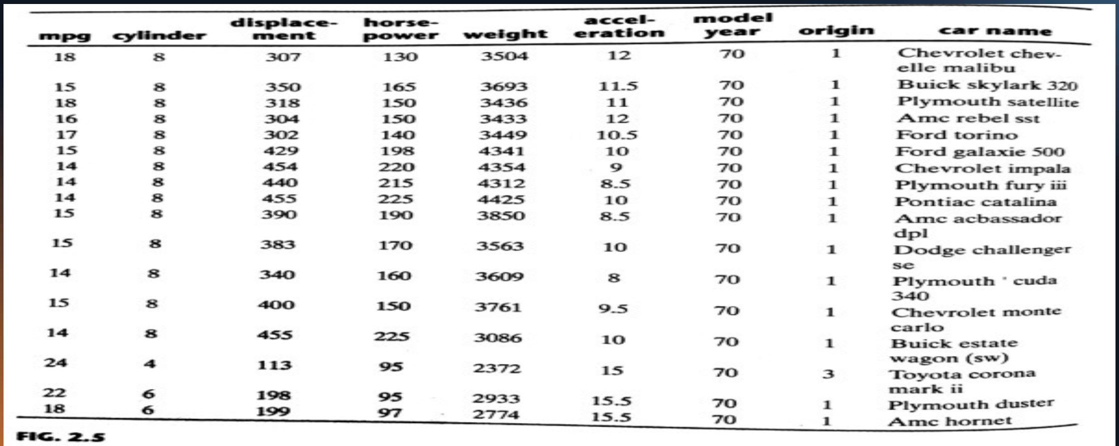


- **Ratio data** represents numeric data for which exact value can be measured.
  - Absolute zero is available for ratio data.
  - Also these variables can be added, subtracted, multiplied, or divided.
  - The central tendency can be measured by mean, median, or mode and methods of dispersion such as standard deviation.
  - Examples of ratio data include height, weight, age, salary, etc.
- Other two approaches :
  - Attributes can also be categorized into types based on a number of values that can be assigned.
  - The attributes can be either discrete or continuous based on this factor.
  - Discrete attributes can assume a finite or countably infinite number of values.
    - Nominal attributes such as roll number, street number, pin code, etc. can have a finite(fixed) number of values whereas numeric attributes such as count, rank of students, etc. can have countably infinite values.
    - A special type of discrete attribute which can assume two values only is called binary attribute.
    - Examples of binary attribute include male/ female, positive/negative, yes/no, etc.
  - Continuous attributes can assume any possible value which is a real number.
    - Examples of continuous attribute include length, height, weight, price etc.

Exploring structure of data

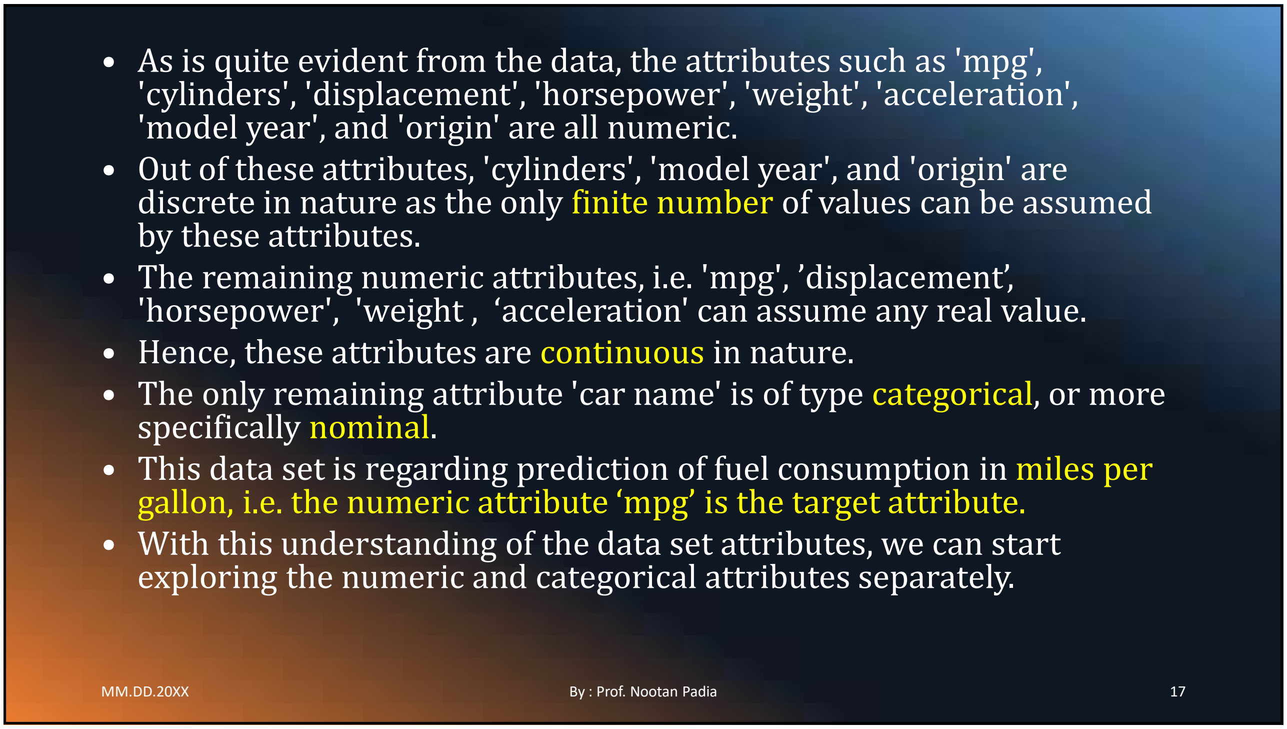
- We need to understand that in a data set, which of the attributes are numeric and which are categorical in nature.
- This is because the approach of exploring numeric data is different than the approach of exploring categorical data.
- In case of a standard data set, we may have the data dictionary available for reference.
- Data dictionary is a metadata repository, i.e. the repository of all information related t to the structure of each data element contained in the data set.
- The data dictionary gives detailed information on each of the attributes — the description as well as the data type and other relevant details.
- In case the data dictionary is not available, we need to use standard library function of the machine learning tool that we are using to get the details.

- The data set that we take as a reference is the Auto MPG data set available in the UCI repository.
- Following figure is a snapshot of the first few rows of the data set.

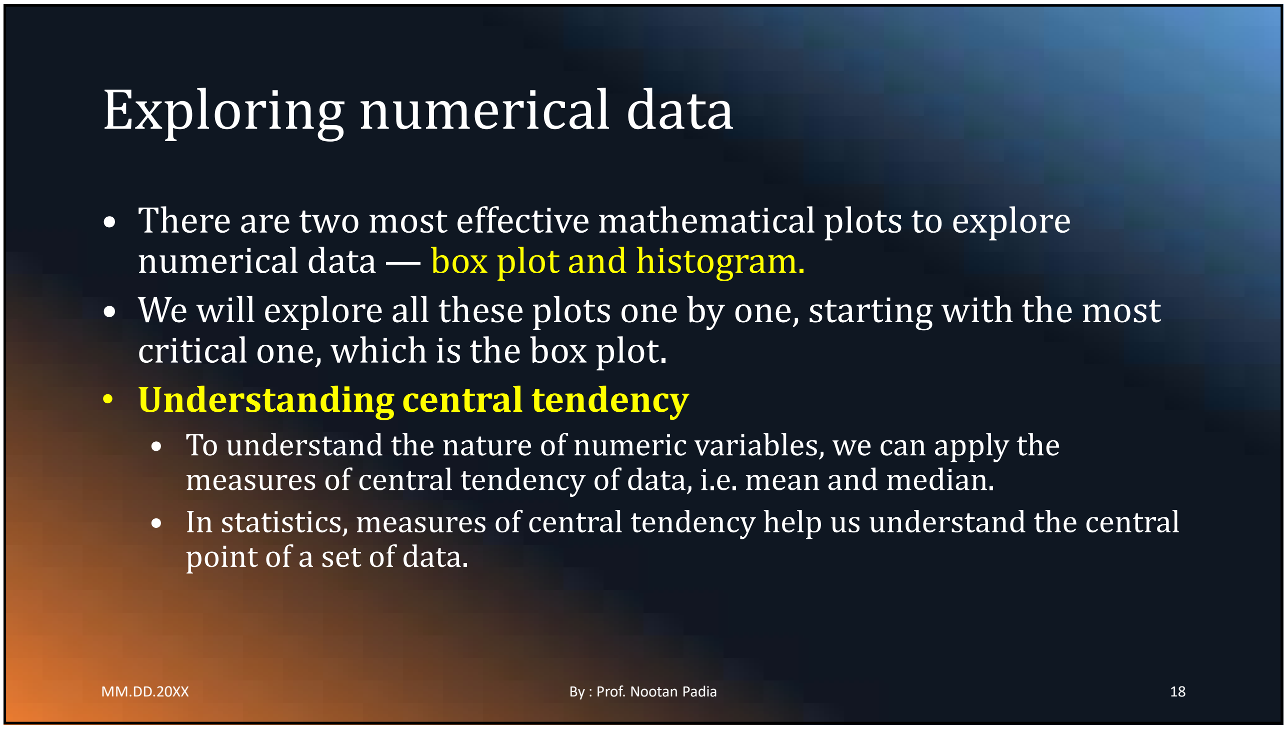


mpg	cylinder	displacement	horsepower	weight	acceleration	model year	origin	car name
18	8	307	130	3504	12	70	1	Chevrolet chevelle malibu
15	8	350	165	3693	11.5	70	1	Buick skylark 320
18	8	318	150	3436	11	70	1	Plymouth satellite
16	8	304	150	3433	12	70	1	Amc rebel sst
17	8	302	140	3449	10.5	70	1	Ford torino
15	8	429	198	4341	10	70	1	Ford galaxie 500
14	8	454	220	4354	9	70	1	Chevrolet impala
14	8	440	215	4312	8.5	70	1	Plymouth fury iii
14	8	455	225	4425	10	70	1	Pontiac catalina
15	8	390	190	3850	8.5	70	1	Amc acbassador dpl
15	8	383	170	3563	10	70	1	Dodge challenger se
14	8	340	160	3609	8	70	1	Plymouth ' cuda 340
15	8	400	150	3761	9.5	70	1	Chevrolet monte carlo
14	8	455	225	3086	10	70	1	Buick estate wagon (sw)
24	4	113	95	2372	15	70	3	Toyota corona mark ii
22	6	198	95	2933	15.5	70	1	Plymouth duster
18	6	199	97	2774	15.5	70	1	Amc hornet

FIG. 2.5
Auto MPG data set

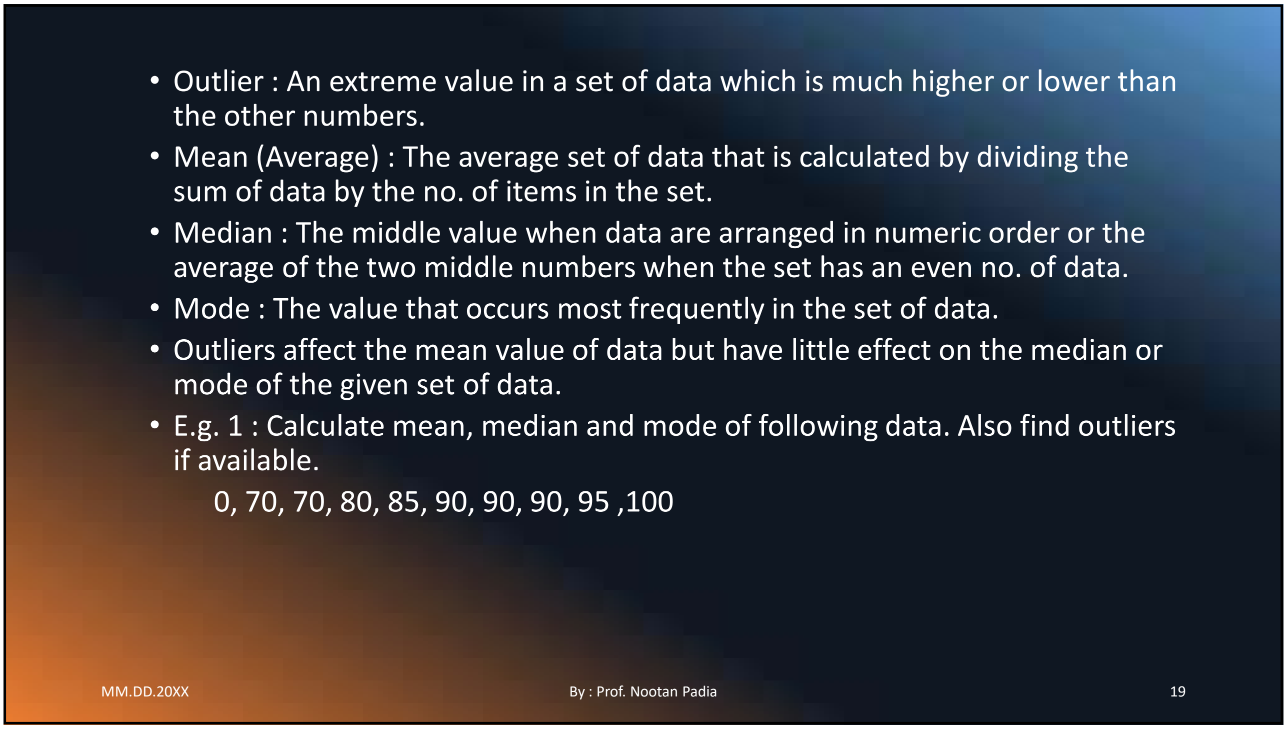


- As is quite evident from the data, the attributes such as 'mpg', 'cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', 'model year', and 'origin' are all numeric.
- Out of these attributes, 'cylinders', 'model year', and 'origin' are discrete in nature as the only finite number of values can be assumed by these attributes.
- The remaining numeric attributes, i.e. 'mpg', ’displacement’, 'horsepower', 'weight , ‘acceleration' can assume any real value.
- Hence, these attributes are continuous in nature.
- The only remaining attribute 'car name' is of type categorical, or more specifically nominal.
- This data set is regarding prediction of fuel consumption in miles per gallon, i.e. the numeric attribute ‘mpg’ is the target attribute.
- With this understanding of the data set attributes, we can start exploring the numeric and categorical attributes separately.



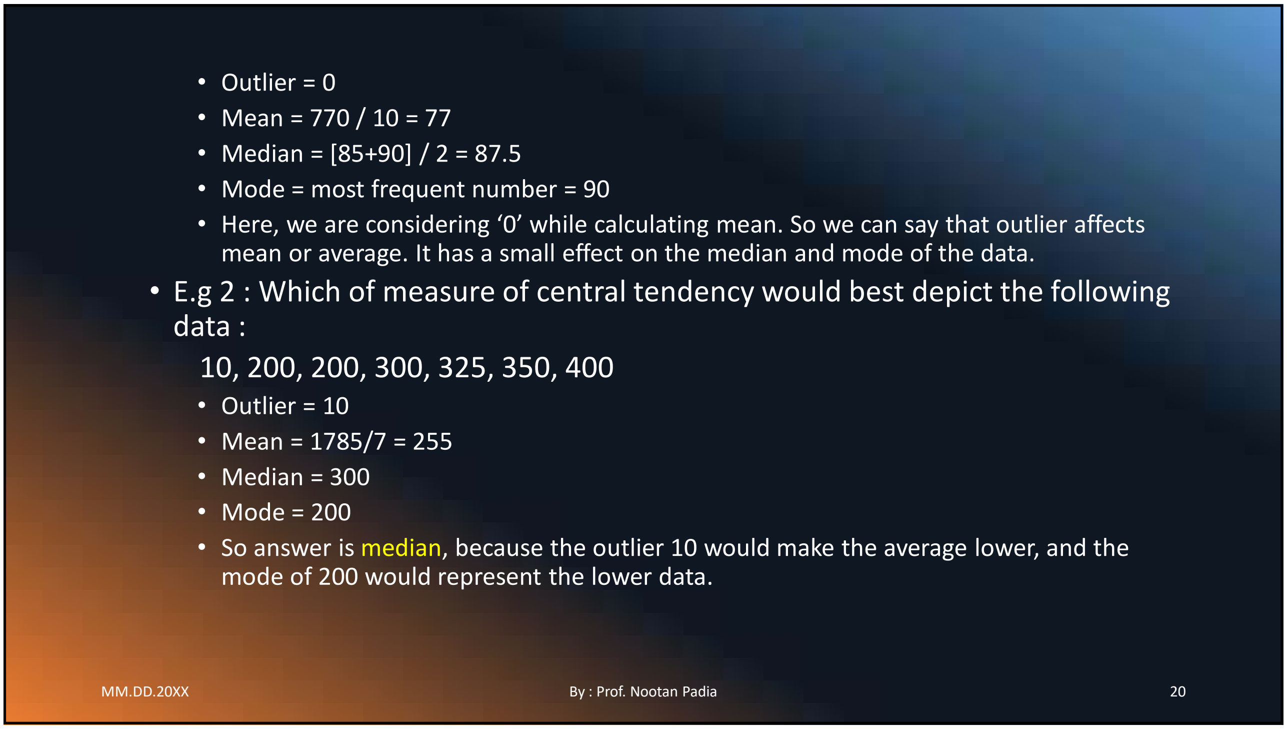
Exploring numerical data

- There are two most effective mathematical plots to explore numerical data — box plot and histogram.
- We will explore all these plots one by one, starting with the most critical one, which is the box plot.
- **Understanding central tendency**
  - To understand the nature of numeric variables, we can apply the measures of central tendency of data, i.e. mean and median.
  - In statistics, measures of central tendency help us understand the central point of a set of data.



- Outlier : An extreme value in a set of data which is much higher or lower than the other numbers.
- Mean (Average) : The average set of data that is calculated by dividing the sum of data by the no. of items in the set.
- Median : The middle value when data are arranged in numeric order or the average of the two middle numbers when the set has an even no. of data.
- Mode : The value that occurs most frequently in the set of data.
- Outliers affect the mean value of data but have little effect on the median or mode of the given set of data.
- E.g. 1 : Calculate mean, median and mode of following data. Also find outliers if available.

  0, 70, 70, 80, 85, 90, 90, 90, 95 ,100

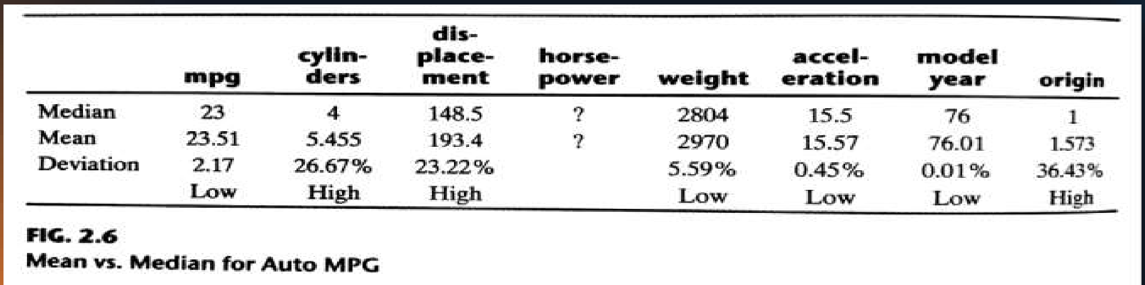


  - Outlier = 0
  - Mean = 770 / 10 = 77
  - Median = [85+90] / 2 = 87.5
  - Mode = most frequent number = 90
  - Here, we are considering ‘0’ while calculating mean. So we can say that outlier affects mean or average. It has a small effect on the median and mode of the data.
- E.g 2 : Which of measure of central tendency would best depict the following data :

  10, 200, 200, 300, 325, 350, 400

  - Outlier = 10
  - Mean = 1785/7 = 255
  - Median = 300
  - Mode = 200
  - So answer is median, because the outlier 10 would make the average lower, and the mode of 200 would represent the lower data.

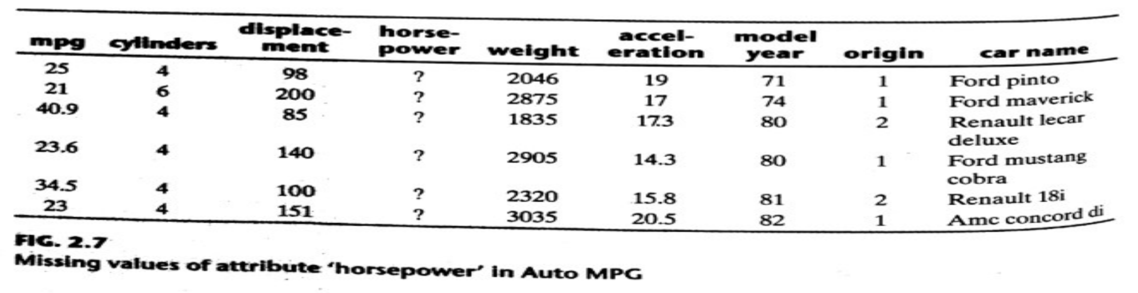
- Mean is likely to get shifted drastically even due to the presence of a small number of outliers.
- If we observe that for certain attributes the deviation between values of mean and median are quite high, we should investigate those attributes further and try to find out the root cause along with the need for remediation.
- In following Figure, the comparison between mean and median for all the attributes has been shown.
- We can see that for the attributes such as 'mpg, 'weight, 'acceleration', and 'model.year' the deviation between mean and median is not significant which means the chance of these attributes having outlier values is less.
- However, the deviation is significant for the attributes 'cylinders', 'displacement' and 'origin'.
- So, we need to further drill down and look at some more statistics for these attributes.



	mpg	cylin-ders	dis-place-ment	horse-power	weight	accel-eration	model year	origin
Median	23	4	148.5	?	2804	15.5	76	1
Mean	23.51	5.455	193.4	?	2970	15.57	76.01	1.573
Deviation	2.17	26.67%	23.22%		5.59%	0.45%	0.01%	36.43%
	Low	High	High		Low	Low	Low	High

FIG. 2.6
Mean vs. Median for Auto MPG

- Also, there is some problem in the values of the attribute 'horsepower' because of which the mean and median calculation is not possible.
- With a bit of investigation, we can find out that the problem is occurring because of the 6 data elements, as shown in Figure 2.7, do not have value for the attribute `horsepower'.
- For that reason, the attribute 'horsepower' is not treated as a numeric.
- That's why the operations applicable on numeric variables, like mean or median, are failing.
- So we have to first remediate the missing values of the attribute 'horsepower' before being able to do any kind of exploration.



mpg	cylinders	displace-ment	horse-power	weight	accel-eration	model year	origin	car name
25	4	98	?	2046	19	71	1	Ford pinto
21	6	200	?	2875	17	74	1	Ford maverick
40.9	4	85	?	1835	17.3	80	2	Renault lecar deluxe
23.6	4	140	?	2905	14.3	80	1	Ford mustang cobra
34.5	4	100	?	2320	15.8	81	2	Renault 18i
23	4	151	?	3035	20.5	82	1	Amc concord di

FIG. 2.7
Missing values of attribute 'horsepower' in Auto MPG

- **Understanding data spread**
  - We have explored the central tendency of the different numeric attributes, so we have a clear idea of which attributes have a large deviation between mean and median.
  - Let's look closely at those attributes.
  - To drill down more, we need to look at the entire range of values of the attributes, though not at the level of data elements as that may be too vast to review manually.
  - So we will take a granular view of the data spread in the form of
    - Dispersion of data
    - Position of the different data values

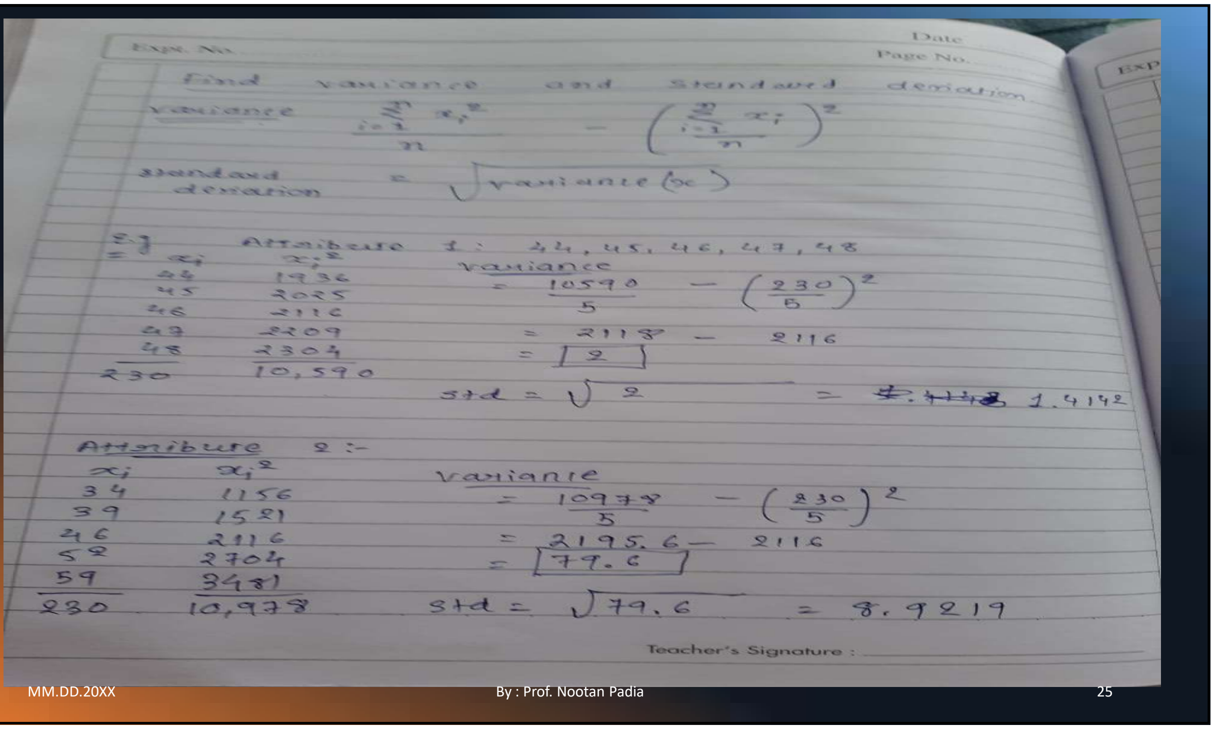
- **Measuring data dispersion**
  - Consider the data values of two attributes
    - Attribute 1 values: 44, 46, 48, 45, and 47
    - Attribute 2 values: 34, 46, 59, 39, and 52
  - Both the set of values have a mean and median of 46.
  - However, the first set of values that is of attribute 1 is more concentrated or clustered around the mean/median value whereas the second set of values of attribute 2 is quite spread out or dispersed.
  - To measure the extent of dispersion of a data, or to find out how much the different values of a data are spread out, the variance of the data is measured.
  - The variance of a data is measured using the formula given below:
-

$$\text{Variance } (x) = \frac{\sum_{i=1}^{n} x_i^2}{n} - \left(\frac{\sum_{i=1}^{n} x_i}{n}\right)^2, \text{where } x \text{ is the variable or attribute w}$$

  - Here x is the variable or attribute whose variance is to be measured and n is the number of observations or values of variable x
  - Standard deviation of a data is measured as below :

$$\text{Standard deviation } (x) = \sqrt{\text{Variance } (x)}$$

  - From calculation of variance, we can say that attribute 1 values are quite concentrated around the mean while attribute 2 values are extremely spread out.



Expt. No.

Date

Page No.

Find variance and standard deviation

variance $\dfrac{\sum_{i=1}^{n} x_i^2}{n} - \left(\dfrac{\sum_{i=1}^{n} x_i}{n}\right)^2$

standard deviation $= \sqrt{\text{variance}(x)}$

E.g Attribute 1 : 44, 45, 46, 47, 48

x_i	x_i^2
44	1936
45	2025
46	2116
47	2209
48	2304
230	10,590

variance

$= \dfrac{10590}{5} - \left(\dfrac{230}{5}\right)^2$

$= 2118 - 2116$

$= \boxed{2}$

std $= \sqrt{2} = 1.4142$

Attribute 2 :-

x_i	x_i^2
34	1156
39	1521
46	2116
52	2704
59	3481
230	10,978

varianle

$= \dfrac{10978}{5} - \left(\dfrac{230}{5}\right)^2$

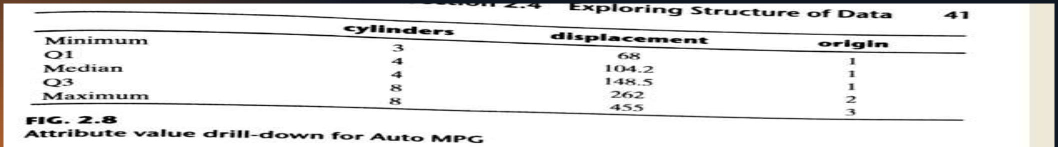
$= 2195.6 - 2116$

$= \boxed{79.6}$

std $= \sqrt{79.6} = 8.9219$

Teacher's Signature :

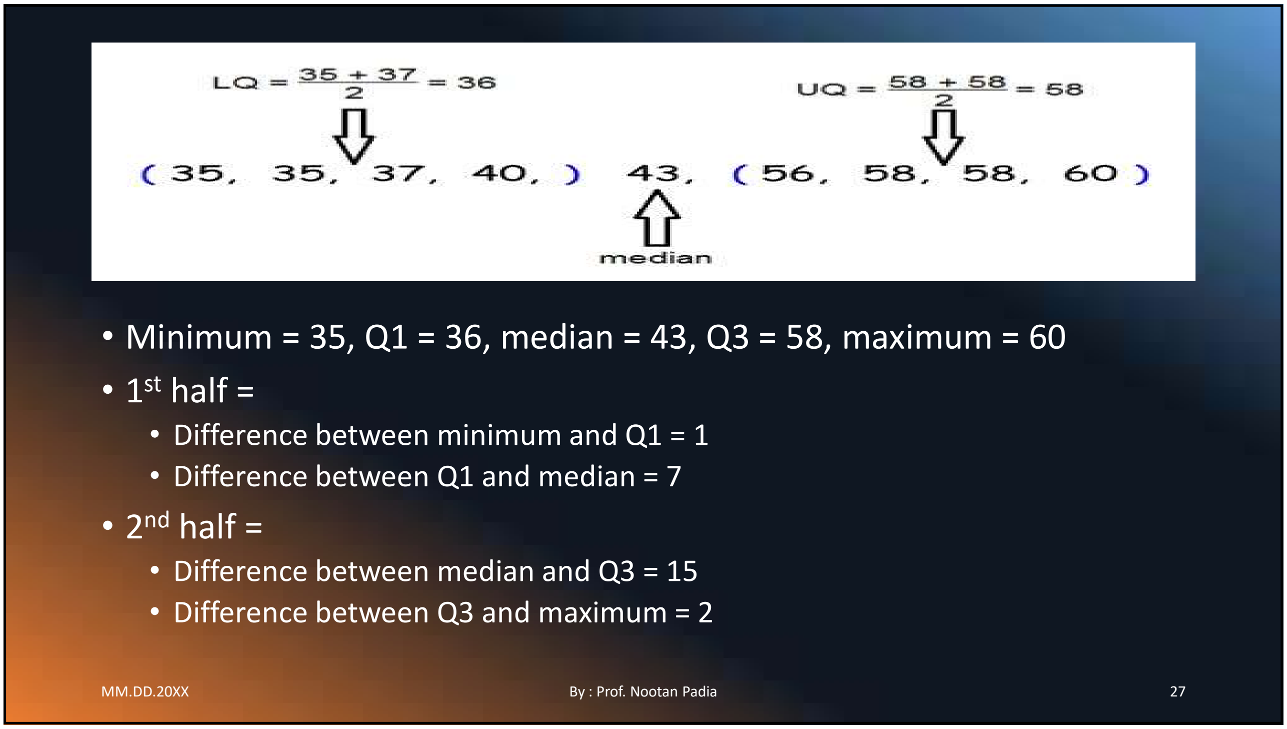
- Measuring data value position
  - When the data values of an attribute are arranged in an increasing order, we have seen earlier that median gives the central data value, which divides the entire data set into two halves.
  - Similarly, if the first half of the data is divided into two halves so that each half consists of one-quarter of the data set, then that median of the first half is known as first quartile or Q1.
  - In the same way, if the second half of the data is divided into two halves, then that median of the second half is known as third quartile or Q3.
  - The overall median is also known as second quartile or Q2.
  - So, any data set has five values - minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum.
  - Let's review these values for the attributes 'cylinders, 'displacement, and 'origin'.
  - Figure 2.8 captures a summary of the range of statistics for the attributes.



Exploring Structure of Data 41

	cylinders	displacement	origin
Minimum	3	68	1
Q1	4	104.2	1
Median	4	148.5	1
Q3	8	262	2
Maximum	8	455	3

FIG. 2.8
Attribute value drill-down for Auto MPG



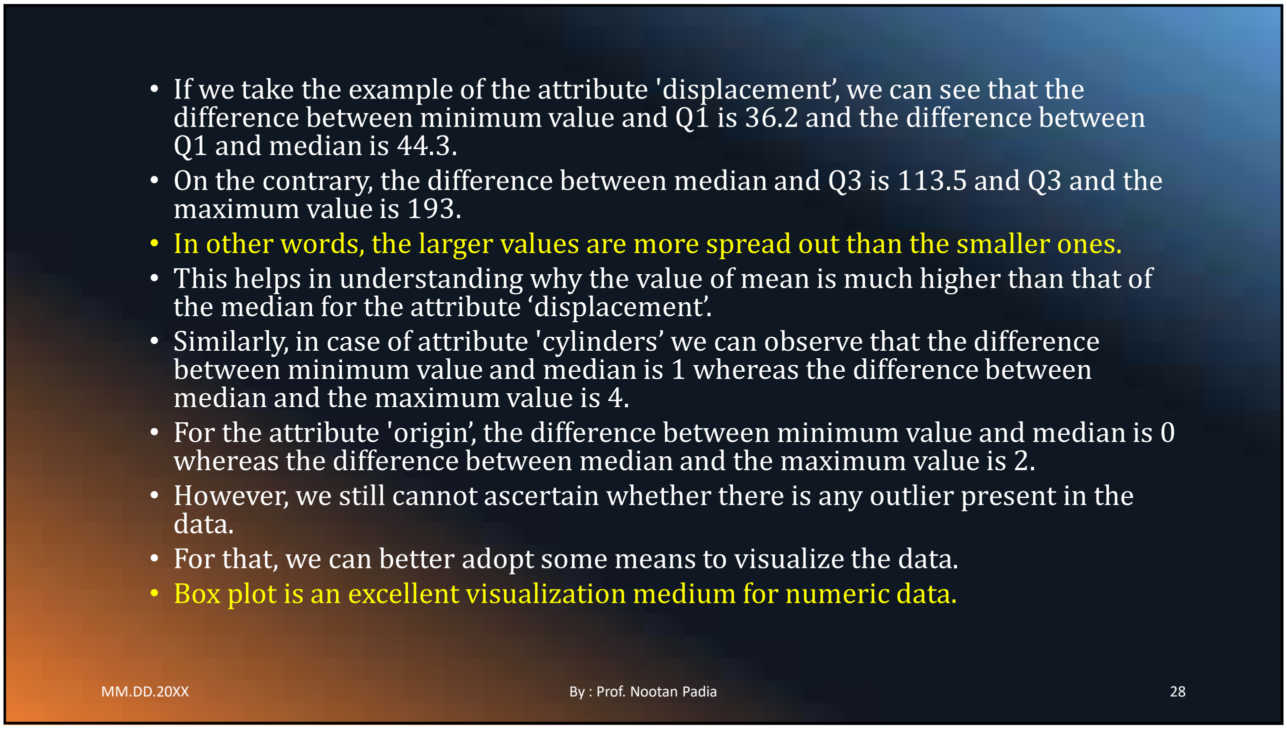
$LQ = \frac{35 + 37}{2} = 36$

$UQ = \frac{58 + 58}{2} = 58$

(35, 35, 37, 40,) 43, (56, 58, 58, 60)

median

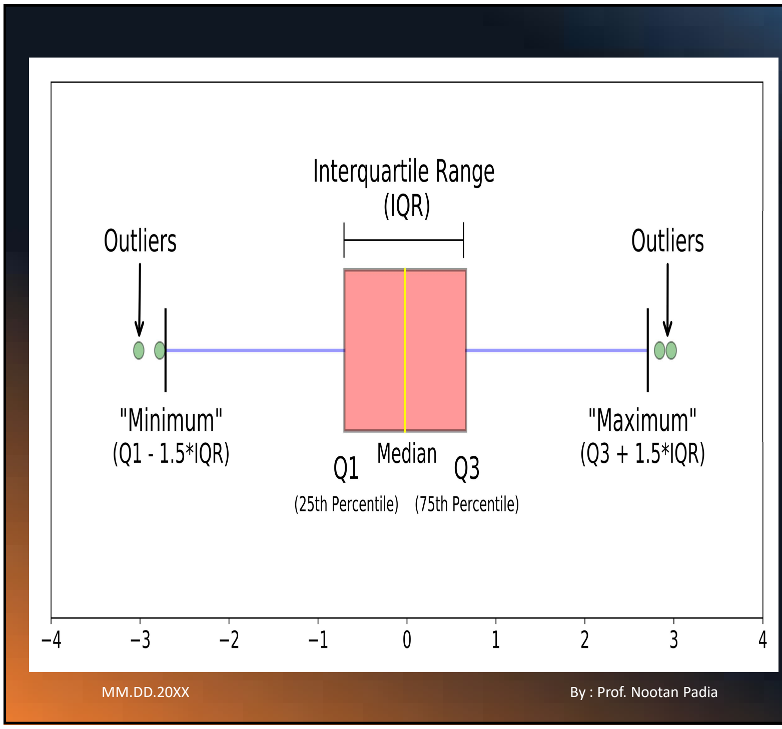
- Minimum = 35, Q1 = 36, median = 43, Q3 = 58, maximum = 60
- 1st half =
  - Difference between minimum and Q1 = 1
  - Difference between Q1 and median = 7
- 2nd half =
  - Difference between median and Q3 = 15
  - Difference between Q3 and maximum = 2



- If we take the example of the attribute 'displacement', we can see that the difference between minimum value and Q1 is 36.2 and the difference between Q1 and median is 44.3.
- On the contrary, the difference between median and Q3 is 113.5 and Q3 and the maximum value is 193.
- In other words, the larger values are more spread out than the smaller ones.
- This helps in understanding why the value of mean is much higher than that of the median for the attribute 'displacement'.
- Similarly, in case of attribute 'cylinders' we can observe that the difference between minimum value and median is 1 whereas the difference between median and the maximum value is 4.
- For the attribute 'origin', the difference between minimum value and median is 0 whereas the difference between median and the maximum value is 2.
- However, we still cannot ascertain whether there is any outlier present in the data.
- For that, we can better adopt some means to visualize the data.
- Box plot is an excellent visualization medium for numeric data.

Plotting and exploring numerical data

- **Box plots**
  - A box plot is an extremely effective mechanism to get a one-shot view and understand the nature of the data.
  - But before we get to review of the box plot for different attributes of Auto MPG data set, let's first try to understand a box plot in general and the interpretation of different aspects in a box plot.
  - As we can see in Figure 2.9, the box plot (also called box and whisker plot) gives a standard visualization of the five-number summary statistics of a data, namely minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum.
  - Below is a detailed interpretation of a box plot.
  - The central rectangle or the box spans from first to third quartile (i.e. Q 1 to Q3), thus giving the inter-quartile range (IQR).



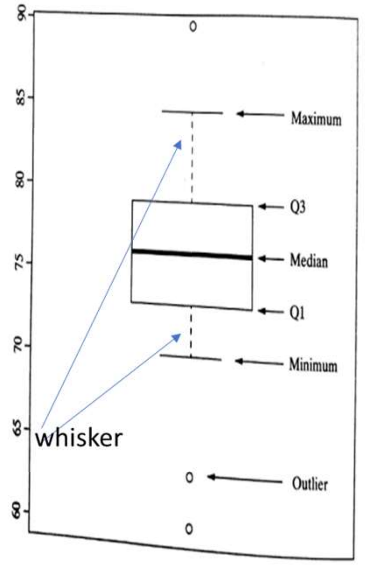


FIG. 2.9
Box plot

- Median is given by the line or band within the box.
- The lower whisker extends up to 1.5 times of the inter-quartile range (or IQR) from the bottom of the box, i.e. the first quartile or Q1.
- Actual length of the lower whisker = Q1 — (1.5 X IQR).
- IQR = (Q3 – Q1)
- E.g. Q1 = 73, median 76, Q3= 79.
- Hence, IQR will be 6
- So, lower whisker can extend maximum till (Q1 — 1.5 x IQR) = 73 — 1.5 x 6 =64.
- Suppose, there are lower range data values such as 70, 63, and 60.
- So, the lower whisker will come at 70 as this is the lowest data value larger than 64.
- The upper whisker extends up to 1.5 as times of the inter-quartile range (or IQR) from the top of the box, i.e. the third quartile or Q3.
- Actual length of the upper whisker = Q3 +1.5 X IQR.
- The data values coming beyond the lower or upper whiskers are the ones which are of unusually low or high values respectively. These are the outliers, which may deserve special consideration.

- There are different variants of box plots. The one covered in last slide is the Tukey box plot.
- Famous mathematician John W. Tukey introduced this type of box plot in 1969.
- Figure presents the respective box plots.
- Let's visualize the box plot for the three attributes - 'cylinders', 'displacement', ' Origin', . We will also review the box plot of another attribute in which the deviation between mean and median is very little and see what the basic difference in the respective box plots is.

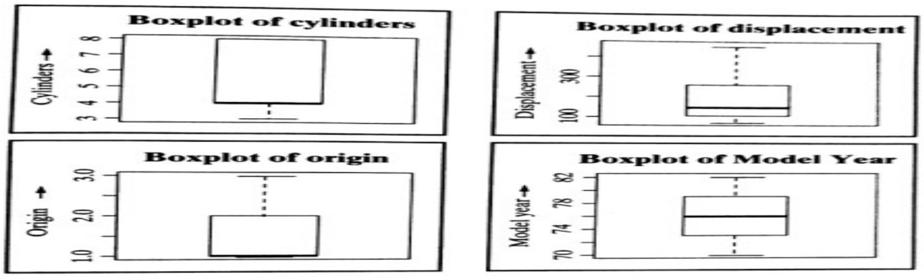


FIG. 2.10
Box plot of Auto MPG attributes

- **Analyzing box plot for 'cylinders'**
  - The boxplot for attribute 'cylinders' looks pretty weird in shape.
  - The upper whisker is missing; the band for median falls at the bottom of the box, even the lower whisker is pretty small compared to the length of the box! Is everything right?
  - The answer is a big YES, and you can figure it out if you delve a little deeper into the actual data values of the attribute.
  - The attribute ‘cylinders’ is discrete in nature having values from 3 to 8.
  - Table 2.2 captures the frequency and cumulative frequency of it.
  - As can be observed in the table, the frequency is extremely high for data value 4.
  - Two other data values where the frequency is quite high are 6 and 8.

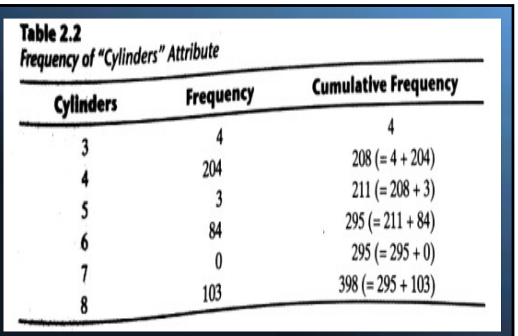
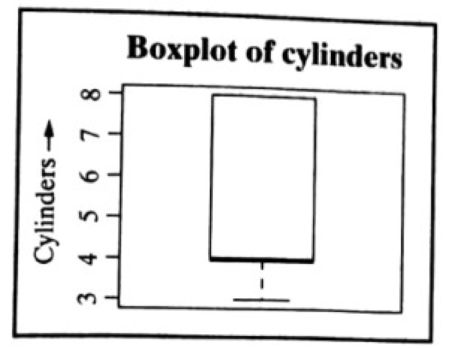
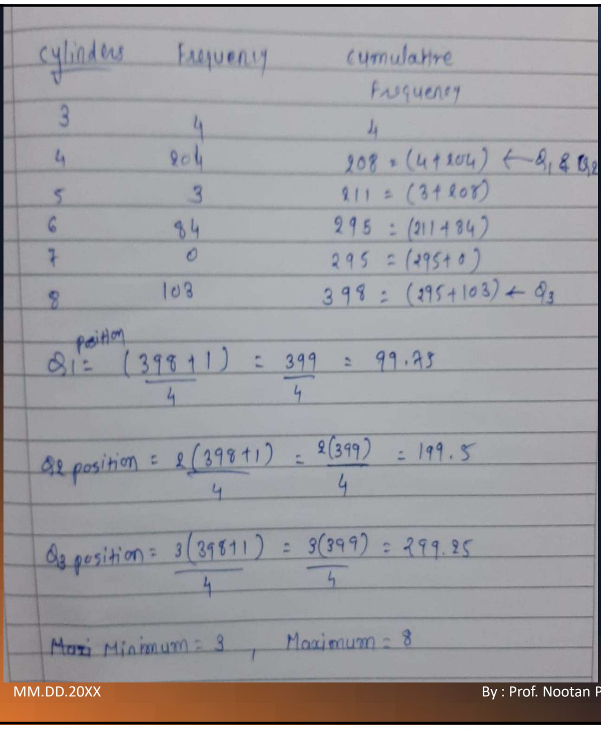


Table 2.2
Frequency of "Cylinders" Attribute

Cylinders	Frequency	Cumulative Frequency
3	4	4
4	204	208 (= 4 + 204)
5	3	211 (= 208 + 3)
6	84	295 (= 211 + 84)
7	0	295 (= 295 + 0)
8	103	398 (= 295 + 103)


Boxplot of cylinders



cylinders	Frequency	cumulative Frequency
3	4	4
4	204	208 = (4+204) ← Q1 & Q2
5	3	211 = (3+208)
6	84	295 = (211+84)
7	0	295 = (295+0)
8	103	398 = (295+103) ← Q3

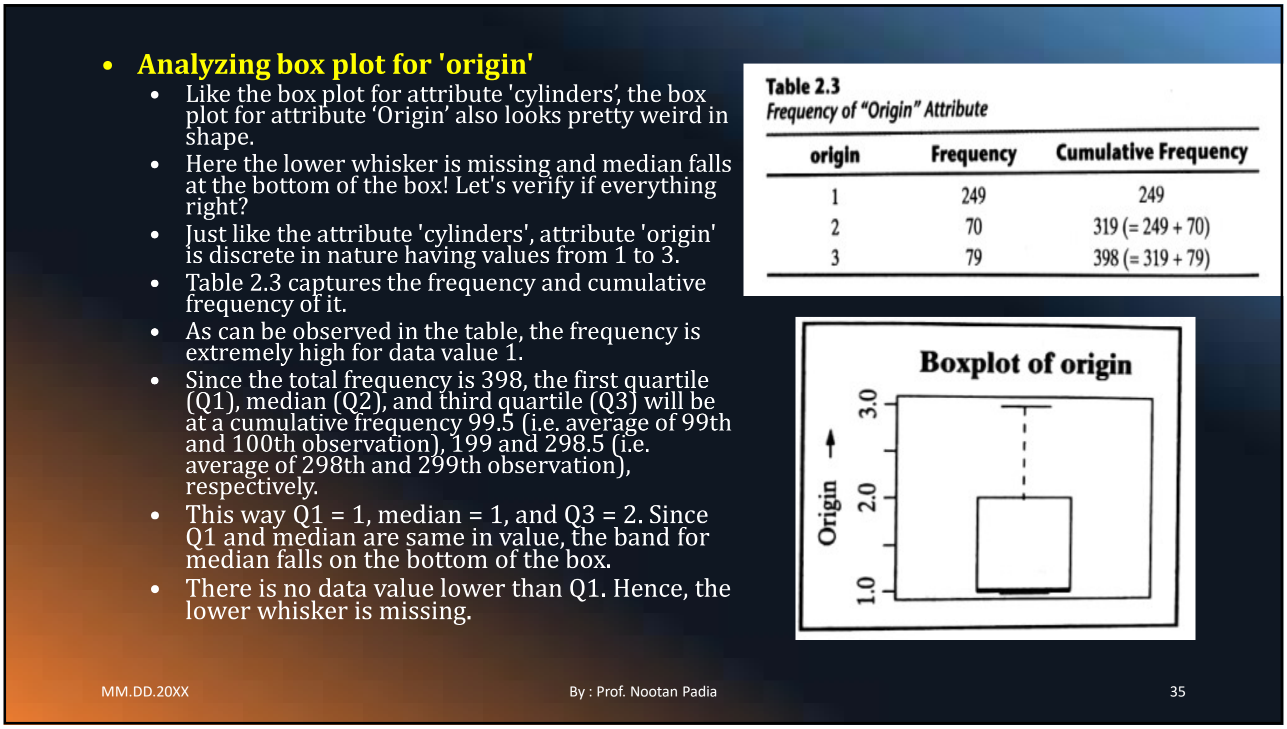
Q_1 position $= \frac{(398+1)}{4} = \frac{399}{4} = 99.75$

Q_2 position $= \frac{2(398+1)}{4} = \frac{2(399)}{4} = 199.5$

Q_3 position $= \frac{3(398+1)}{4} = \frac{3(399)}{4} = 299.25$

~~Maxi~~ Minimum = 3 , Maximum = 8

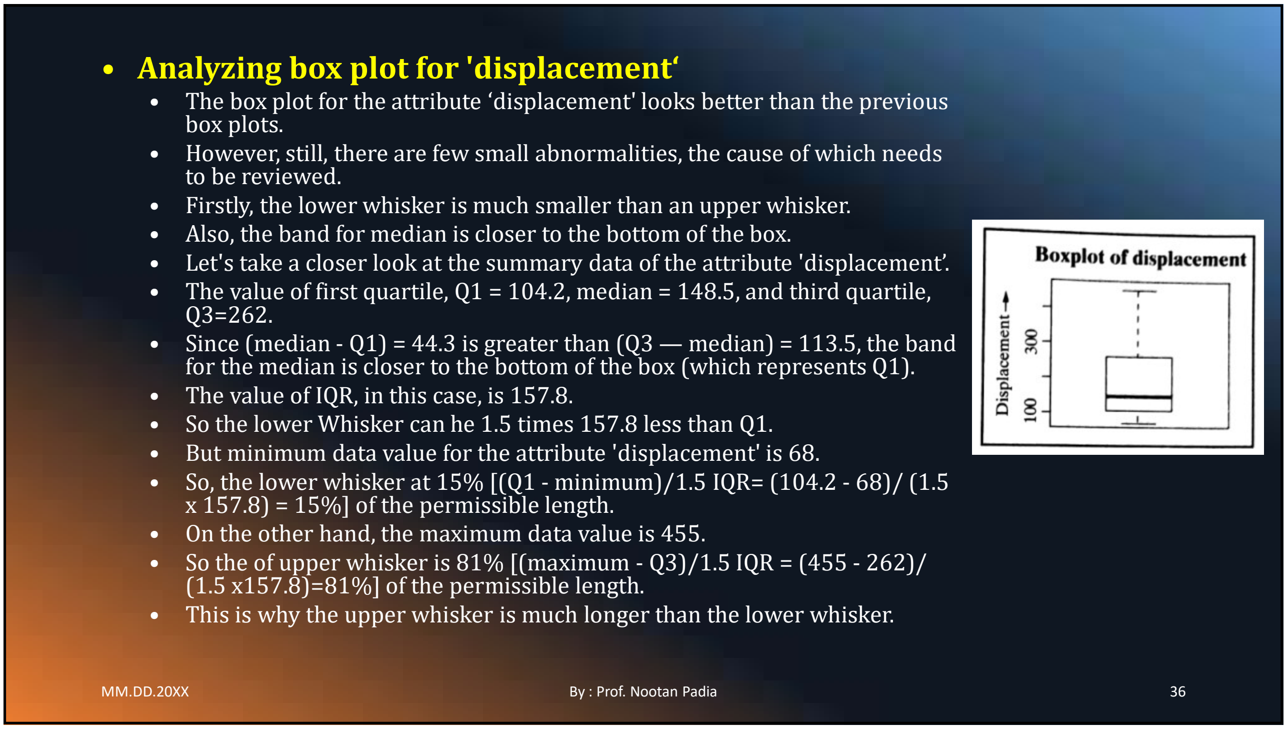
- Since there is no data value beyond 8, there is no upper whisker.
- Also, since both Q1 and median are 4, the band for median falls on the bottom of the box.
- Same way, though the lower whisker could have extended till -2 (Q1- 1.5 x IQR = 4 - 1.5 x 4 = -2), in reality, there is no data value lower than 3.
- Hence, the lower whisker is also short. In any case, a value of cylinders less than 1 is not possible.



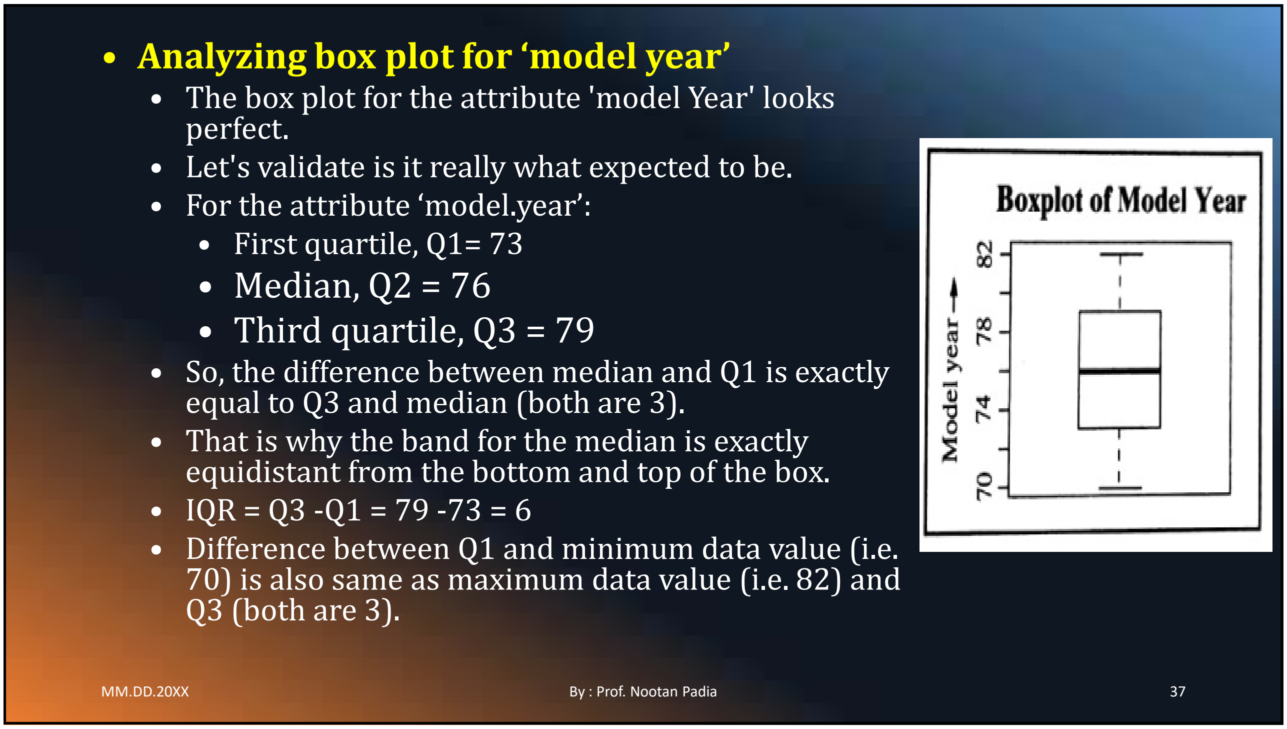
- **Analyzing box plot for 'origin'**
  - Like the box plot for attribute 'cylinders', the box plot for attribute 'Origin' also looks pretty weird in shape.
  - Here the lower whisker is missing and median falls at the bottom of the box! Let's verify if everything right?
  - Just like the attribute 'cylinders', attribute 'origin' is discrete in nature having values from 1 to 3.
  - Table 2.3 captures the frequency and cumulative frequency of it.
  - As can be observed in the table, the frequency is extremely high for data value 1.
  - Since the total frequency is 398, the first quartile (Q1), median (Q2), and third quartile (Q3) will be at a cumulative frequency 99.5 (i.e. average of 99th and 100th observation), 199 and 298.5 (i.e. average of 298th and 299th observation), respectively.
  - This way Q1 = 1, median = 1, and Q3 = 2. Since Q1 and median are same in value, the band for median falls on the bottom of the box.
  - There is no data value lower than Q1. Hence, the lower whisker is missing.

Table 2.3
Frequency of "Origin" Attribute

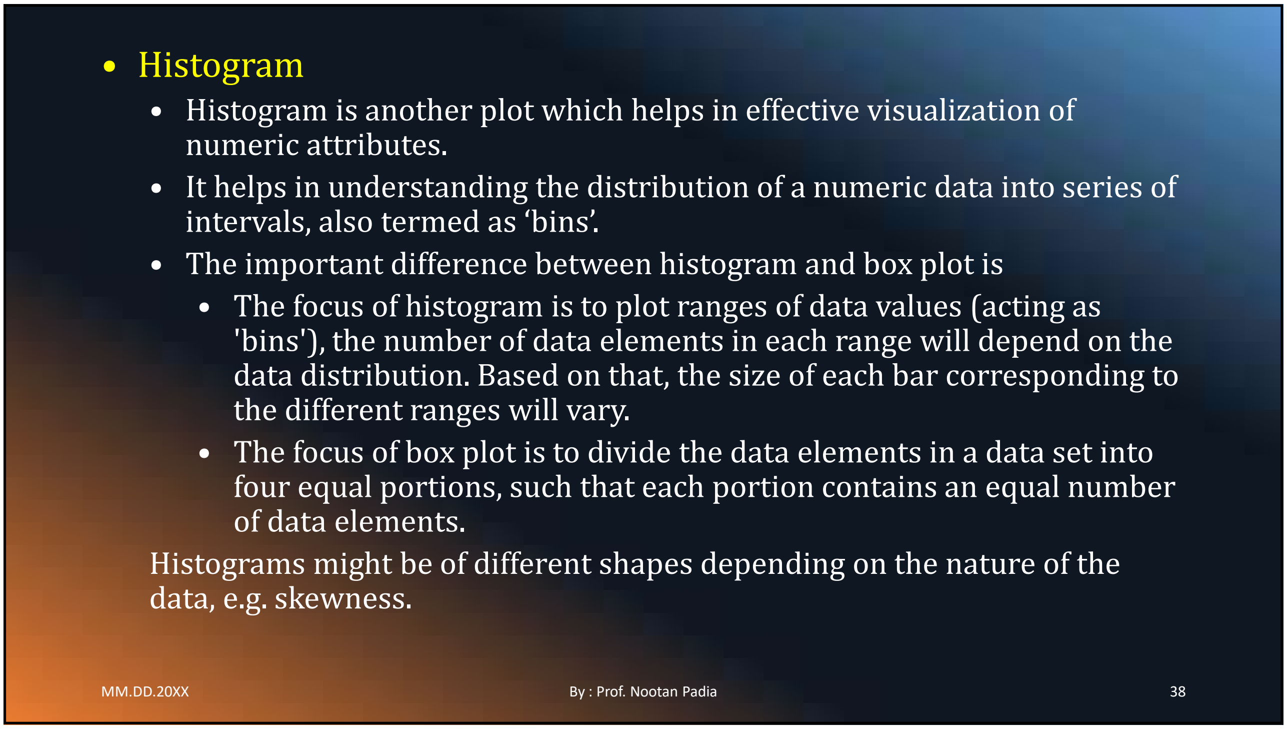
origin	Frequency	Cumulative Frequency
1	249	249
2	70	319 (= 249 + 70)
3	79	398 (= 319 + 79)



- **Analyzing box plot for 'displacement'**
  - The box plot for the attribute 'displacement' looks better than the previous box plots.
  - However, still, there are few small abnormalities, the cause of which needs to be reviewed.
  - Firstly, the lower whisker is much smaller than an upper whisker.
  - Also, the band for median is closer to the bottom of the box.
  - Let's take a closer look at the summary data of the attribute 'displacement'.
  - The value of first quartile, Q1 = 104.2, median = 148.5, and third quartile, Q3=262.
  - Since (median - Q1) = 44.3 is greater than (Q3 — median) = 113.5, the band for the median is closer to the bottom of the box (which represents Q1).
  - The value of IQR, in this case, is 157.8.
  - So the lower Whisker can he 1.5 times 157.8 less than Q1.
  - But minimum data value for the attribute 'displacement' is 68.
  - So, the lower whisker at 15% [(Q1 - minimum)/1.5 IQR= (104.2 - 68)/ (1.5 x 157.8) = 15%] of the permissible length.
  - On the other hand, the maximum data value is 455.
  - So the of upper whisker is 81% [(maximum - Q3)/1.5 IQR = (455 - 262)/ (1.5 x157.8)=81%] of the permissible length.
  - This is why the upper whisker is much longer than the lower whisker.

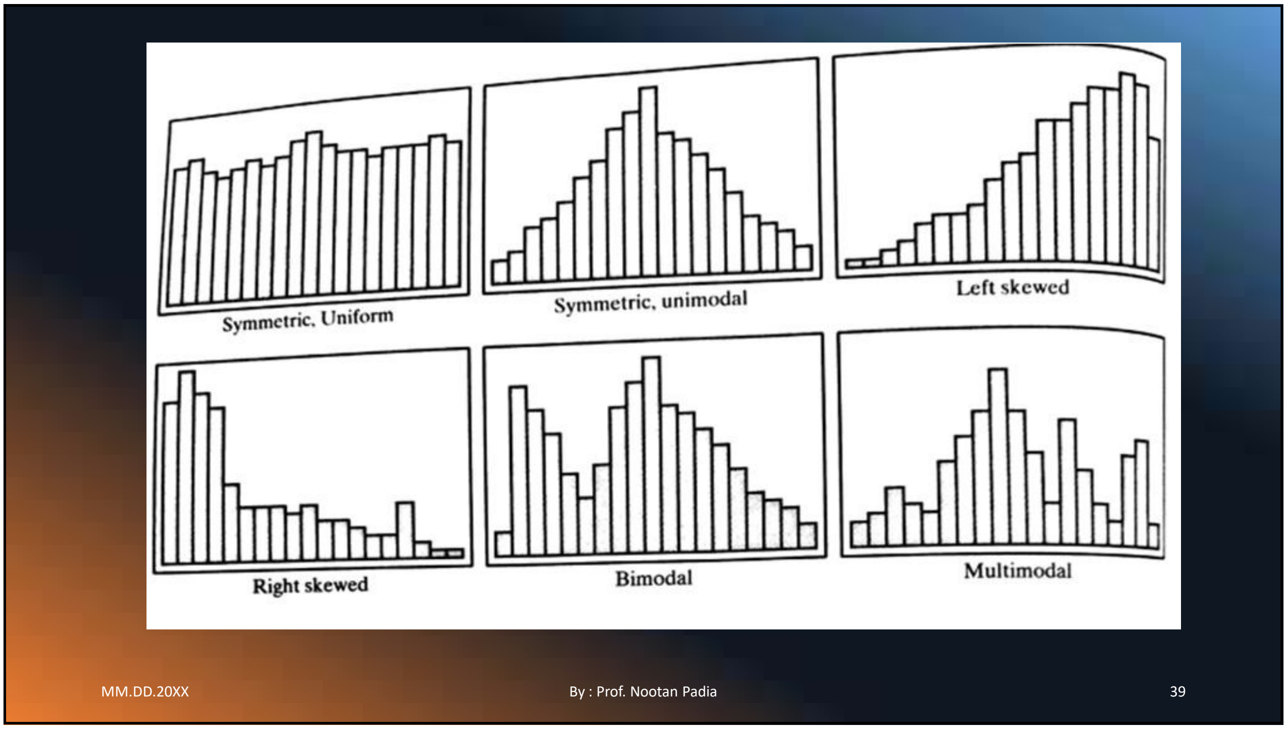


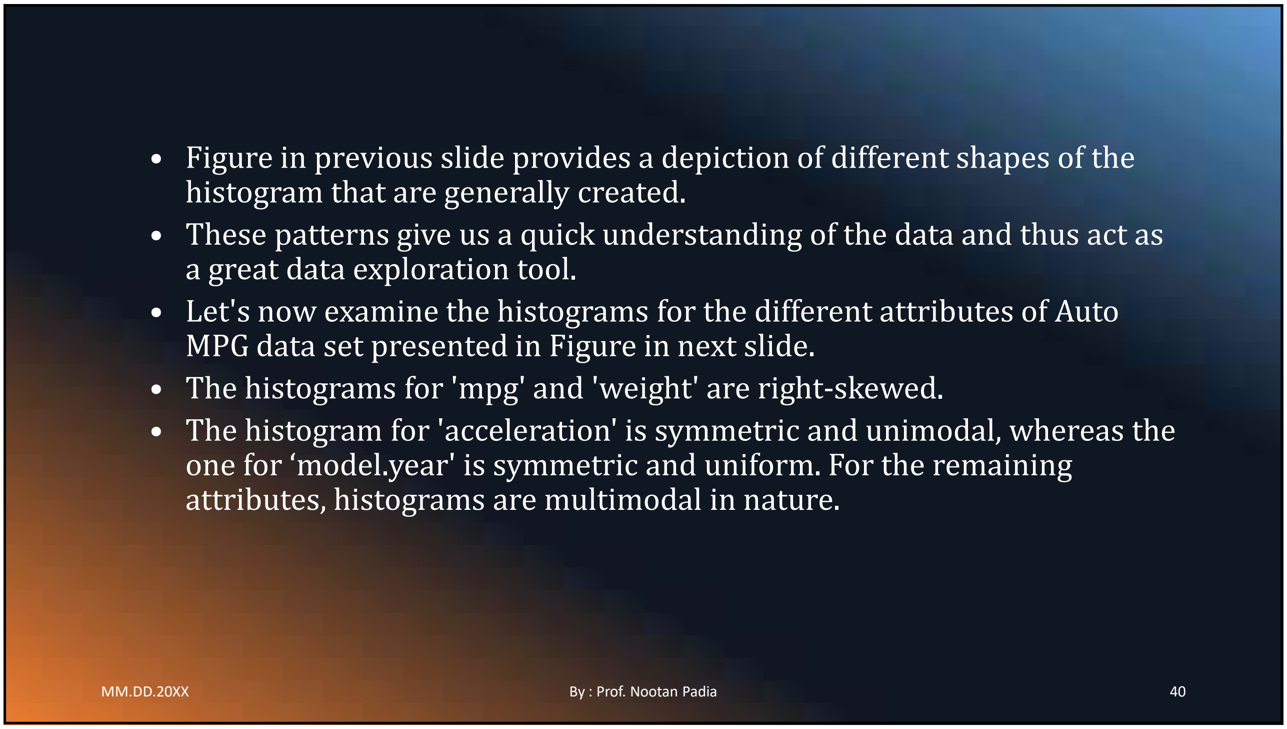
- **Analyzing box plot for 'model year'**
  - The box plot for the attribute 'model Year' looks perfect.
  - Let's validate is it really what expected to be.
  - For the attribute 'model.year':
    - First quartile, Q1= 73
    - Median, Q2 = 76
    - Third quartile, Q3 = 79
  - So, the difference between median and Q1 is exactly equal to Q3 and median (both are 3).
  - That is why the band for the median is exactly equidistant from the bottom and top of the box.
  - IQR = Q3 -Q1 = 79 -73 = 6
  - Difference between Q1 and minimum data value (i.e. 70) is also same as maximum data value (i.e. 82) and Q3 (both are 3).



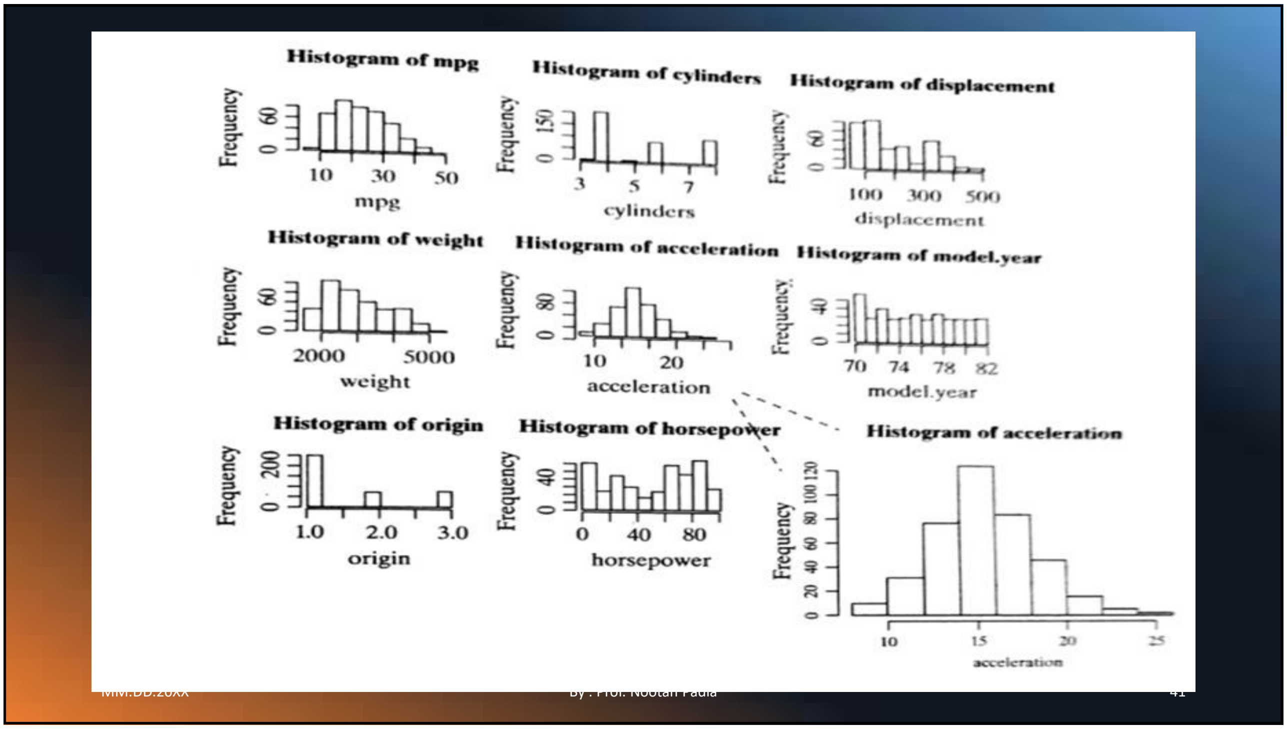
- Histogram
  - Histogram is another plot which helps in effective visualization of numeric attributes.
  - It helps in understanding the distribution of a numeric data into series of intervals, also termed as 'bins'.
  - The important difference between histogram and box plot is
    - The focus of histogram is to plot ranges of data values (acting as 'bins'), the number of data elements in each range will depend on the data distribution. Based on that, the size of each bar corresponding to the different ranges will vary.
    - The focus of box plot is to divide the data elements in a data set into four equal portions, such that each portion contains an equal number of data elements.

  Histograms might be of different shapes depending on the nature of the data, e.g. skewness.





- Figure in previous slide provides a depiction of different shapes of the histogram that are generally created.
- These patterns give us a quick understanding of the data and thus act as a great data exploration tool.
- Let's now examine the histograms for the different attributes of Auto MPG data set presented in Figure in next slide.
- The histograms for 'mpg' and 'weight' are right-skewed.
- The histogram for 'acceleration' is symmetric and unimodal, whereas the one for ‘model.year' is symmetric and uniform. For the remaining attributes, histograms are multimodal in nature.



Exploring categorical data

- Unlike numerical data, there are not many options for exploring categorical data.
- In the Auto MPG data set, attribute ‘car.name’ is categorical in nature.
- We may consider ‘cylinders’ as a categorical variable instead of a numeric variable.
- The first summary which we may be interested in noting is how many unique names are there for the attribute ‘car name’ or how many unique values are there for ‘cylinders’ attribute.
- For attribute car name : Chevrolet chevelle malibu ,Buick skylark 320, Plymouth satellite , Amc rebel sst, Ford torino , Ford galaxie 500 etc
- For attribute 'cylinders' 8 4 6 3 5
- We may also look for a little more details and want to get a table consisting the categories of the attribute and count of the data elements falling into that category.

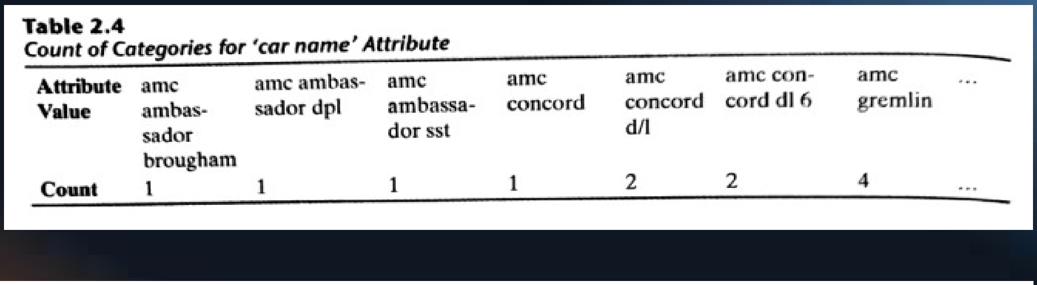


Table 2.4
Count of Categories for ‘car name’ Attribute

Attribute Value	amc ambassador brougham	amc ambassador dpl	amc ambassador sst	amc concord	amc concord d/l	amc concord dl 6	amc gremlin	...
Count	1	1	1	1	2	2	4	...

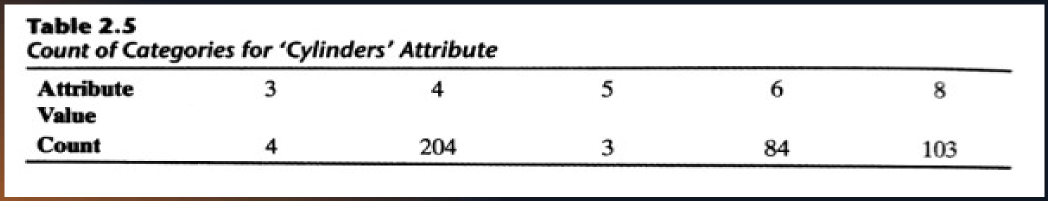


Table 2.5
Count of Categories for ‘Cylinders’ Attribute

Attribute Value	3	4	5	6	8
Count	4	204	3	84	103

- In the same way, we may also be interested to know the proportion (or percentage) of count of data elements belonging to a category.
- Say, e.g., for the attributes ‘cylinders’, the proportion of data elements belonging to the category 4 is 204 ÷ 398 = 0.513, i.e. 51.3%.
- Tables 2.6 and 2.7 contain the summarization of the categorical attributes by proportion of data elements.

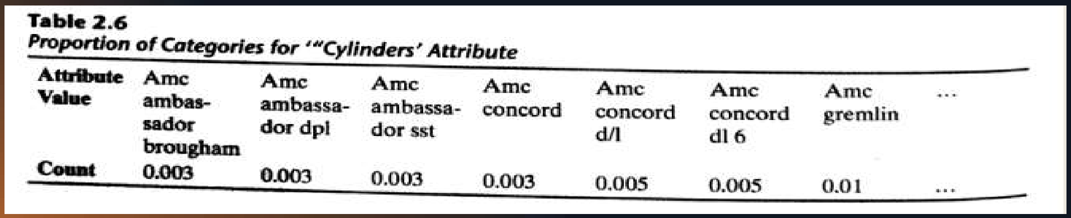


Table 2.6
Proportion of Categories for ‘“Cylinders’ Attribute

Attribute Value	Amc ambassador brougham	Amc ambassador dpl	Amc ambassador sst	Amc concord	Amc concord d/l	Amc concord dl 6	Amc gremlin	...
Count	0.003	0.003	0.003	0.003	0.005	0.005	0.01	...

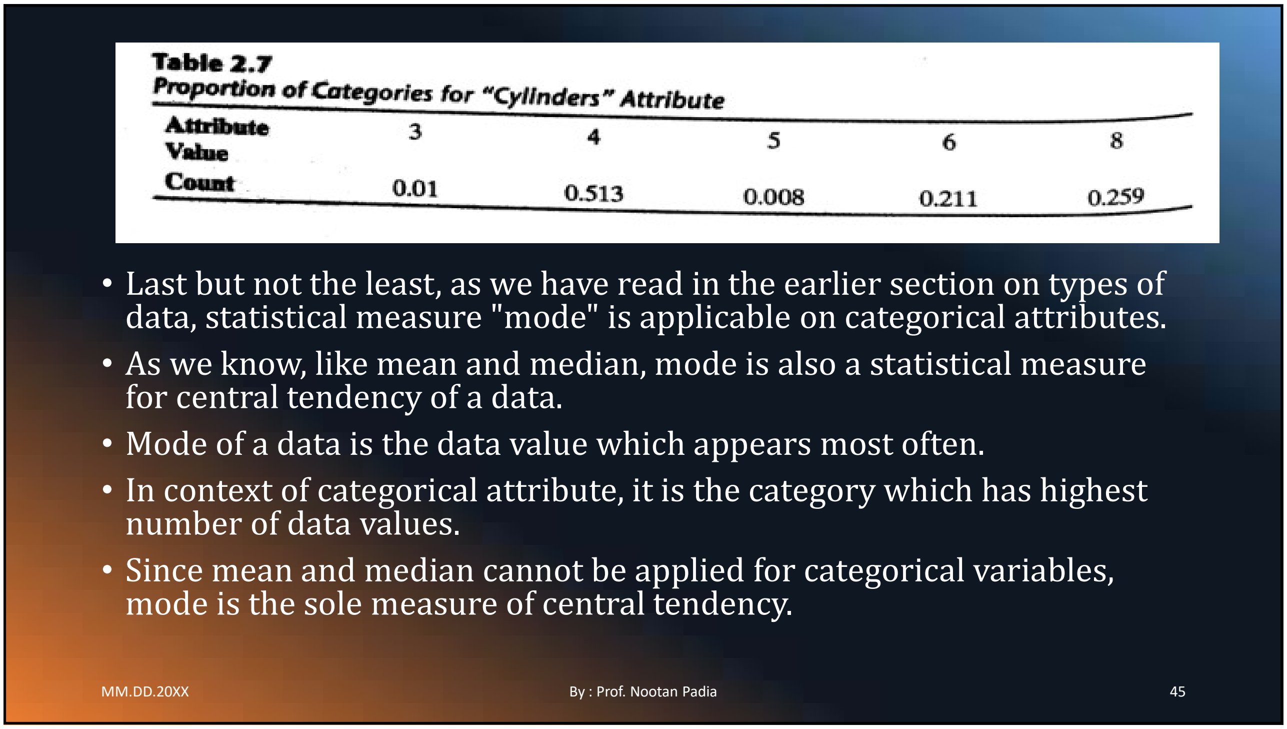
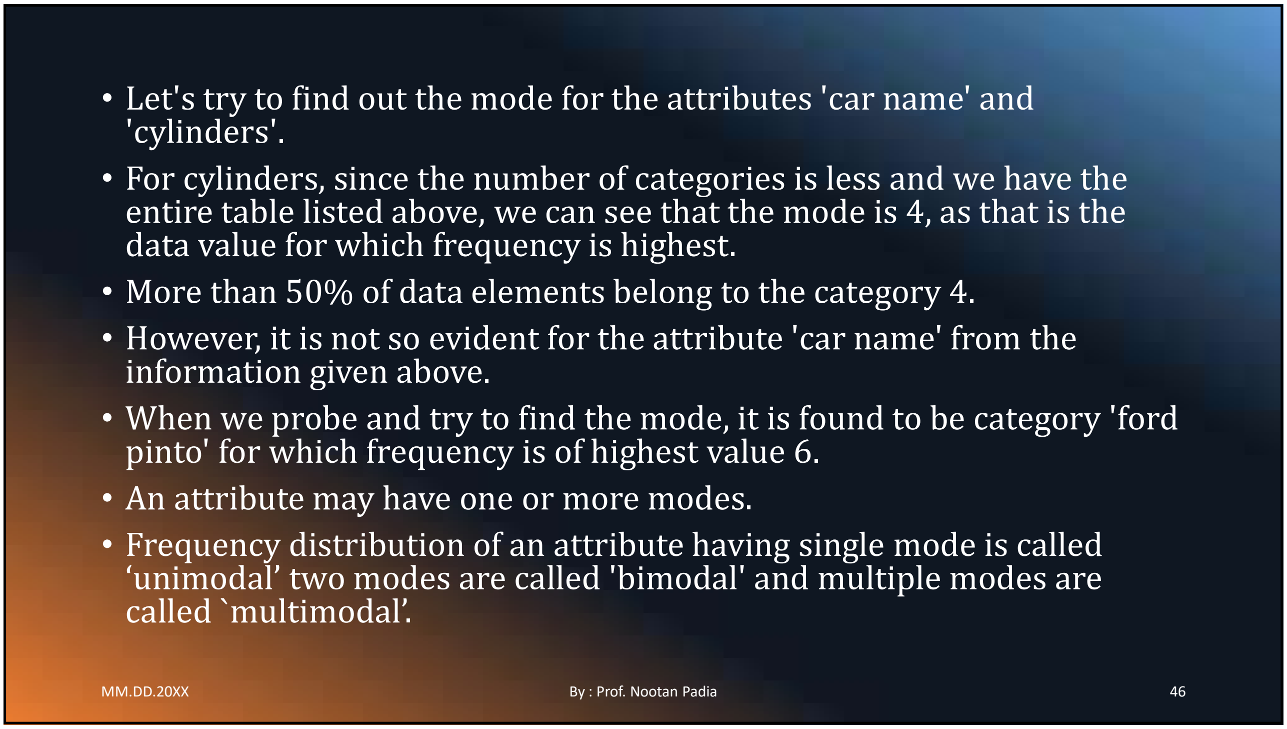


Table 2.7
Proportion of Categories for "Cylinders" Attribute

Attribute Value	3	4	5	6	8
Count	0.01	0.513	0.008	0.211	0.259

- Last but not the least, as we have read in the earlier section on types of data, statistical measure "mode" is applicable on categorical attributes.
- As we know, like mean and median, mode is also a statistical measure for central tendency of a data.
- Mode of a data is the data value which appears most often.
- In context of categorical attribute, it is the category which has highest number of data values.
- Since mean and median cannot be applied for categorical variables, mode is the sole measure of central tendency.



- Let's try to find out the mode for the attributes 'car name' and 'cylinders'.
- For cylinders, since the number of categories is less and we have the entire table listed above, we can see that the mode is 4, as that is the data value for which frequency is highest.
- More than 50% of data elements belong to the category 4.
- However, it is not so evident for the attribute 'car name' from the information given above.
- When we probe and try to find the mode, it is found to be category 'ford pinto' for which frequency is of highest value 6.
- An attribute may have one or more modes.
- Frequency distribution of an attribute having single mode is called 'unimodal' two modes are called 'bimodal' and multiple modes are called `multimodal'.

Exploring relationship between variables

- One more important angle of data exploration is to explore relationship between attributes.
- There are multiple plots to enable us explore the relationship between variables.
- The basic and most commonly used plot is scatter plot.
- **Scatter plot**
  - A scatter plot helps in visualizing bivariate relationships, i.e. relationship between two variables.
  - It is a two-dimensional plot in which points or dots are drawn on coordinates provided by values of the attributes.
  - For example, in a data set there are two attributes — attr_1 and attr_2.
  - We want to understand the relationship between two attributes, i.e. with a change in value of one attribute, say attr_1, how does the value of the other attribute, say attr_2, changes.

- We can draw a scatter plot, with attr_1 mapped to x-axis and attr_2 mapped in y-axis.
- So, every point in the plot will have value of attr_1 in the x-coordinate and value of attr_2 in the y-coordinate.
- As in a two-dimensional plot attr_1 is said to be the independent variable and attr_2 as the dependent variable.
- Let's take a real example in this context. In the data set Auto MPG, there is expected to be some relation between the attributes ‘displacement' and ‘mpg’, try to verify our intuition using the scatter plot of 'displacement' and 'mpg’.
- Let's map displacement' as the x-coordinate and 'mpg' as the y-coordinate.
- The scatter plot comes as in Figure:

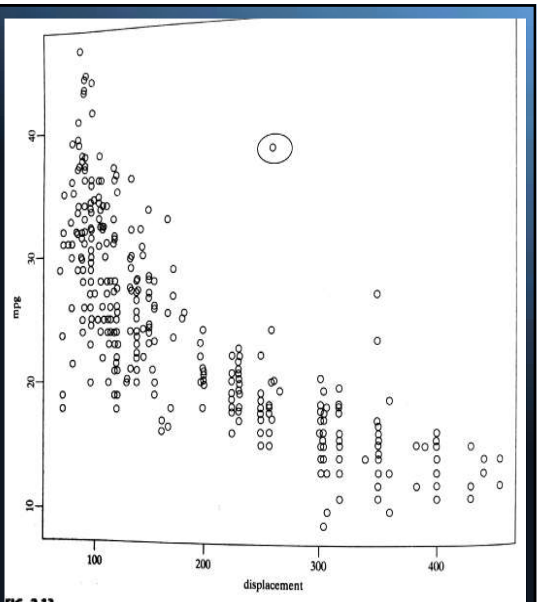


FIG. 2.13
Scatter plot of ‘displacement’ and ‘mpg’

- As is evident in the scatter plot, there is a definite relation between the two variables.
- The value of 'mpg' seems to steadily decrease with the increase in the value of ‘displacement’.
- It may come in our mind that what is the extent of relationship?
- Well, it can be reviewed by calculating the correlation between the variables.
- One more interesting fact to notice is that there are certain data values which stand-out of the others.
- For example, there is one data element which has a mpg of 37 for a displacement of 250.
- This record is completely different from other data elements having similar displacement value but mpg value in the range of 15 to 25.
- This gives an indication that of presence of outlier data values.
- In Figure in next slide, the pair wise relationship among the features — 'mpg', 'displacement', `horsepower', 'weight', and 'acceleration' have been captured.
- As you can see, in most of the cases, there is a significant relationship between the attribute pairs.
- However, in some cases, e.g. between attributes 'weight' and 'acceleration', the relationship doesn’t seem to be very strong.

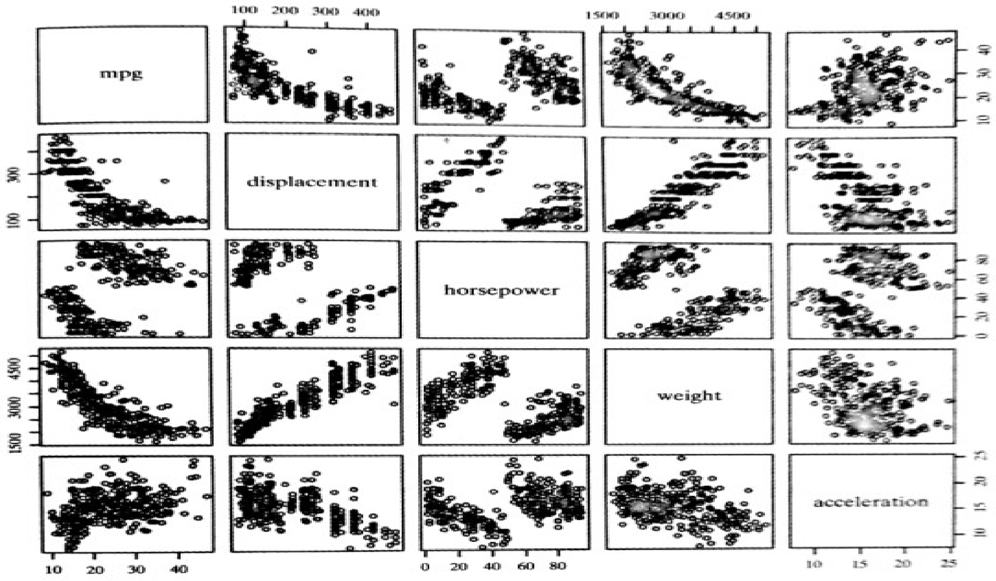
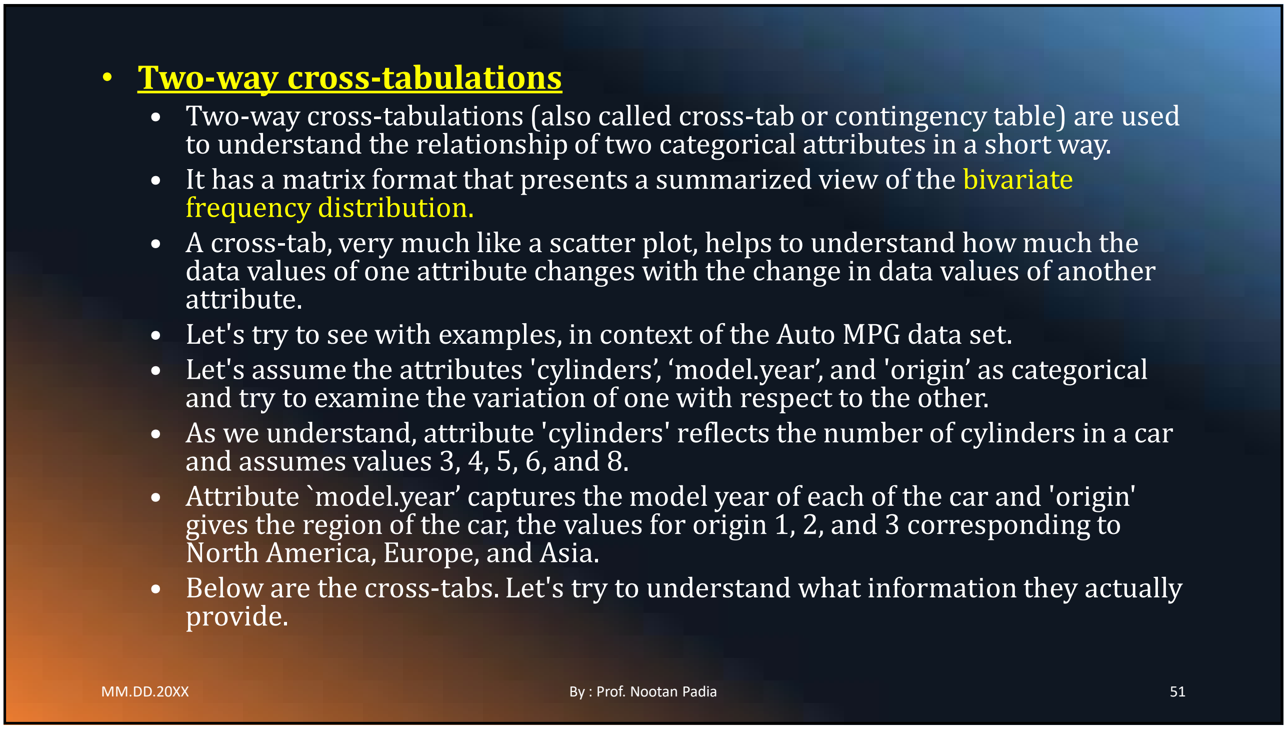
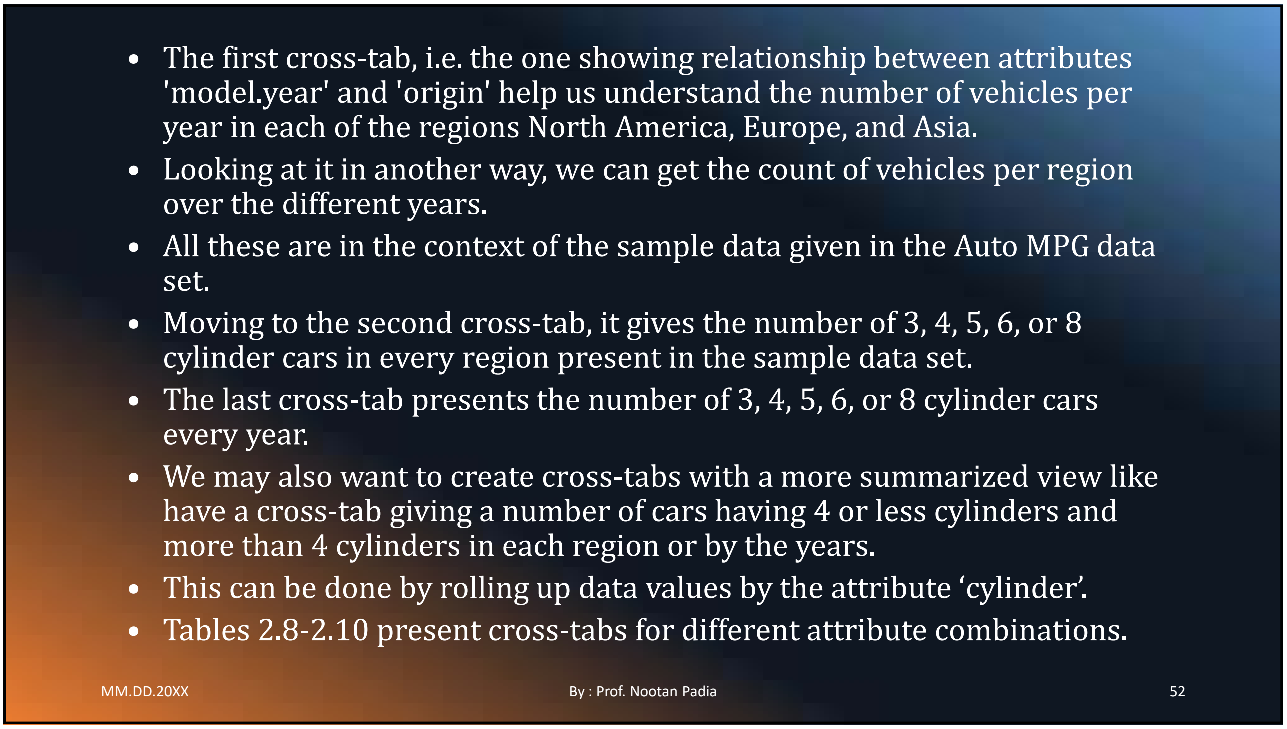


FIG. 2.14
Pair wise scatter plot between different attributes of Auto MPG



- **Two-way cross-tabulations**
  - Two-way cross-tabulations (also called cross-tab or contingency table) are used to understand the relationship of two categorical attributes in a short way.
  - It has a matrix format that presents a summarized view of the bivariate frequency distribution.
  - A cross-tab, very much like a scatter plot, helps to understand how much the data values of one attribute changes with the change in data values of another attribute.
  - Let's try to see with examples, in context of the Auto MPG data set.
  - Let's assume the attributes 'cylinders', ‘model.year', and 'origin' as categorical and try to examine the variation of one with respect to the other.
  - As we understand, attribute 'cylinders' reflects the number of cylinders in a car and assumes values 3, 4, 5, 6, and 8.
  - Attribute `model.year' captures the model year of each of the car and 'origin' gives the region of the car, the values for origin 1, 2, and 3 corresponding to North America, Europe, and Asia.
  - Below are the cross-tabs. Let's try to understand what information they actually provide.



- The first cross-tab, i.e. the one showing relationship between attributes 'model.year' and 'origin' help us understand the number of vehicles per year in each of the regions North America, Europe, and Asia.
- Looking at it in another way, we can get the count of vehicles per region over the different years.
- All these are in the context of the sample data given in the Auto MPG data set.
- Moving to the second cross-tab, it gives the number of 3, 4, 5, 6, or 8 cylinder cars in every region present in the sample data set.
- The last cross-tab presents the number of 3, 4, 5, 6, or 8 cylinder cars every year.
- We may also want to create cross-tabs with a more summarized view like have a cross-tab giving a number of cars having 4 or less cylinders and more than 4 cylinders in each region or by the years.
- This can be done by rolling up data values by the attribute ‘cylinder'.
- Tables 2.8-2.10 present cross-tabs for different attribute combinations.

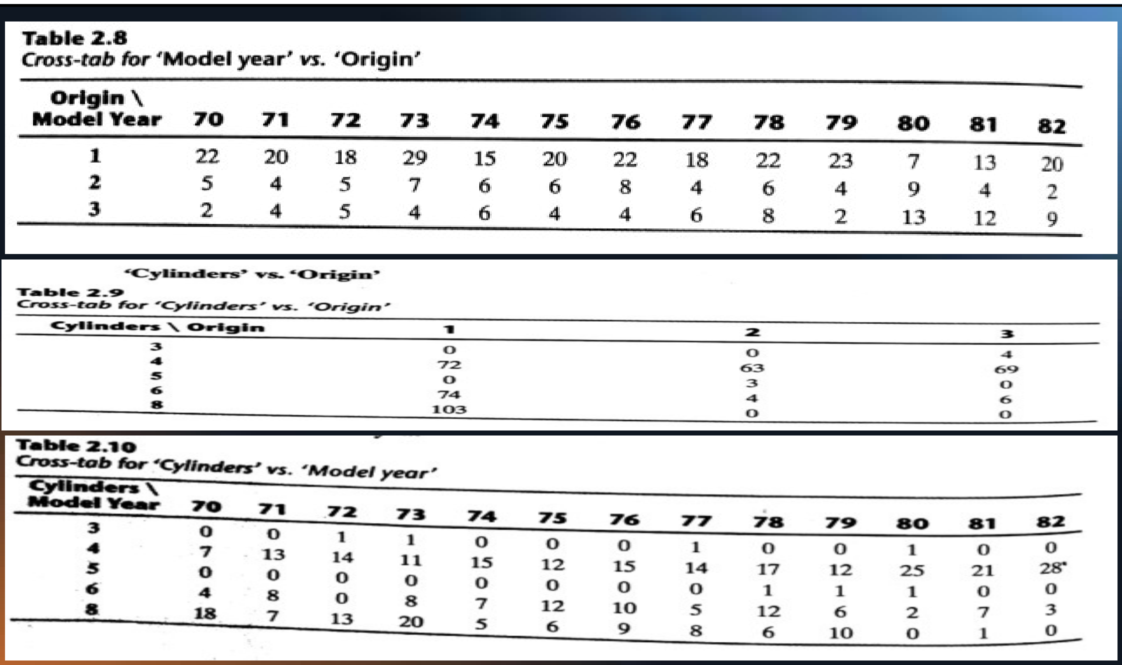


Table 2.8
Cross-tab for 'Model year' vs. 'Origin'

Origin \ Model Year	70	71	72	73	74	75	76	77	78	79	80	81	82
1	22	20	18	29	15	20	22	18	22	23	7	13	20
2	5	4	5	7	6	6	8	4	6	4	9	4	2
3	2	4	5	4	6	4	4	6	8	2	13	12	9

'Cylinders' vs. 'Origin'

Table 2.9
Cross-tab for 'Cylinders' vs. 'Origin'

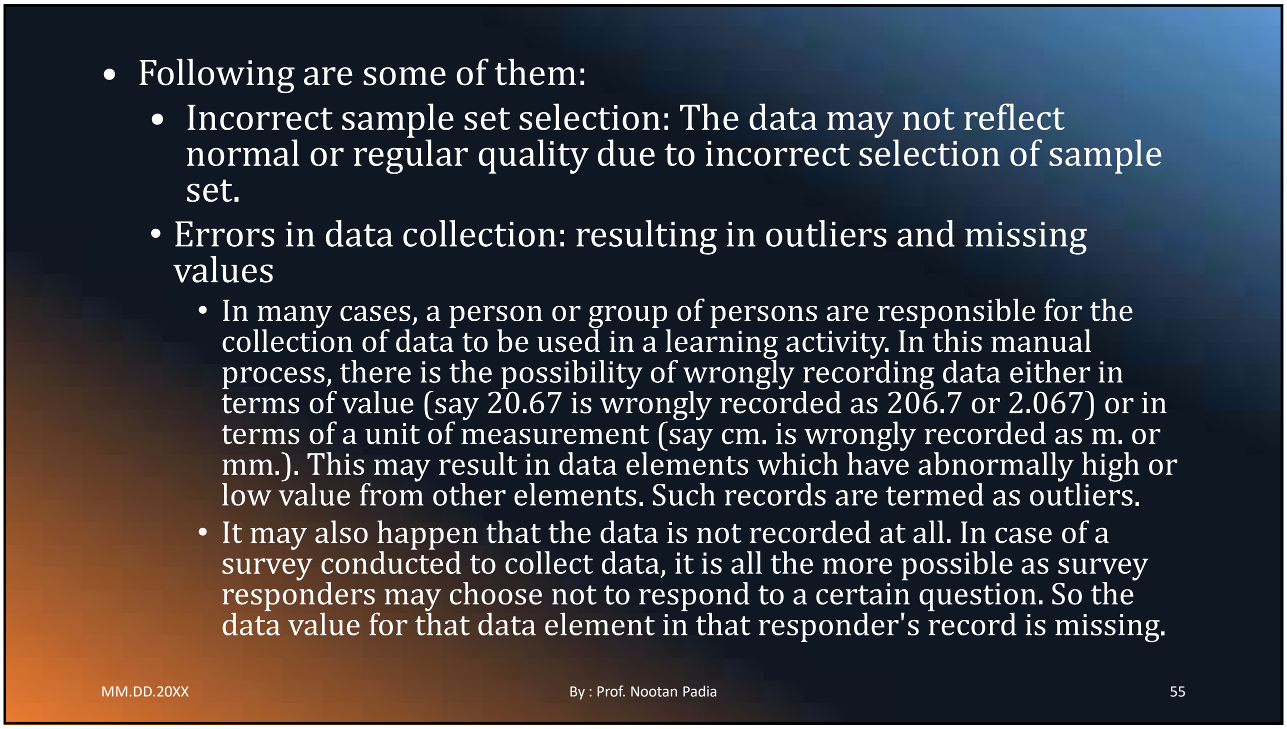
Cylinders \ Origin	1	2	3
3	0	0	4
4	72	63	69
5	0	3	0
6	74	4	6
8	103	0	0

Table 2.10
Cross-tab for 'Cylinders' vs. 'Model year'

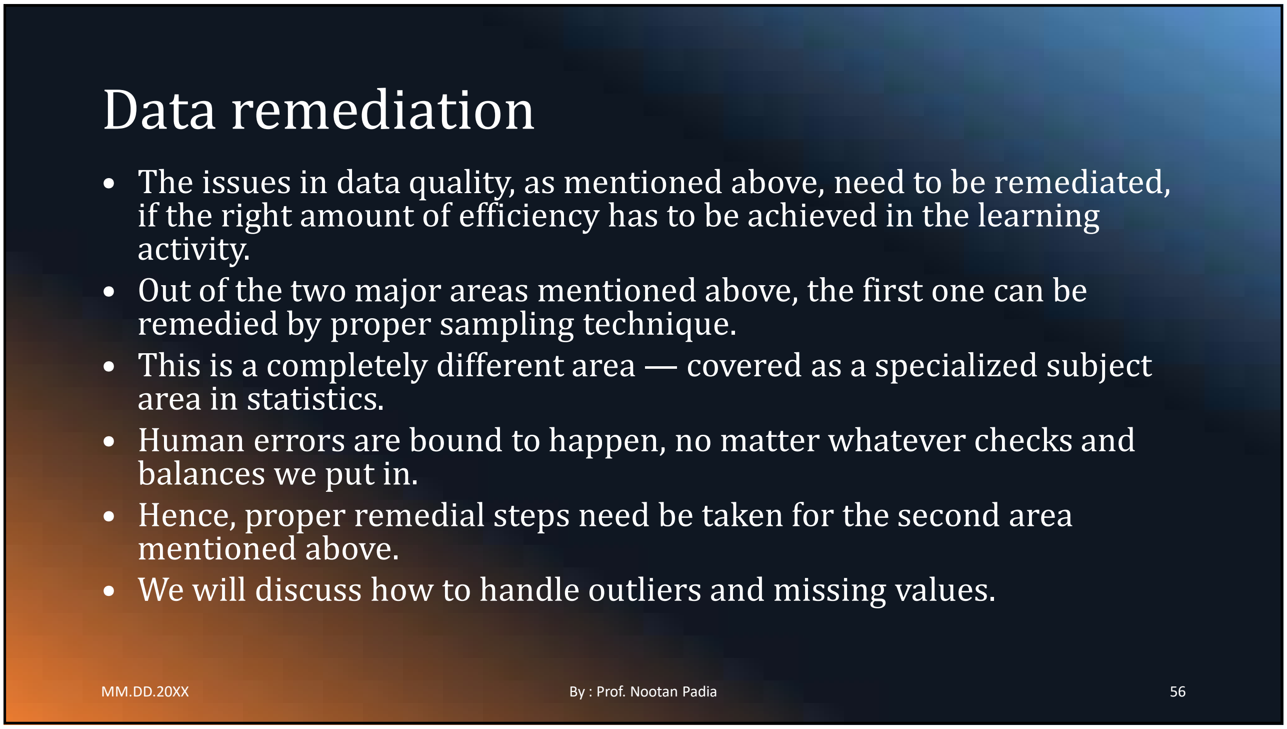
Cylinders \ Model Year	70	71	72	73	74	75	76	77	78	79	80	81	82
3	0	0	1	1	0	0	0	1	0	0	1	0	0
4	7	13	14	11	15	12	15	14	17	12	25	21	28*
5	0	0	0	0	0	0	0	0	1	1	1	0	0
6	4	8	0	8	7	12	10	5	12	6	2	7	3
8	18	7	13	20	5	6	9	8	6	10	0	1	0

Data quality

- Success of machine learning depends largely on the quality of data.
- A data which has the right quality helps to achieve better prediction accuracy, in case of supervised learning.
- However, it is not realistic to expect that the data will be flawless.
- We have already come across at least two types of problems:
  - Certain data elements without a value or data with a missing value.
  - Data elements having value surprisingly different from the other elements, which we term as outliers.
- There are multiple factors which lead to these data quality issues.

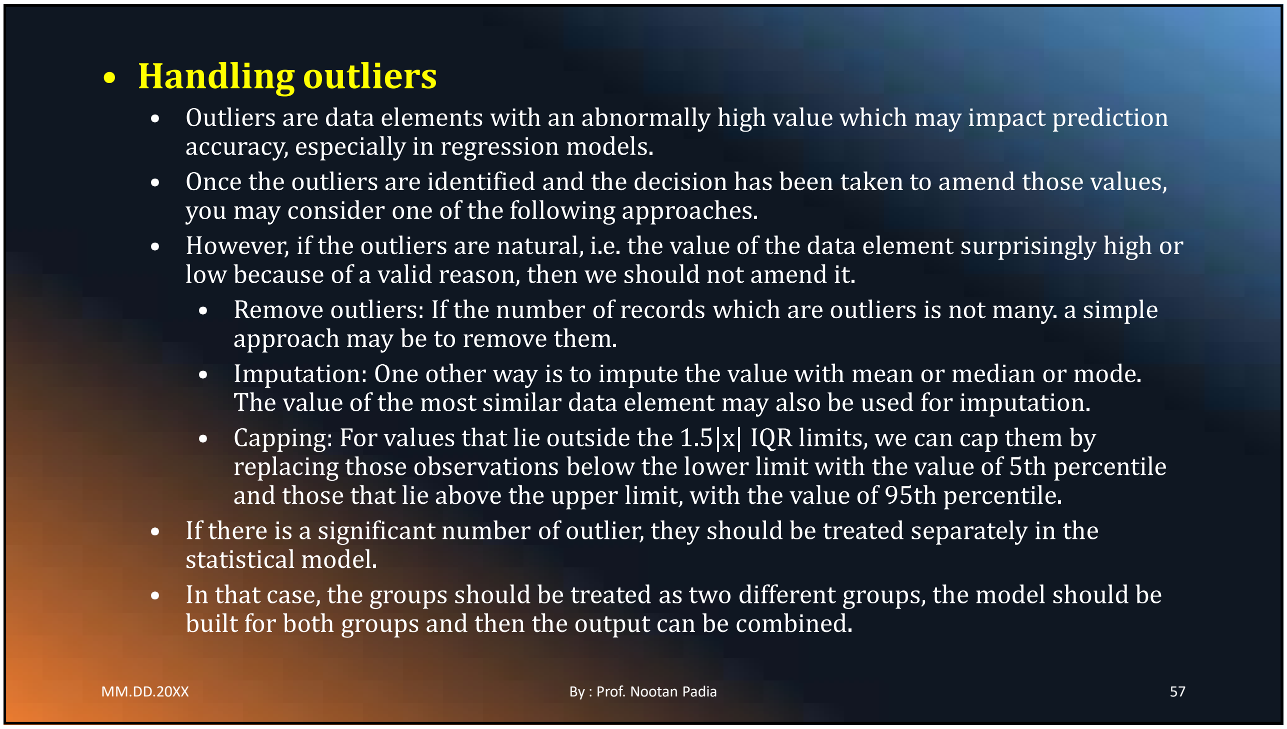


- Following are some of them:
  - Incorrect sample set selection: The data may not reflect normal or regular quality due to incorrect selection of sample set.
  - Errors in data collection: resulting in outliers and missing values
    - In many cases, a person or group of persons are responsible for the collection of data to be used in a learning activity. In this manual process, there is the possibility of wrongly recording data either in terms of value (say 20.67 is wrongly recorded as 206.7 or 2.067) or in terms of a unit of measurement (say cm. is wrongly recorded as m. or mm.). This may result in data elements which have abnormally high or low value from other elements. Such records are termed as outliers.
    - It may also happen that the data is not recorded at all. In case of a survey conducted to collect data, it is all the more possible as survey responders may choose not to respond to a certain question. So the data value for that data element in that responder's record is missing.

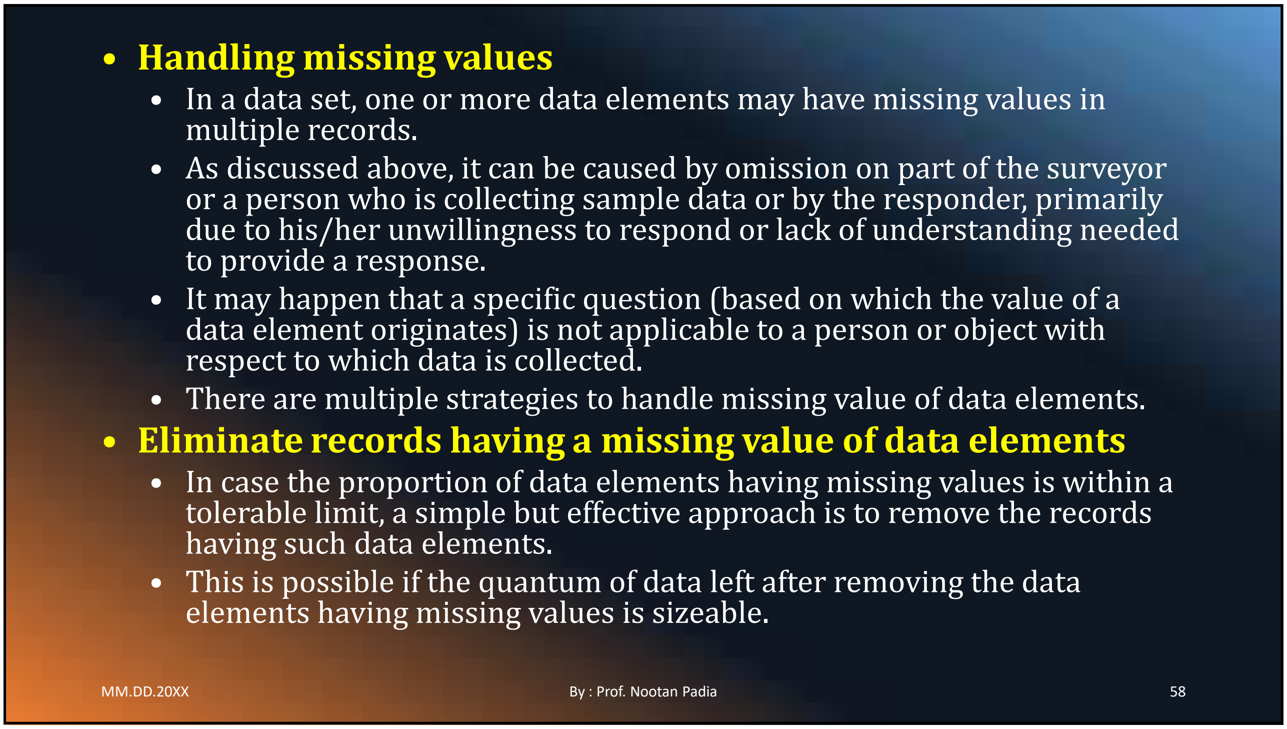


Data remediation

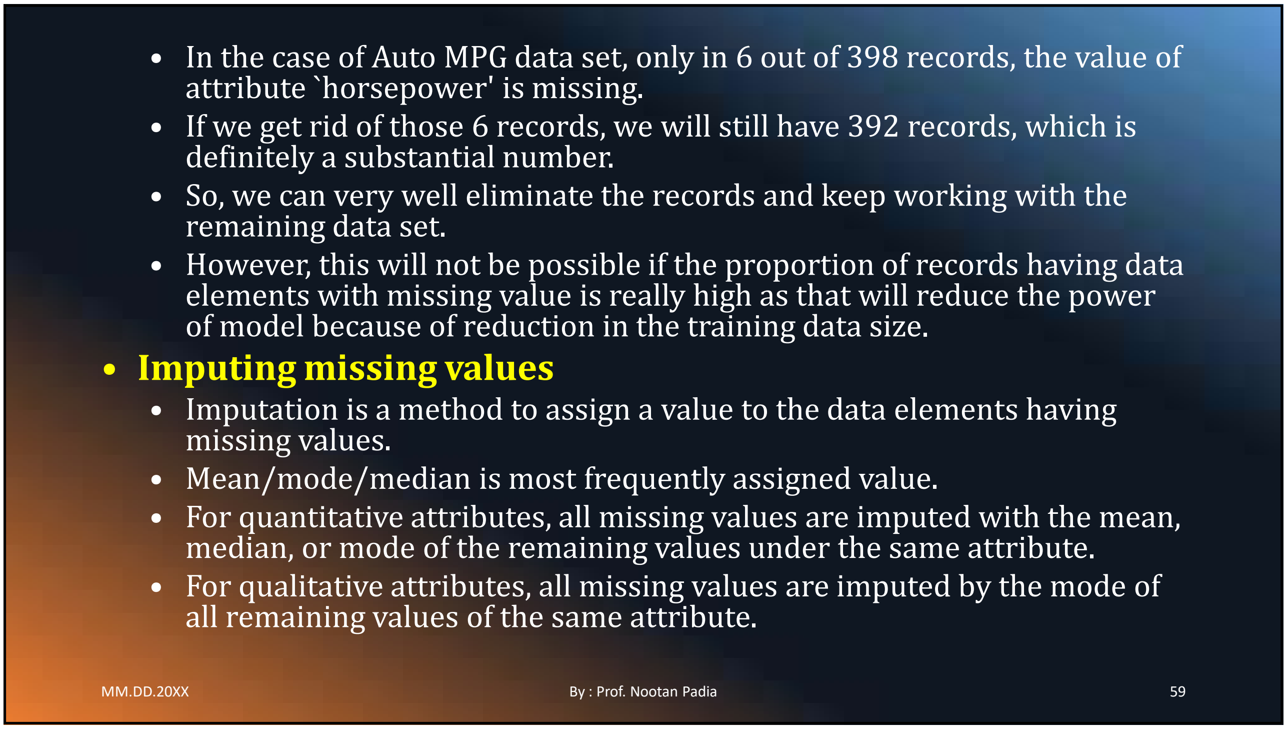
- The issues in data quality, as mentioned above, need to be remediated, if the right amount of efficiency has to be achieved in the learning activity.
- Out of the two major areas mentioned above, the first one can be remedied by proper sampling technique.
- This is a completely different area — covered as a specialized subject area in statistics.
- Human errors are bound to happen, no matter whatever checks and balances we put in.
- Hence, proper remedial steps need be taken for the second area mentioned above.
- We will discuss how to handle outliers and missing values.



- **Handling outliers**
  - Outliers are data elements with an abnormally high value which may impact prediction accuracy, especially in regression models.
  - Once the outliers are identified and the decision has been taken to amend those values, you may consider one of the following approaches.
  - However, if the outliers are natural, i.e. the value of the data element surprisingly high or low because of a valid reason, then we should not amend it.
    - Remove outliers: If the number of records which are outliers is not many. a simple approach may be to remove them.
    - Imputation: One other way is to impute the value with mean or median or mode. The value of the most similar data element may also be used for imputation.
    - Capping: For values that lie outside the 1.5|x| IQR limits, we can cap them by replacing those observations below the lower limit with the value of 5th percentile and those that lie above the upper limit, with the value of 95th percentile.
  - If there is a significant number of outlier, they should be treated separately in the statistical model.
  - In that case, the groups should be treated as two different groups, the model should be built for both groups and then the output can be combined.

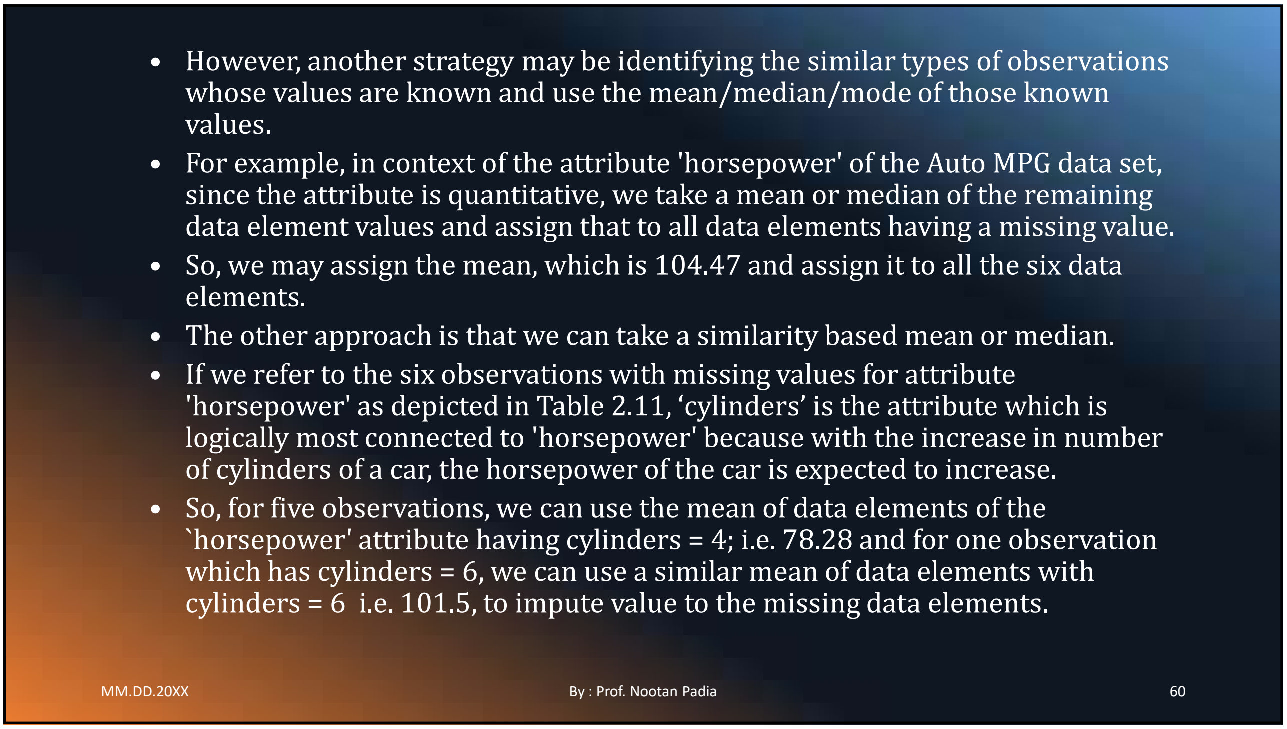


- **Handling missing values**
  - In a data set, one or more data elements may have missing values in multiple records.
  - As discussed above, it can be caused by omission on part of the surveyor or a person who is collecting sample data or by the responder, primarily due to his/her unwillingness to respond or lack of understanding needed to provide a response.
  - It may happen that a specific question (based on which the value of a data element originates) is not applicable to a person or object with respect to which data is collected.
  - There are multiple strategies to handle missing value of data elements.
- **Eliminate records having a missing value of data elements**
  - In case the proportion of data elements having missing values is within a tolerable limit, a simple but effective approach is to remove the records having such data elements.
  - This is possible if the quantum of data left after removing the data elements having missing values is sizeable.



- In the case of Auto MPG data set, only in 6 out of 398 records, the value of attribute `horsepower' is missing.
- If we get rid of those 6 records, we will still have 392 records, which is definitely a substantial number.
- So, we can very well eliminate the records and keep working with the remaining data set.
- However, this will not be possible if the proportion of records having data elements with missing value is really high as that will reduce the power of model because of reduction in the training data size.

- **Imputing missing values**
  - Imputation is a method to assign a value to the data elements having missing values.
  - Mean/mode/median is most frequently assigned value.
  - For quantitative attributes, all missing values are imputed with the mean, median, or mode of the remaining values under the same attribute.
  - For qualitative attributes, all missing values are imputed by the mode of all remaining values of the same attribute.



- However, another strategy may be identifying the similar types of observations whose values are known and use the mean/median/mode of those known values.
- For example, in context of the attribute 'horsepower' of the Auto MPG data set, since the attribute is quantitative, we take a mean or median of the remaining data element values and assign that to all data elements having a missing value.
- So, we may assign the mean, which is 104.47 and assign it to all the six data elements.
- The other approach is that we can take a similarity based mean or median.
- If we refer to the six observations with missing values for attribute 'horsepower' as depicted in Table 2.11, 'cylinders' is the attribute which is logically most connected to 'horsepower' because with the increase in number of cylinders of a car, the horsepower of the car is expected to increase.
- So, for five observations, we can use the mean of data elements of the `horsepower' attribute having cylinders = 4; i.e. 78.28 and for one observation which has cylinders = 6, we can use a similar mean of data elements with cylinders = 6 i.e. 101.5, to impute value to the missing data elements.

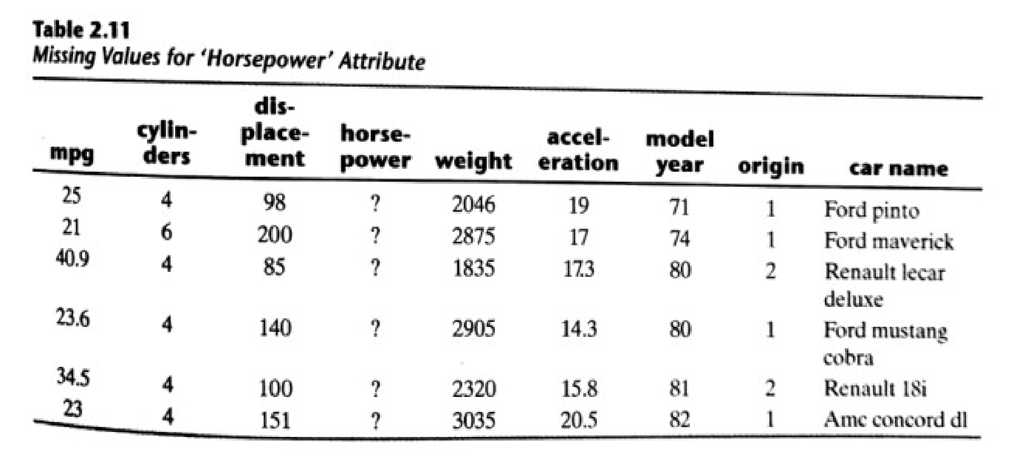
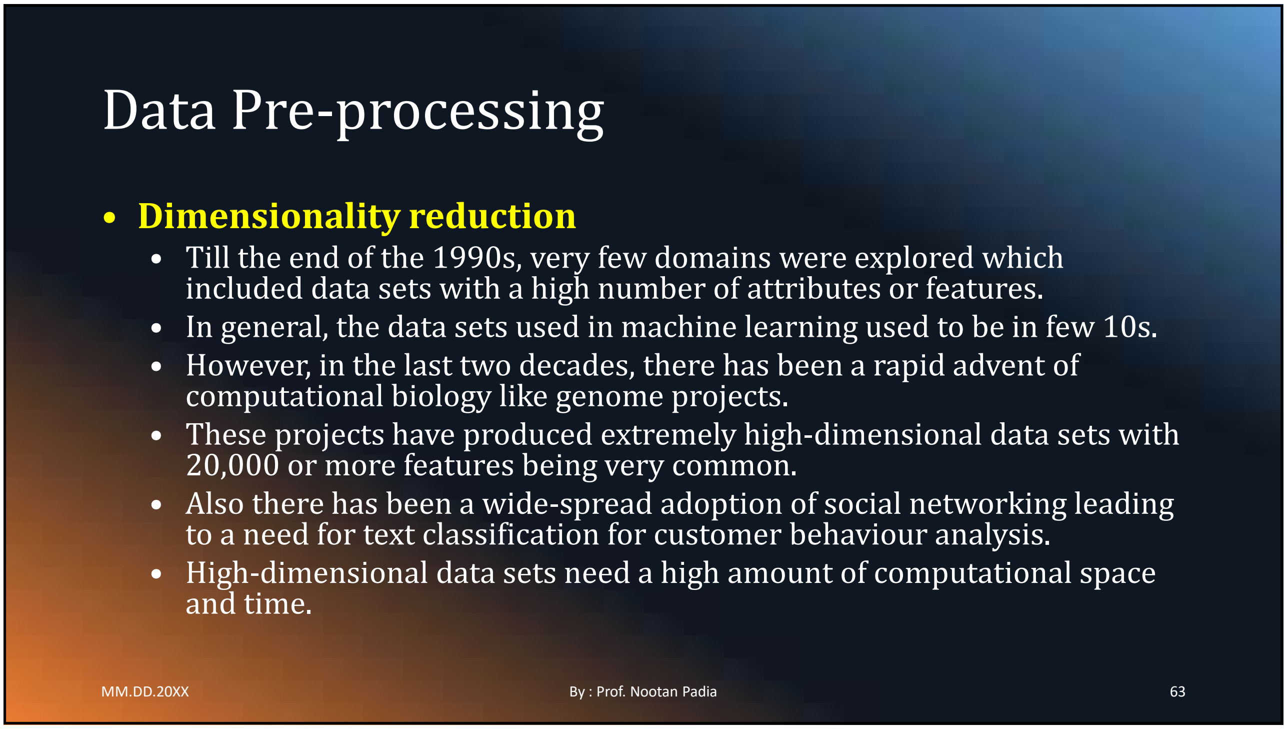


Table 2.11
Missing Values for 'Horsepower' Attribute

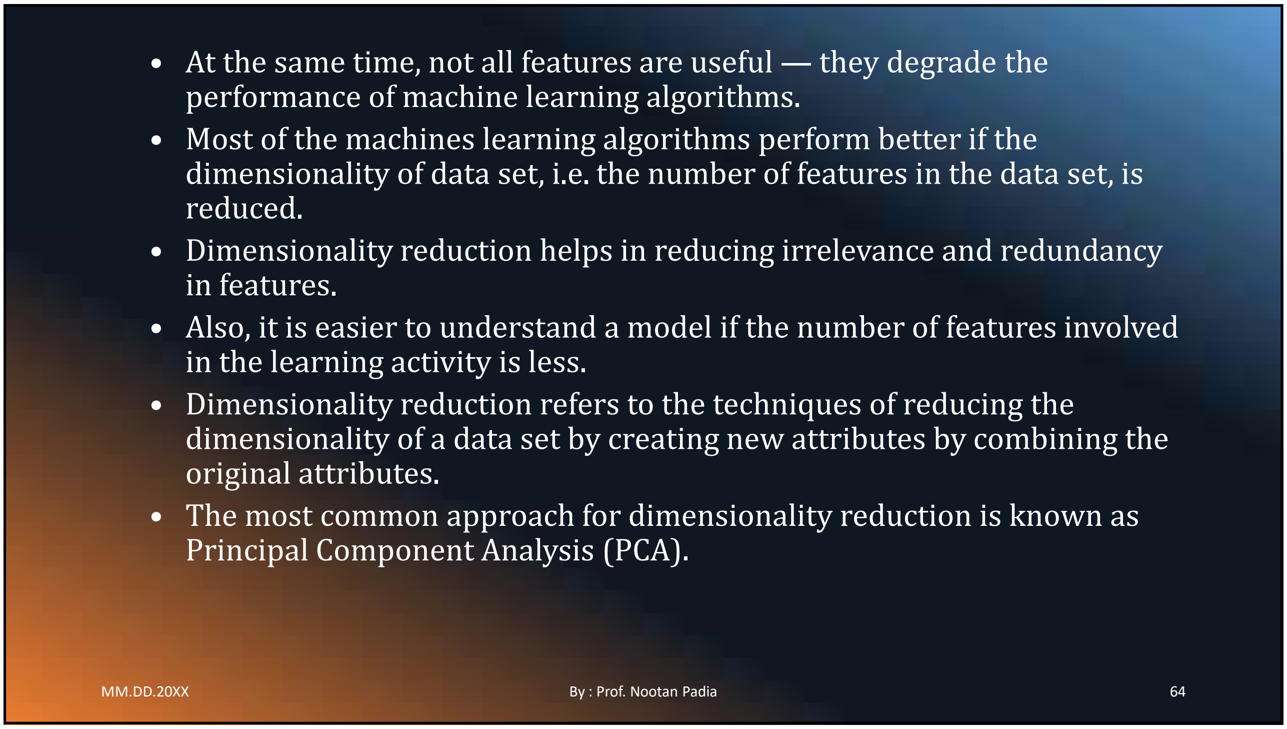
mpg	cylin-ders	dis-place-ment	horse-power	weight	accel-eration	model year	origin	car name
25	4	98	?	2046	19	71	1	Ford pinto
21	6	200	?	2875	17	74	1	Ford maverick
40.9	4	85	?	1835	17.3	80	2	Renault lecar deluxe
23.6	4	140	?	2905	14.3	80	1	Ford mustang cobra
34.5	4	100	?	2320	15.8	81	2	Renault 18i
23	4	151	?	3035	20.5	82	1	Amc concord dl

- **Estimate missing values**
  - If there are data points similar to the ones with missing attribute values, then the attribute values from those similar data points can be planted in place of the missing value.
  - For finding similar data points or observations, distance function can be used.
  - For example, let's assume that the weight of a Russian student having age 12 years and height 5 ft. is missing.
  - Then the weight of any other Russian student having age close to 12 rears and height close to 5 ft. can be assigned.

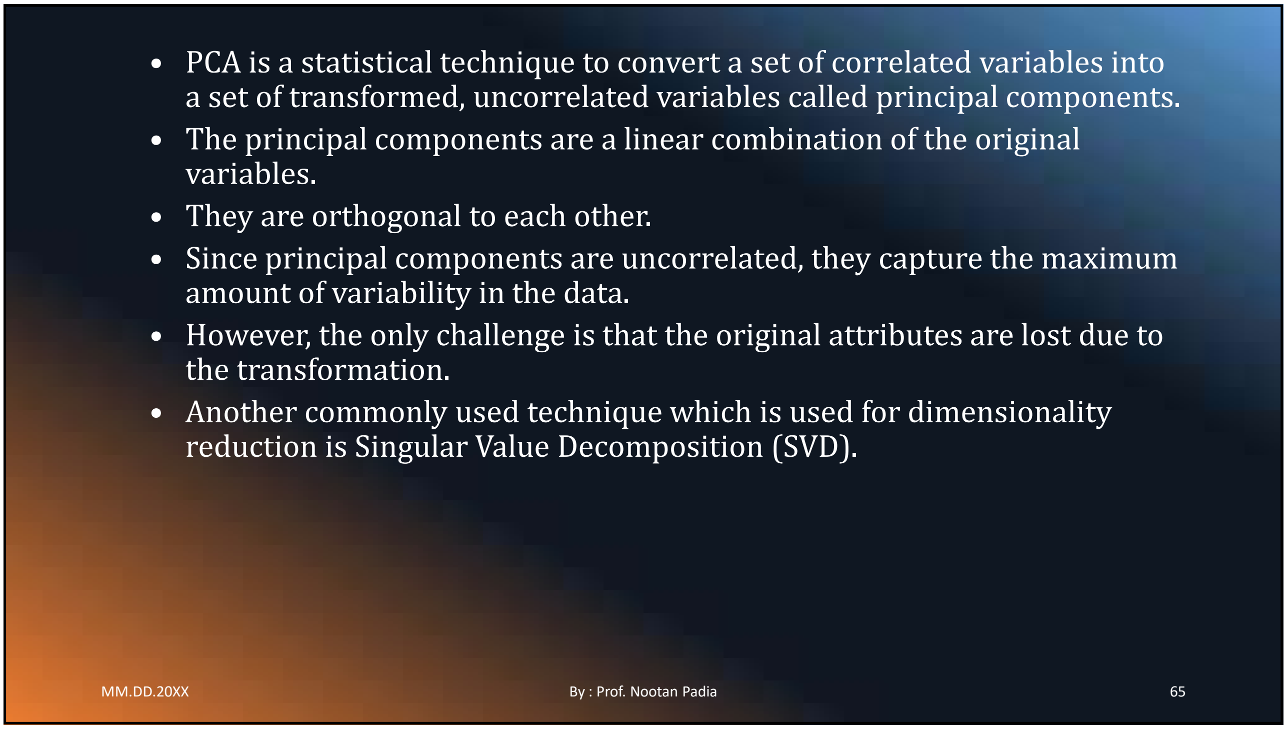


Data Pre-processing

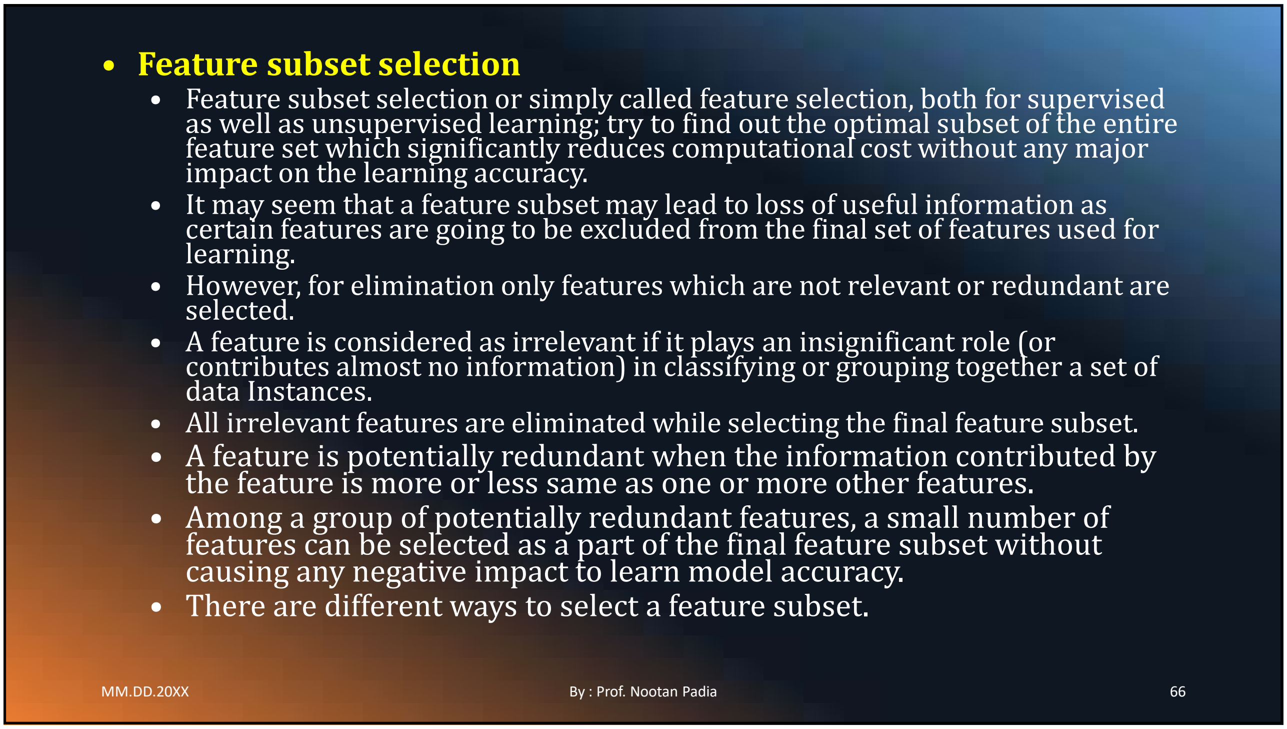
- **Dimensionality reduction**
  - Till the end of the 1990s, very few domains were explored which included data sets with a high number of attributes or features.
  - In general, the data sets used in machine learning used to be in few 10s.
  - However, in the last two decades, there has been a rapid advent of computational biology like genome projects.
  - These projects have produced extremely high-dimensional data sets with 20,000 or more features being very common.
  - Also there has been a wide-spread adoption of social networking leading to a need for text classification for customer behaviour analysis.
  - High-dimensional data sets need a high amount of computational space and time.



- At the same time, not all features are useful — they degrade the performance of machine learning algorithms.
- Most of the machines learning algorithms perform better if the dimensionality of data set, i.e. the number of features in the data set, is reduced.
- Dimensionality reduction helps in reducing irrelevance and redundancy in features.
- Also, it is easier to understand a model if the number of features involved in the learning activity is less.
- Dimensionality reduction refers to the techniques of reducing the dimensionality of a data set by creating new attributes by combining the original attributes.
- The most common approach for dimensionality reduction is known as Principal Component Analysis (PCA).



- PCA is a statistical technique to convert a set of correlated variables into a set of transformed, uncorrelated variables called principal components.
- The principal components are a linear combination of the original variables.
- They are orthogonal to each other.
- Since principal components are uncorrelated, they capture the maximum amount of variability in the data.
- However, the only challenge is that the original attributes are lost due to the transformation.
- Another commonly used technique which is used for dimensionality reduction is Singular Value Decomposition (SVD).

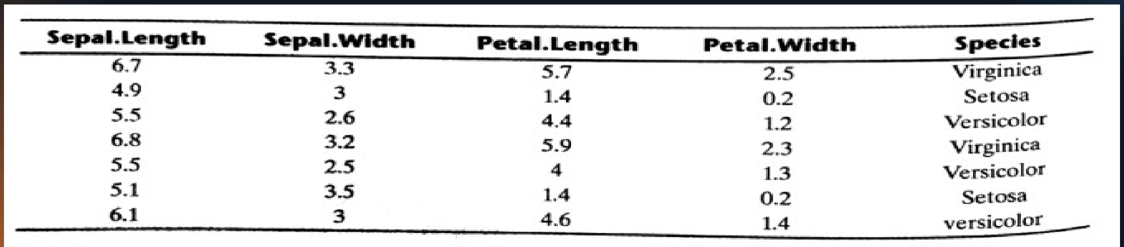


- **Feature subset selection**
  - Feature subset selection or simply called feature selection, both for supervised as well as unsupervised learning; try to find out the optimal subset of the entire feature set which significantly reduces computational cost without any major impact on the learning accuracy.
  - It may seem that a feature subset may lead to loss of useful information as certain features are going to be excluded from the final set of features used for learning.
  - However, for elimination only features which are not relevant or redundant are selected.
  - A feature is considered as irrelevant if it plays an insignificant role (or contributes almost no information) in classifying or grouping together a set of data Instances.
  - All irrelevant features are eliminated while selecting the final feature subset.
  - A feature is potentially redundant when the information contributed by the feature is more or less same as one or more other features.
  - Among a group of potentially redundant features, a small number of features can be selected as a part of the final feature subset without causing any negative impact to learn model accuracy.
  - There are different ways to select a feature subset.

What is feature?

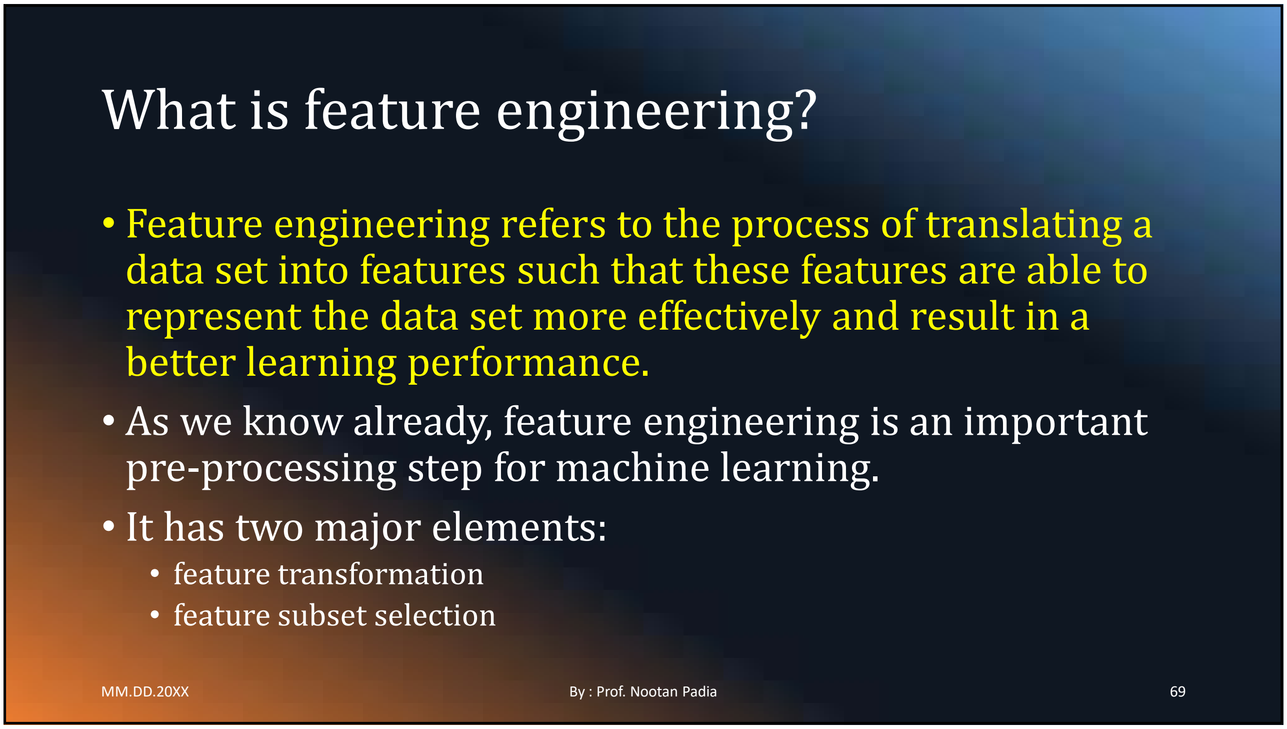
- A feature is an attribute of a data set that is used in a machine learning process.
- There is a view amongst certain machine learning practitioners that only those attributes which are meaningful to a machine learning problem are to be called as features, but this view has to be taken with a pinch of salt.
- In fact, selection of the subset of features which are meaningful for machine learning is a sub-area of feature engineering which draws a lot of research interest.
- The features in a data set are also called its dimensions. So a data set having 'n' features are called an n-dimensional data set.
- Let's take the example of a famous machine learning data set, Iris, introduced by the British statistician and biologist Ronald Fisher, partly shown in Figure in next slide.

- It has five attributes or features namely Sepal.Length, Sepal.Width, Petal.Length, Petal.Width and Species.
- Out of these, the feature ‘Species’ represents the class variable and the remaining features are the predictor variables.
- It is a five-dimensional data set.



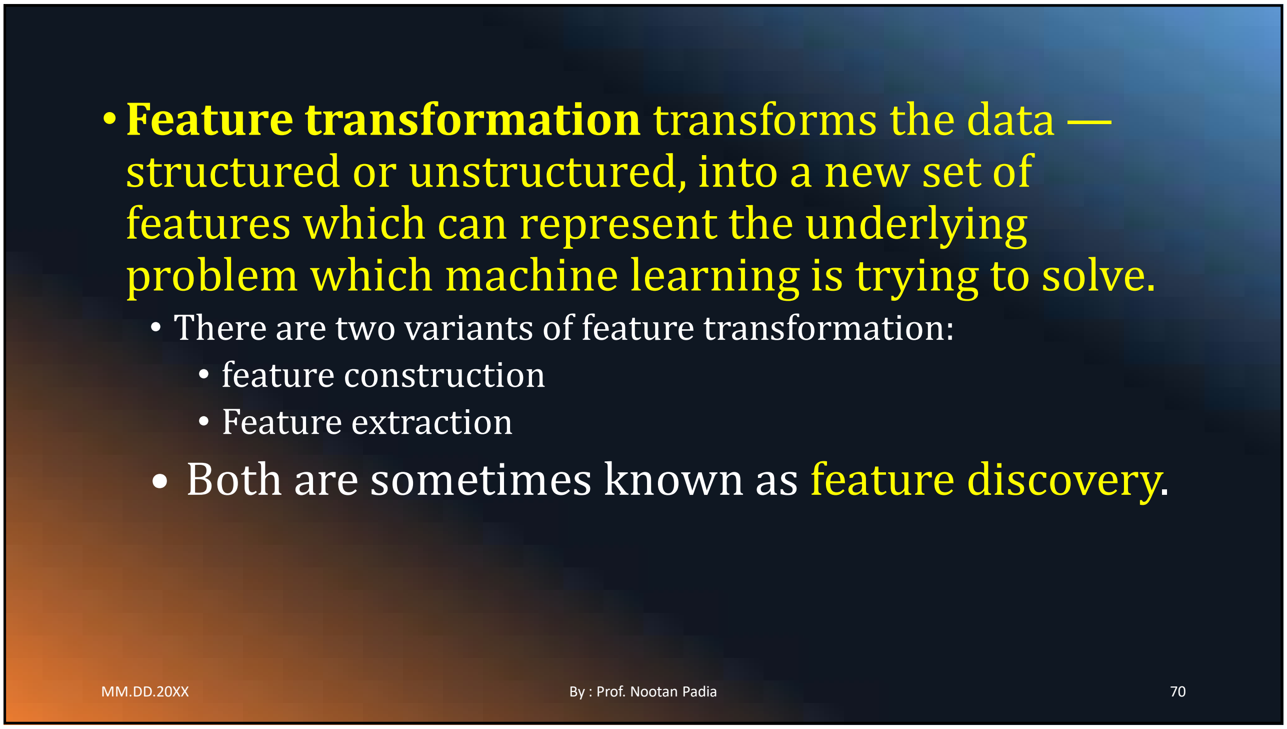
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
6.7	3.3	5.7	2.5	Virginica
4.9	3	1.4	0.2	Setosa
5.5	2.6	4.4	1.2	Versicolor
6.8	3.2	5.9	2.3	Virginica
5.5	2.5	4	1.3	Versicolor
5.1	3.5	1.4	0.2	Setosa
6.1	3	4.6	1.4	versicolor

FIG. 4.1
Data set features

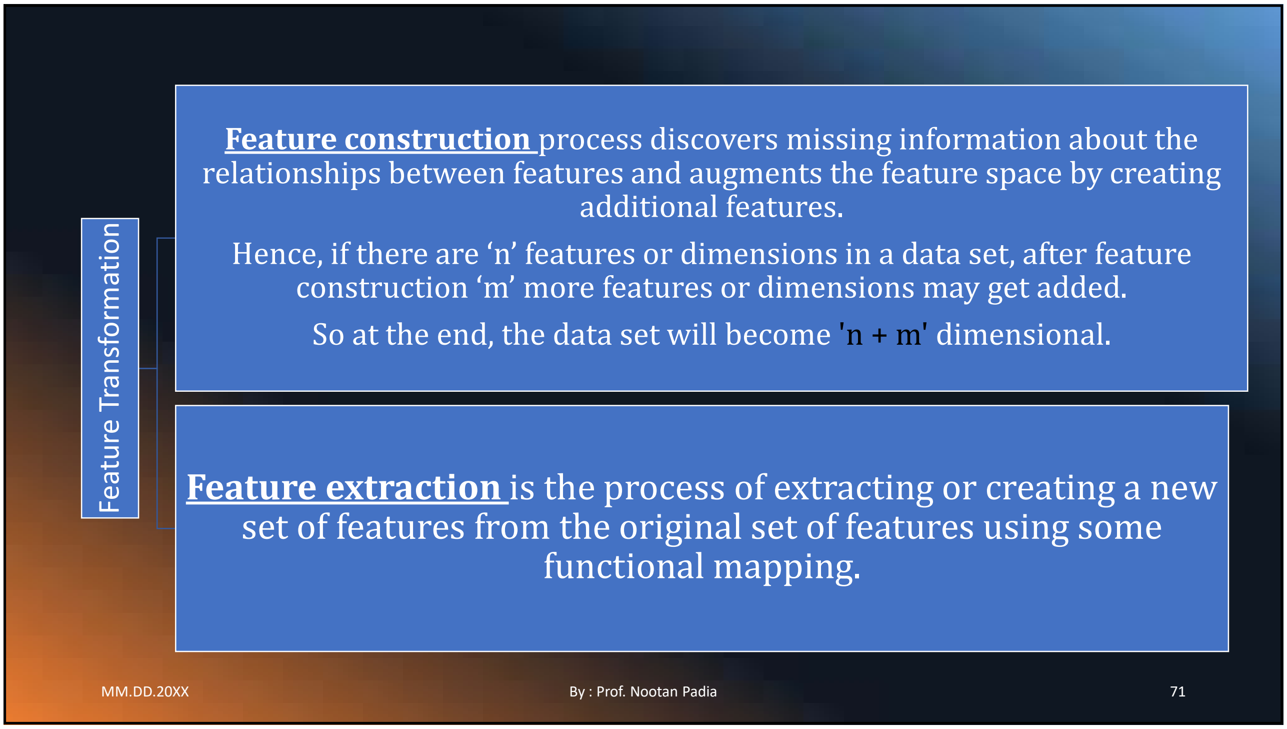


What is feature engineering?

- Feature engineering refers to the process of translating a data set into features such that these features are able to represent the data set more effectively and result in a better learning performance.
- As we know already, feature engineering is an important pre-processing step for machine learning.
- It has two major elements:
  - feature transformation
  - feature subset selection



- **Feature transformation** transforms the data — structured or unstructured, into a new set of features which can represent the underlying problem which machine learning is trying to solve.
  - There are two variants of feature transformation:
    - feature construction
    - Feature extraction
  - Both are sometimes known as feature discovery.



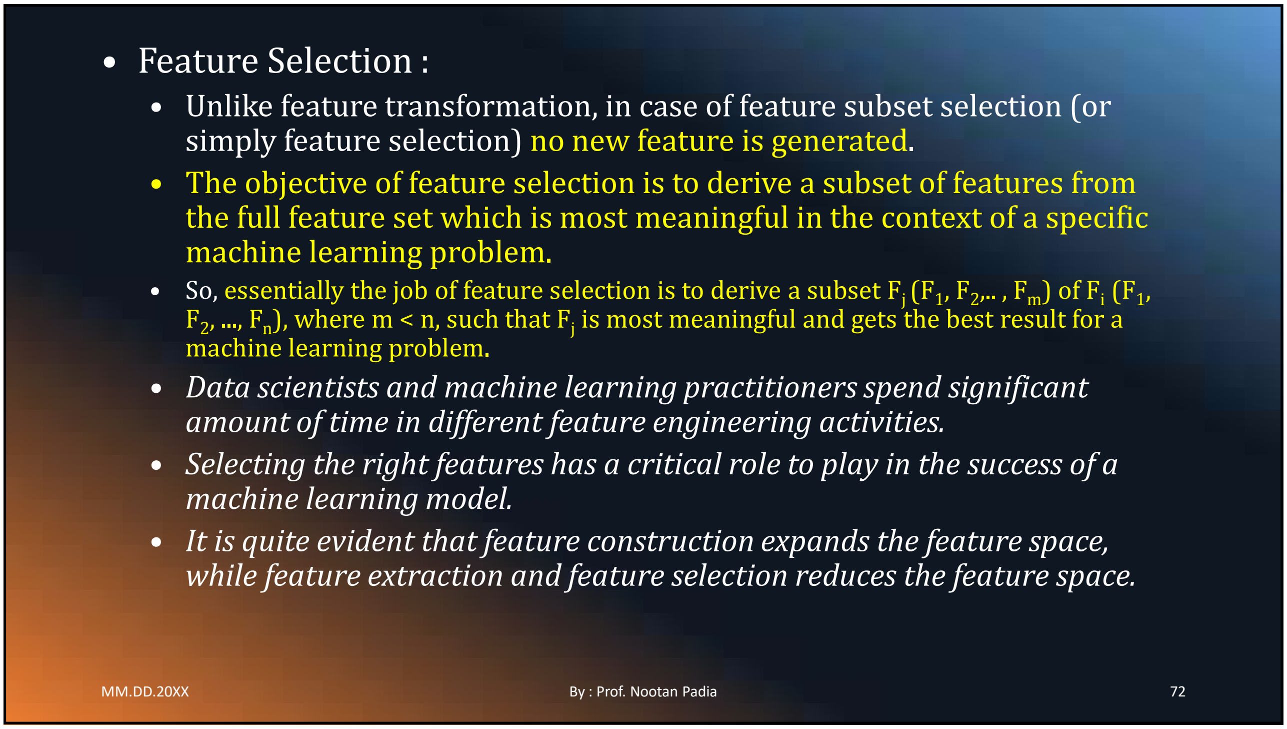
Feature Transformation

Feature construction process discovers missing information about the relationships between features and augments the feature space by creating additional features.

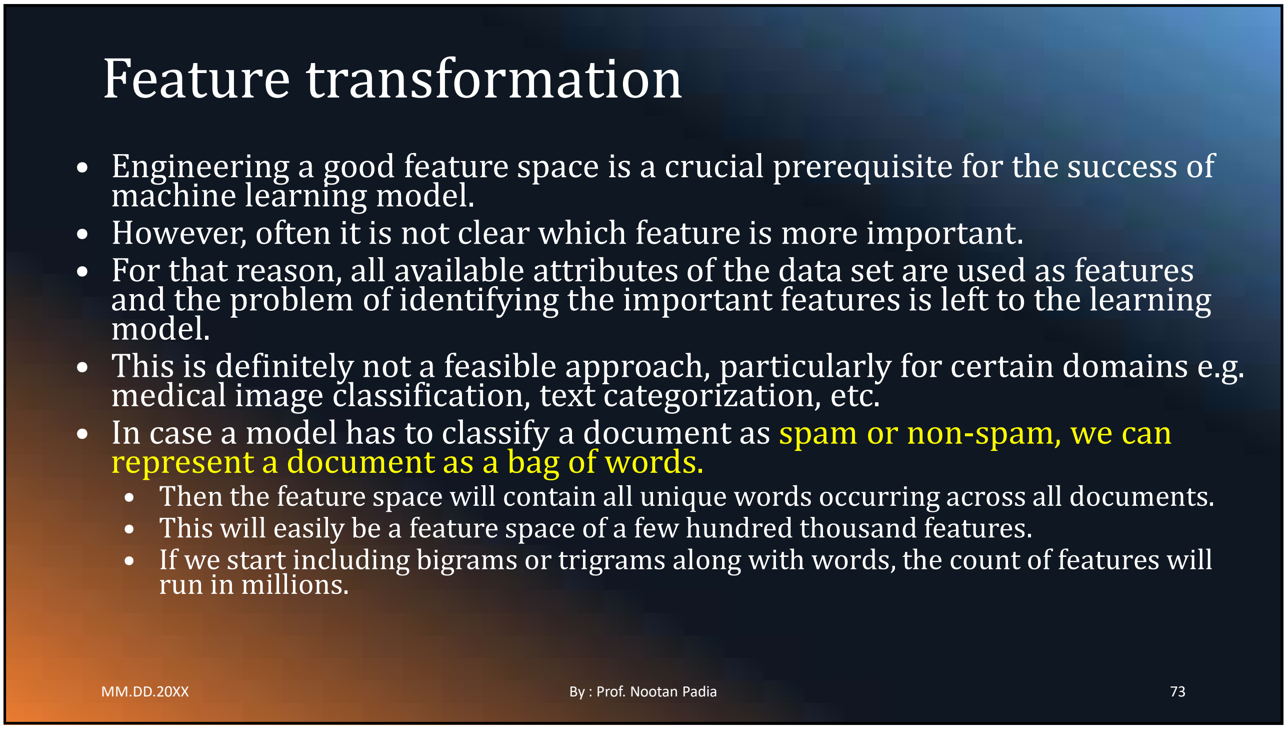
Hence, if there are ‘n’ features or dimensions in a data set, after feature construction ‘m’ more features or dimensions may get added.

So at the end, the data set will become 'n + m' dimensional.

Feature extraction is the process of extracting or creating a new set of features from the original set of features using some functional mapping.

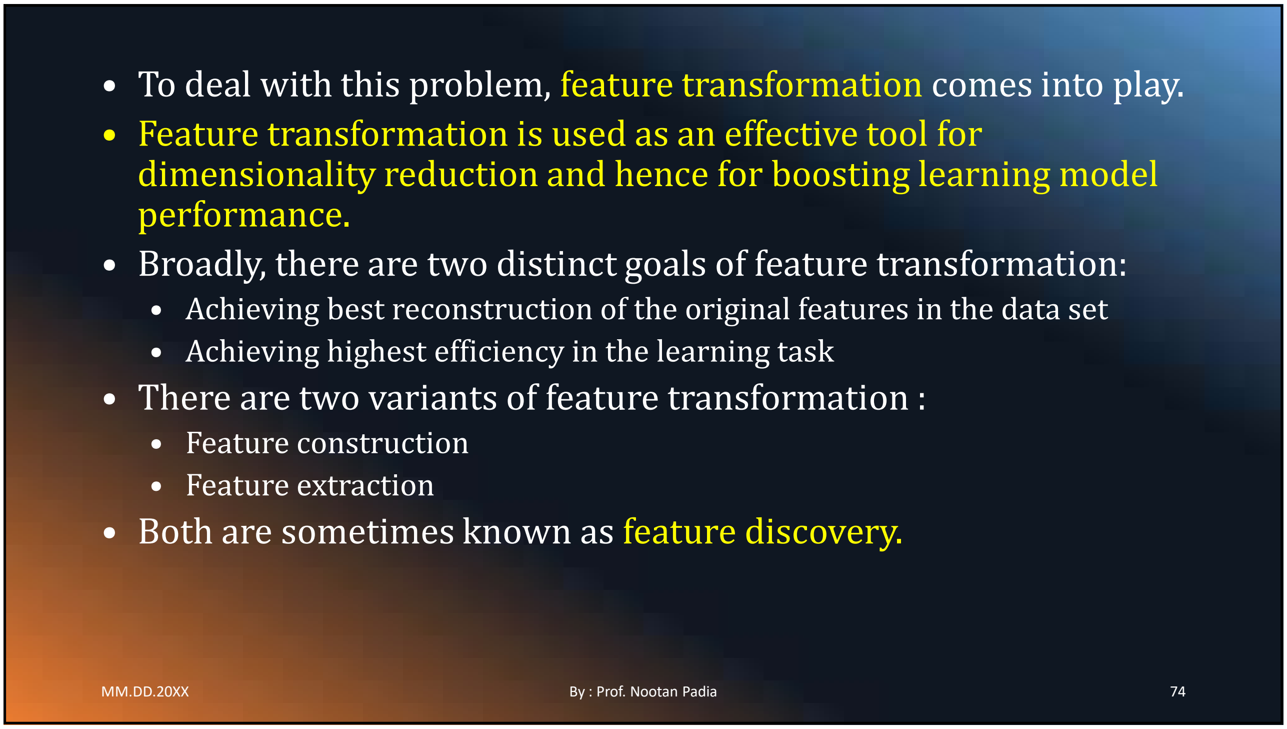


- Feature Selection :
  - Unlike feature transformation, in case of feature subset selection (or simply feature selection) no new feature is generated.
  - The objective of feature selection is to derive a subset of features from the full feature set which is most meaningful in the context of a specific machine learning problem.
  - So, essentially the job of feature selection is to derive a subset F_j $(F_1, F_2,.. , F_m)$ of F_i $(F_1, F_2, ..., F_n)$, where $m < n$, such that F_j is most meaningful and gets the best result for a machine learning problem.
  - *Data scientists and machine learning practitioners spend significant amount of time in different feature engineering activities.*
  - *Selecting the right features has a critical role to play in the success of a machine learning model.*
  - *It is quite evident that feature construction expands the feature space, while feature extraction and feature selection reduces the feature space.*



Feature transformation

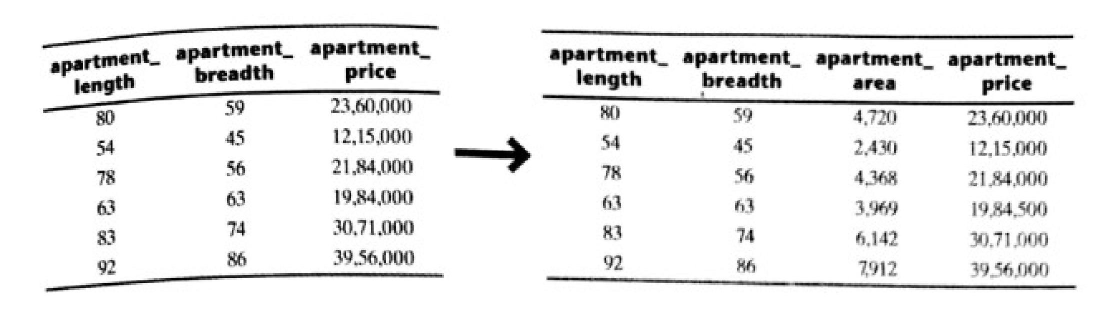
- Engineering a good feature space is a crucial prerequisite for the success of machine learning model.
- However, often it is not clear which feature is more important.
- For that reason, all available attributes of the data set are used as features and the problem of identifying the important features is left to the learning model.
- This is definitely not a feasible approach, particularly for certain domains e.g. medical image classification, text categorization, etc.
- In case a model has to classify a document as spam or non-spam, we can represent a document as a bag of words.
  - Then the feature space will contain all unique words occurring across all documents.
  - This will easily be a feature space of a few hundred thousand features.
  - If we start including bigrams or trigrams along with words, the count of features will run in millions.



- To deal with this problem, feature transformation comes into play.
- Feature transformation is used as an effective tool for dimensionality reduction and hence for boosting learning model performance.
- Broadly, there are two distinct goals of feature transformation:
  - Achieving best reconstruction of the original features in the data set
  - Achieving highest efficiency in the learning task
- There are two variants of feature transformation :
  - Feature construction
  - Feature extraction
- Both are sometimes known as feature discovery.

Feature Construction

- Feature construction involves transforming a given set of input features to generate a new set of more powerful features.
- Let's take the example of a real estate data set having details of all apartments sold in a specific region.
    - The data set has three features — apartment length apartment breadth, and price, of the apartment.
    - If it is used as an input to a regression problem, such data can be training data for the regression model.
    - So given the training data, the model should be able to predict the price of an apartment whose price is not known or which has just come up for sale.
    - However instead of using length and breadth or the apartment as a predictor, it is much convenient and makes more sense to use the area of the apartment, which is not an existing feature of the data set.
    - So such a feature, namely apartment area, can be added to the data set.
    - In other words, we transform the three-dimensional data set to a four-dimensional data set, with the newly 'discovered' feature apartment area being added to the original data set.

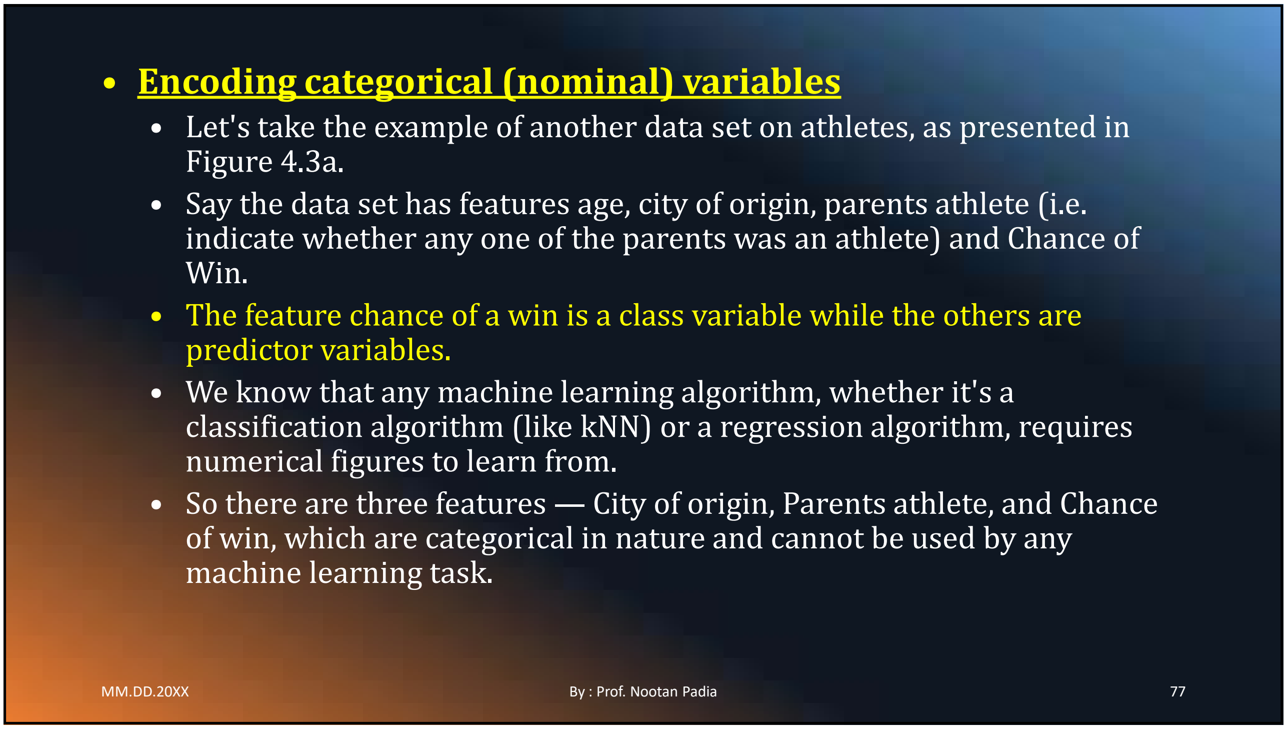


apartment_ length	apartment_ breadth	apartment_ price
80	59	23,60,000
54	45	12,15,000
78	56	21,84,000
63	63	19,84,000
83	74	30,71,000
92	86	39,56,000

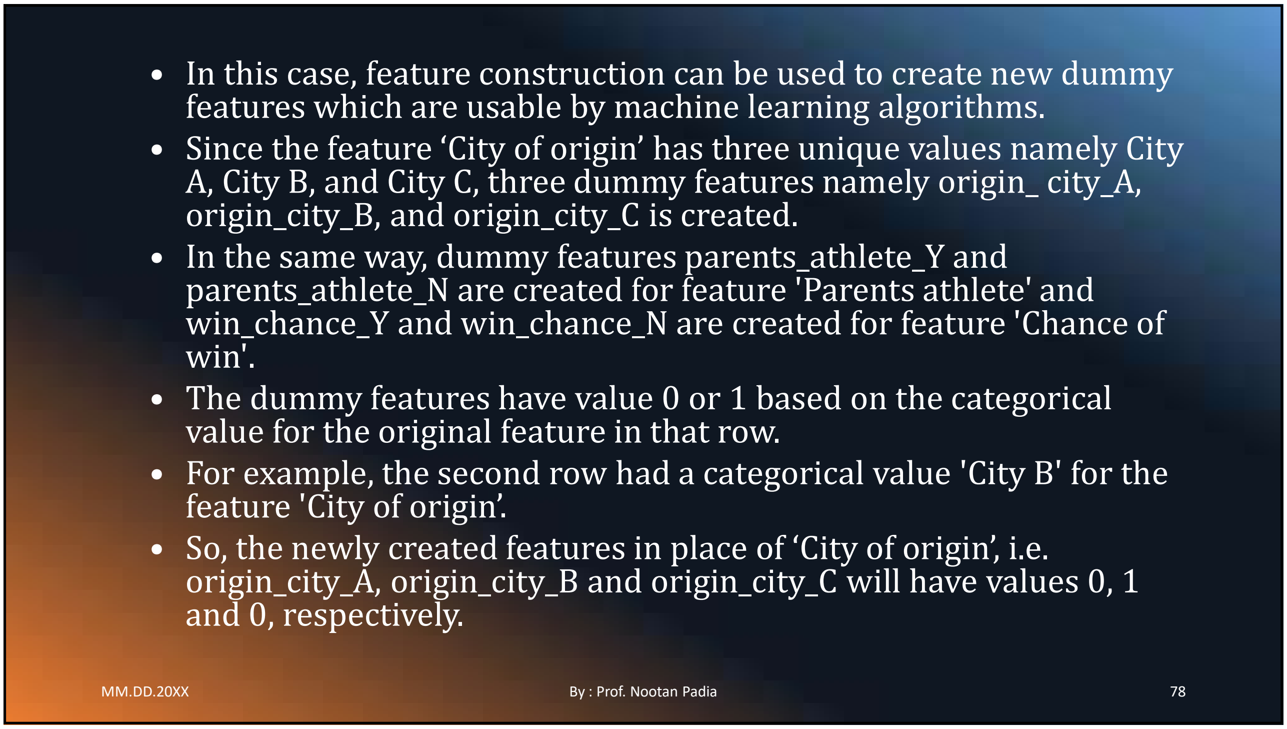
→

apartment_ length	apartment_ breadth	apartment_ area	apartment_ price
80	59	4,720	23,60,000
54	45	2,430	12,15,000
78	56	4,368	21,84,000
63	63	3,969	19,84,500
83	74	6,142	30,71,000
92	86	7,912	39,56,000

- There are certain situations where feature construction is an essential activity before we can start with the machine learning task.
- These situations are
    - when features have categorical value and machine learning needs numeric value inputs
    - when features having numeric (continuous) values and need to be converted to ordinal values
    - when text-specific feature construction needs to be done

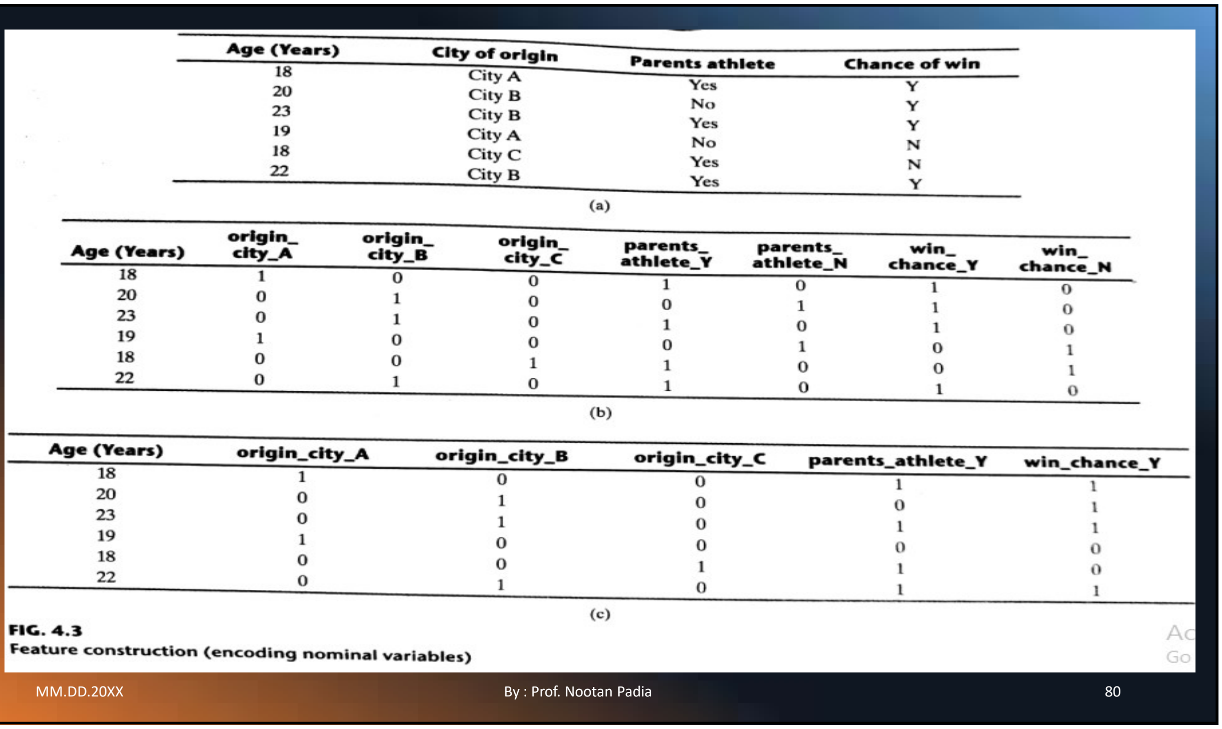


- **Encoding categorical (nominal) variables**
  - Let's take the example of another data set on athletes, as presented in Figure 4.3a.
  - Say the data set has features age, city of origin, parents athlete (i.e. indicate whether any one of the parents was an athlete) and Chance of Win.
  - The feature chance of a win is a class variable while the others are predictor variables.
  - We know that any machine learning algorithm, whether it's a classification algorithm (like kNN) or a regression algorithm, requires numerical figures to learn from.
  - So there are three features — City of origin, Parents athlete, and Chance of win, which are categorical in nature and cannot be used by any machine learning task.



- In this case, feature construction can be used to create new dummy features which are usable by machine learning algorithms.
- Since the feature ‘City of origin’ has three unique values namely City A, City B, and City C, three dummy features namely origin_ city_A, origin_city_B, and origin_city_C is created.
- In the same way, dummy features parents_athlete_Y and parents_athlete_N are created for feature 'Parents athlete' and win_chance_Y and win_chance_N are created for feature 'Chance of win'.
- The dummy features have value 0 or 1 based on the categorical value for the original feature in that row.
- For example, the second row had a categorical value 'City B' for the feature 'City of origin’.
- So, the newly created features in place of ‘City of origin’, i.e. origin_city_A, origin_city_B and origin_city_C will have values 0, 1 and 0, respectively.

- In the same way, parents_athlete_Y and parents_athlete_N will have values 0 and 1, respectively in row 2 as the original feature 'Parents athlete' had a categorical value 'No' in row 2.
- The entire set of transformation for athletes' data set is shown in Figure 4.3b.
- However, examining closely, we see that the features ‘Parents athlete' and 'Chance of win' in the original data set can have two values only.
- So creating two features from them is a kind of duplication, since the value of one feature can be decided from the value of the other.
- To avoid this duplication, we can just leave one feature and eliminate the other, as shown in Figure 4.3c.



Age (Years)	City of origin	Parents athlete	Chance of win
18	City A	Yes	Y
20	City B	No	Y
23	City B	Yes	Y
19	City A	No	N
18	City C	Yes	N
22	City B	Yes	Y

(a)

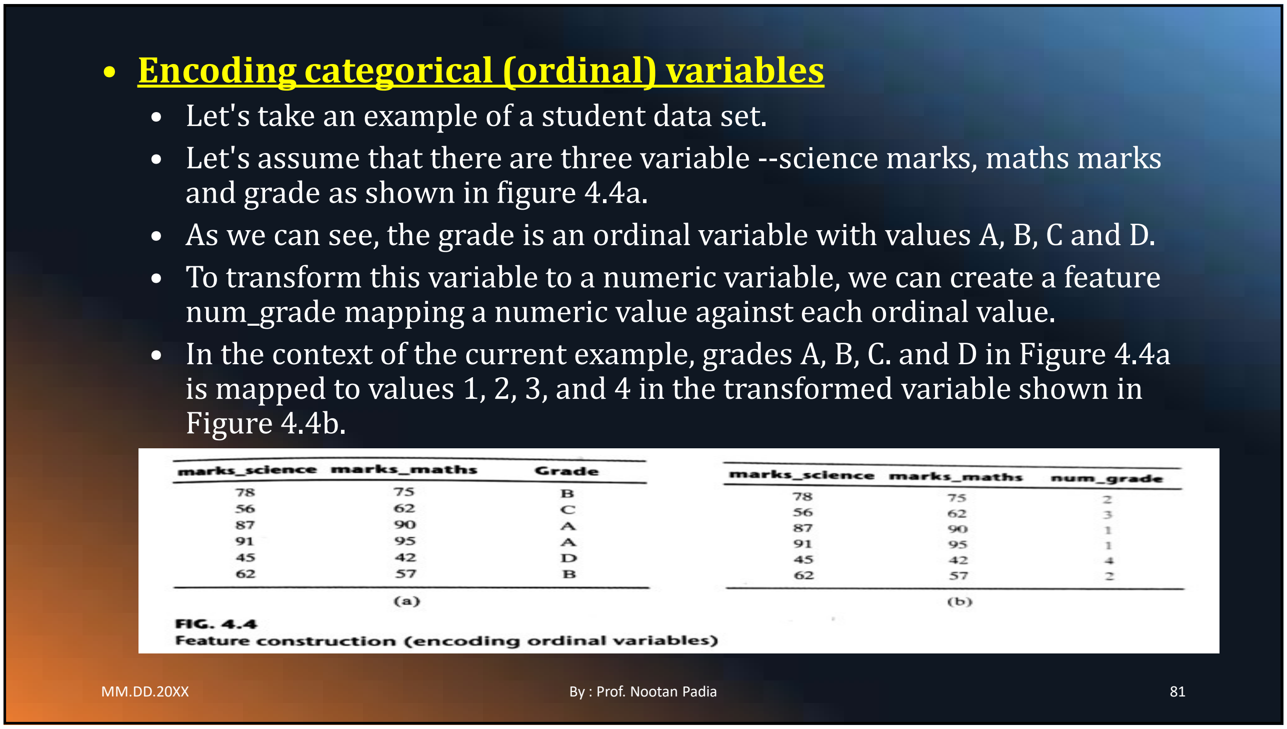
Age (Years)	origin_city_A	origin_city_B	origin_city_C	parents_athlete_Y	parents_athlete_N	win_chance_Y	win_chance_N
18	1	0	0	1	0	1	0
20	0	1	0	0	1	1	0
23	0	1	0	1	0	1	0
19	1	0	0	0	1	0	1
18	0	0	1	1	0	0	1
22	0	1	0	1	0	1	0

(b)

Age (Years)	origin_city_A	origin_city_B	origin_city_C	parents_athlete_Y	win_chance_Y
18	1	0	0	1	1
20	0	1	0	0	1
23	0	1	0	1	1
19	1	0	0	0	0
18	0	0	1	1	0
22	0	1	0	1	1

(c)

FIG. 4.3
Feature construction (encoding nominal variables)



Encoding categorical (ordinal) variables

- Let's take an example of a student data set.
- Let's assume that there are three variable --science marks, maths marks and grade as shown in figure 4.4a.
- As we can see, the grade is an ordinal variable with values A, B, C and D.
- To transform this variable to a numeric variable, we can create a feature num_grade mapping a numeric value against each ordinal value.
- In the context of the current example, grades A, B, C. and D in Figure 4.4a is mapped to values 1, 2, 3, and 4 in the transformed variable shown in Figure 4.4b.

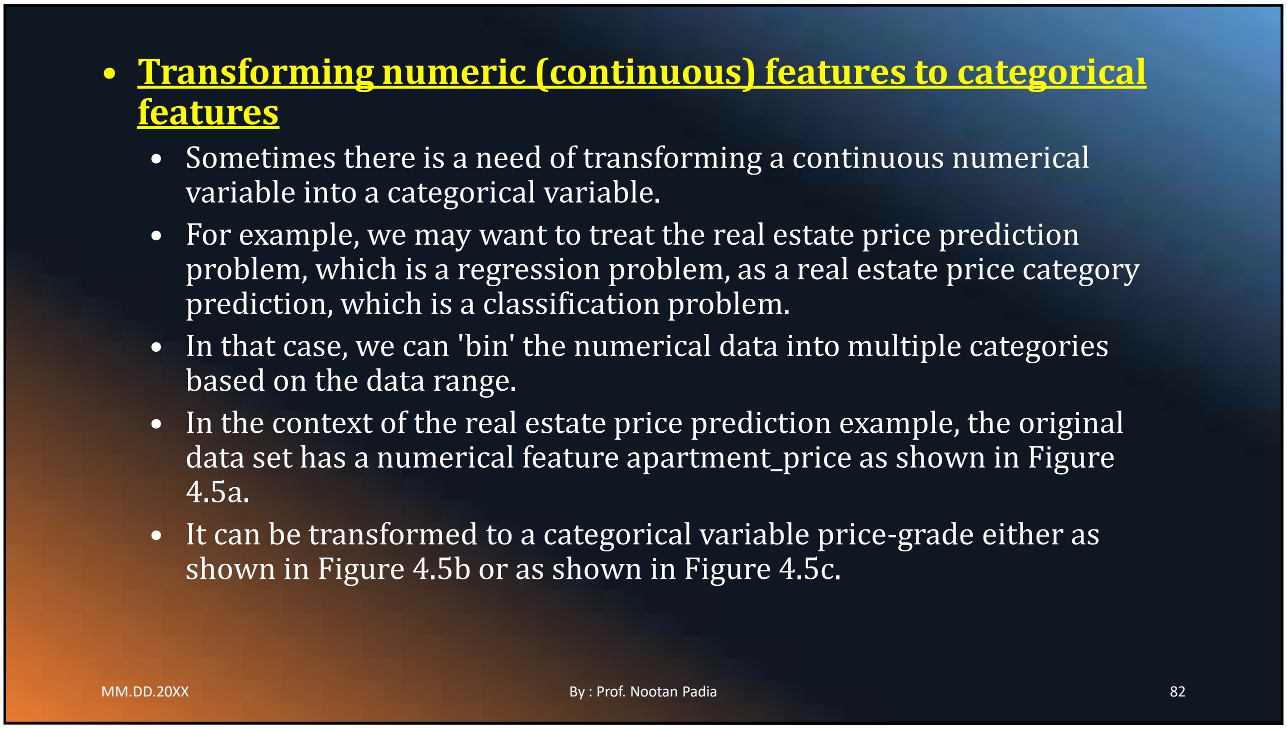
marks_science	marks_maths	Grade
78	75	B
56	62	C
87	90	A
91	95	A
45	42	D
62	57	B

(a)

marks_science	marks_maths	num_grade
78	75	2
56	62	3
87	90	1
91	95	1
45	42	4
62	57	2

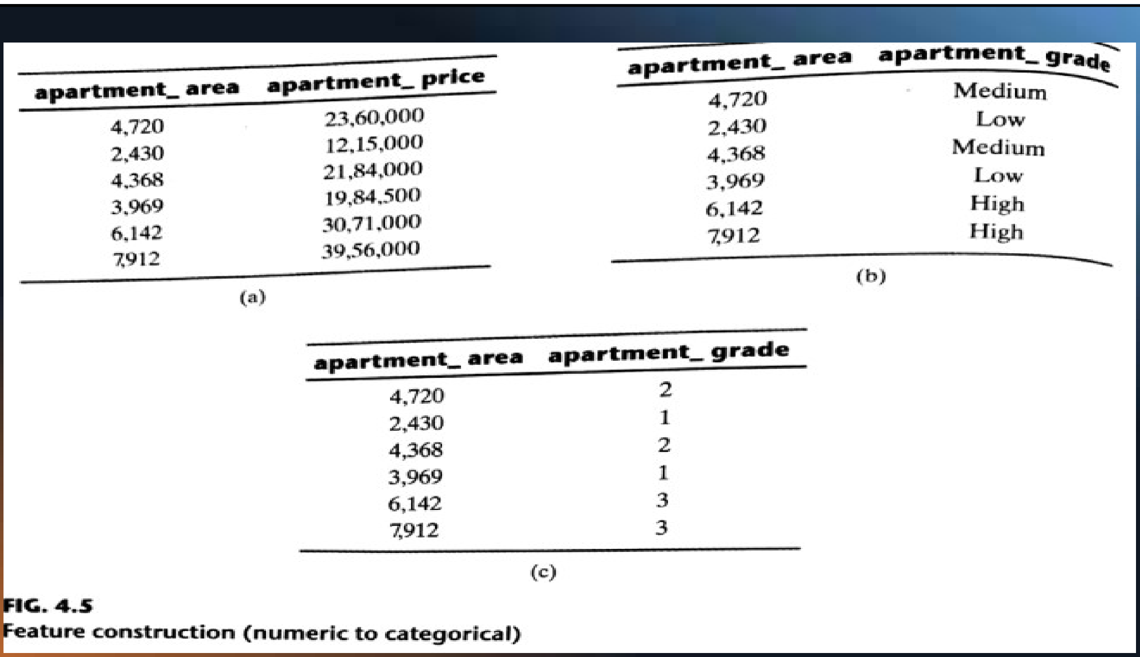
(b)

FIG. 4.4
Feature construction (encoding ordinal variables)



Transforming numeric (continuous) features to categorical features

- Sometimes there is a need of transforming a continuous numerical variable into a categorical variable.
- For example, we may want to treat the real estate price prediction problem, which is a regression problem, as a real estate price category prediction, which is a classification problem.
- In that case, we can 'bin' the numerical data into multiple categories based on the data range.
- In the context of the real estate price prediction example, the original data set has a numerical feature apartment_price as shown in Figure 4.5a.
- It can be transformed to a categorical variable price-grade either as shown in Figure 4.5b or as shown in Figure 4.5c.



apartment_ area	apartment_ price
4,720	23,60,000
2,430	12,15,000
4,368	21,84,000
3,969	19,84,500
6,142	30,71,000
7,912	39,56,000

(a)

apartment_ area	apartment_ grade
4,720	Medium
2,430	Low
4,368	Medium
3,969	Low
6,142	High
7,912	High

(b)

apartment_ area	apartment_ grade
4,720	2
2,430	1
4,368	2
3,969	1
6,142	3
7,912	3

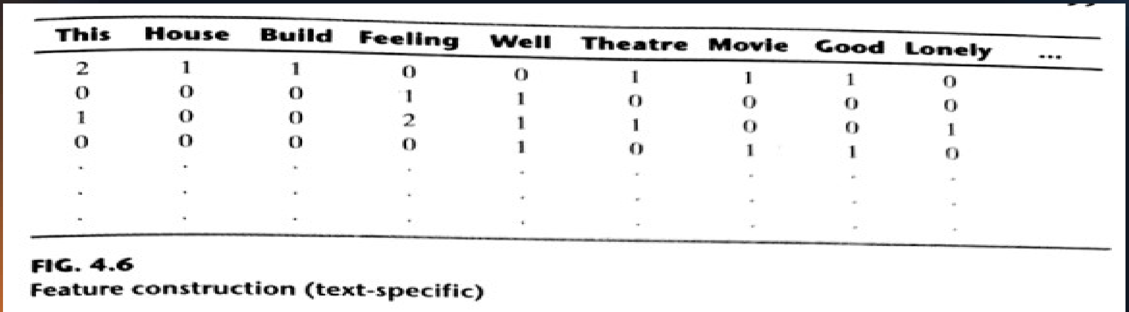
(c)

FIG. 4.5
Feature construction (numeric to categorical)

- **Text-specific feature construction**
  - In the current world, text is arguably the most predominant medium of communication.
  - Whether we think about social networks like Facebook or micro-blogging channels like Twitter or emails or short messaging services such as Whatsapp, text plays a major role in the flow of information.
  - Hence, text mining is an important area of research - not only for technology practitioners, but also for industry practitioners.
  - However, making sense of text data, due to the inherent unstructured nature of the data, is not so straightforward.
  - In the first place, the text data chunks that we can think about do not have readily available features, like structured data sets, on which machine learning tasks can be executed.
  - All machine learning models need numerical data as input.

- So the text data in the data sets need to be transformed into numerical features.
- Text data, or corpus, is converted to a numerical representation following a process is known as vectorization.
- In this process, word occurrences in all documents belonging to the corpus are consolidated in the form of bag-of-words.
- There are three major steps that are followed:
  - tokenize
  - count
  - Normalize
- In order to tokenize a corpus, the blank spaces and punctuations are used as delimiters to separate out the words, or tokens.
- Then the number of occurrences of each token is counted, for each document.
- Lastly, tokens are weighted with reducing importance when they occur in the majority of the documents.

- A matrix is then formed with each token representing a column and a specific document of the corpus representing each row.
- Each cell contains the count of occurrence of the token in a specific document.
- This matrix is known as a document-term matrix (also known as a term-document matrix).
- Figure 4.6 represents a typical document-term matrix which forms an input to a machine learning model.



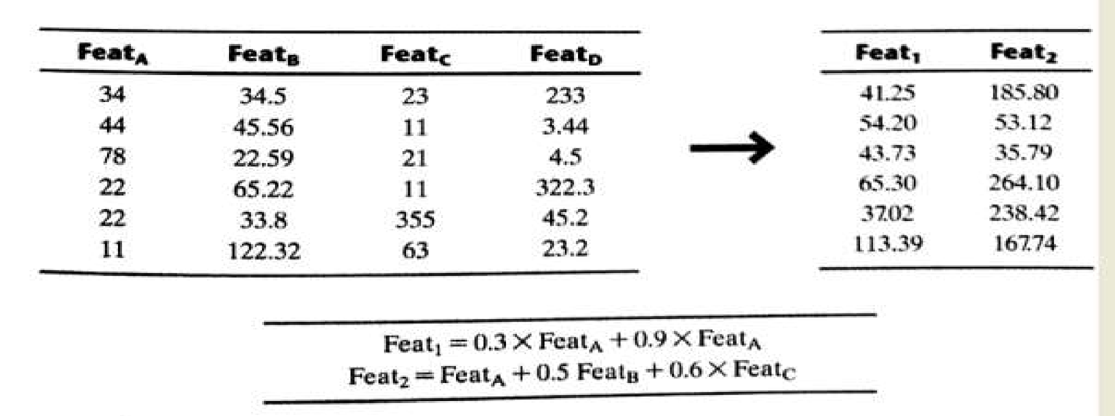
This	House	Build	Feeling	Well	Theatre	Movie	Good	Lonely	...
2	1	1	0	0	1	1	1	0	
0	0	0	1	1	0	0	0	0	
1	0	0	2	1	1	0	0	1	
0	0	0	0	1	0	1	1	0	
.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	

FIG. 4.6
Feature construction (text-specific)

Feature extraction

- In feature extraction, new features are created from a combination of original features.
- Some of the commonly used operators for combining the original features include
- For Boolean features: Conjunctions, Disjunctions, Negation, etc.
- For nominal features: Cartesian product, M of N, etc.
- For numerical features: Min, Max, Addition, Subtraction, Multiplication, Division, Average, Equivalence, Inequality, etc.
- Let's take an example and try to understand.
- Say, we have a data set with a feature set F_i (F_1, F_2, ..., F_n).
- After feature extraction using a mapping function f (F1, F2, ..., F_n) say, we will have a set of features F_i'(F_1', F_2',,F_m') such that F'_i =f(F_i) and m < n.

- For example, $F'_1 = k_1F_1 + k_2F_2$.
- This is depicted in Figure 4.7.



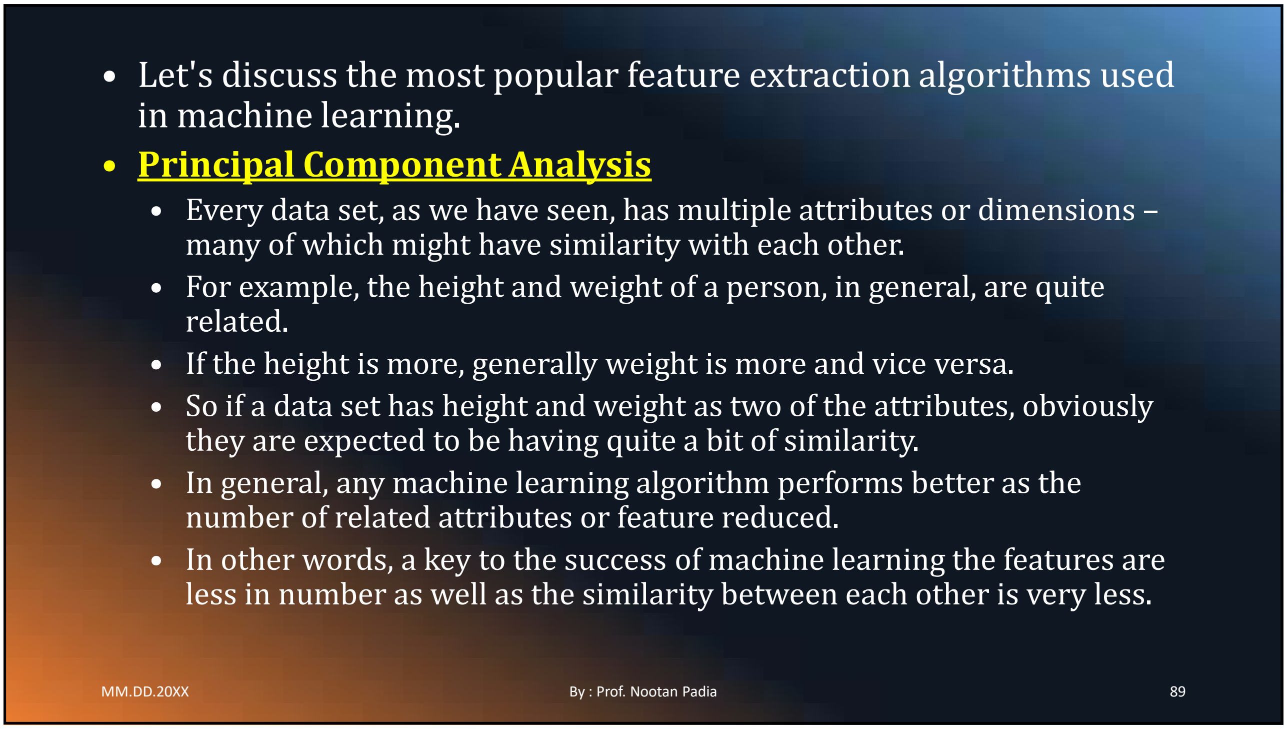
$Feat_A$	$Feat_B$	$Feat_C$	$Feat_D$
34	34.5	23	233
44	45.56	11	3.44
78	22.59	21	4.5
22	65.22	11	322.3
22	33.8	355	45.2
11	122.32	63	23.2

→

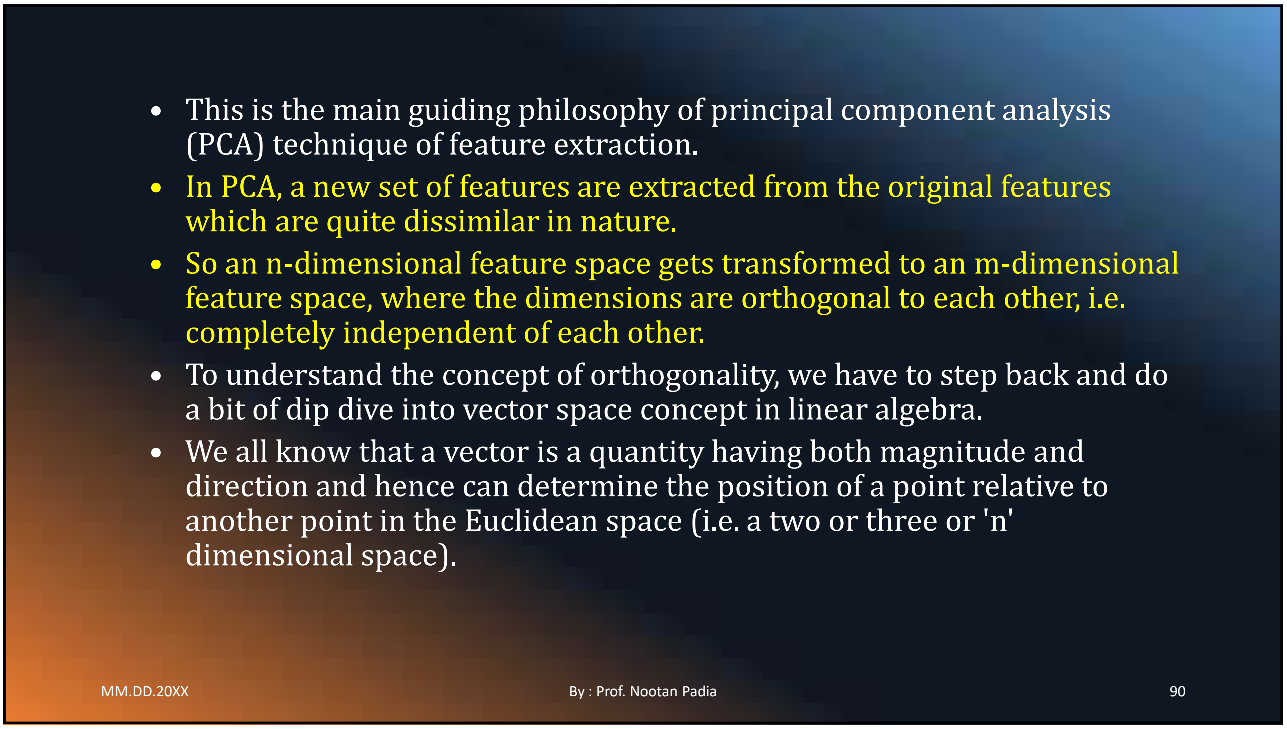
$Feat_1$	$Feat_2$
41.25	185.80
54.20	53.12
43.73	35.79
65.30	264.10
37.02	238.42
113.39	167.74

$$Feat_1 = 0.3 \times Feat_A + 0.9 \times Feat_A$$
$$Feat_2 = Feat_A + 0.5\ Feat_B + 0.6 \times Feat_C$$

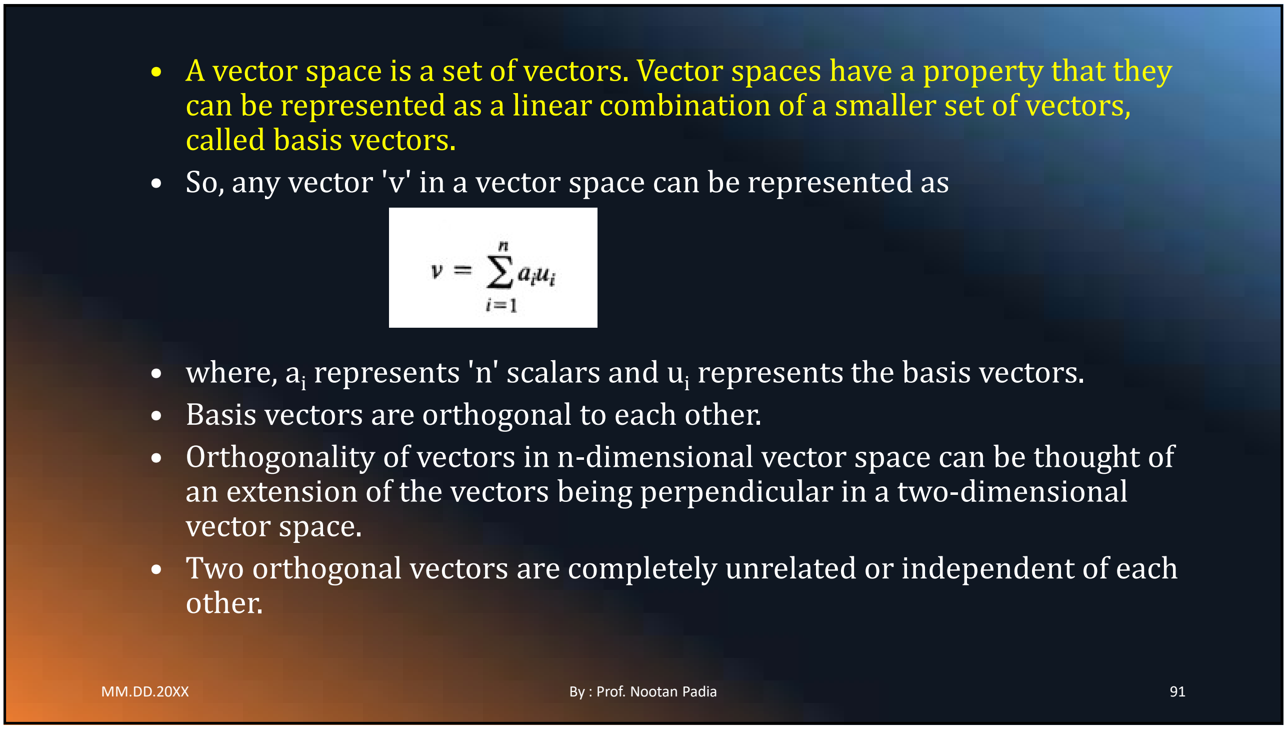
FIG. 4.7 Feature extraction



- Let's discuss the most popular feature extraction algorithms used in machine learning.
- **Principal Component Analysis**
  - Every data set, as we have seen, has multiple attributes or dimensions – many of which might have similarity with each other.
  - For example, the height and weight of a person, in general, are quite related.
  - If the height is more, generally weight is more and vice versa.
  - So if a data set has height and weight as two of the attributes, obviously they are expected to be having quite a bit of similarity.
  - In general, any machine learning algorithm performs better as the number of related attributes or feature reduced.
  - In other words, a key to the success of machine learning the features are less in number as well as the similarity between each other is very less.



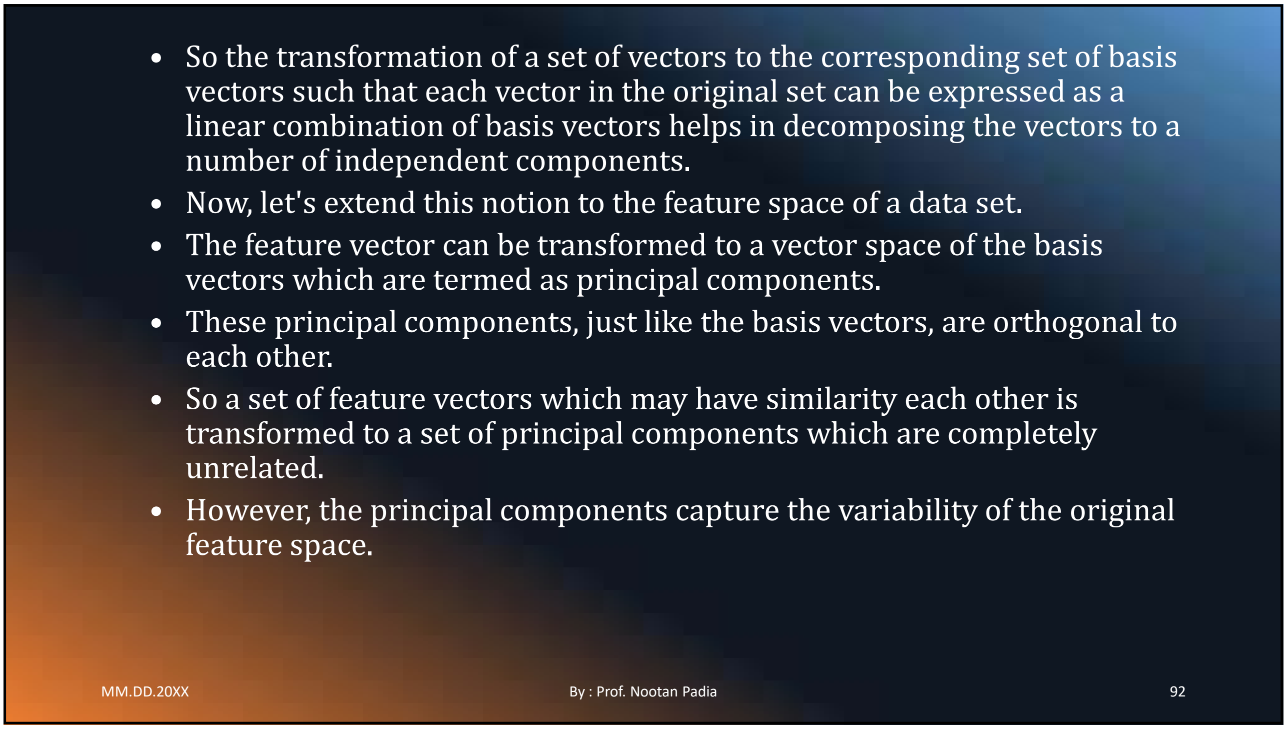
- This is the main guiding philosophy of principal component analysis (PCA) technique of feature extraction.
- In PCA, a new set of features are extracted from the original features which are quite dissimilar in nature.
- So an n-dimensional feature space gets transformed to an m-dimensional feature space, where the dimensions are orthogonal to each other, i.e. completely independent of each other.
- To understand the concept of orthogonality, we have to step back and do a bit of dip dive into vector space concept in linear algebra.
- We all know that a vector is a quantity having both magnitude and direction and hence can determine the position of a point relative to another point in the Euclidean space (i.e. a two or three or 'n' dimensional space).



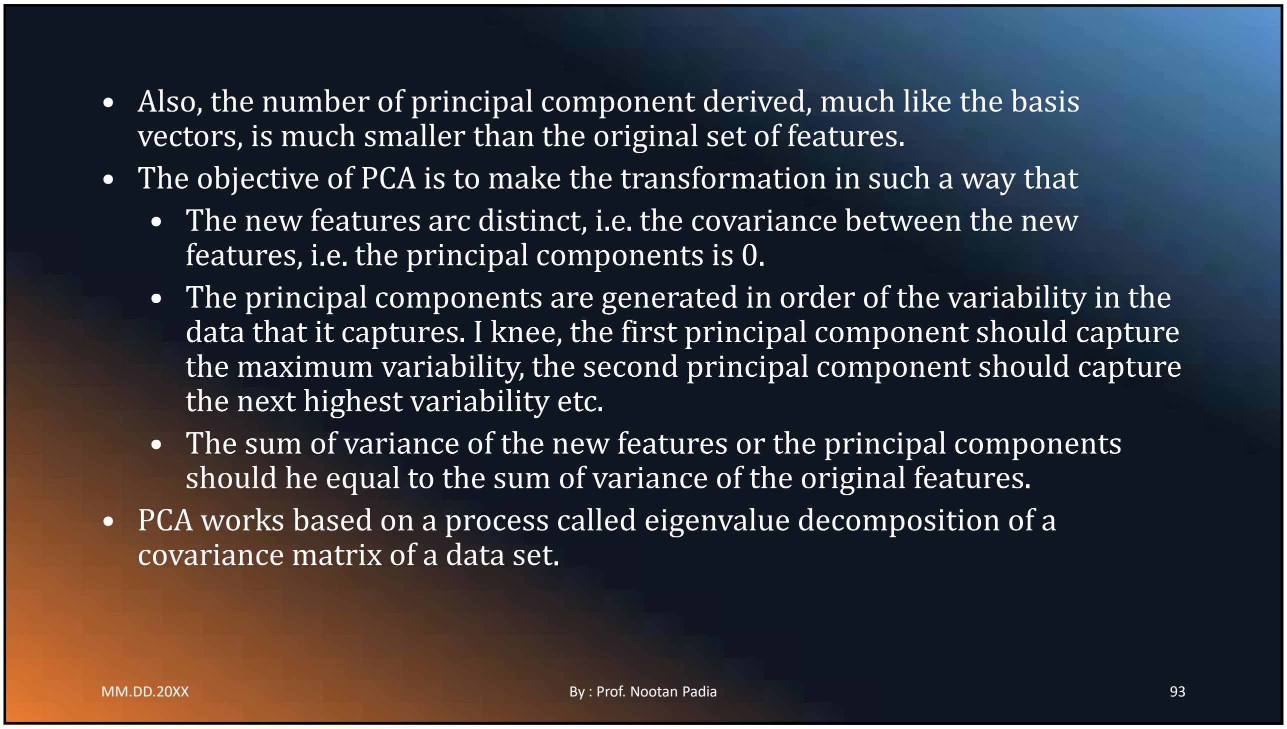
- A vector space is a set of vectors. Vector spaces have a property that they can be represented as a linear combination of a smaller set of vectors, called basis vectors.
- So, any vector 'v' in a vector space can be represented as

$$v = \sum_{i=1}^{n} a_i u_i$$

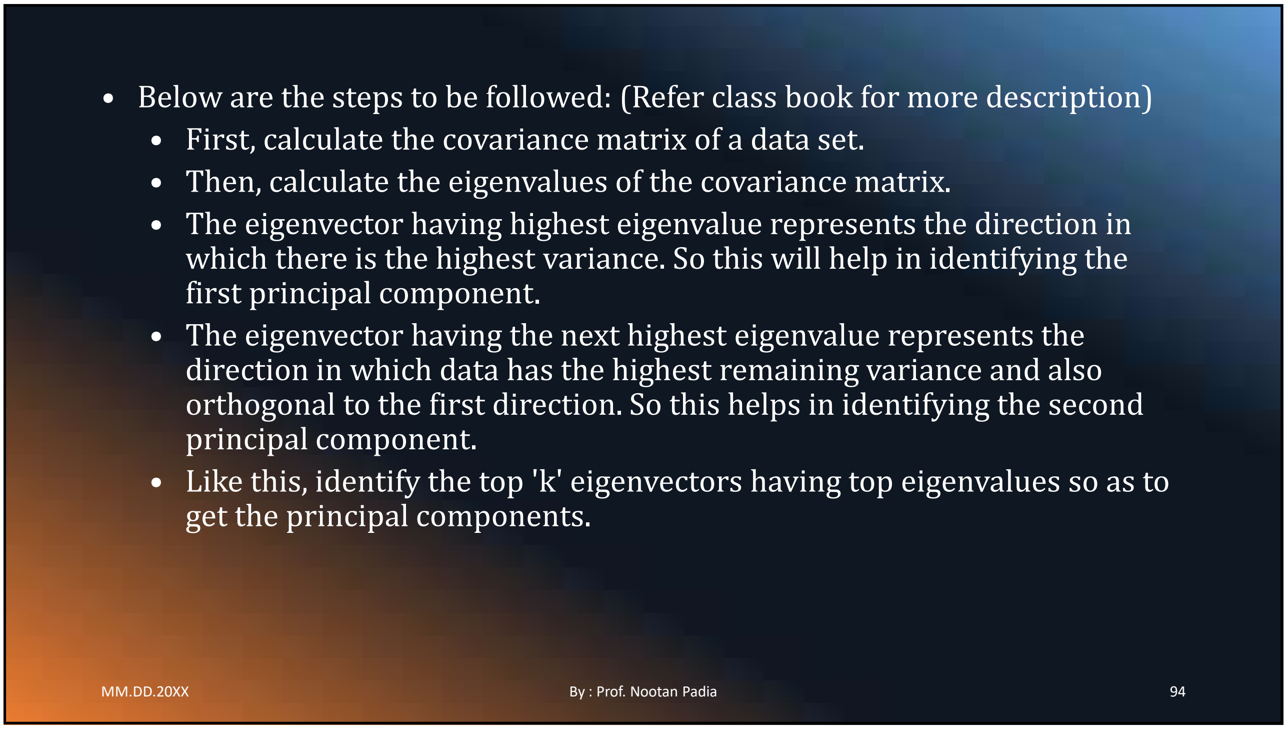
- where, a_i represents 'n' scalars and u_i represents the basis vectors.
- Basis vectors are orthogonal to each other.
- Orthogonality of vectors in n-dimensional vector space can be thought of an extension of the vectors being perpendicular in a two-dimensional vector space.
- Two orthogonal vectors are completely unrelated or independent of each other.



- So the transformation of a set of vectors to the corresponding set of basis vectors such that each vector in the original set can be expressed as a linear combination of basis vectors helps in decomposing the vectors to a number of independent components.
- Now, let's extend this notion to the feature space of a data set.
- The feature vector can be transformed to a vector space of the basis vectors which are termed as principal components.
- These principal components, just like the basis vectors, are orthogonal to each other.
- So a set of feature vectors which may have similarity each other is transformed to a set of principal components which are completely unrelated.
- However, the principal components capture the variability of the original feature space.

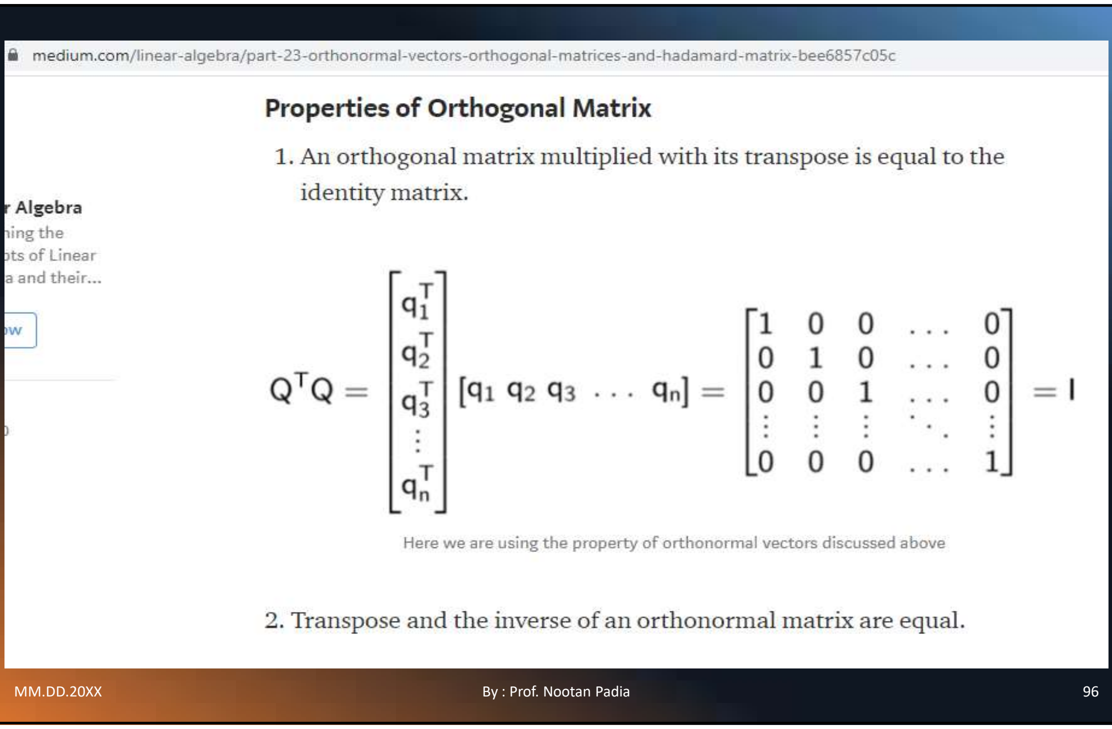


- Also, the number of principal component derived, much like the basis vectors, is much smaller than the original set of features.
- The objective of PCA is to make the transformation in such a way that
  - The new features arc distinct, i.e. the covariance between the new features, i.e. the principal components is 0.
  - The principal components are generated in order of the variability in the data that it captures. I knee, the first principal component should capture the maximum variability, the second principal component should capture the next highest variability etc.
  - The sum of variance of the new features or the principal components should he equal to the sum of variance of the original features.
- PCA works based on a process called eigenvalue decomposition of a covariance matrix of a data set.



- Below are the steps to be followed: (Refer class book for more description)
  - First, calculate the covariance matrix of a data set.
  - Then, calculate the eigenvalues of the covariance matrix.
  - The eigenvector having highest eigenvalue represents the direction in which there is the highest variance. So this will help in identifying the first principal component.
  - The eigenvector having the next highest eigenvalue represents the direction in which data has the highest remaining variance and also orthogonal to the first direction. So this helps in identifying the second principal component.
  - Like this, identify the top 'k' eigenvectors having top eigenvalues so as to get the principal components.

- **Singular value decomposition**
  - Singular value decomposition (SVD) is a matrix factorization technique commonly used in linear algebra.
  - SVD of a matrix A (m x n) is a factorization of the form:
    - A = U ∑V
    - where, U and V are orthonormal matrices, U is an m x m unitary matrix, V is an n x n unitary matrix and ∑ is an m x n rectangular diagonal matrix.
  - The diagonal entries of ∑ are known as singular values of matrix A.
  - The columns of U and V are called the left-singular and right-singular vectors of matrix A, respectively.
  - SVD is generally used in PCA, once the mean of each variable has been removed.


medium.com/linear-algebra/part-23-orthonormal-vectors-orthogonal-matrices-and-hadamard-matrix-bee6857c05c
r Algebra
hing the
pts of Linear
a and their...
ow

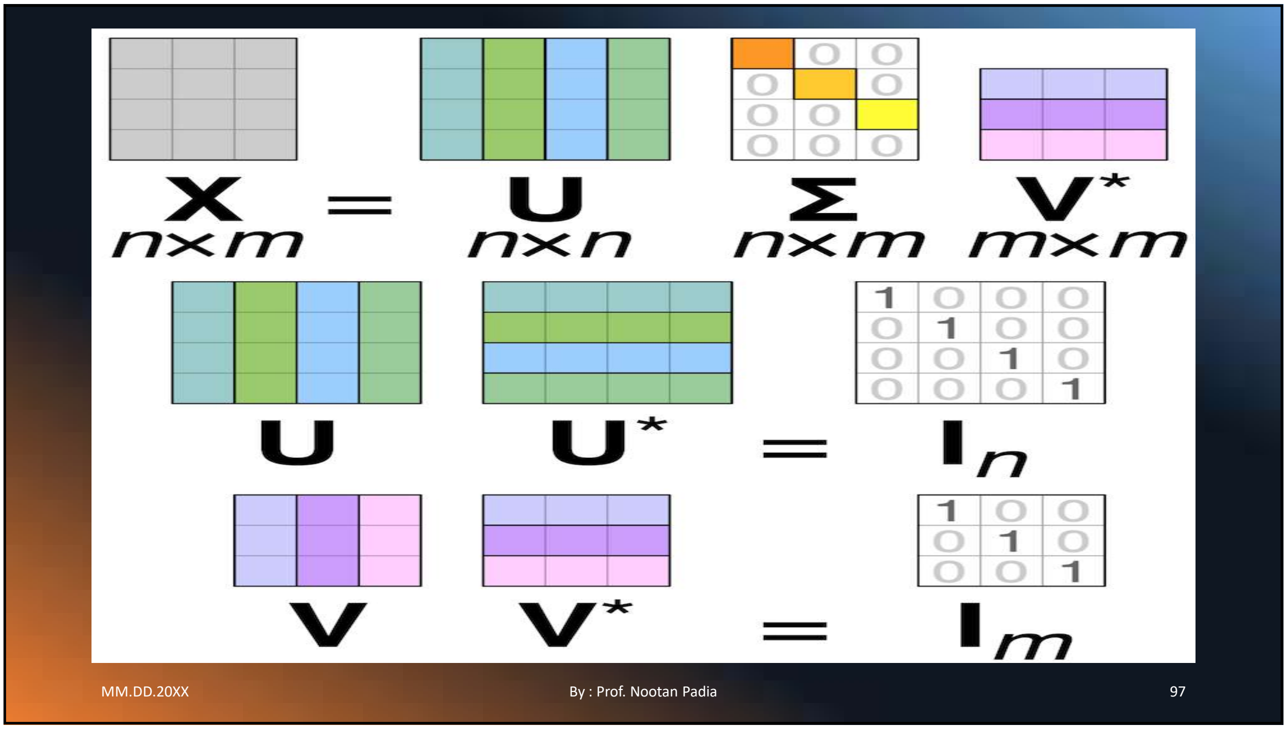
Properties of Orthogonal Matrix

1. An orthogonal matrix multiplied with its transpose is equal to the identity matrix.

$$Q^TQ = \begin{bmatrix} q_1^T \\ q_2^T \\ q_3^T \\ \vdots \\ q_n^T \end{bmatrix} [q_1\ q_2\ q_3\ \ldots\ q_n] = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix} = I$$

Here we are using the property of orthonormal vectors discussed above

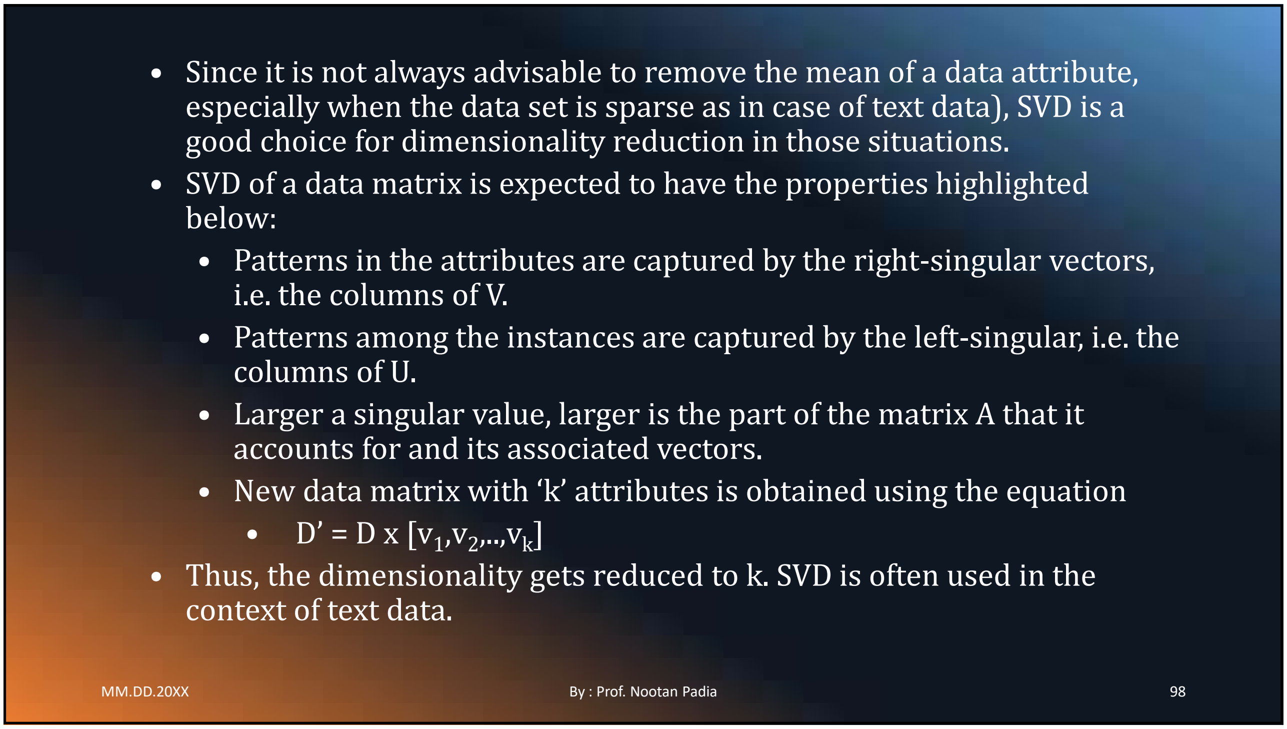
2. Transpose and the inverse of an orthonormal matrix are equal.



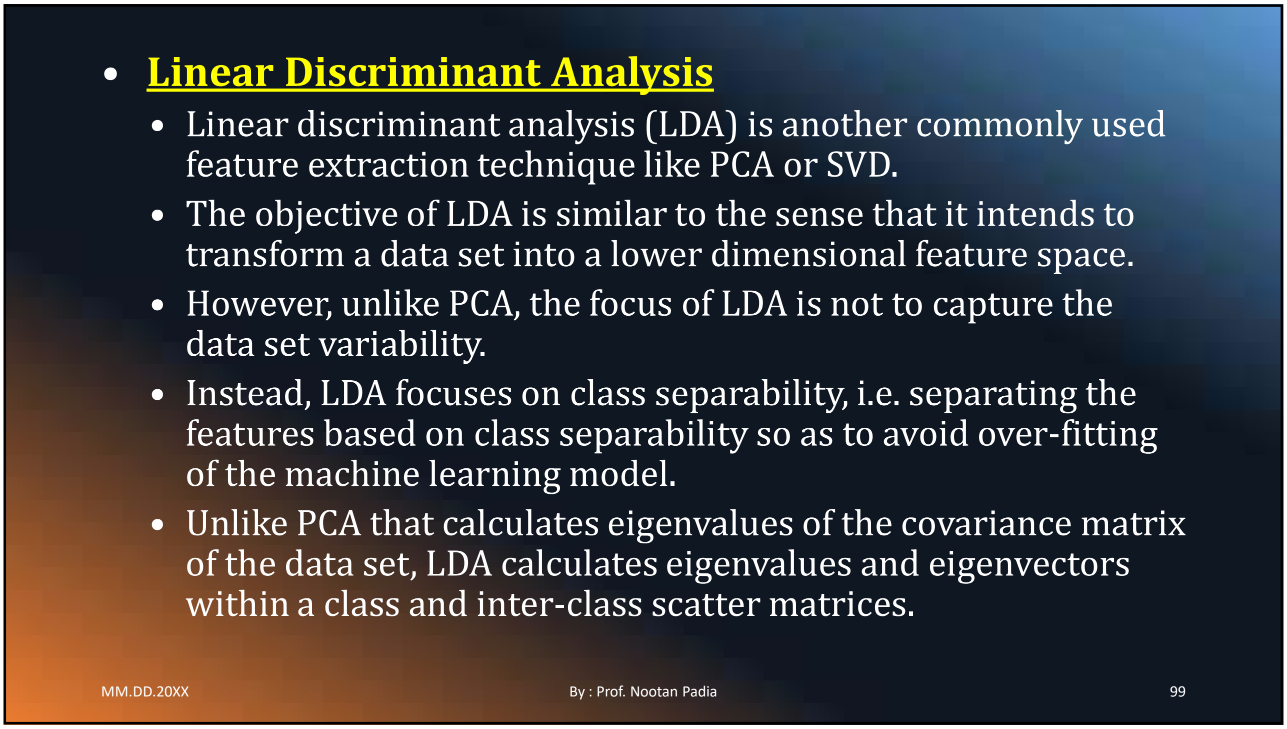
$$X_{n\times m} = U_{n\times n}\,\Sigma_{n\times m}\,V^*_{m\times m}$$

$$U\,U^* = I_n$$

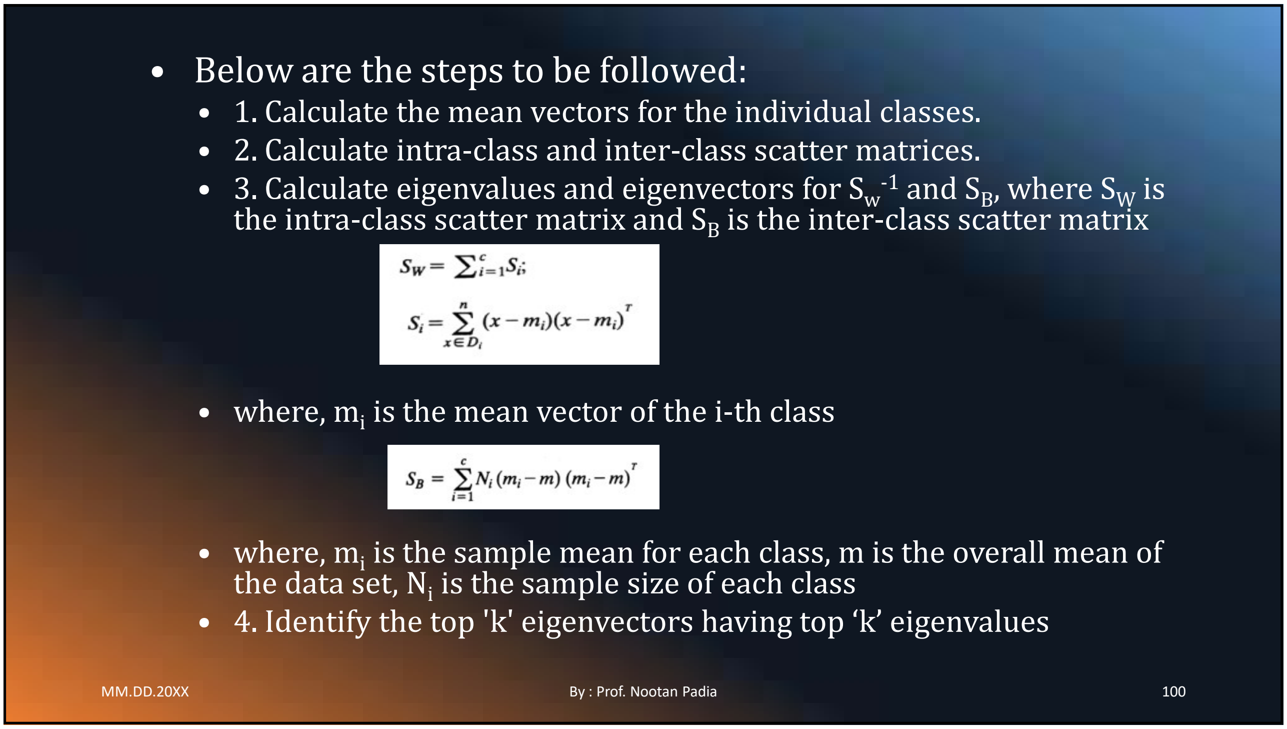
$$V\,V^* = I_m$$



- Since it is not always advisable to remove the mean of a data attribute, especially when the data set is sparse as in case of text data), SVD is a good choice for dimensionality reduction in those situations.
- SVD of a data matrix is expected to have the properties highlighted below:
  - Patterns in the attributes are captured by the right-singular vectors, i.e. the columns of V.
  - Patterns among the instances are captured by the left-singular, i.e. the columns of U.
  - Larger a singular value, larger is the part of the matrix A that it accounts for and its associated vectors.
  - New data matrix with ‘k’ attributes is obtained using the equation
    - D’ = D x [v_1,v_2,..,v_k]
- Thus, the dimensionality gets reduced to k. SVD is often used in the context of text data.



- **Linear Discriminant Analysis**
  - Linear discriminant analysis (LDA) is another commonly used feature extraction technique like PCA or SVD.
  - The objective of LDA is similar to the sense that it intends to transform a data set into a lower dimensional feature space.
  - However, unlike PCA, the focus of LDA is not to capture the data set variability.
  - Instead, LDA focuses on class separability, i.e. separating the features based on class separability so as to avoid over-fitting of the machine learning model.
  - Unlike PCA that calculates eigenvalues of the covariance matrix of the data set, LDA calculates eigenvalues and eigenvectors within a class and inter-class scatter matrices.



- Below are the steps to be followed:
  - 1. Calculate the mean vectors for the individual classes.
  - 2. Calculate intra-class and inter-class scatter matrices.
  - 3. Calculate eigenvalues and eigenvectors for S_W^{-1} and S_B, where S_W is the intra-class scatter matrix and S_B is the inter-class scatter matrix

$$S_W = \sum_{i=1}^{c} S_i;$$

$$S_i = \sum_{x \in D_i}^{n} (x - m_i)(x - m_i)^T$$

  - where, m_i is the mean vector of the i-th class

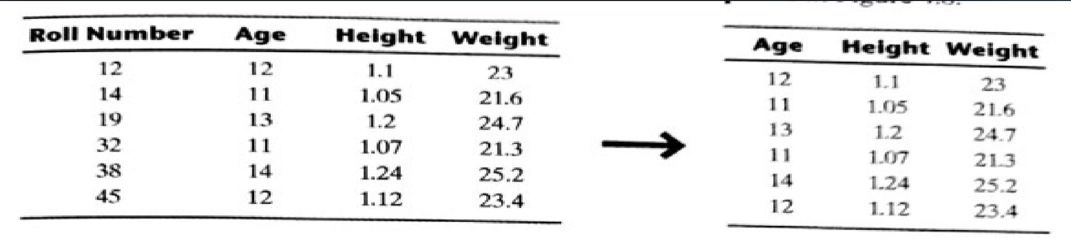
$$S_B = \sum_{i=1}^{c} N_i (m_i - m)(m_i - m)^T$$

  - where, m_i is the sample mean for each class, m is the overall mean of the data set, N_i is the sample size of each class
  - 4. Identify the top 'k' eigenvectors having top ‘k’ eigenvalues

Feature subset selection

- Feature selection is arguably the most critical pre-processing activity in any machine learning project.
- It intends to select a subset of system attributes or features which makes a most meaningful contribution in a machine learning activity.
- Let's quickly discuss a practical example to understand the philosophy behind feature selection.
- Say we are trying to predict the weight of students based on past information about similar students, which is captured in a 'student weight' data set.
- The student weight data set has features such as Roll Number, Age, Height, and Weight.
- We can well understand that roll number can have no bearing, whatsoever, in predicting student weight.

- So we can eliminate the feature roll number and build a feature subset to be considered in this machine learning problem.
- The subset of features is expected to give better results than the full set. The same has been depicted in Figure 4.8.

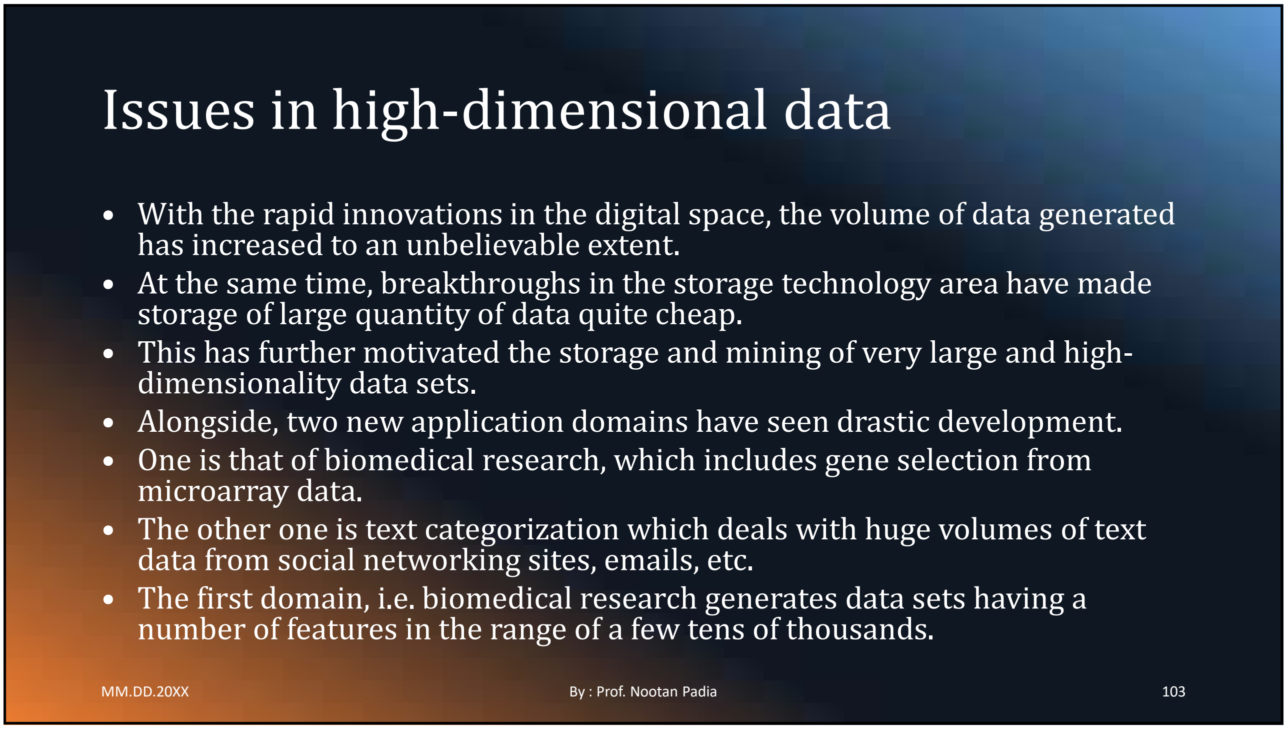


Roll Number	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3
38	14	1.24	25.2
45	12	1.12	23.4

→

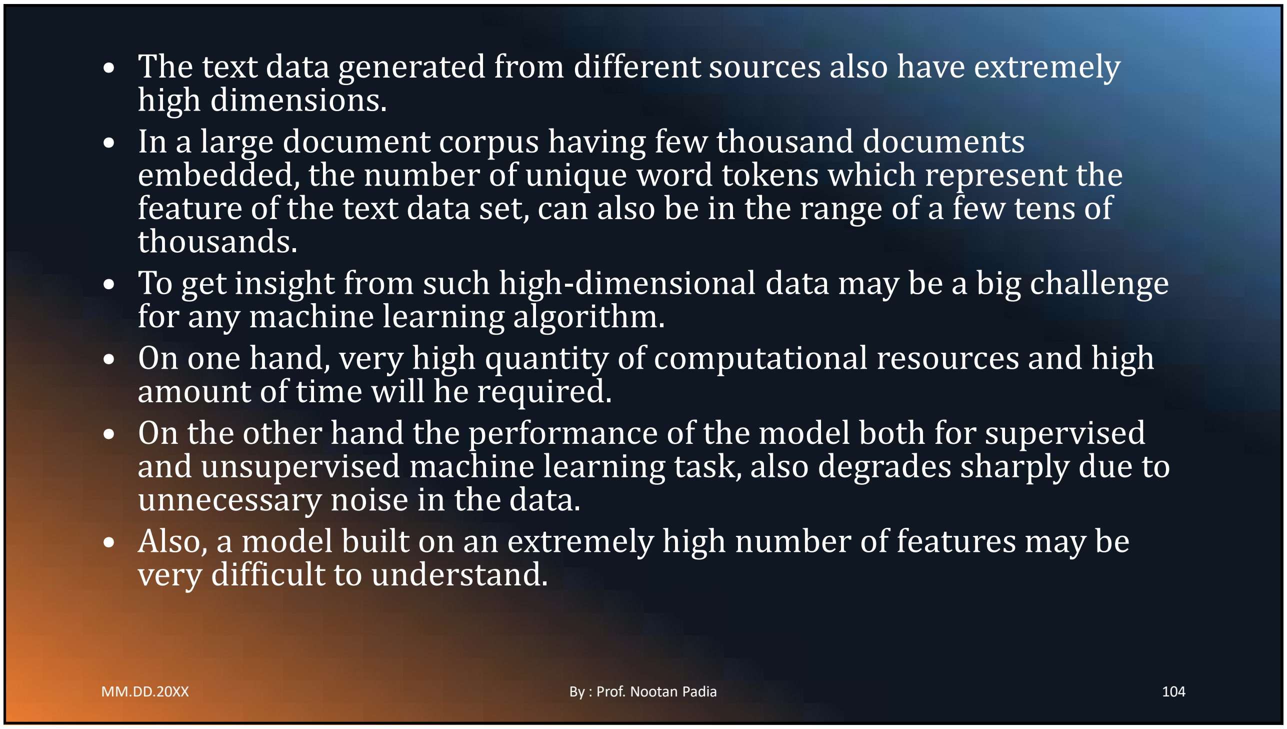
Age	Height	Weight
12	1.1	23
11	1.05	21.6
13	1.2	24.7
11	1.07	21.3
14	1.24	25.2
12	1.12	23.4

FIG. 4.8
Feature selection

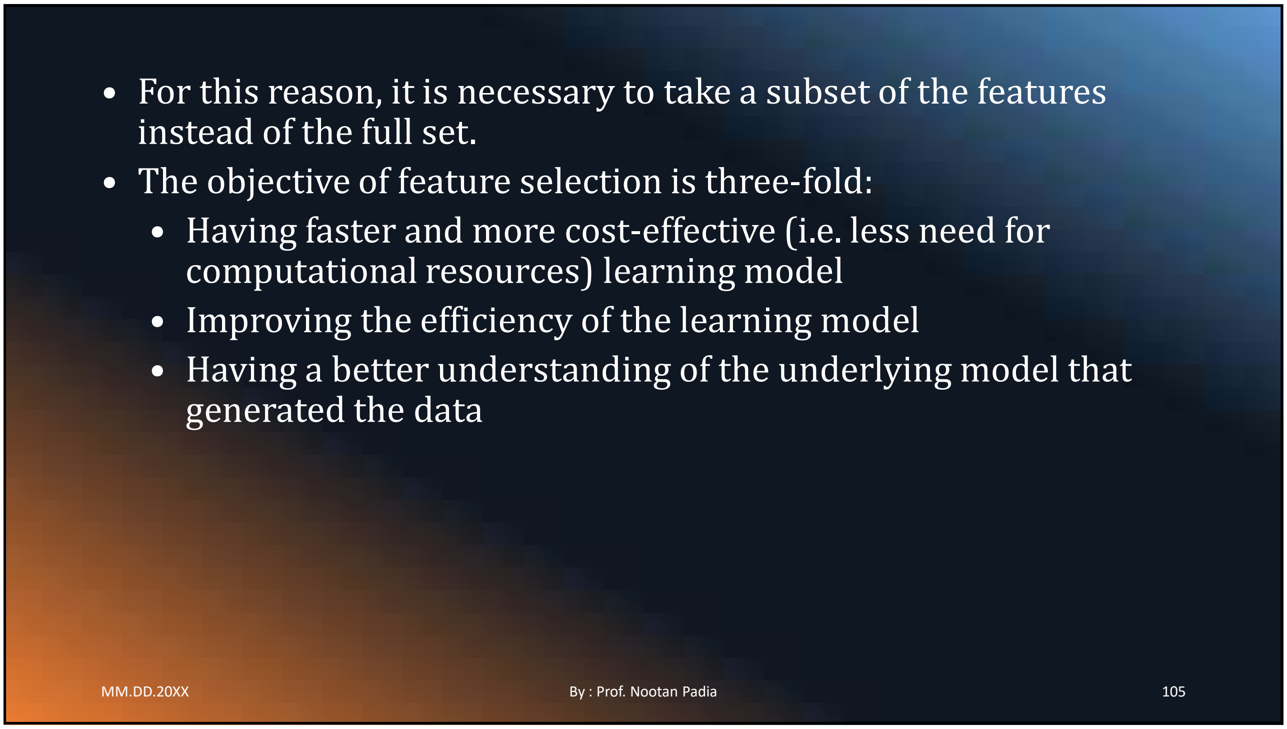


Issues in high-dimensional data

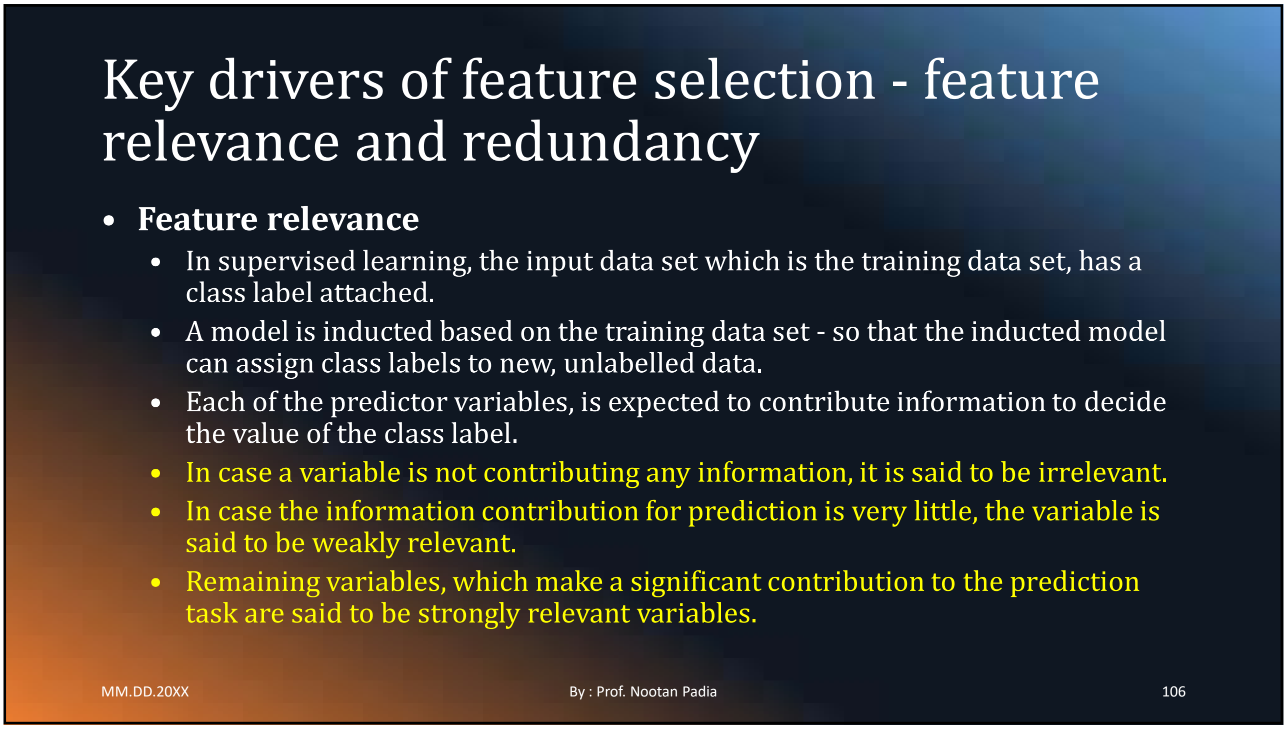
- With the rapid innovations in the digital space, the volume of data generated has increased to an unbelievable extent.
- At the same time, breakthroughs in the storage technology area have made storage of large quantity of data quite cheap.
- This has further motivated the storage and mining of very large and high-dimensionality data sets.
- Alongside, two new application domains have seen drastic development.
- One is that of biomedical research, which includes gene selection from microarray data.
- The other one is text categorization which deals with huge volumes of text data from social networking sites, emails, etc.
- The first domain, i.e. biomedical research generates data sets having a number of features in the range of a few tens of thousands.



- The text data generated from different sources also have extremely high dimensions.
- In a large document corpus having few thousand documents embedded, the number of unique word tokens which represent the feature of the text data set, can also be in the range of a few tens of thousands.
- To get insight from such high-dimensional data may be a big challenge for any machine learning algorithm.
- On one hand, very high quantity of computational resources and high amount of time will he required.
- On the other hand the performance of the model both for supervised and unsupervised machine learning task, also degrades sharply due to unnecessary noise in the data.
- Also, a model built on an extremely high number of features may be very difficult to understand.

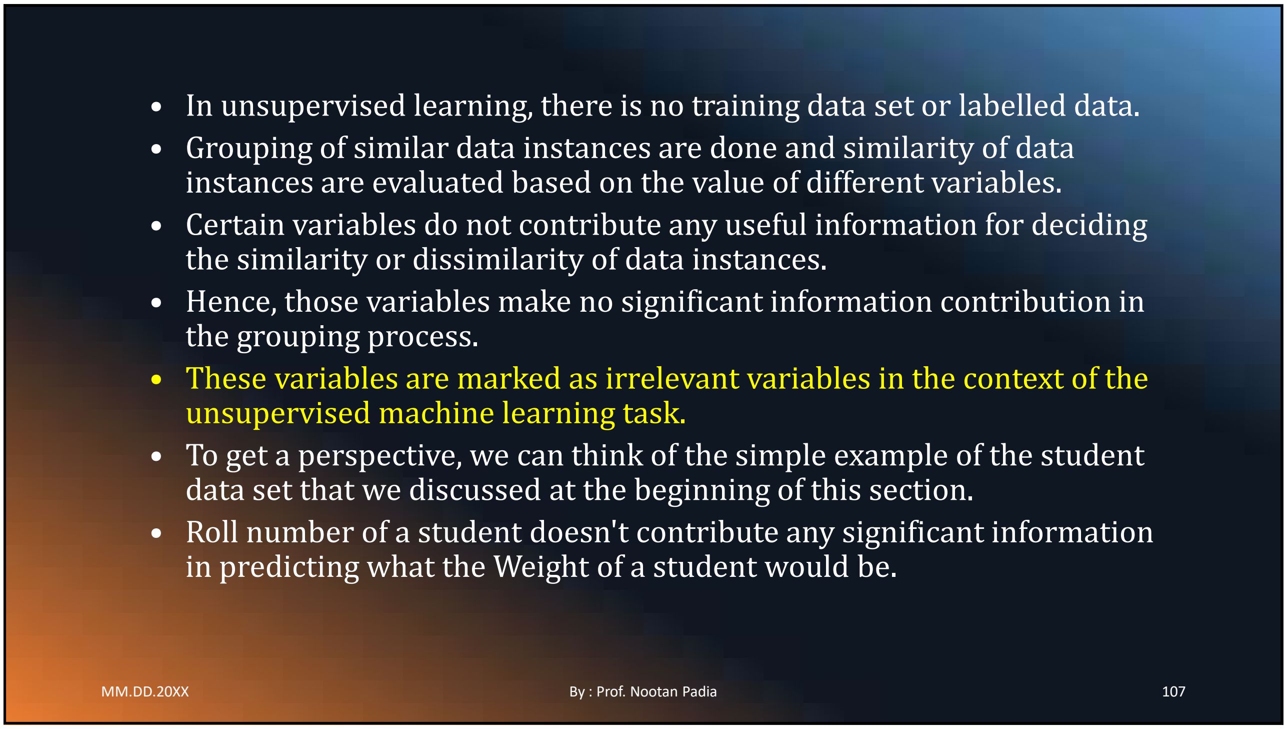


- For this reason, it is necessary to take a subset of the features instead of the full set.
- The objective of feature selection is three-fold:
  - Having faster and more cost-effective (i.e. less need for computational resources) learning model
  - Improving the efficiency of the learning model
  - Having a better understanding of the underlying model that generated the data

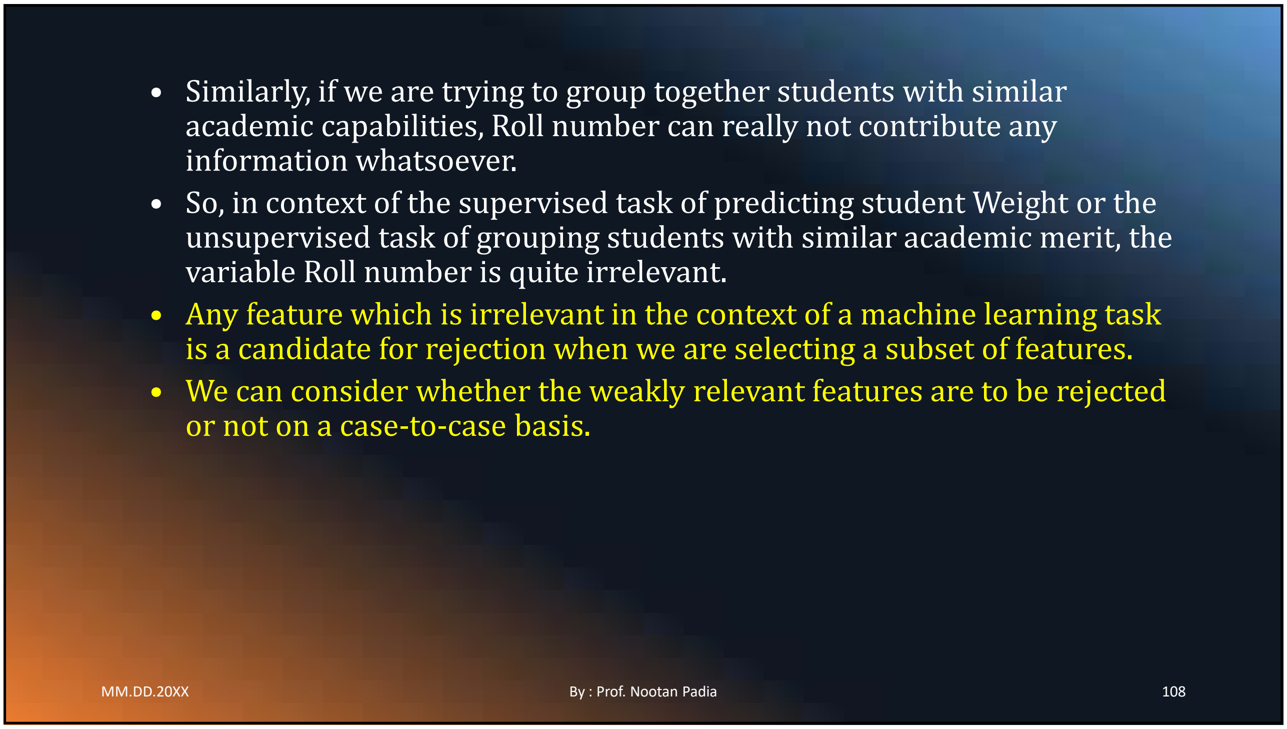


Key drivers of feature selection - feature relevance and redundancy

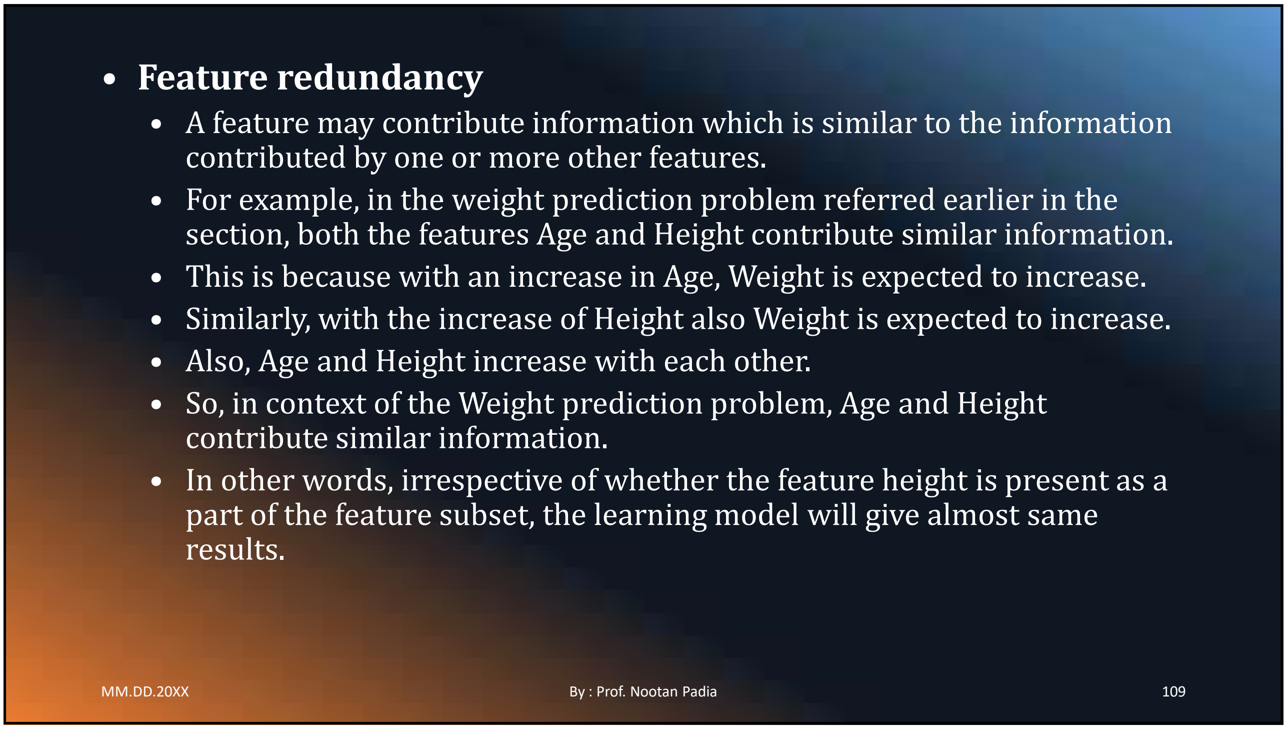
- **Feature relevance**
  - In supervised learning, the input data set which is the training data set, has a class label attached.
  - A model is inducted based on the training data set - so that the inducted model can assign class labels to new, unlabelled data.
  - Each of the predictor variables, is expected to contribute information to decide the value of the class label.
  - In case a variable is not contributing any information, it is said to be irrelevant.
  - In case the information contribution for prediction is very little, the variable is said to be weakly relevant.
  - Remaining variables, which make a significant contribution to the prediction task are said to be strongly relevant variables.



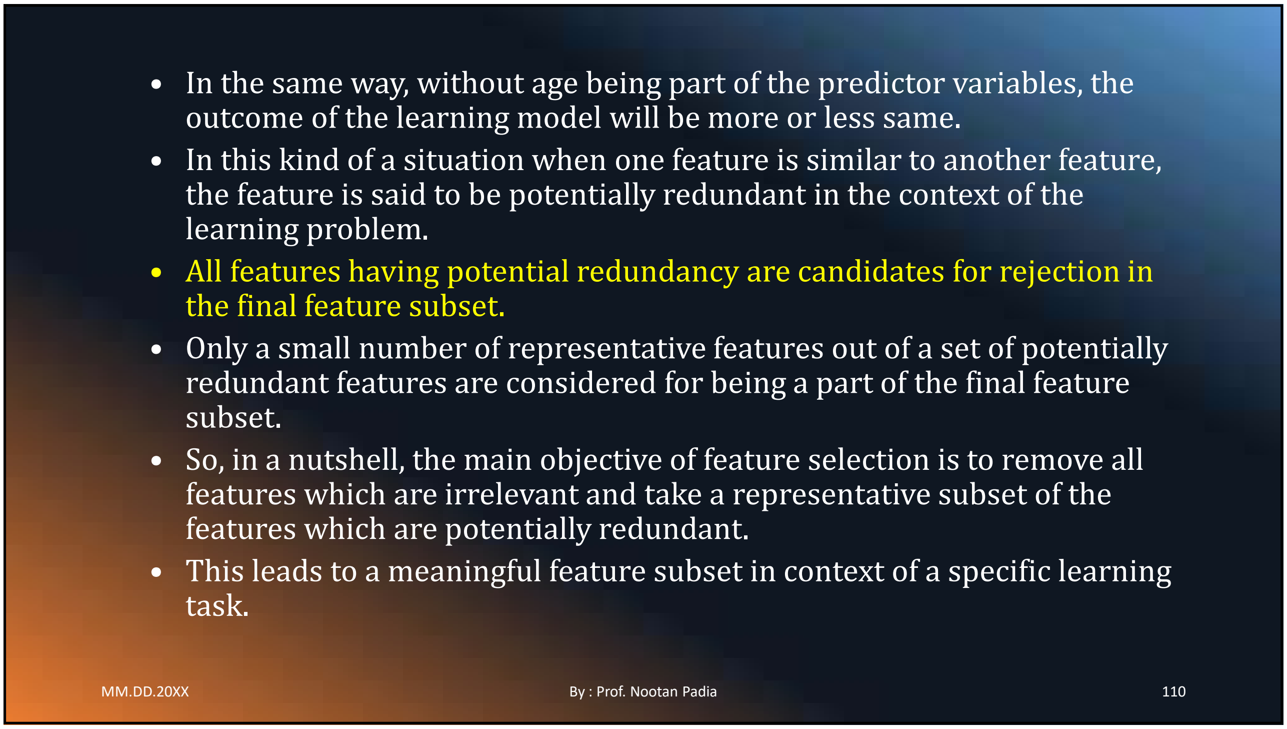
- In unsupervised learning, there is no training data set or labelled data.
- Grouping of similar data instances are done and similarity of data instances are evaluated based on the value of different variables.
- Certain variables do not contribute any useful information for deciding the similarity or dissimilarity of data instances.
- Hence, those variables make no significant information contribution in the grouping process.
- These variables are marked as irrelevant variables in the context of the unsupervised machine learning task.
- To get a perspective, we can think of the simple example of the student data set that we discussed at the beginning of this section.
- Roll number of a student doesn't contribute any significant information in predicting what the Weight of a student would be.



- Similarly, if we are trying to group together students with similar academic capabilities, Roll number can really not contribute any information whatsoever.
- So, in context of the supervised task of predicting student Weight or the unsupervised task of grouping students with similar academic merit, the variable Roll number is quite irrelevant.
- Any feature which is irrelevant in the context of a machine learning task is a candidate for rejection when we are selecting a subset of features.
- We can consider whether the weakly relevant features are to be rejected or not on a case-to-case basis.



- **Feature redundancy**
  - A feature may contribute information which is similar to the information contributed by one or more other features.
  - For example, in the weight prediction problem referred earlier in the section, both the features Age and Height contribute similar information.
  - This is because with an increase in Age, Weight is expected to increase.
  - Similarly, with the increase of Height also Weight is expected to increase.
  - Also, Age and Height increase with each other.
  - So, in context of the Weight prediction problem, Age and Height contribute similar information.
  - In other words, irrespective of whether the feature height is present as a part of the feature subset, the learning model will give almost same results.



- In the same way, without age being part of the predictor variables, the outcome of the learning model will be more or less same.
- In this kind of a situation when one feature is similar to another feature, the feature is said to be potentially redundant in the context of the learning problem.
- All features having potential redundancy are candidates for rejection in the final feature subset.
- Only a small number of representative features out of a set of potentially redundant features are considered for being a part of the final feature subset.
- So, in a nutshell, the main objective of feature selection is to remove all features which are irrelevant and take a representative subset of the features which are potentially redundant.
- This leads to a meaningful feature subset in context of a specific learning task.

- **Measures of feature relevance**
  - As mentioned earlier, feature relevance is to be gauged by the amount of information contributed by a feature.
  - For supervised learning, mutual information is considered as a good measure of information contribution of a feature to decide the value of the class label.
  - That’s why it is a good indicator of the relevance of a feature with respect to the class variable.
  - Higher the value of mutual information of a feature, more relevant is that feature. Mutual information can be calculated as follows:

$$MI(C,f) = H(C) + H(f) - H(C,f)$$

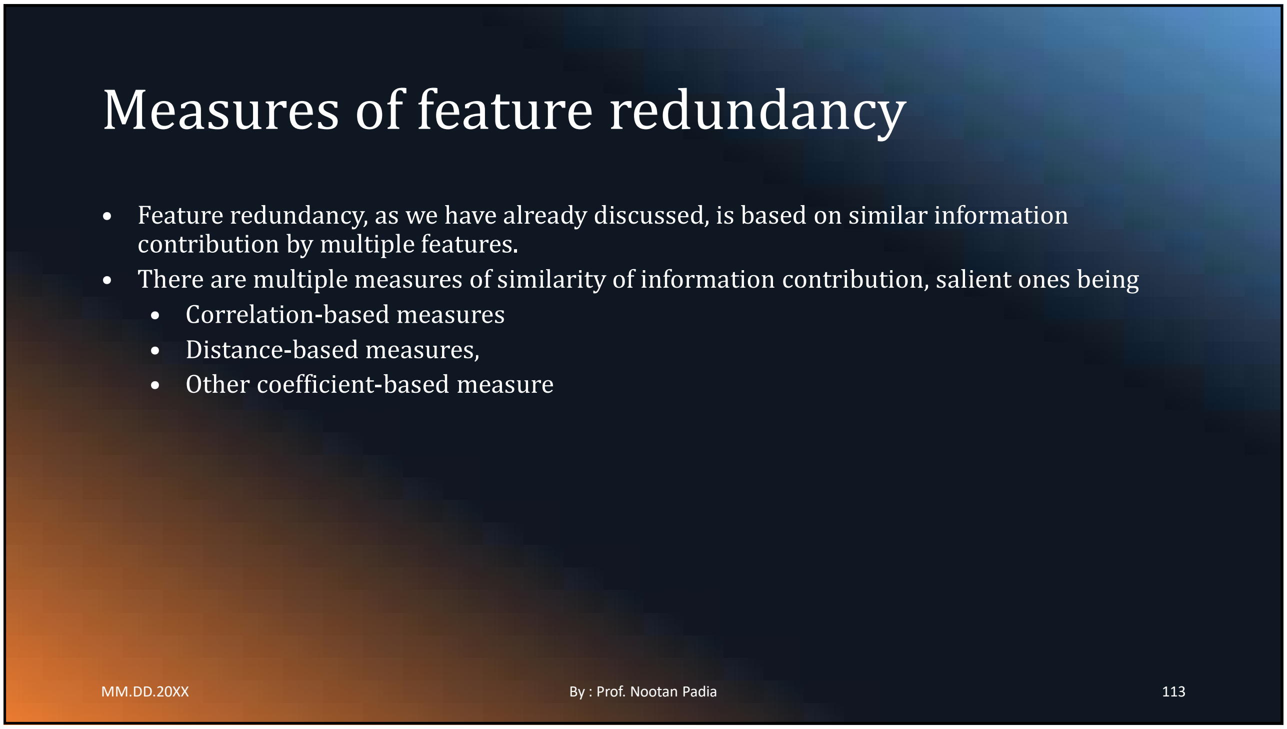
where, marginal entropy of the class, $H(C) = -\sum_{i=1}^{k} p(C_i)\log_2 p(C_i)$

marginal entropy of the feature ‘x’, $H(f) = -\sum_{c} p(f = x)\log_2 p(f = x)$

- K = number of classes, C = class variable, f= feature set that take discrete values.
- In case of unsupervised learning, there is no class variable.
- Hence, feature-to-class mutual information cannot be used to measure the information contribution of the features.
- In case of unsupervised learning, the entropy of the set of features without one feature at a time is calculated for all the features.
- Then, the features are ranked in a descending order of information gain from a feature and top ‘p’ percentage (value of β is a design parameter of the algorithm) of features are Selected as relevant features.
- The entropy of a feature f is calculated using Shannon’s formula below:

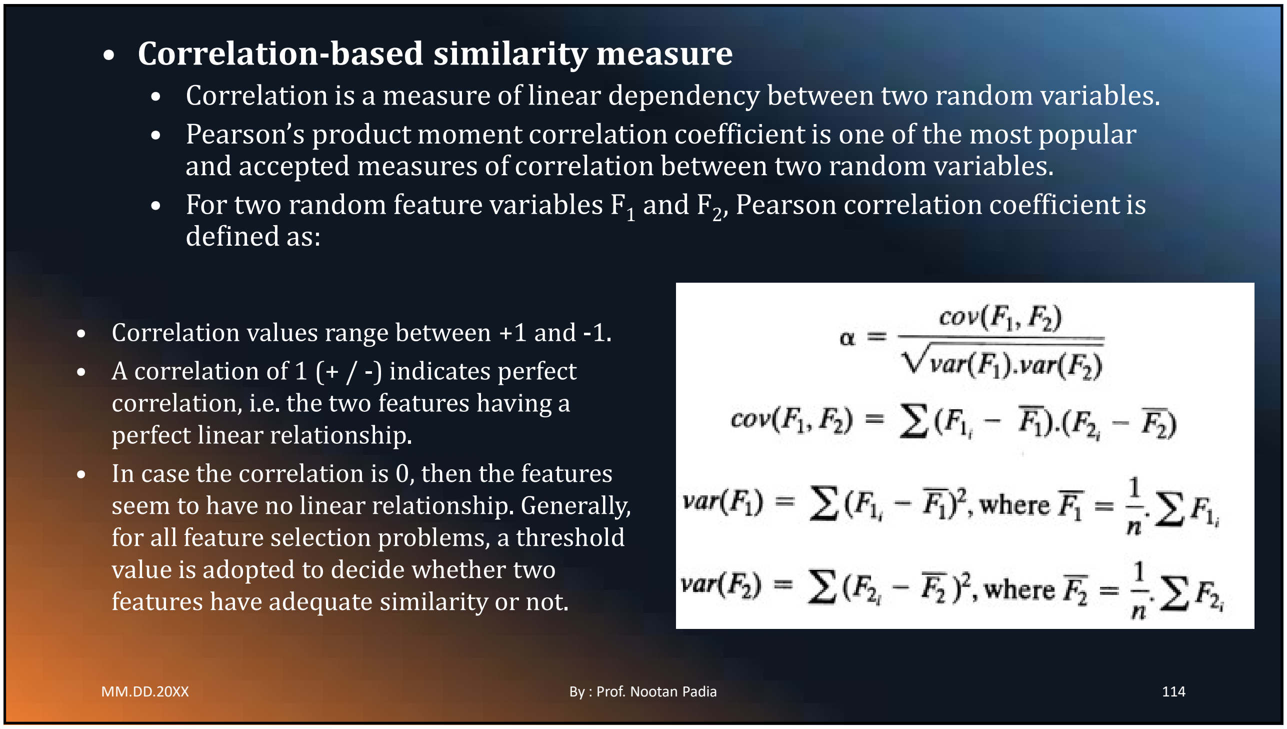
$$H(f) = -\sum_{x} p(f = x)\log_2 p(f = x)$$

  - $\sum_x$ is used only for features that take discrete values.
  - For continuous features, it should be replaced by discretization performed first to estimate probabilities *p(f=x)*



Measures of feature redundancy

- Feature redundancy, as we have already discussed, is based on similar information contribution by multiple features.
- There are multiple measures of similarity of information contribution, salient ones being
  - Correlation-based measures
  - Distance-based measures,
  - Other coefficient-based measure



- **Correlation-based similarity measure**
  - Correlation is a measure of linear dependency between two random variables.
  - Pearson’s product moment correlation coefficient is one of the most popular and accepted measures of correlation between two random variables.
  - For two random feature variables F_1 and F_2, Pearson correlation coefficient is defined as:

- Correlation values range between +1 and -1.
- A correlation of 1 (+ / -) indicates perfect correlation, i.e. the two features having a perfect linear relationship.
- In case the correlation is 0, then the features seem to have no linear relationship. Generally, for all feature selection problems, a threshold value is adopted to decide whether two features have adequate similarity or not.

$$\alpha = \frac{cov(F_1, F_2)}{\sqrt{var(F_1).var(F_2)}}$$

$$cov(F_1, F_2) = \sum (F_{1_i} - \overline{F_1}).(F_{2_i} - \overline{F_2})$$

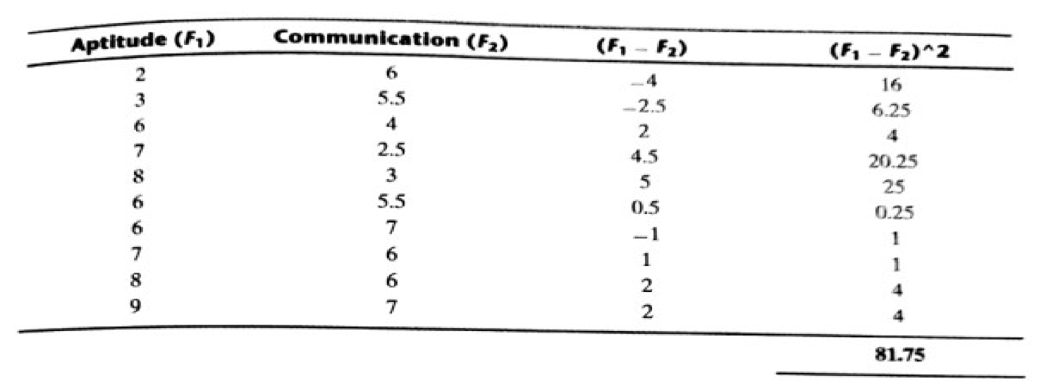
$$var(F_1) = \sum (F_{1_i} - \overline{F_1})^2, \text{where } \overline{F_1} = \frac{1}{n}.\sum F_{1_i}$$

$$var(F_2) = \sum (F_{2_i} - \overline{F_2})^2, \text{where } \overline{F_2} = \frac{1}{n}.\sum F_{2_i}$$

- **Distance-based similarity measure**
  - The most common distance measure is the Euclidean distance, which, between two features F_1 and F_2 are calculated as:

$$d(F_1, F_2) = \sqrt{\sum_{i=1}^{n} (F_{1_i} - F_{2_i})^2}$$

  - where F_1 and F_2- are features of an n-dimensional data set.
  - Refer to the Figure in next slide.
  - The data set has two features, aptitude (F_1) and communication (F_2) under consideration.
  - The Euclidean distance between the features has been calculated using the formula provided above.



Aptitude (F_1)	Communication (F_2)	($F_1 - F_2$)	($F_1 - F_2$)^2
2	6	−4	16
3	5.5	−2.5	6.25
6	4	2	4
7	2.5	4.5	20.25
8	3	5	25
6	5.5	0.5	0.25
6	7	−1	1
7	6	1	1
8	6	2	4
9	7	2	4
			81.75

Distance calculation between features

Sqrt(81.75)

- A more generalized form of the Euclidean distance is the Minkowski distance, measured as

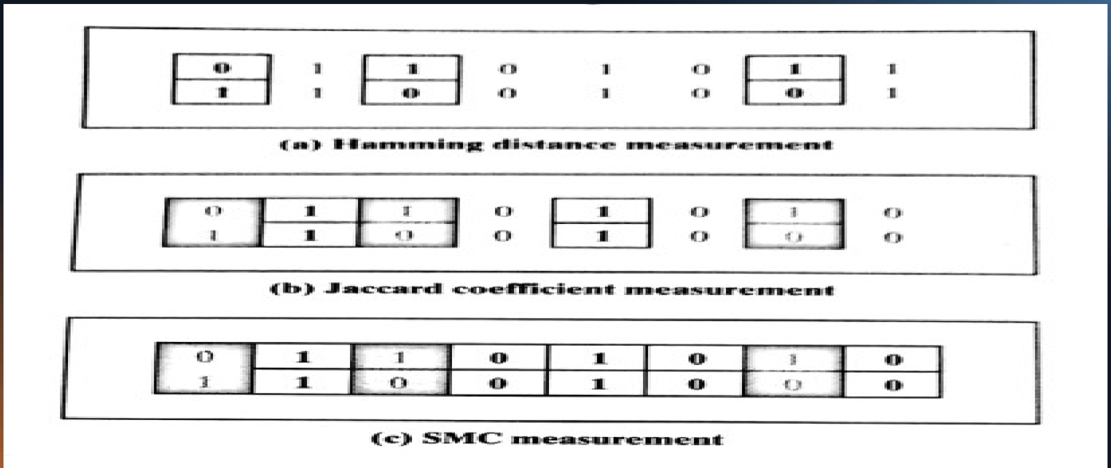
$$d(F_1, F_2) = \sqrt[r]{\sum_{i=1}^{n} (F_{1_i} - F_{2_i})^r}$$

- Minkowski distance takes the form of Euclidean distance (also called L2 norm) when r = 2.
- At r = 1, it takes the form of Manhattan distance (also called L1 norm), as shown below:

$$d(F_1, F_2) = \sum_{i=1}^{n} |F_{1_i} - F_{2_i}|$$

- A specific example of Manhattan distance, used more frequently to calculate the distance between binary vectors is the Hamming distance.

- For example, the Hamming distance between two vectors 01101011 and 11001001 is 3, as illustrated in Figure a.



(a) Hamming distance measurement

(b) Jaccard coefficient measurement

(c) SMC measurement

Distance measures between features

- **Other similarity measures**
  - Jaccard index/coefficient is used as a measure of similarity between two features.
  - The Jaccard distance, a measure of dissimilarity between two features, is complementary of Jaccard index.
  - For two features having binary values, Jaccard index is measured as

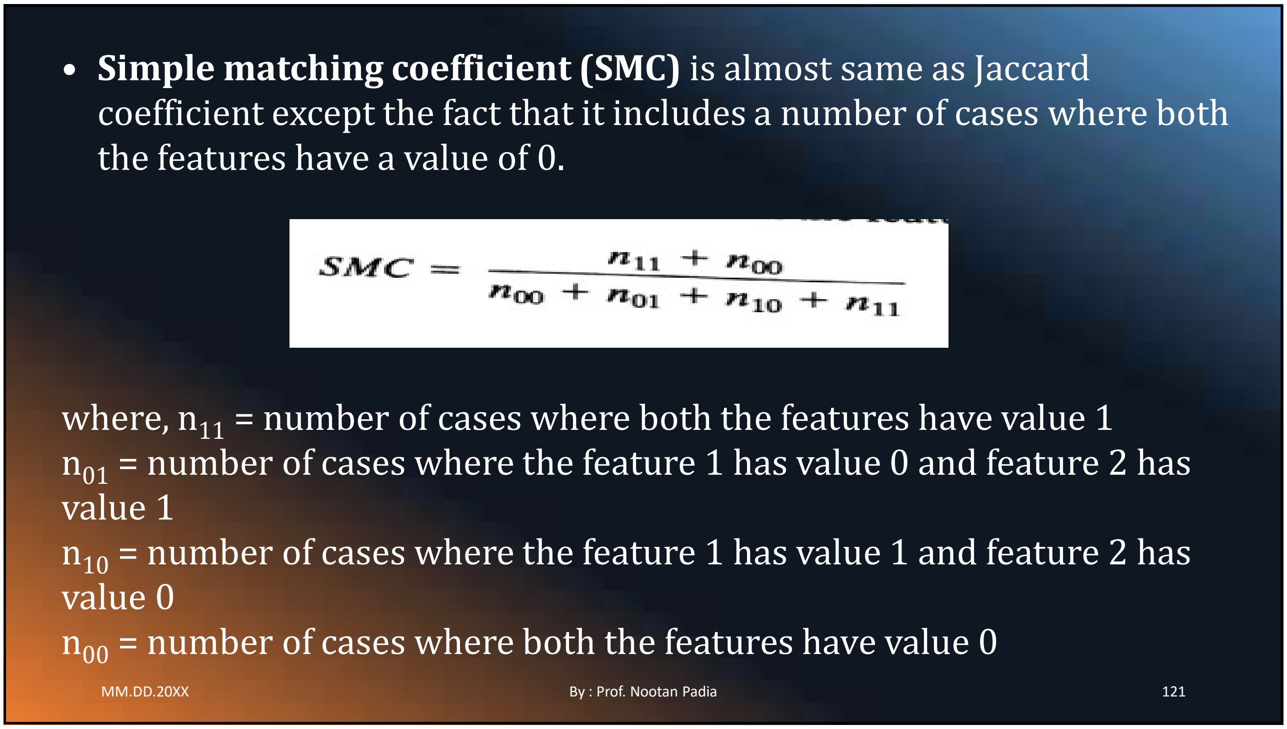
$$J = \frac{n_{11}}{n_{01} + n_{10} + n_{11}}$$

where, n_{11} = number of cases where both the features have value 1
n_{01} = number of cases where the feature 1 has value 0 and feature 2 has value 1
n_{10} = number of cases where the feature 1 has value 1 and feature 2 has value 0
Jaccard distance, d_j =1- J

- Let's consider two features F1 and F2 having values (0, 1, 1, 0, 1, 0, 1, 0) and (1, 1, 0, 0, 1, 0, 0, 0).
- Figure 4.10b shows the identification of the values of n_{11}, n_{01} and n_{10}.
- As shown, the cases where both the values are 0 have been left out without border — as an indication of the fact that they will be excluded in the calculation of Jaccard coefficient.
- Jaccard coefficient of F1 and F2,

$$J = \frac{n_{11}}{n_{01} + n_{10} + n_{11}} = \frac{2}{1 + 2 + 2} = \frac{2}{5} \text{ or } 0.4.$$

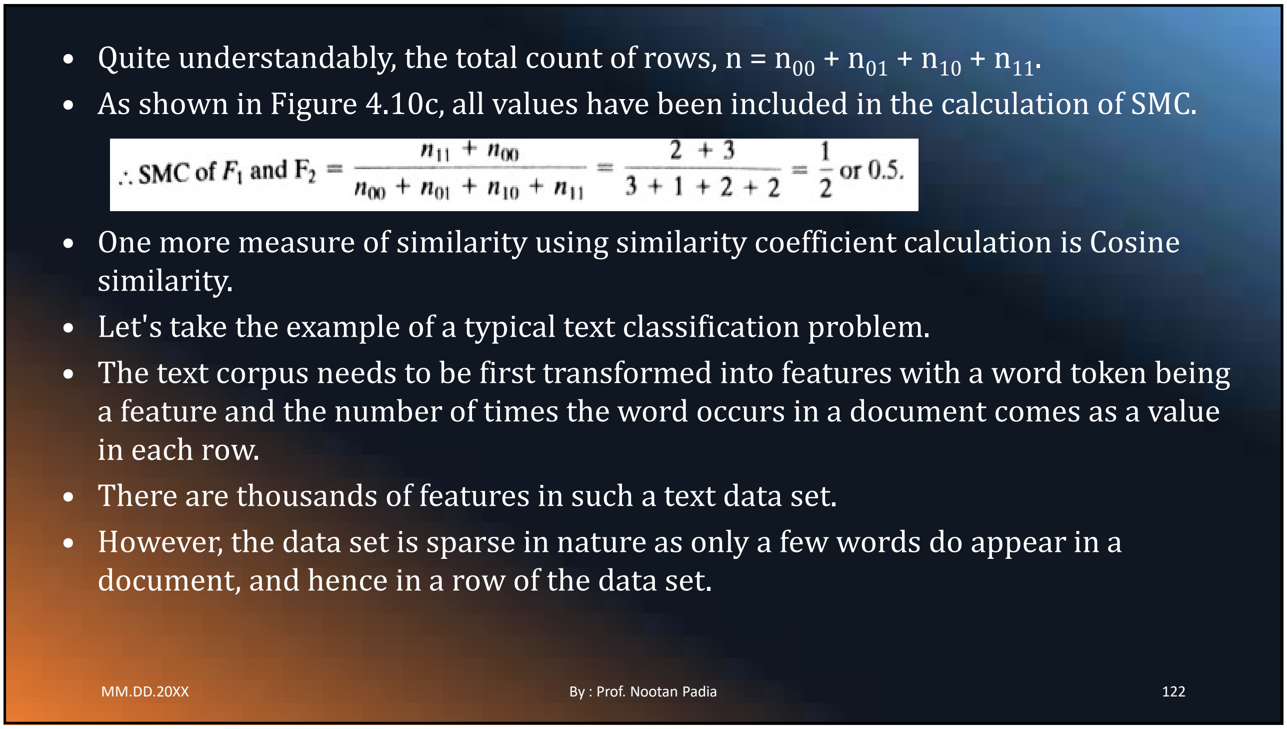
- Jaccard distance between F1 and F2, d_j = 1 — J = 1/2 or 0.6.



- **Simple matching coefficient (SMC)** is almost same as Jaccard coefficient except the fact that it includes a number of cases where both the features have a value of 0.

$$SMC = \frac{n_{11} + n_{00}}{n_{00} + n_{01} + n_{10} + n_{11}}$$

where, n_{11} = number of cases where both the features have value 1
n_{01} = number of cases where the feature 1 has value 0 and feature 2 has value 1
n_{10} = number of cases where the feature 1 has value 1 and feature 2 has value 0
n_{00} = number of cases where both the features have value 0



- Quite understandably, the total count of rows, $n = n_{00} + n_{01} + n_{10} + n_{11}$.
- As shown in Figure 4.10c, all values have been included in the calculation of SMC.

$$\therefore \text{SMC of } F_1 \text{ and } F_2 = \frac{n_{11} + n_{00}}{n_{00} + n_{01} + n_{10} + n_{11}} = \frac{2 + 3}{3 + 1 + 2 + 2} = \frac{1}{2} \text{ or } 0.5.$$

- One more measure of similarity using similarity coefficient calculation is Cosine similarity.
- Let's take the example of a typical text classification problem.
- The text corpus needs to be first transformed into features with a word token being a feature and the number of times the word occurs in a document comes as a value in each row.
- There are thousands of features in such a text data set.
- However, the data set is sparse in nature as only a few words do appear in a document, and hence in a row of the data set.

- So each row has very few non-zero values.
- However, the non-zero values can be anything integer value as the same word may occur any number of times.
- Also. considering the sparsity of the data set, the 0-0 matches (which obviously is going to be pretty high) need to be ignored.

$$cos\ (x,\ y) = \frac{x.y}{\|x\|.\|y\|}$$

$$\text{where, } x.y = \text{vector dot product of } x \text{ and } y = \sum\nolimits_{i=1}^{n} x_i y_i$$

$$\|x\| = \sqrt{\sum\nolimits_{i=1}^{n} x_i^2} \text{ and } \|y\| = \sqrt{\sum\nolimits_{i=1}^{n} y_i^2}$$

- Let's calculate the cosine similarity of x and y, where x = (2, 4, 0, 0, 2, 1, 3, 0, 0) and y = (2, 1, 0, 0, 3, 2, 1, 0, 1).

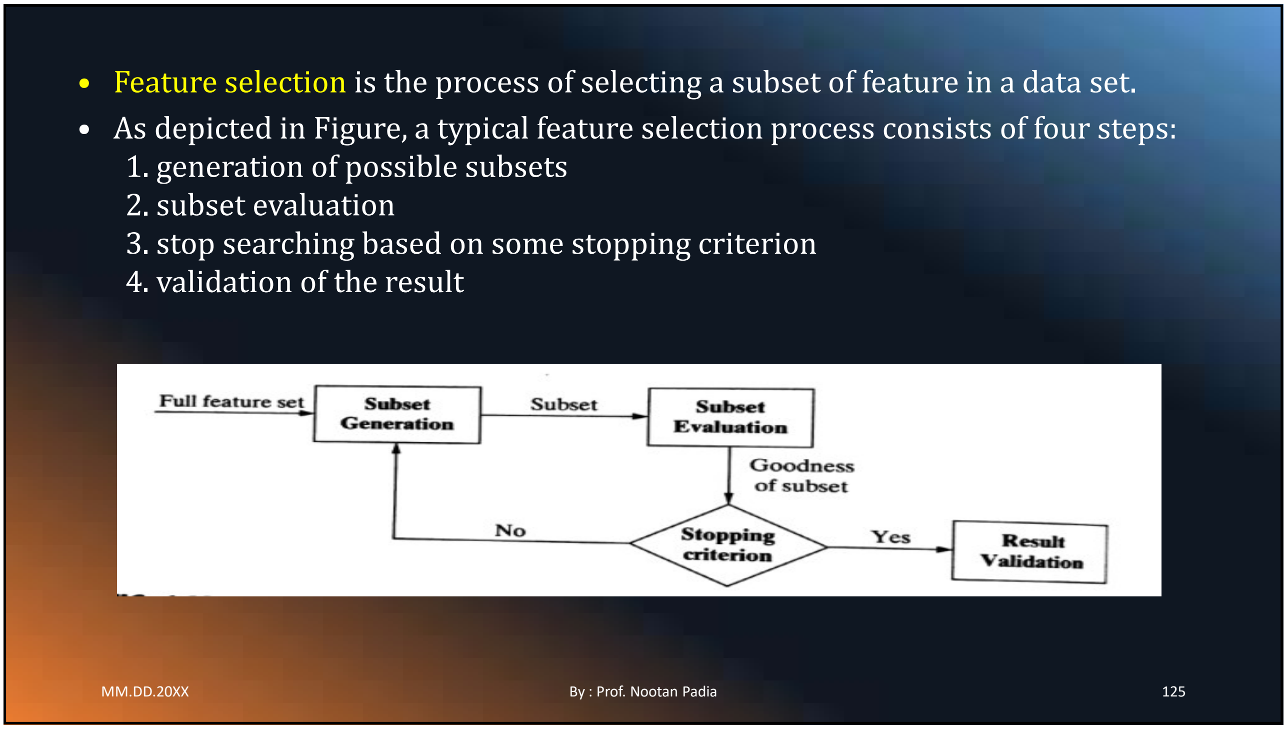
- In this case, x.y = 2*2 + 4*1 + 0*0 + 0*0 + 2*3 + 1*2 + 3*1 + 0*0 + 0*1 = 19

$$\|x\| = \sqrt{2^2 + 4^2 + 0^2 + 0^2 + 2^2 + 1^2 + 3^2 + 0^2 + 0^2} = \sqrt{34} = 5.83$$

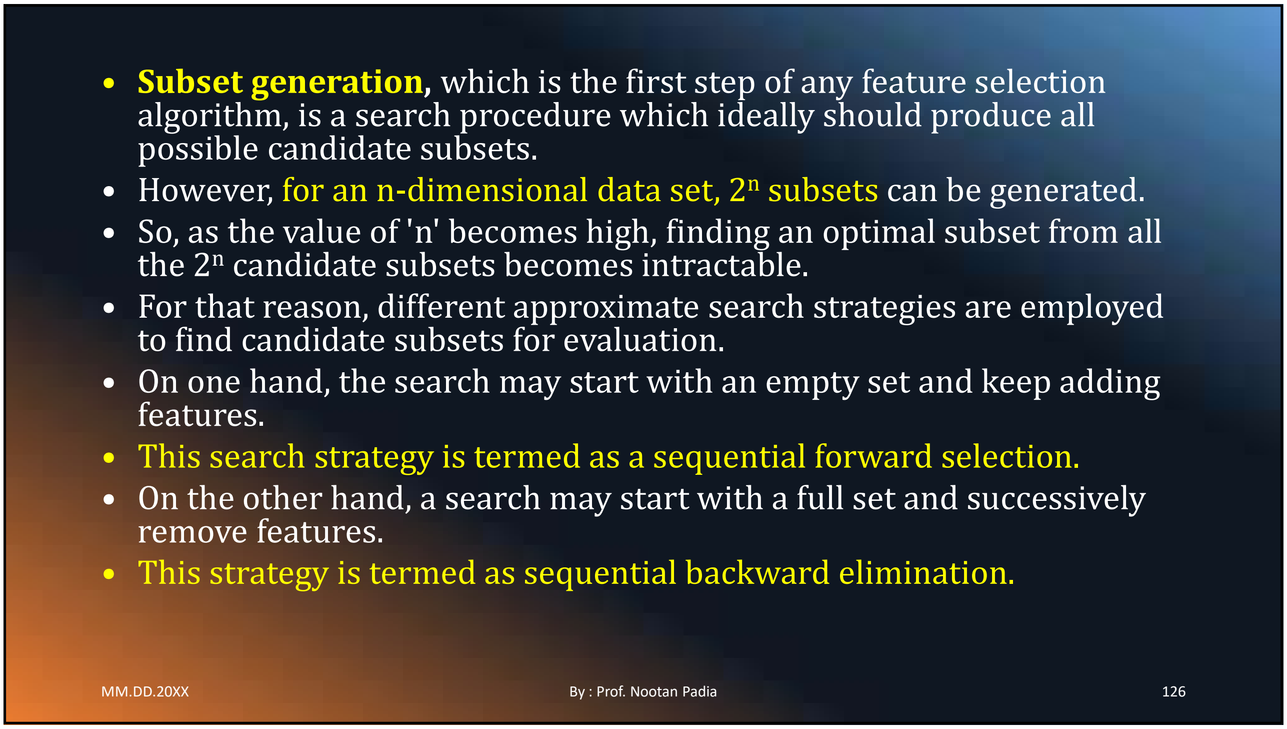
$$\|y\| = \sqrt{2^2 + 1^2 + 0^2 + 0^2 + 3^2 + 2^2 + 1^2 + 0^2 + 1^2} = \sqrt{20} = 4.47$$

$$\therefore cos\ (x, y) = \frac{19}{5.83 * 4.47} = 0.729$$

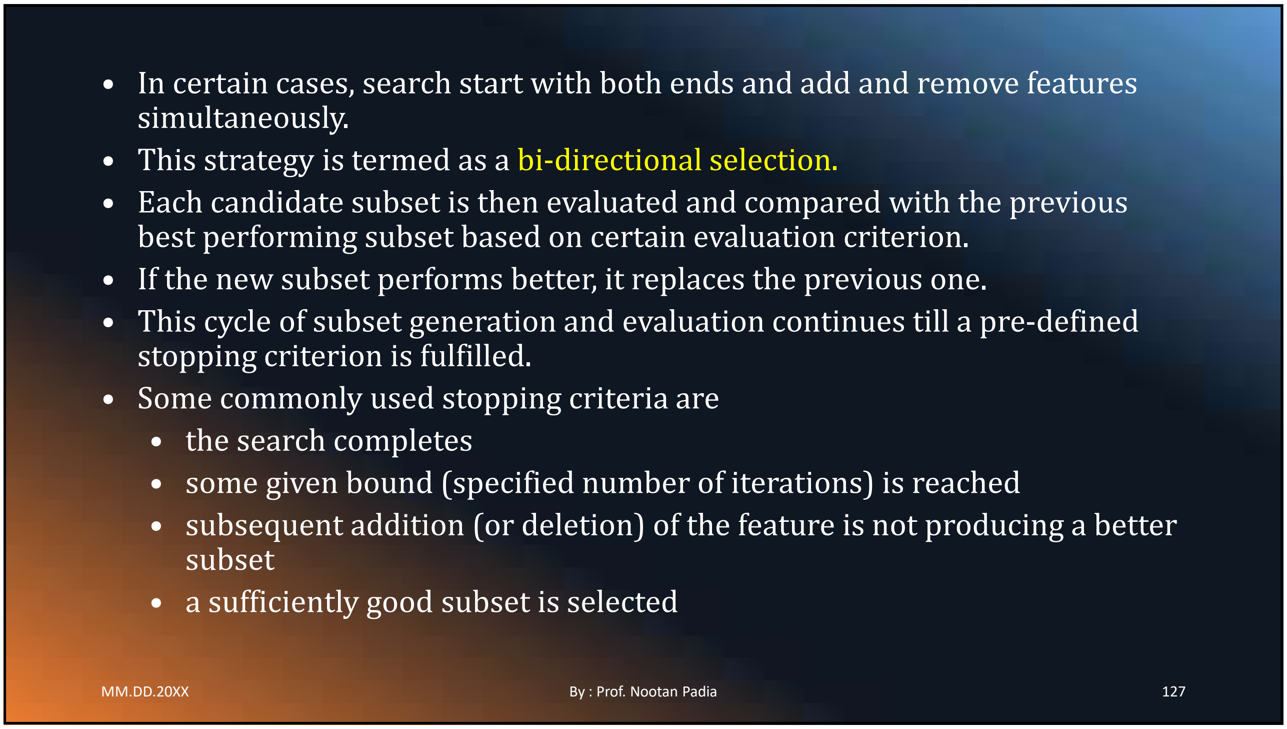
- Cosine similarity actually measures the angle between x and y vectors.
- Hence, if cosine similarity has a value 1, the angle between x and y is 0° which means x and y are same except for the magnitude.
- If cosine similarity is 0, the angle between x and y is 90°.
- Hence, they do not share any similarity.



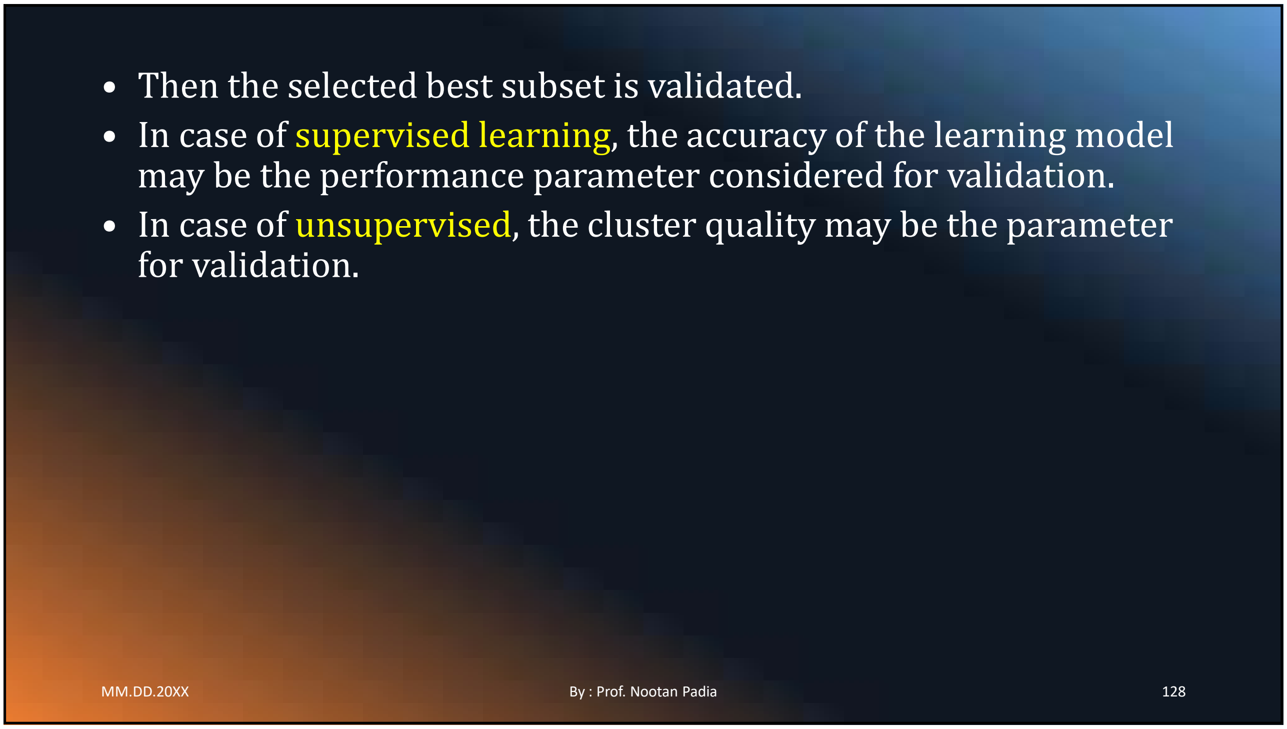
- Feature selection is the process of selecting a subset of feature in a data set.
- As depicted in Figure, a typical feature selection process consists of four steps:
  1. generation of possible subsets
  2. subset evaluation
  3. stop searching based on some stopping criterion
  4. validation of the result



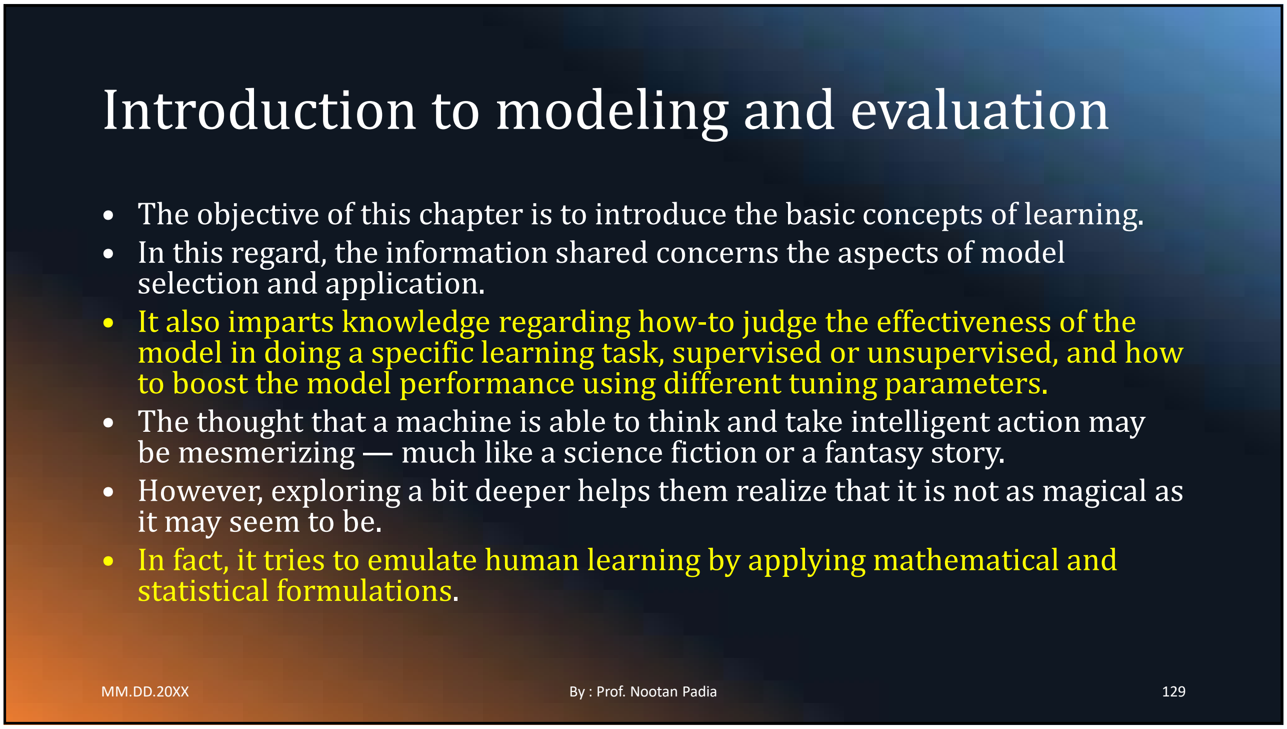
- **Subset generation,** which is the first step of any feature selection algorithm, is a search procedure which ideally should produce all possible candidate subsets.
- However, for an n-dimensional data set, 2^n subsets can be generated.
- So, as the value of 'n' becomes high, finding an optimal subset from all the 2^n candidate subsets becomes intractable.
- For that reason, different approximate search strategies are employed to find candidate subsets for evaluation.
- On one hand, the search may start with an empty set and keep adding features.
- This search strategy is termed as a sequential forward selection.
- On the other hand, a search may start with a full set and successively remove features.
- This strategy is termed as sequential backward elimination.



- In certain cases, search start with both ends and add and remove features simultaneously.
- This strategy is termed as a bi-directional selection.
- Each candidate subset is then evaluated and compared with the previous best performing subset based on certain evaluation criterion.
- If the new subset performs better, it replaces the previous one.
- This cycle of subset generation and evaluation continues till a pre-defined stopping criterion is fulfilled.
- Some commonly used stopping criteria are
    - the search completes
    - some given bound (specified number of iterations) is reached
    - subsequent addition (or deletion) of the feature is not producing a better subset
    - a sufficiently good subset is selected

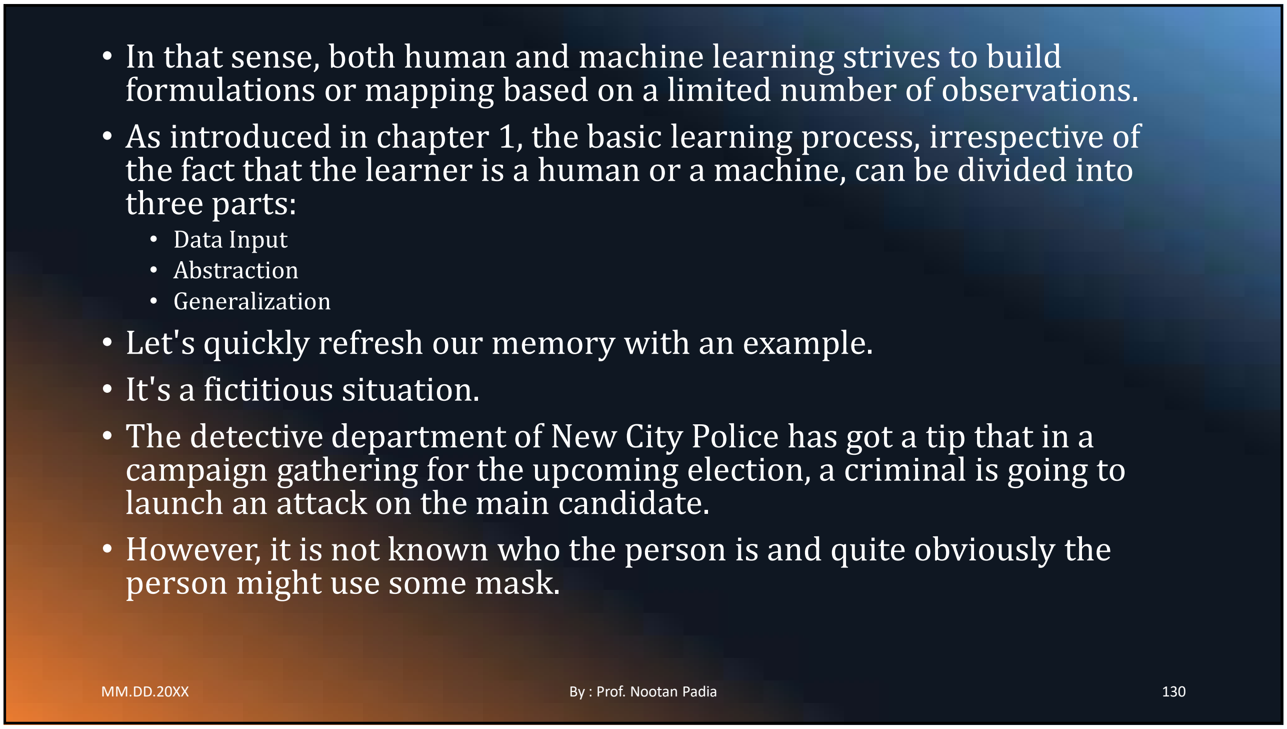


- Then the selected best subset is validated.
- In case of supervised learning, the accuracy of the learning model may be the performance parameter considered for validation.
- In case of unsupervised, the cluster quality may be the parameter for validation.

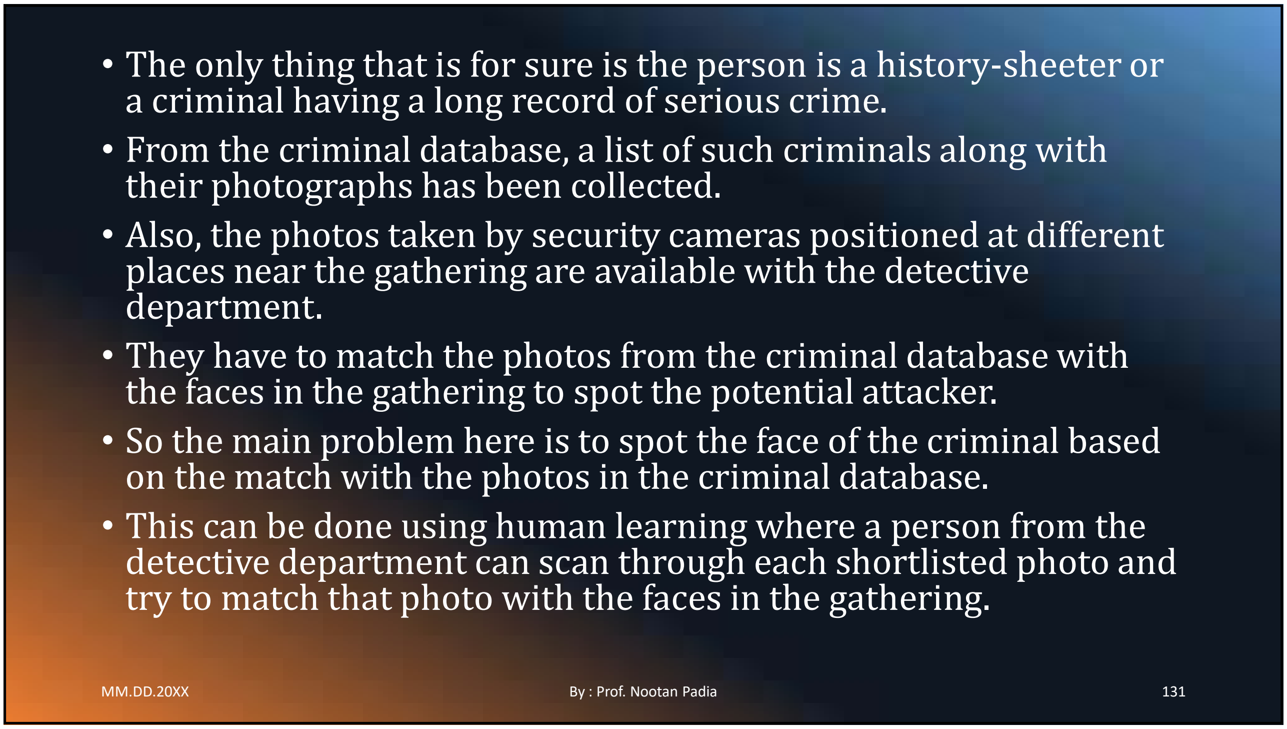


Introduction to modeling and evaluation

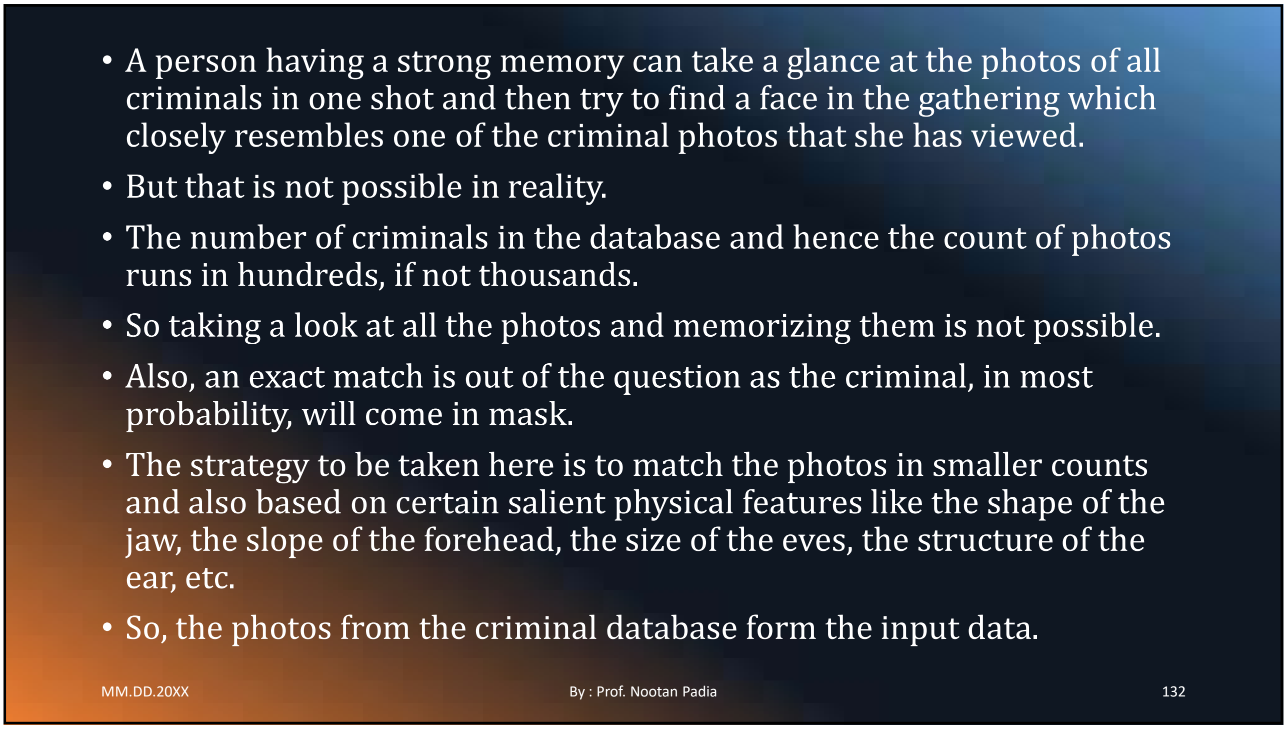
- The objective of this chapter is to introduce the basic concepts of learning.
- In this regard, the information shared concerns the aspects of model selection and application.
- It also imparts knowledge regarding how-to judge the effectiveness of the model in doing a specific learning task, supervised or unsupervised, and how to boost the model performance using different tuning parameters.
- The thought that a machine is able to think and take intelligent action may be mesmerizing — much like a science fiction or a fantasy story.
- However, exploring a bit deeper helps them realize that it is not as magical as it may seem to be.
- In fact, it tries to emulate human learning by applying mathematical and statistical formulations.



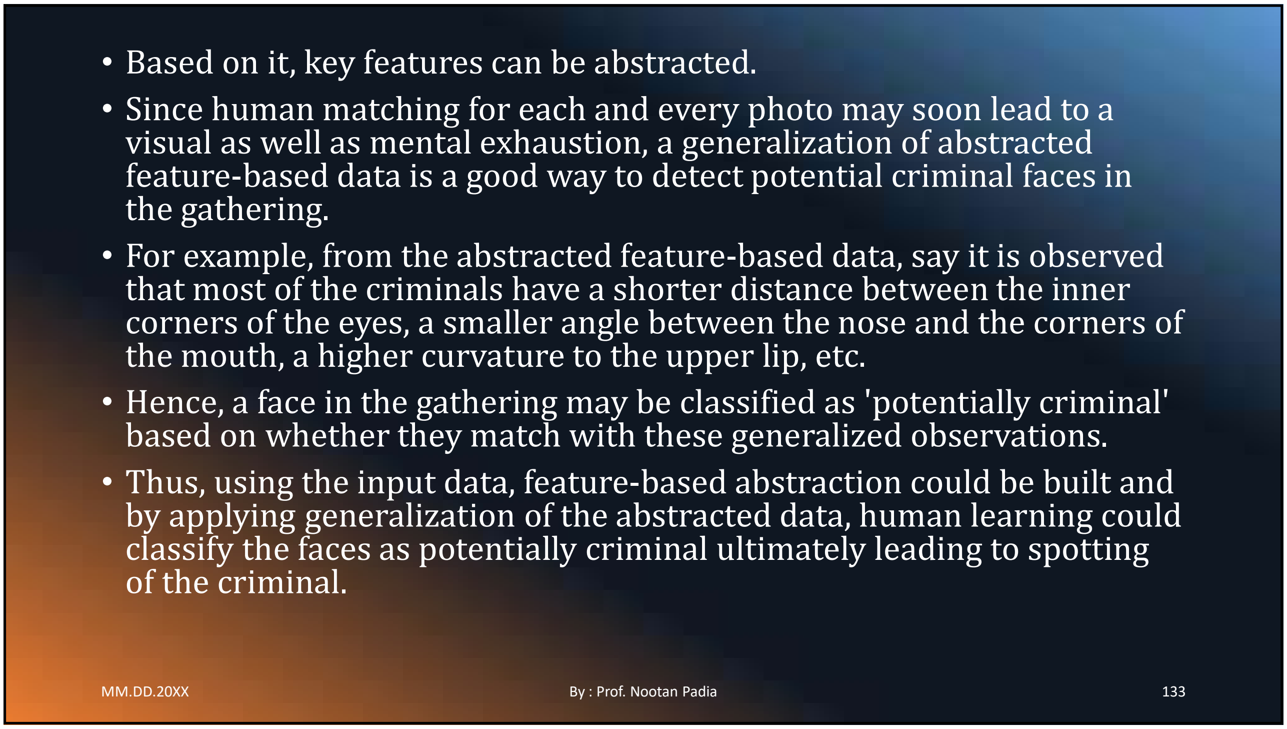
- In that sense, both human and machine learning strives to build formulations or mapping based on a limited number of observations.
- As introduced in chapter 1, the basic learning process, irrespective of the fact that the learner is a human or a machine, can be divided into three parts:
  - Data Input
  - Abstraction
  - Generalization
- Let's quickly refresh our memory with an example.
- It's a fictitious situation.
- The detective department of New City Police has got a tip that in a campaign gathering for the upcoming election, a criminal is going to launch an attack on the main candidate.
- However, it is not known who the person is and quite obviously the person might use some mask.



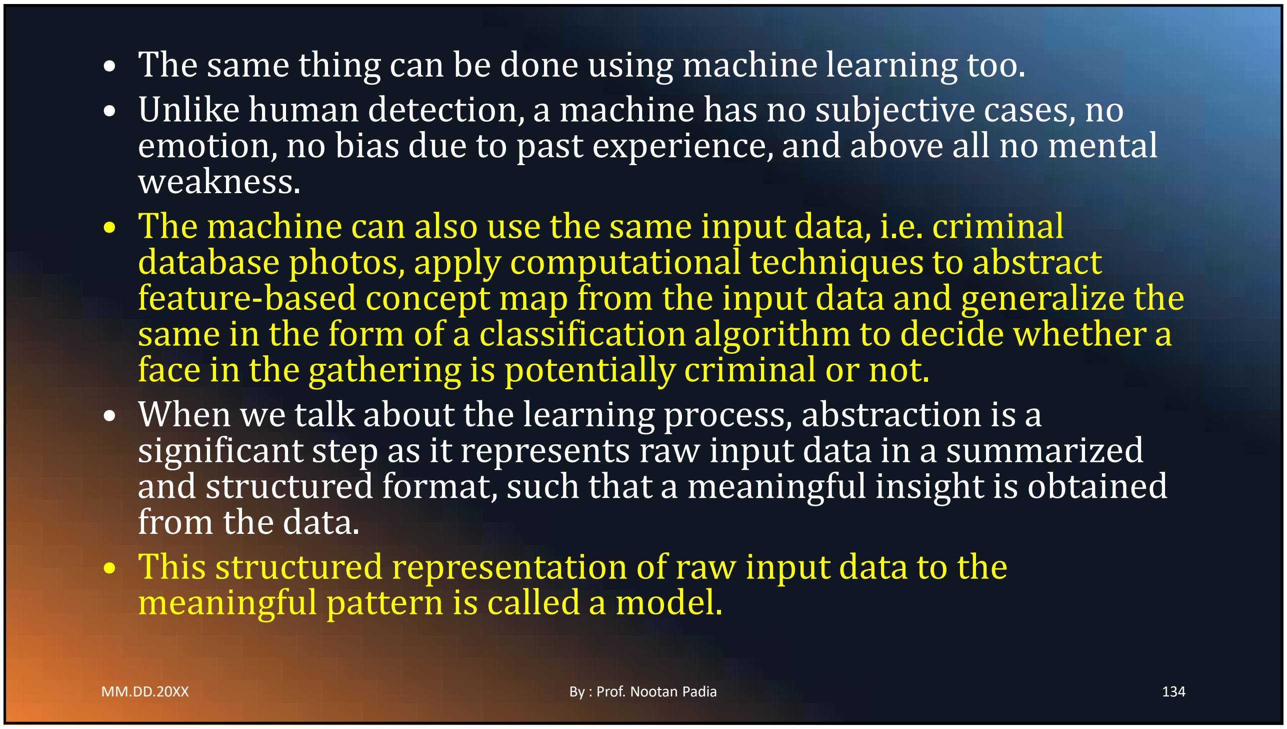
- The only thing that is for sure is the person is a history-sheeter or a criminal having a long record of serious crime.
- From the criminal database, a list of such criminals along with their photographs has been collected.
- Also, the photos taken by security cameras positioned at different places near the gathering are available with the detective department.
- They have to match the photos from the criminal database with the faces in the gathering to spot the potential attacker.
- So the main problem here is to spot the face of the criminal based on the match with the photos in the criminal database.
- This can be done using human learning where a person from the detective department can scan through each shortlisted photo and try to match that photo with the faces in the gathering.



- A person having a strong memory can take a glance at the photos of all criminals in one shot and then try to find a face in the gathering which closely resembles one of the criminal photos that she has viewed.
- But that is not possible in reality.
- The number of criminals in the database and hence the count of photos runs in hundreds, if not thousands.
- So taking a look at all the photos and memorizing them is not possible.
- Also, an exact match is out of the question as the criminal, in most probability, will come in mask.
- The strategy to be taken here is to match the photos in smaller counts and also based on certain salient physical features like the shape of the jaw, the slope of the forehead, the size of the eves, the structure of the ear, etc.
- So, the photos from the criminal database form the input data.



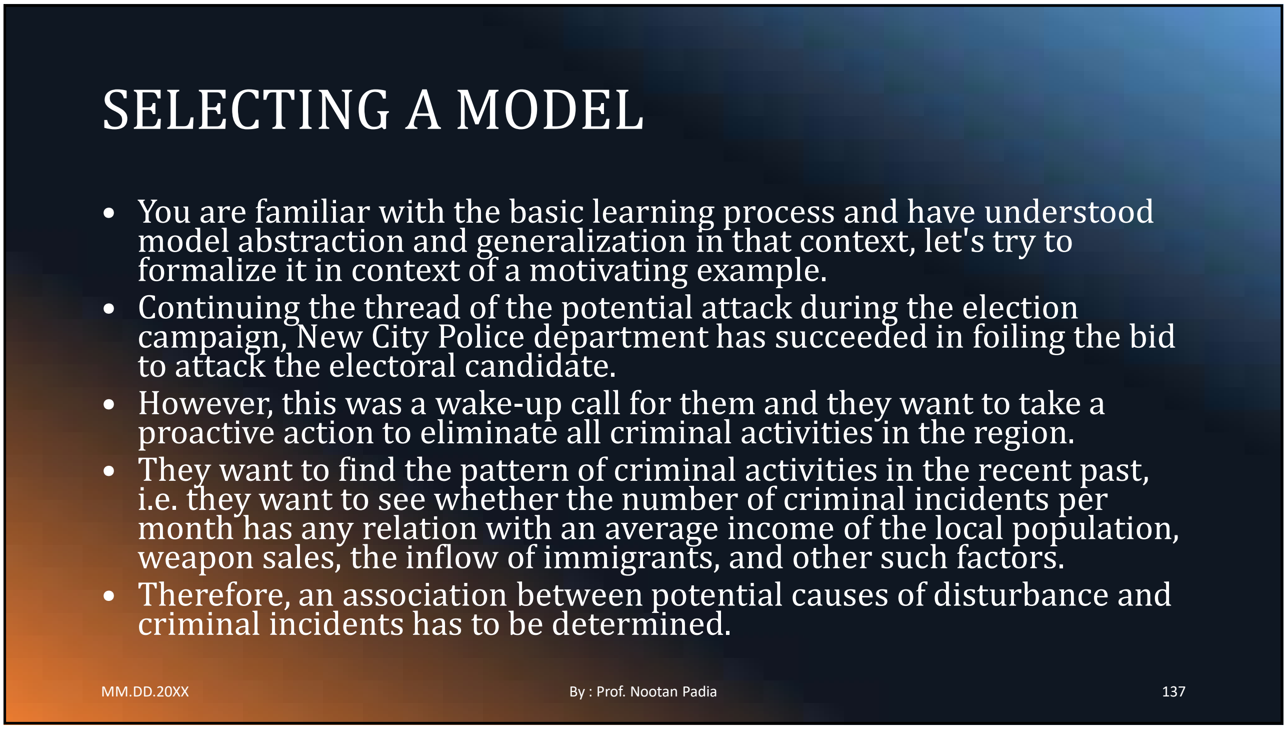
- Based on it, key features can be abstracted.
- Since human matching for each and every photo may soon lead to a visual as well as mental exhaustion, a generalization of abstracted feature-based data is a good way to detect potential criminal faces in the gathering.
- For example, from the abstracted feature-based data, say it is observed that most of the criminals have a shorter distance between the inner corners of the eyes, a smaller angle between the nose and the corners of the mouth, a higher curvature to the upper lip, etc.
- Hence, a face in the gathering may be classified as 'potentially criminal' based on whether they match with these generalized observations.
- Thus, using the input data, feature-based abstraction could be built and by applying generalization of the abstracted data, human learning could classify the faces as potentially criminal ultimately leading to spotting of the criminal.



- The same thing can be done using machine learning too.
- Unlike human detection, a machine has no subjective cases, no emotion, no bias due to past experience, and above all no mental weakness.
- The machine can also use the same input data, i.e. criminal database photos, apply computational techniques to abstract feature-based concept map from the input data and generalize the same in the form of a classification algorithm to decide whether a face in the gathering is potentially criminal or not.
- When we talk about the learning process, abstraction is a significant step as it represents raw input data in a summarized and structured format, such that a meaningful insight is obtained from the data.
- This structured representation of raw input data to the meaningful pattern is called a model.

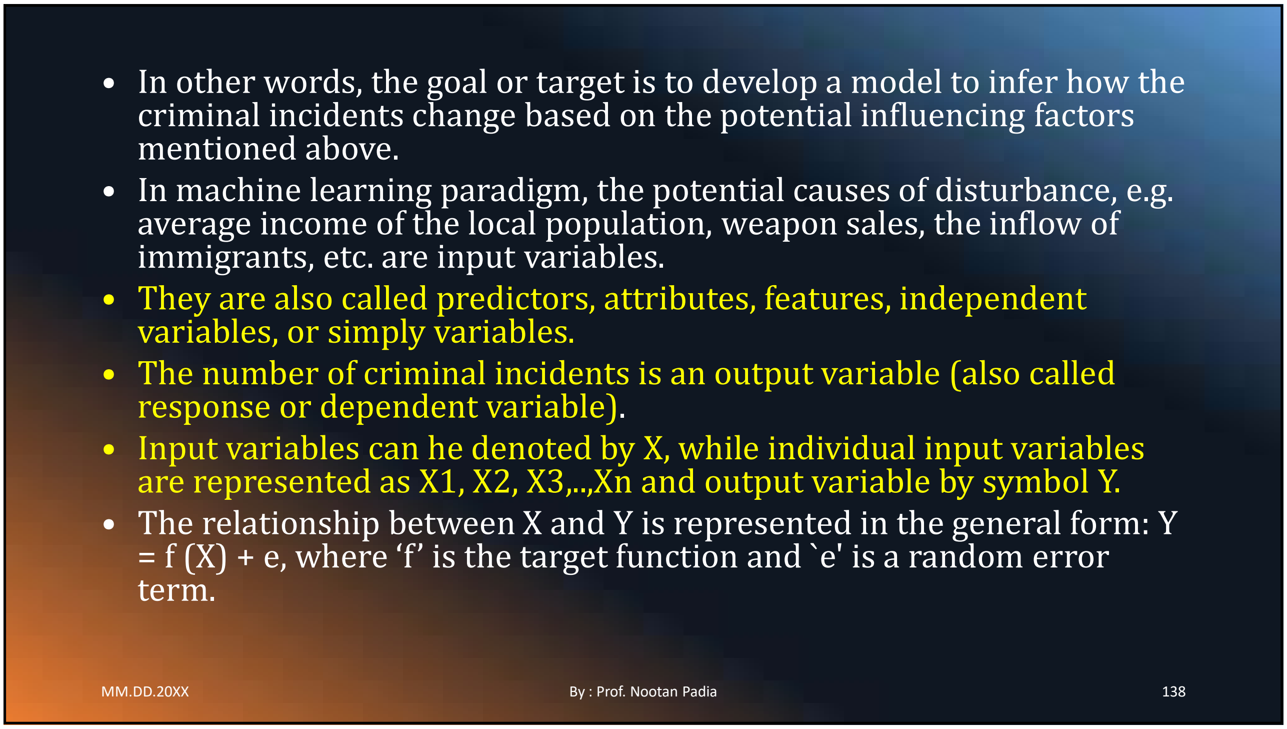
- The model might have different forms.
- It might be a mathematical equation, it might be a graph or tree structure, it might be a computational block, etc.
- The decision regarding which model is to be selected for a specific data set is taken by the learning task, based on the problem to be solved and the type of data.
- For example, when the problem is related to prediction and the target field is numeric and continuous, the regression model is assigned.
- The process of assigning a model, and fitting a specific model to a data set is called model training.
- Once the model is trained, the raw input data is summarized into an abstracted form.
- However, with abstraction, the learner is able to only summarize the knowledge.

- This knowledge might be still very broad-based — consisting of a huge number of feature-based data and inter-relations.
- To generate actionable insight from such broad-based knowledge is very difficult.
- This is where generalization comes into play.
- Generalization searches through the huge set of abstracted knowledge to come up with a small and manageable set of key findings. It is not possible to do an exhaustive search by reviewing each of the abstracted findings one-by-one.
- A heuristic search is employed, an approach which is also used for human learning (often termed as `gut-feel').
- It is quite obvious that the heuristics sometimes result in erroneous result.
- If the outcome is systematically incorrect, the learning is said to have a bias.

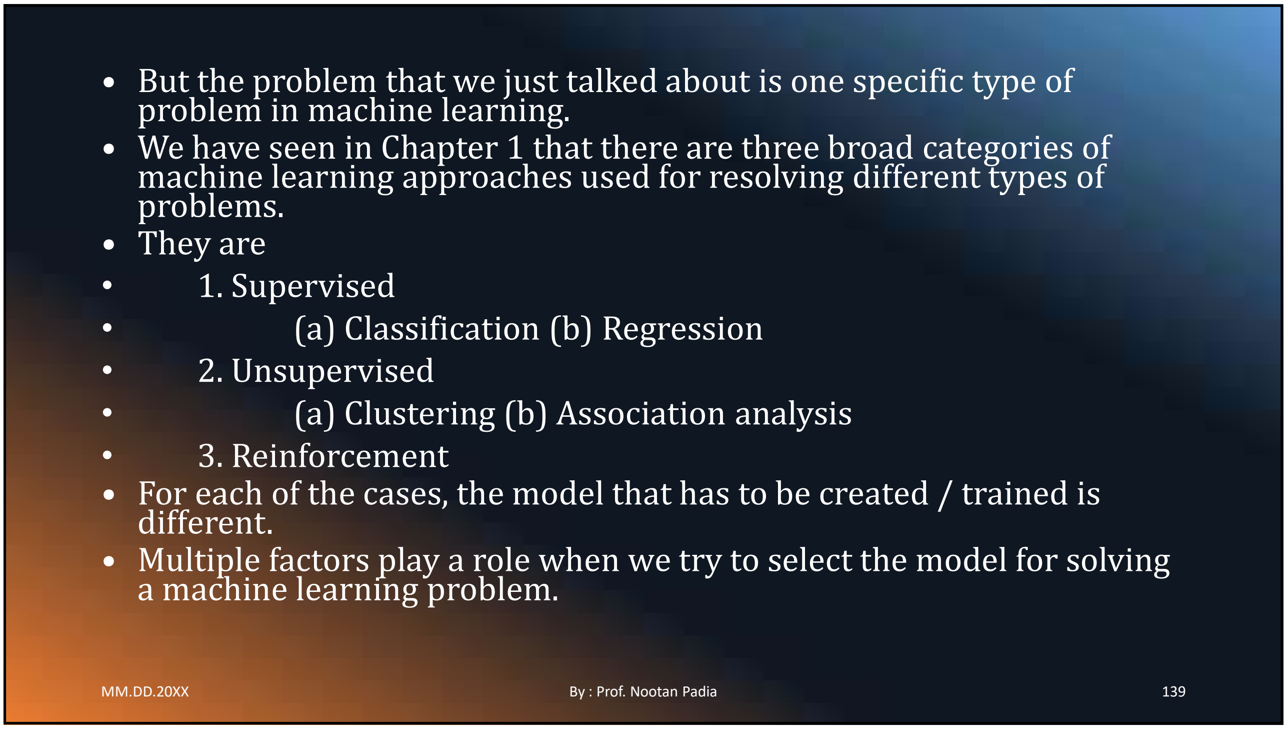


SELECTING A MODEL

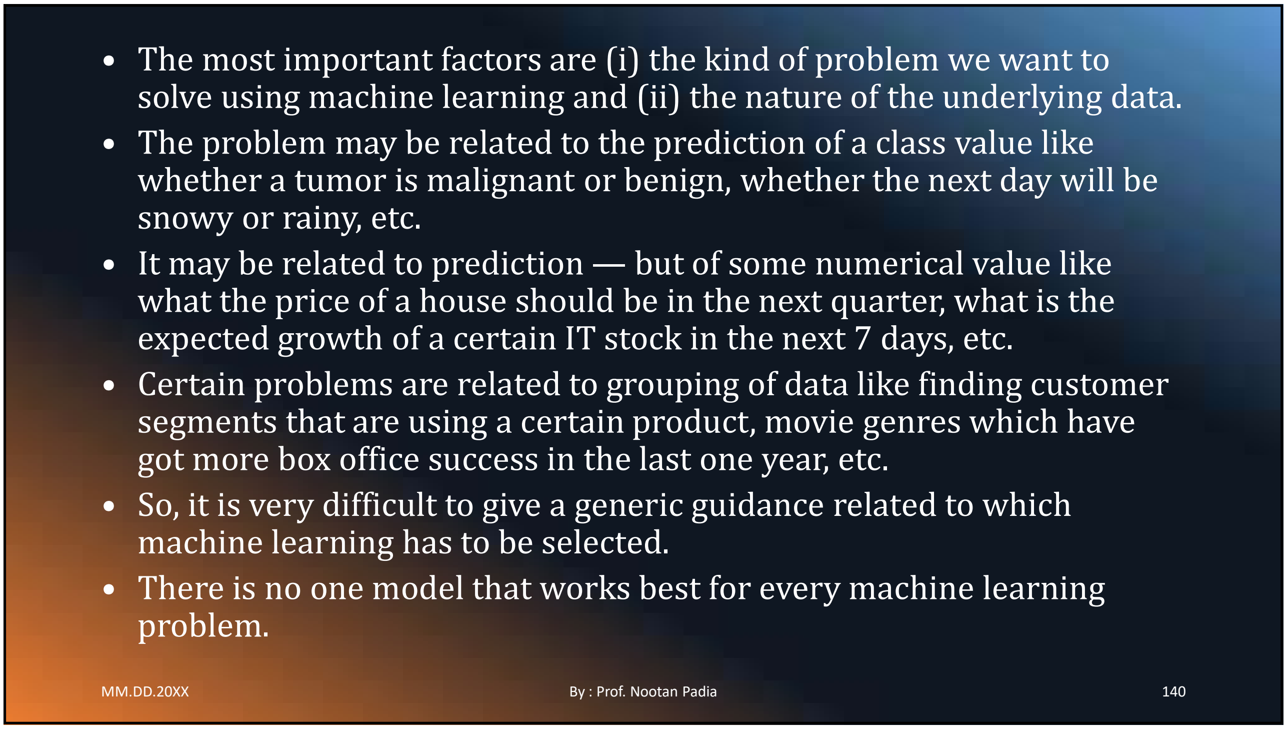
- You are familiar with the basic learning process and have understood model abstraction and generalization in that context, let's try to formalize it in context of a motivating example.
- Continuing the thread of the potential attack during the election campaign, New City Police department has succeeded in foiling the bid to attack the electoral candidate.
- However, this was a wake-up call for them and they want to take a proactive action to eliminate all criminal activities in the region.
- They want to find the pattern of criminal activities in the recent past, i.e. they want to see whether the number of criminal incidents per month has any relation with an average income of the local population, weapon sales, the inflow of immigrants, and other such factors.
- Therefore, an association between potential causes of disturbance and criminal incidents has to be determined.



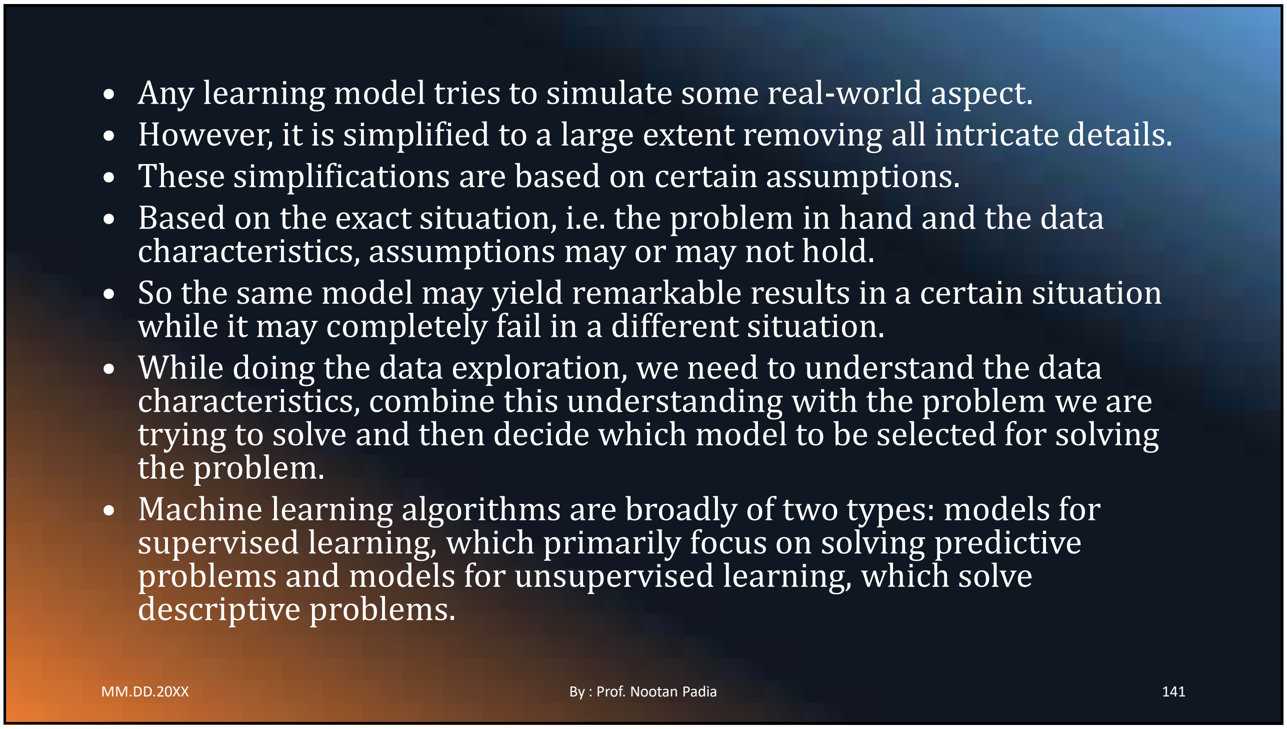
- In other words, the goal or target is to develop a model to infer how the criminal incidents change based on the potential influencing factors mentioned above.
- In machine learning paradigm, the potential causes of disturbance, e.g. average income of the local population, weapon sales, the inflow of immigrants, etc. are input variables.
- They are also called predictors, attributes, features, independent variables, or simply variables.
- The number of criminal incidents is an output variable (also called response or dependent variable).
- Input variables can he denoted by X, while individual input variables are represented as X1, X2, X3,..,Xn and output variable by symbol Y.
- The relationship between X and Y is represented in the general form: Y = f (X) + e, where ‘f’ is the target function and `e' is a random error term.



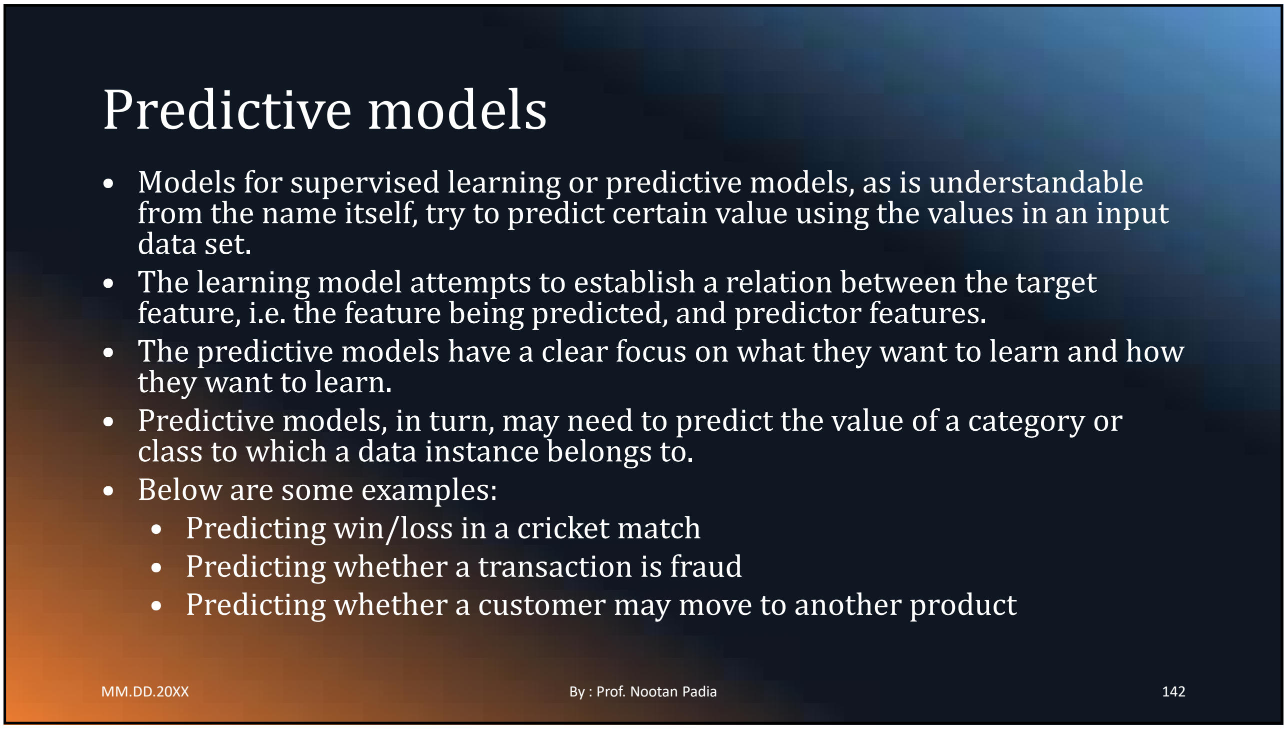
- But the problem that we just talked about is one specific type of problem in machine learning.
- We have seen in Chapter 1 that there are three broad categories of machine learning approaches used for resolving different types of problems.
- They are
-     1. Supervised
-             (a) Classification (b) Regression
-     2. Unsupervised
-             (a) Clustering (b) Association analysis
-     3. Reinforcement
- For each of the cases, the model that has to be created / trained is different.
- Multiple factors play a role when we try to select the model for solving a machine learning problem.



- The most important factors are (i) the kind of problem we want to solve using machine learning and (ii) the nature of the underlying data.
- The problem may be related to the prediction of a class value like whether a tumor is malignant or benign, whether the next day will be snowy or rainy, etc.
- It may be related to prediction — but of some numerical value like what the price of a house should be in the next quarter, what is the expected growth of a certain IT stock in the next 7 days, etc.
- Certain problems are related to grouping of data like finding customer segments that are using a certain product, movie genres which have got more box office success in the last one year, etc.
- So, it is very difficult to give a generic guidance related to which machine learning has to be selected.
- There is no one model that works best for every machine learning problem.

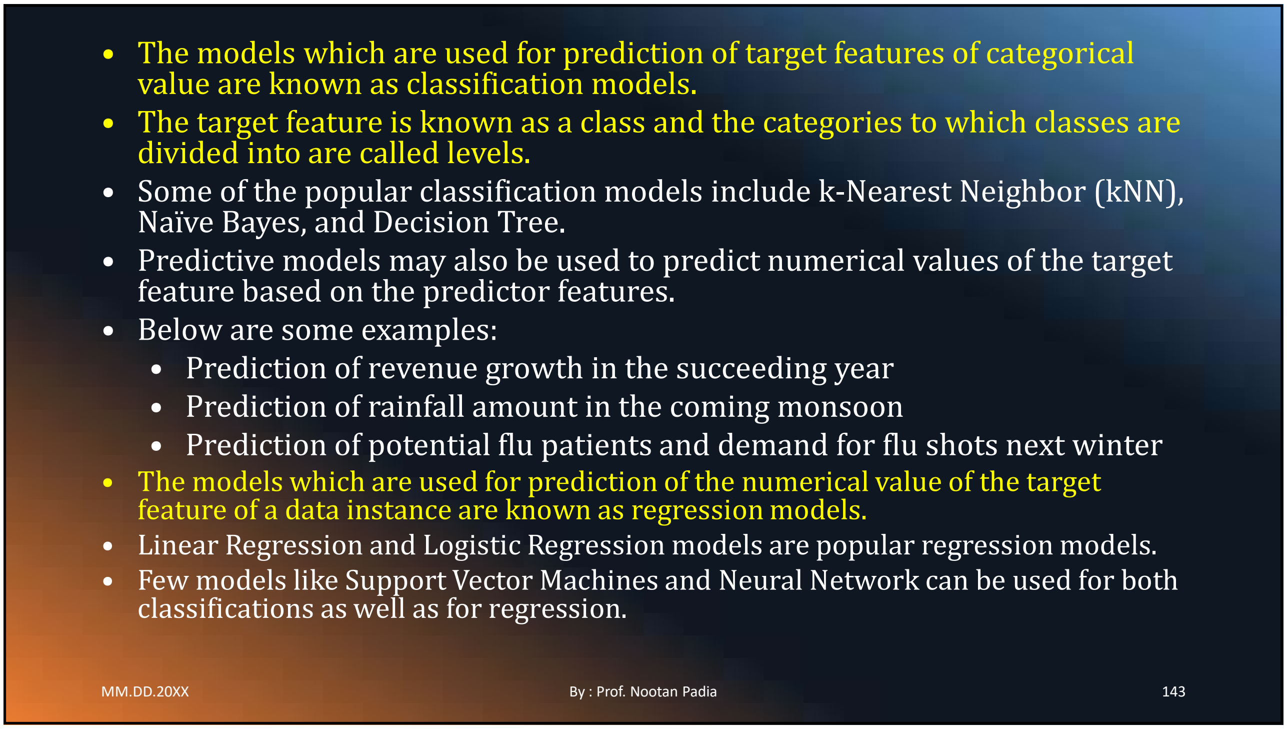


- Any learning model tries to simulate some real-world aspect.
- However, it is simplified to a large extent removing all intricate details.
- These simplifications are based on certain assumptions.
- Based on the exact situation, i.e. the problem in hand and the data characteristics, assumptions may or may not hold.
- So the same model may yield remarkable results in a certain situation while it may completely fail in a different situation.
- While doing the data exploration, we need to understand the data characteristics, combine this understanding with the problem we are trying to solve and then decide which model to be selected for solving the problem.
- Machine learning algorithms are broadly of two types: models for supervised learning, which primarily focus on solving predictive problems and models for unsupervised learning, which solve descriptive problems.

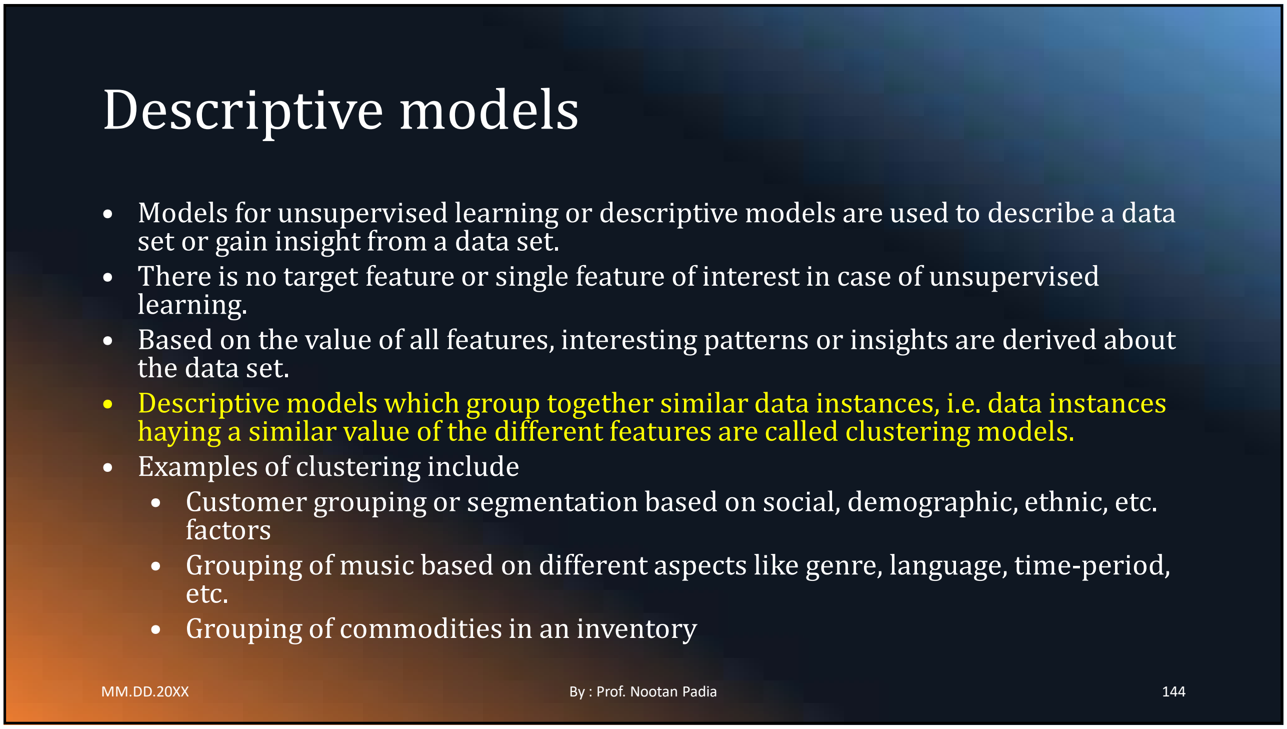


Predictive models

- Models for supervised learning or predictive models, as is understandable from the name itself, try to predict certain value using the values in an input data set.
- The learning model attempts to establish a relation between the target feature, i.e. the feature being predicted, and predictor features.
- The predictive models have a clear focus on what they want to learn and how they want to learn.
- Predictive models, in turn, may need to predict the value of a category or class to which a data instance belongs to.
- Below are some examples:
  - Predicting win/loss in a cricket match
  - Predicting whether a transaction is fraud
  - Predicting whether a customer may move to another product

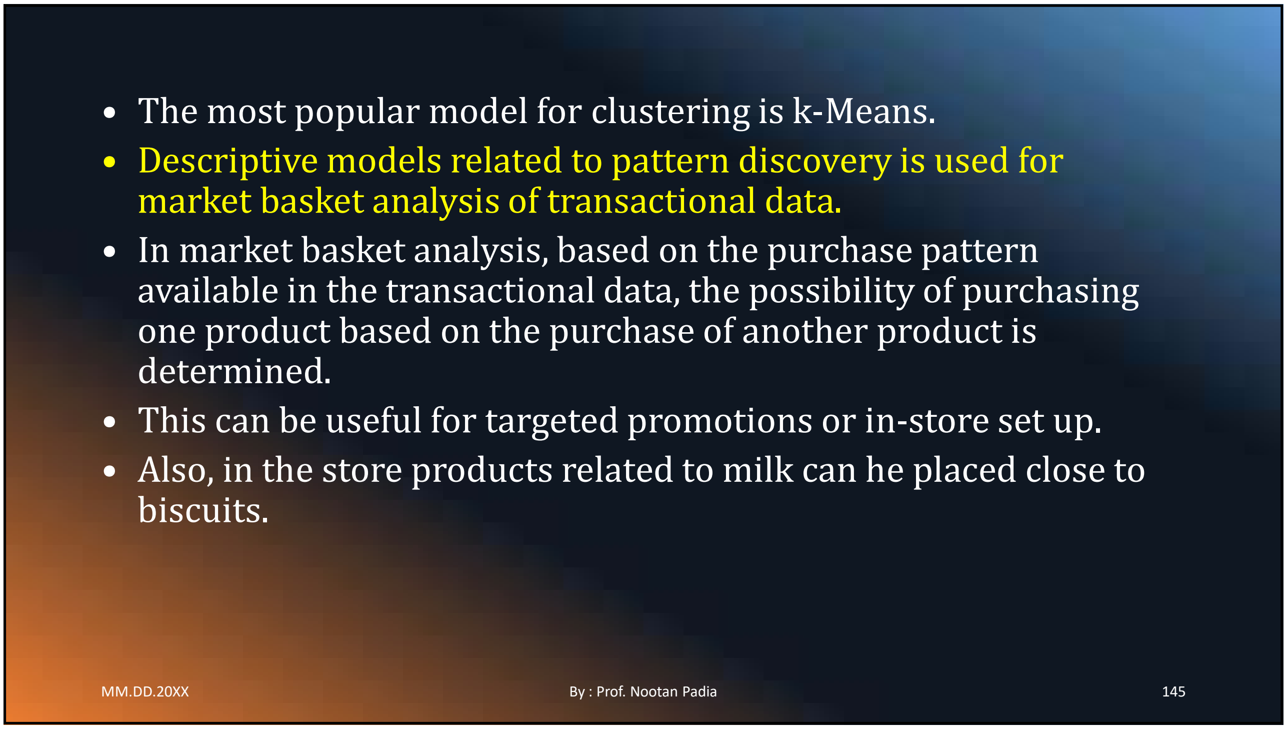


- The models which are used for prediction of target features of categorical value are known as classification models.
- The target feature is known as a class and the categories to which classes are divided into are called levels.
- Some of the popular classification models include k-Nearest Neighbor (kNN), Naïve Bayes, and Decision Tree.
- Predictive models may also be used to predict numerical values of the target feature based on the predictor features.
- Below are some examples:
  - Prediction of revenue growth in the succeeding year
  - Prediction of rainfall amount in the coming monsoon
  - Prediction of potential flu patients and demand for flu shots next winter
- The models which are used for prediction of the numerical value of the target feature of a data instance are known as regression models.
- Linear Regression and Logistic Regression models are popular regression models.
- Few models like Support Vector Machines and Neural Network can be used for both classifications as well as for regression.

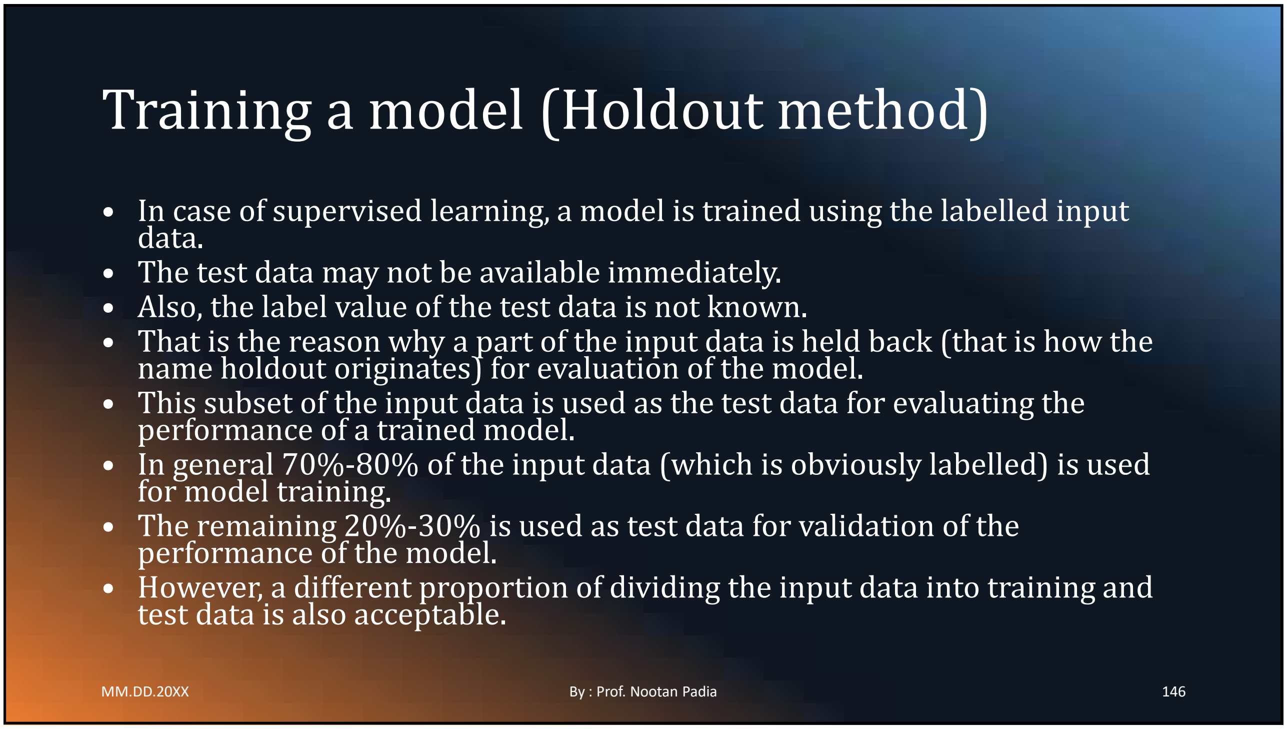


Descriptive models

- Models for unsupervised learning or descriptive models are used to describe a data set or gain insight from a data set.
- There is no target feature or single feature of interest in case of unsupervised learning.
- Based on the value of all features, interesting patterns or insights are derived about the data set.
- Descriptive models which group together similar data instances, i.e. data instances haying a similar value of the different features are called clustering models.
- Examples of clustering include
  - Customer grouping or segmentation based on social, demographic, ethnic, etc. factors
  - Grouping of music based on different aspects like genre, language, time-period, etc.
  - Grouping of commodities in an inventory

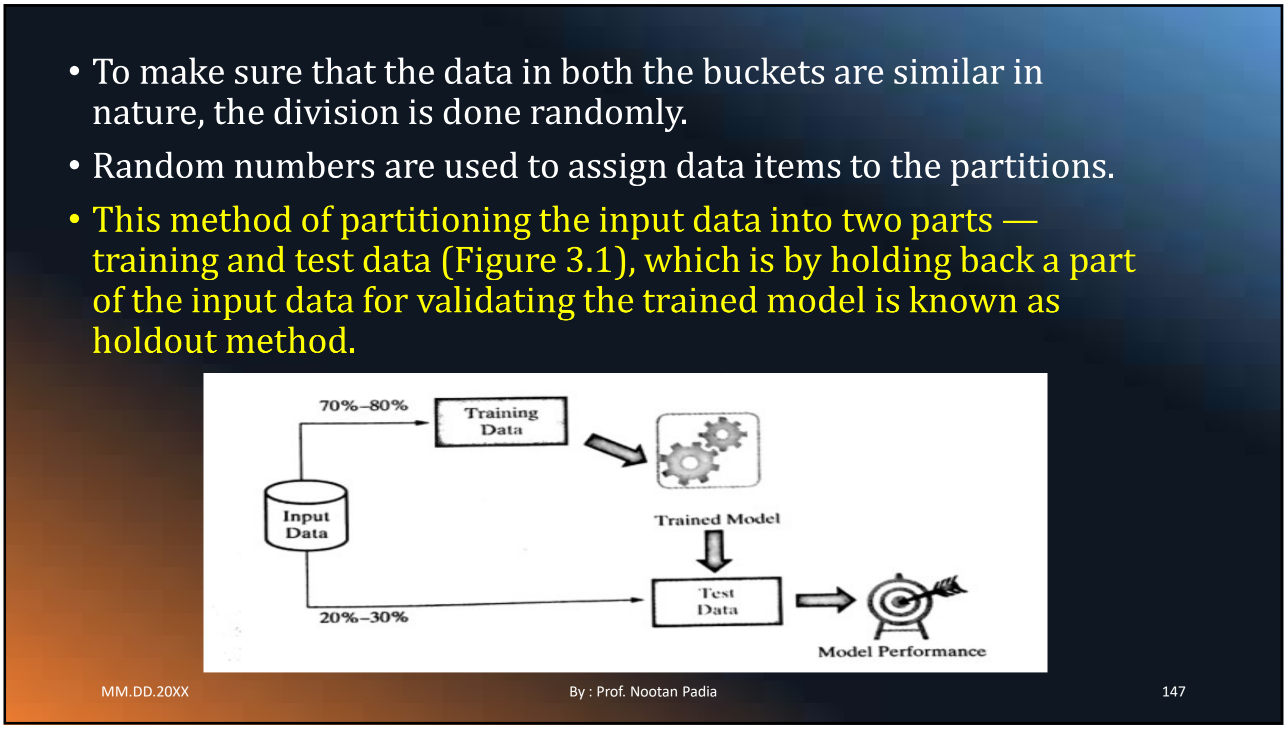


- The most popular model for clustering is k-Means.
- Descriptive models related to pattern discovery is used for market basket analysis of transactional data.
- In market basket analysis, based on the purchase pattern available in the transactional data, the possibility of purchasing one product based on the purchase of another product is determined.
- This can be useful for targeted promotions or in-store set up.
- Also, in the store products related to milk can he placed close to biscuits.

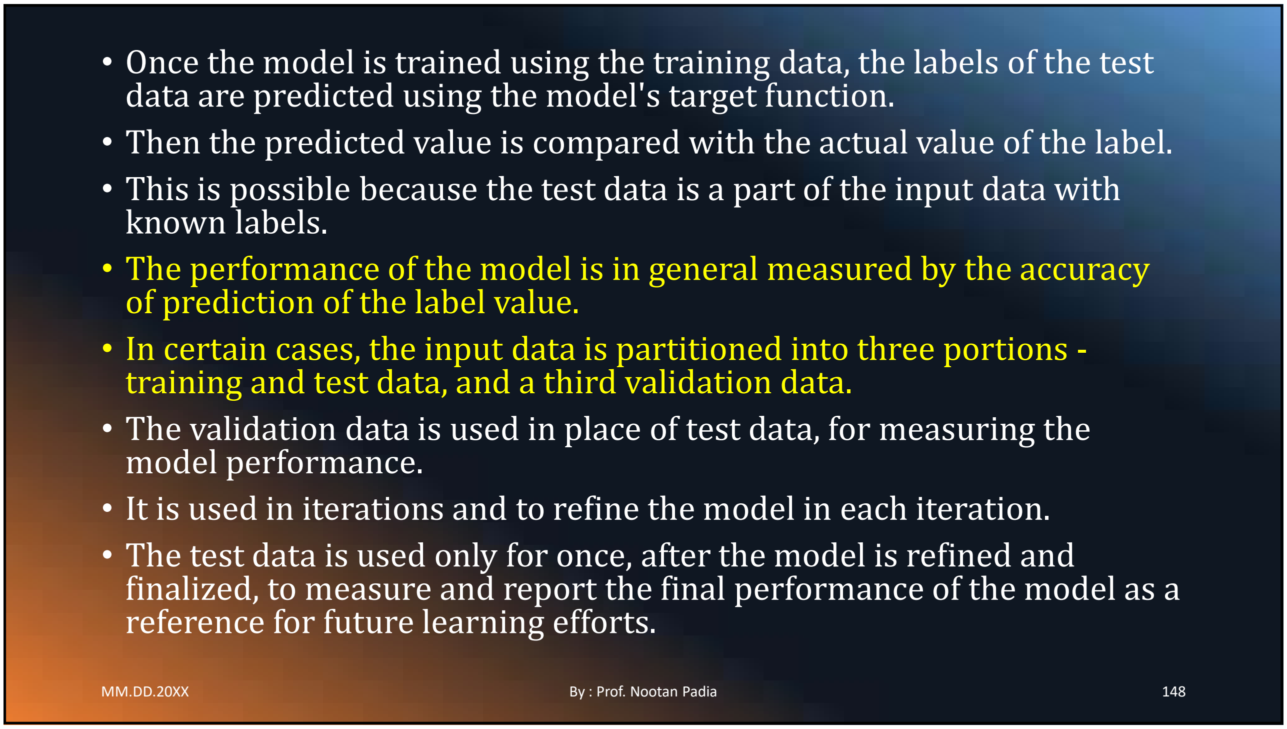


Training a model (Holdout method)

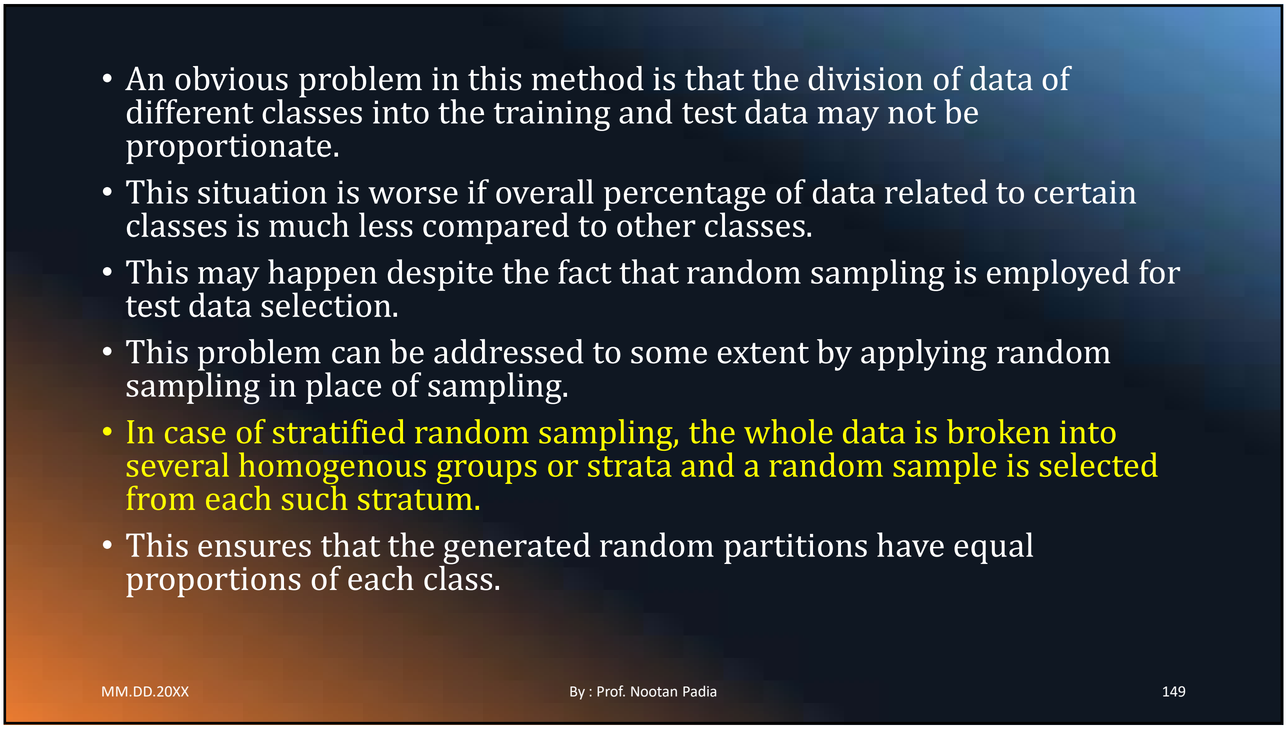
- In case of supervised learning, a model is trained using the labelled input data.
- The test data may not be available immediately.
- Also, the label value of the test data is not known.
- That is the reason why a part of the input data is held back (that is how the name holdout originates) for evaluation of the model.
- This subset of the input data is used as the test data for evaluating the performance of a trained model.
- In general 70%-80% of the input data (which is obviously labelled) is used for model training.
- The remaining 20%-30% is used as test data for validation of the performance of the model.
- However, a different proportion of dividing the input data into training and test data is also acceptable.



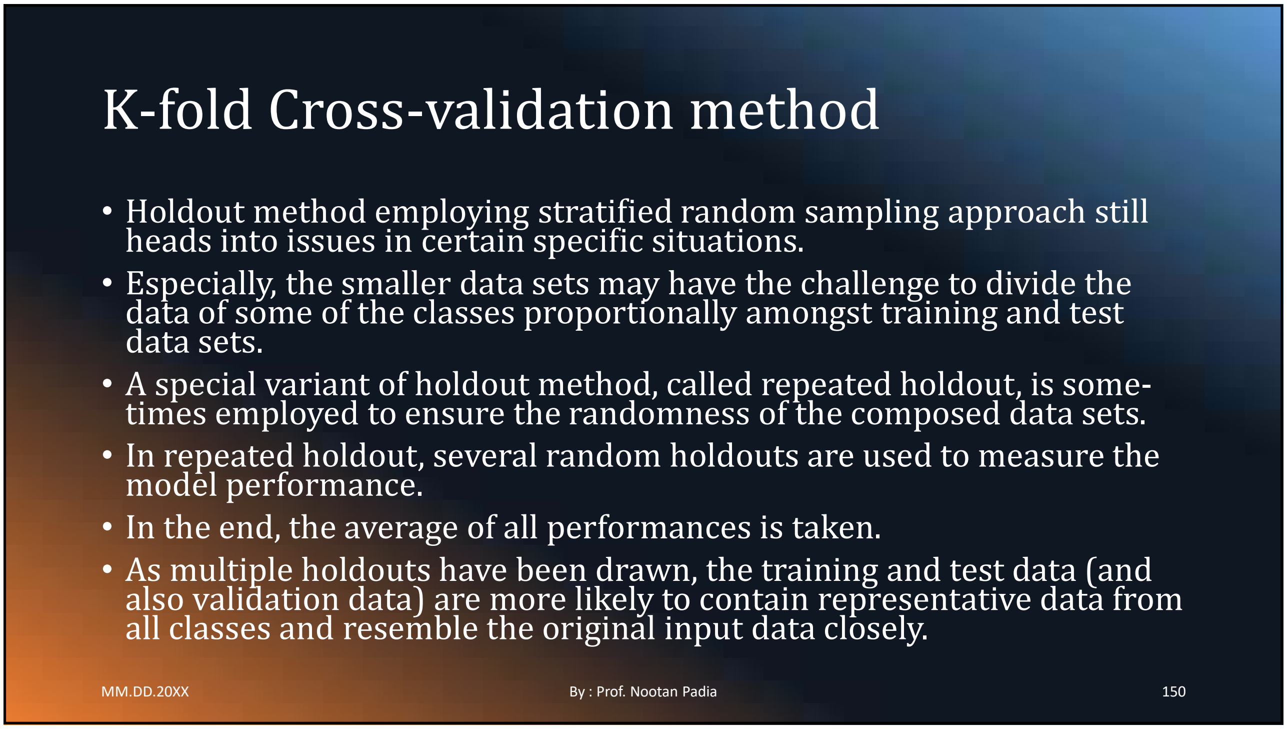
- To make sure that the data in both the buckets are similar in nature, the division is done randomly.
- Random numbers are used to assign data items to the partitions.
- This method of partitioning the input data into two parts — training and test data (Figure 3.1), which is by holding back a part of the input data for validating the trained model is known as holdout method.



- Once the model is trained using the training data, the labels of the test data are predicted using the model's target function.
- Then the predicted value is compared with the actual value of the label.
- This is possible because the test data is a part of the input data with known labels.
- The performance of the model is in general measured by the accuracy of prediction of the label value.
- In certain cases, the input data is partitioned into three portions - training and test data, and a third validation data.
- The validation data is used in place of test data, for measuring the model performance.
- It is used in iterations and to refine the model in each iteration.
- The test data is used only for once, after the model is refined and finalized, to measure and report the final performance of the model as a reference for future learning efforts.



- An obvious problem in this method is that the division of data of different classes into the training and test data may not be proportionate.
- This situation is worse if overall percentage of data related to certain classes is much less compared to other classes.
- This may happen despite the fact that random sampling is employed for test data selection.
- This problem can be addressed to some extent by applying random sampling in place of sampling.
- In case of stratified random sampling, the whole data is broken into several homogenous groups or strata and a random sample is selected from each such stratum.
- This ensures that the generated random partitions have equal proportions of each class.

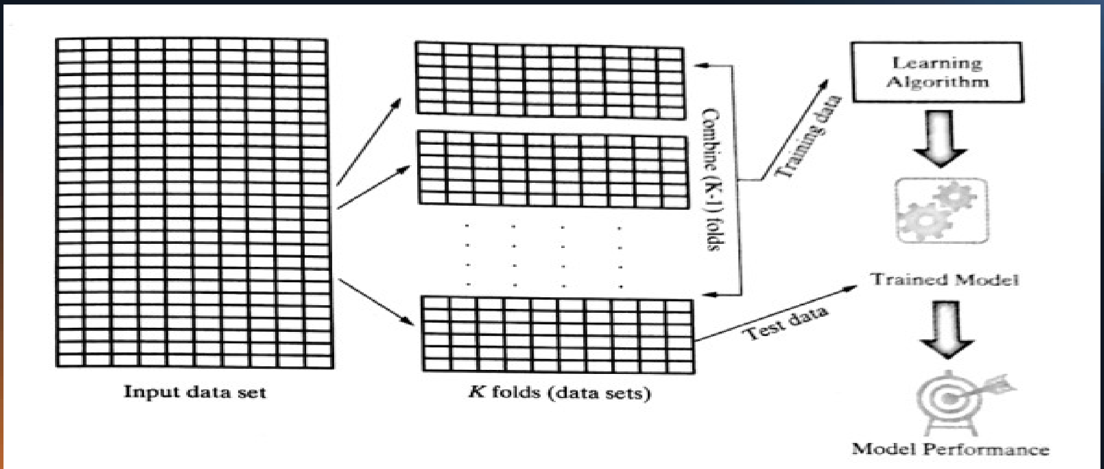


K-fold Cross-validation method

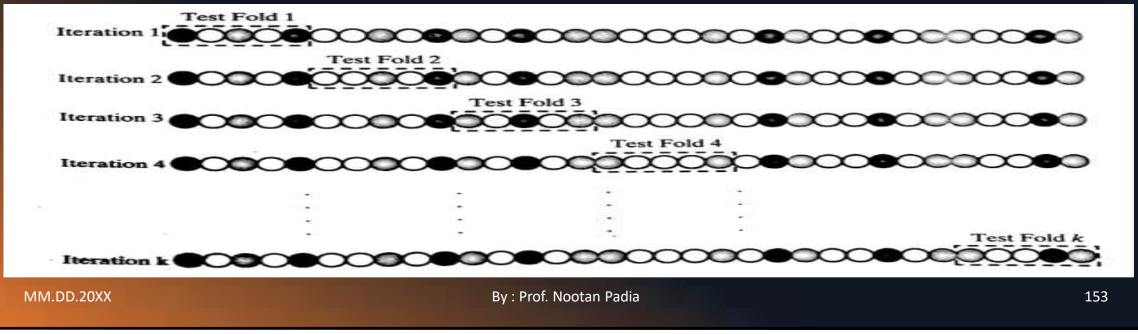
- Holdout method employing stratified random sampling approach still heads into issues in certain specific situations.
- Especially, the smaller data sets may have the challenge to divide the data of some of the classes proportionally amongst training and test data sets.
- A special variant of holdout method, called repeated holdout, is sometimes employed to ensure the randomness of the composed data sets.
- In repeated holdout, several random holdouts are used to measure the model performance.
- In the end, the average of all performances is taken.
- As multiple holdouts have been drawn, the training and test data (and also validation data) are more likely to contain representative data from all classes and resemble the original input data closely.

- This process of repeated holdout is the basis of k-fold cross-validation technique.
- In k-fold cross-validation, the data set is divided into k-completely distinct or non-overlapping random partitions called folds.
- Figure in next slide depicts an overall approach for k-fold cross-validation.
- The value of 'k' in k-fold cross-validation can be set to any number.
- However, there are two approaches which are extremely popular:
  - 10-fold cross-validation (10-fold CV)
  - Leave-one-out cross-validation (LOOCV)
- 10-fold cross-validation is by far the most popular approach.
- In this approach, for each of the 10-folds, each comprising of approximately 10% of the data, one of the folds is used as the test data for validating model performance trained based on the remaining 9 folds (or 90% of the data).

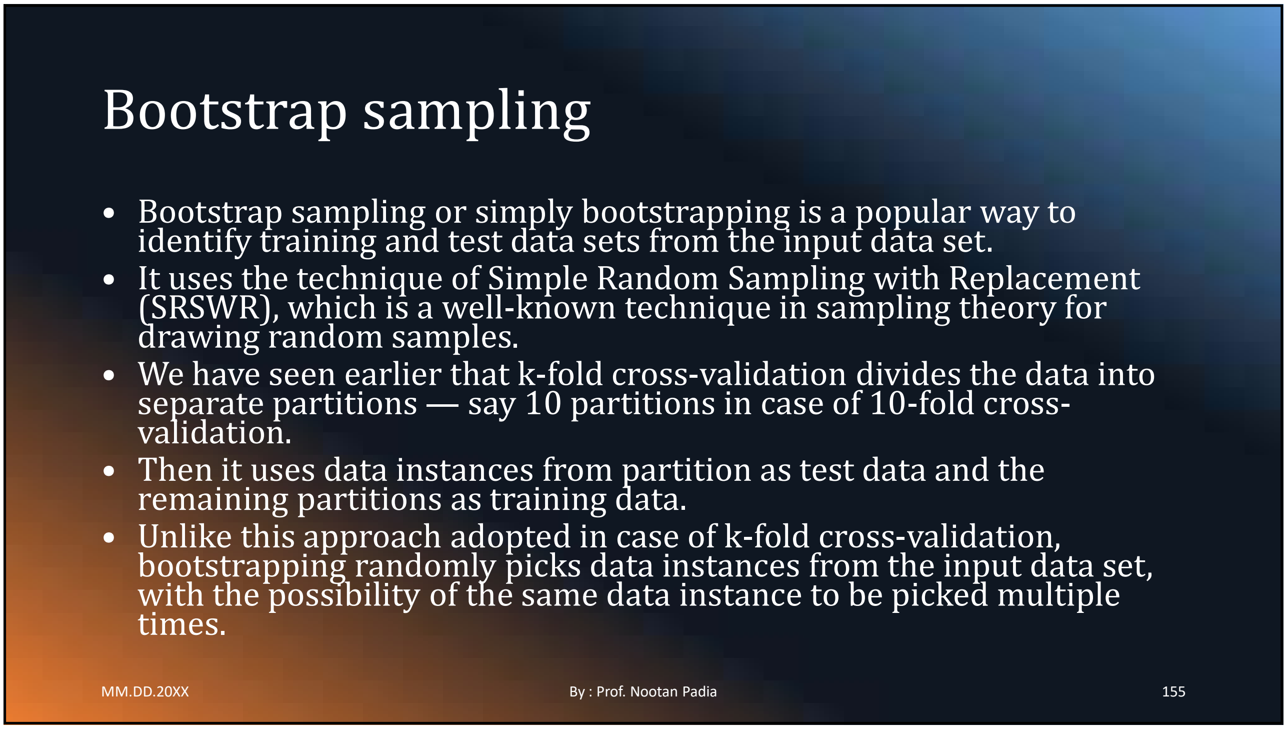
Overall approaches for K-fold cross validation



- This is repeated 10 times, once for each of the 10 folds being used as the test data and the remaining folds as the training data.
- The average performance across all folds is being reported. Following figure depicts the detailed approach of selecting the 'k' folds in k-fold cross-validation.
- As can be observed in the figure, each of the circles resembles a record in the input data set whereas the different colors indicate the different classes that the records belong to.
- Detailed approach for fold selection

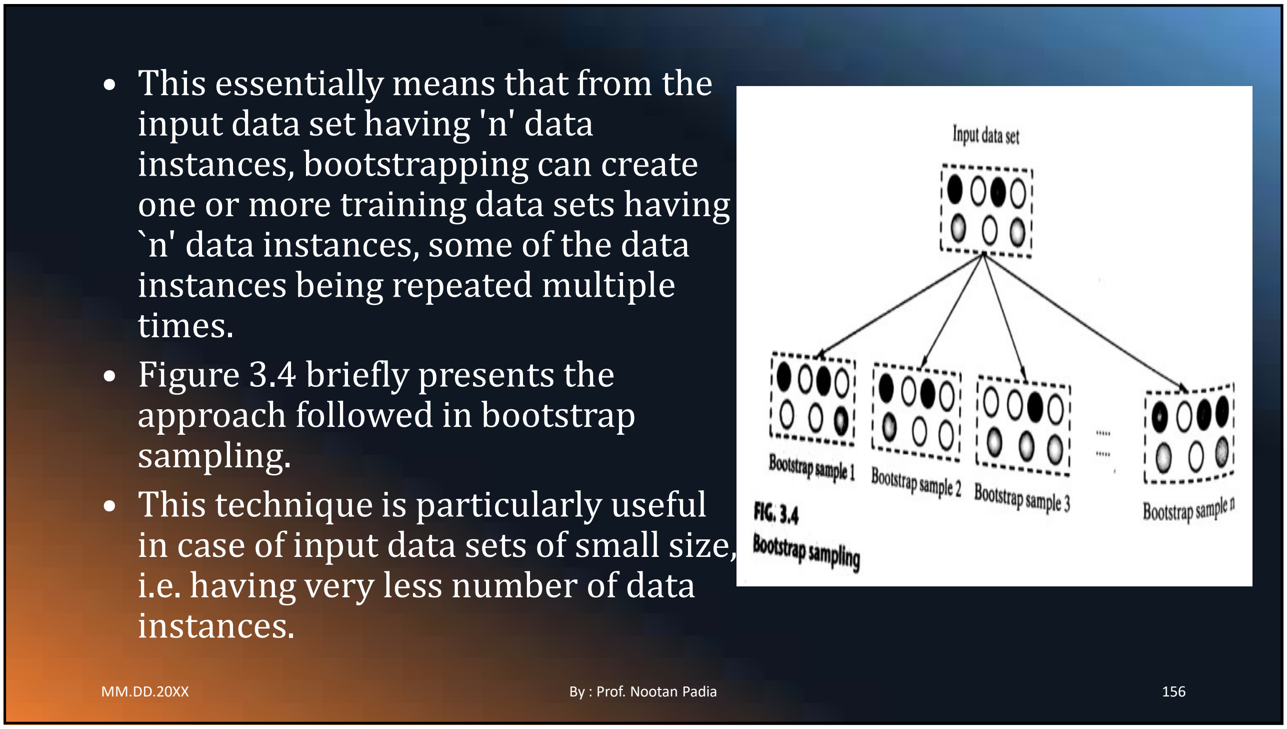


- The entire data set is broken into 'k' folds out of which one fold is selected in each iteration as the test data set.
- The fold selected as test data set in each of the 'k' iterations is different.
- Also, note that though in figure in previous slide the circles resemble the records in the input data set, the contiguous circles represented as folds do not mean that they are subsequent records in the data set.
- This is more a virtual representation and not a physical representation.
- As already mentioned, the records in a fold are drawn by using random sampling technique.
- Leave-one-out cross-validation (LOOCV) is an extreme case of k-fold cross-validation using one record or data instance at a time as a test data.
- This is done to maximize the count of data used to train the model.
- It is obvious that the number of iterations for which it has to be run is equal to the total number of data in the input data set.
- Hence, obviously, it is computationally very expensive and not used much in practice.



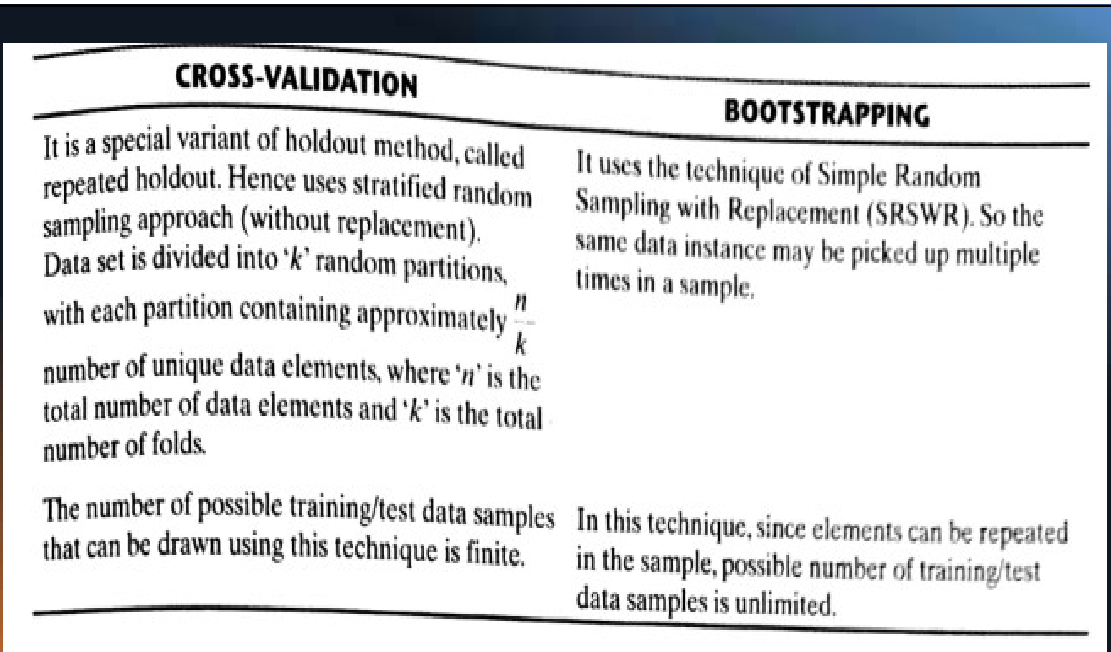
Bootstrap sampling

- Bootstrap sampling or simply bootstrapping is a popular way to identify training and test data sets from the input data set.
- It uses the technique of Simple Random Sampling with Replacement (SRSWR), which is a well-known technique in sampling theory for drawing random samples.
- We have seen earlier that k-fold cross-validation divides the data into separate partitions — say 10 partitions in case of 10-fold cross-validation.
- Then it uses data instances from partition as test data and the remaining partitions as training data.
- Unlike this approach adopted in case of k-fold cross-validation, bootstrapping randomly picks data instances from the input data set, with the possibility of the same data instance to be picked multiple times.



- This essentially means that from the input data set having 'n' data instances, bootstrapping can create one or more training data sets having `n' data instances, some of the data instances being repeated multiple times.
- Figure 3.4 briefly presents the approach followed in bootstrap sampling.
- This technique is particularly useful in case of input data sets of small size, i.e. having very less number of data instances.

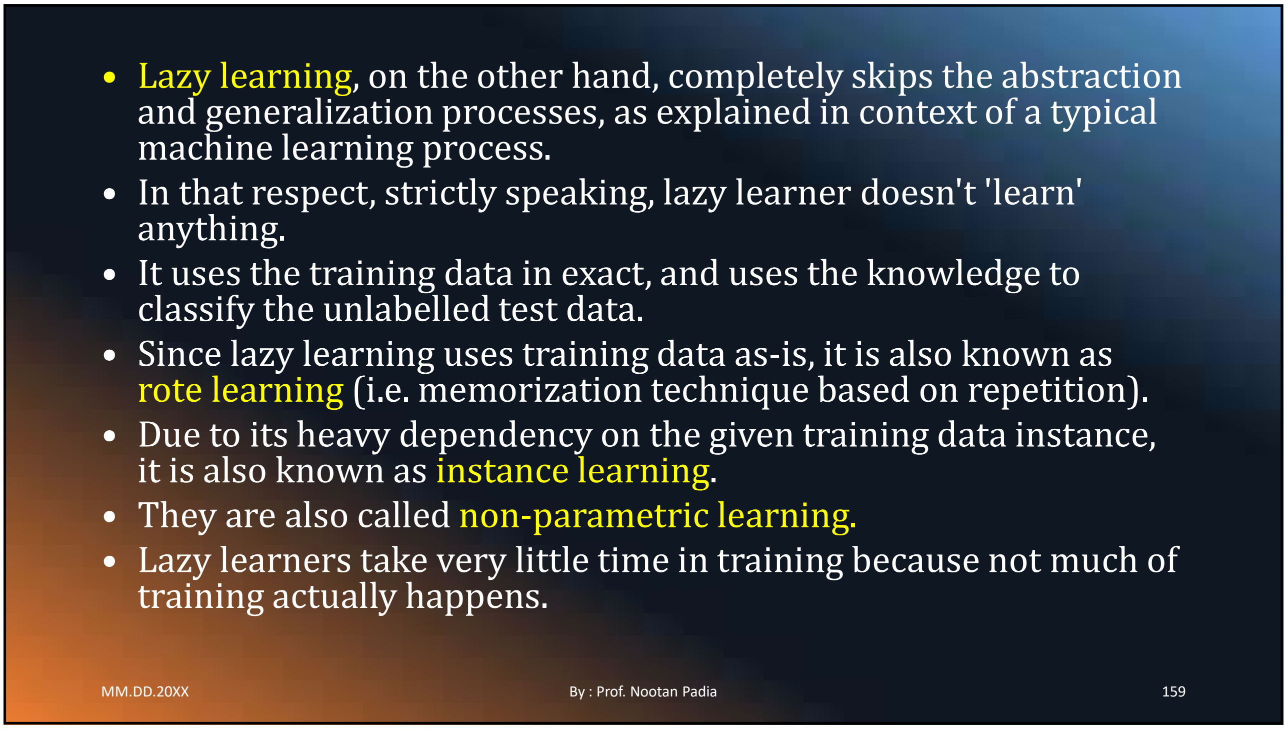
FIG. 3.4
Bootstrap sampling



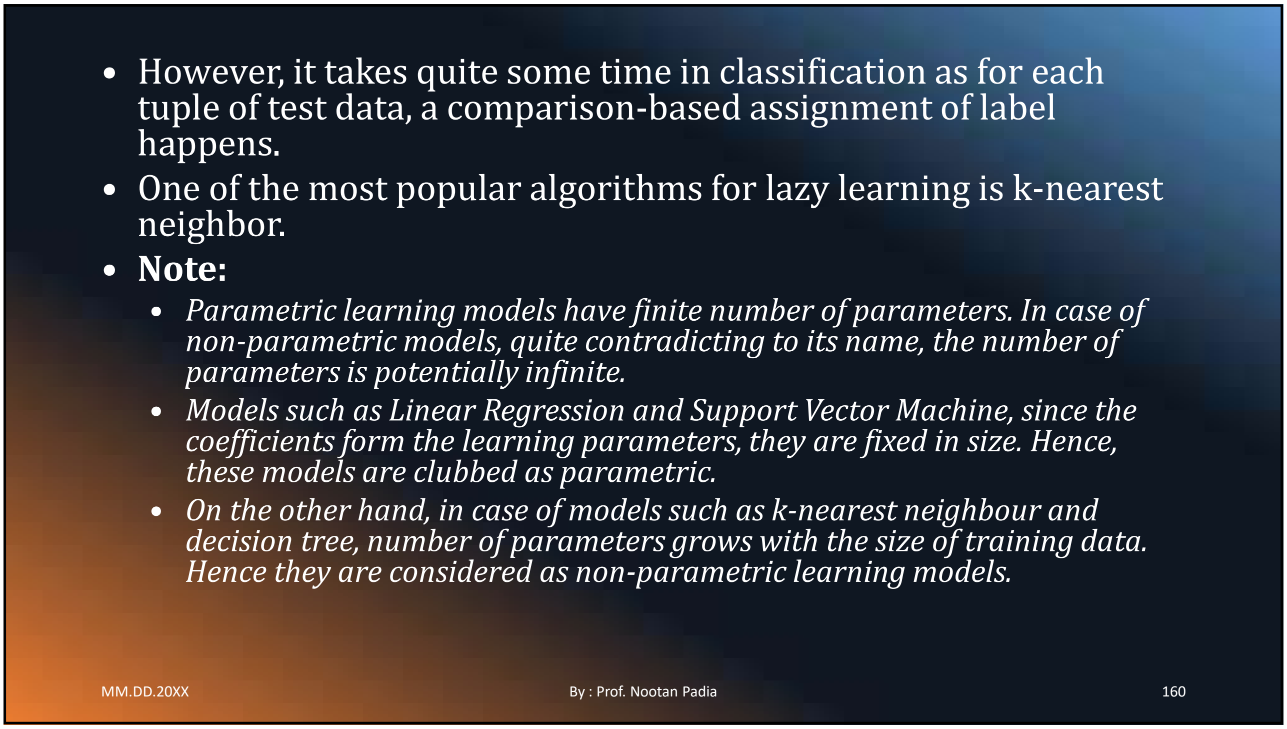
CROSS-VALIDATION	BOOTSTRAPPING
It is a special variant of holdout method, called repeated holdout. Hence uses stratified random sampling approach (without replacement). Data set is divided into 'k' random partitions, with each partition containing approximately $\frac{n}{k}$ number of unique data elements, where 'n' is the total number of data elements and 'k' is the total number of folds.	It uses the technique of Simple Random Sampling with Replacement (SRSWR). So the same data instance may be picked up multiple times in a sample.
The number of possible training/test data samples that can be drawn using this technique is finite.	In this technique, since elements can be repeated in the sample, possible number of training/test data samples is unlimited.

Lazy vs. Eager learner

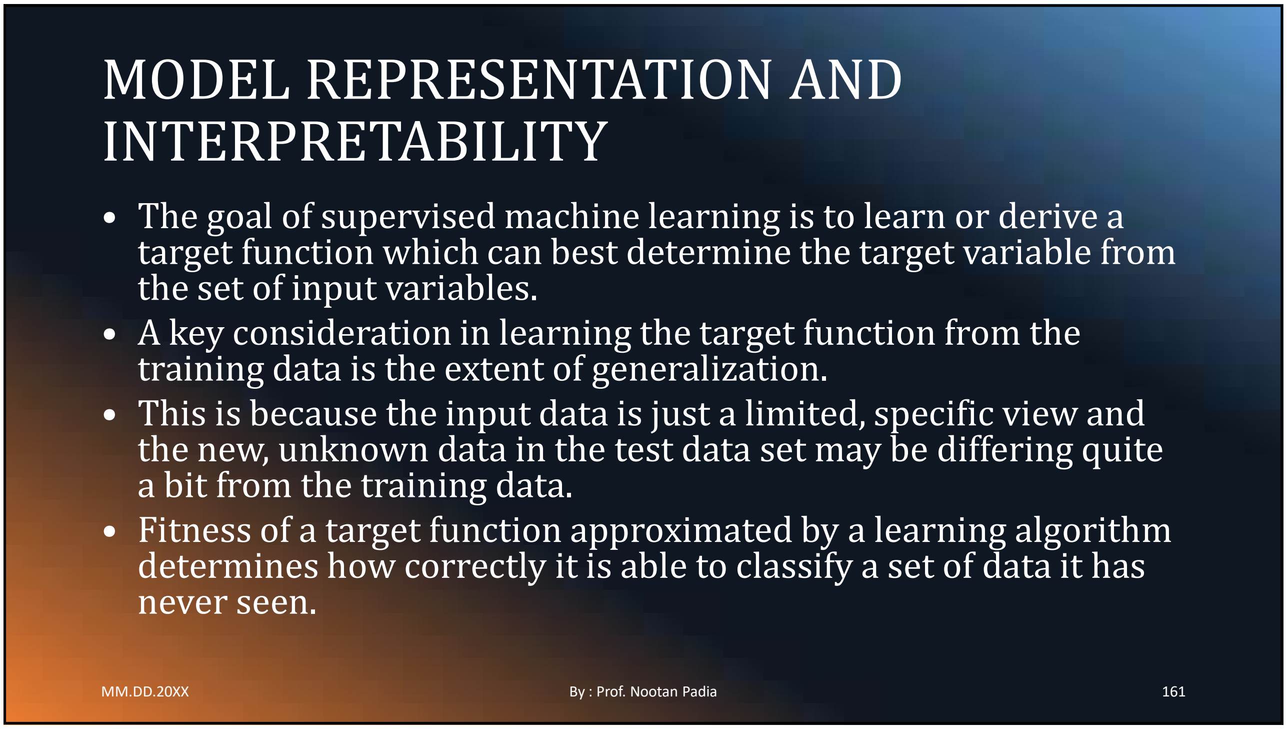
- Eager learning follows the general principles of machine learning — it tries to construct a generalized, input-independent target function during the model training phase.
- It follows the typical steps of machine learning, i.e. abstraction and generalization and comes up with a trained model at the end of the learning phase.
- Hence, when the test data comes in for classification, the eager learner is ready with the model and doesn't need to refer back to the training data.
- Eager learners take more time in the learning phase than the lazy learners.
- Some of the algorithms which adopt eager learning approach include Decision Tree, Support Vector Machine, Neural Network, etc.



- Lazy learning, on the other hand, completely skips the abstraction and generalization processes, as explained in context of a typical machine learning process.
- In that respect, strictly speaking, lazy learner doesn't 'learn' anything.
- It uses the training data in exact, and uses the knowledge to classify the unlabelled test data.
- Since lazy learning uses training data as-is, it is also known as rote learning (i.e. memorization technique based on repetition).
- Due to its heavy dependency on the given training data instance, it is also known as instance learning.
- They are also called non-parametric learning.
- Lazy learners take very little time in training because not much of training actually happens.

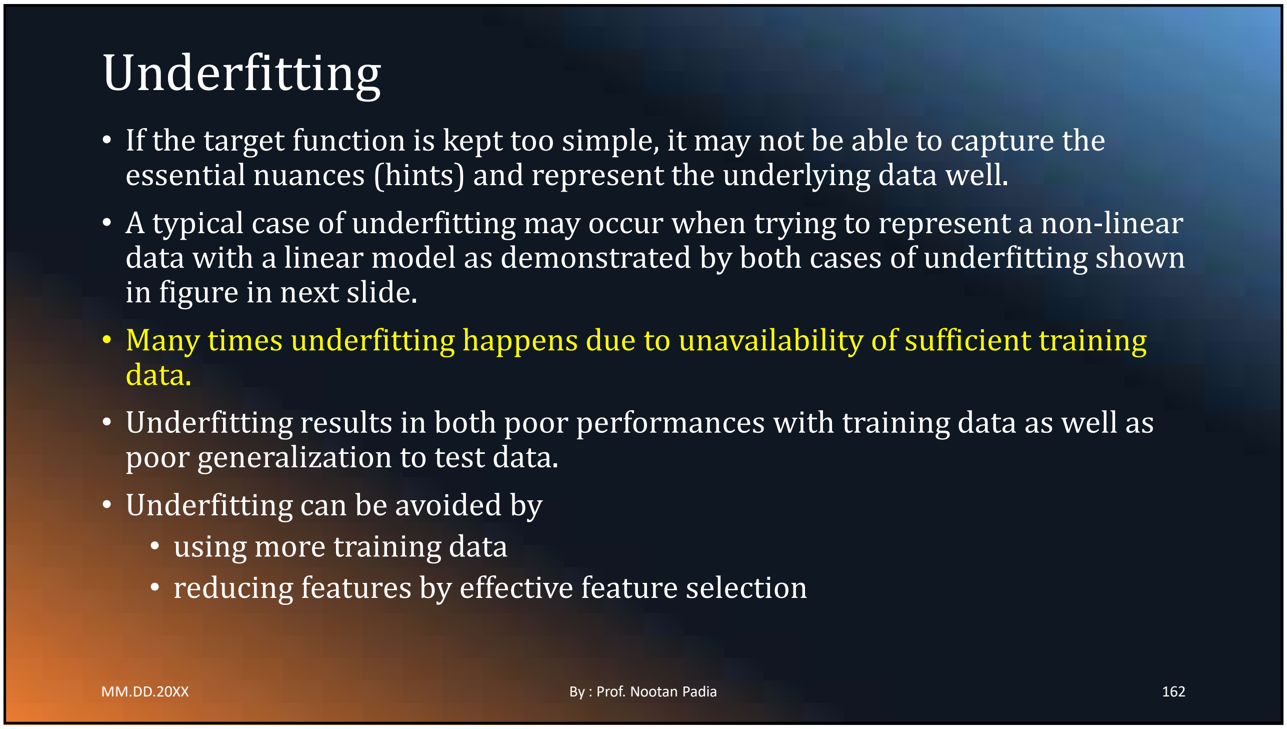


- However, it takes quite some time in classification as for each tuple of test data, a comparison-based assignment of label happens.
- One of the most popular algorithms for lazy learning is k-nearest neighbor.
- **Note:**
  - *Parametric learning models have finite number of parameters. In case of non-parametric models, quite contradicting to its name, the number of parameters is potentially infinite.*
  - *Models such as Linear Regression and Support Vector Machine, since the coefficients form the learning parameters, they are fixed in size. Hence, these models are clubbed as parametric.*
  - *On the other hand, in case of models such as k-nearest neighbour and decision tree, number of parameters grows with the size of training data. Hence they are considered as non-parametric learning models.*



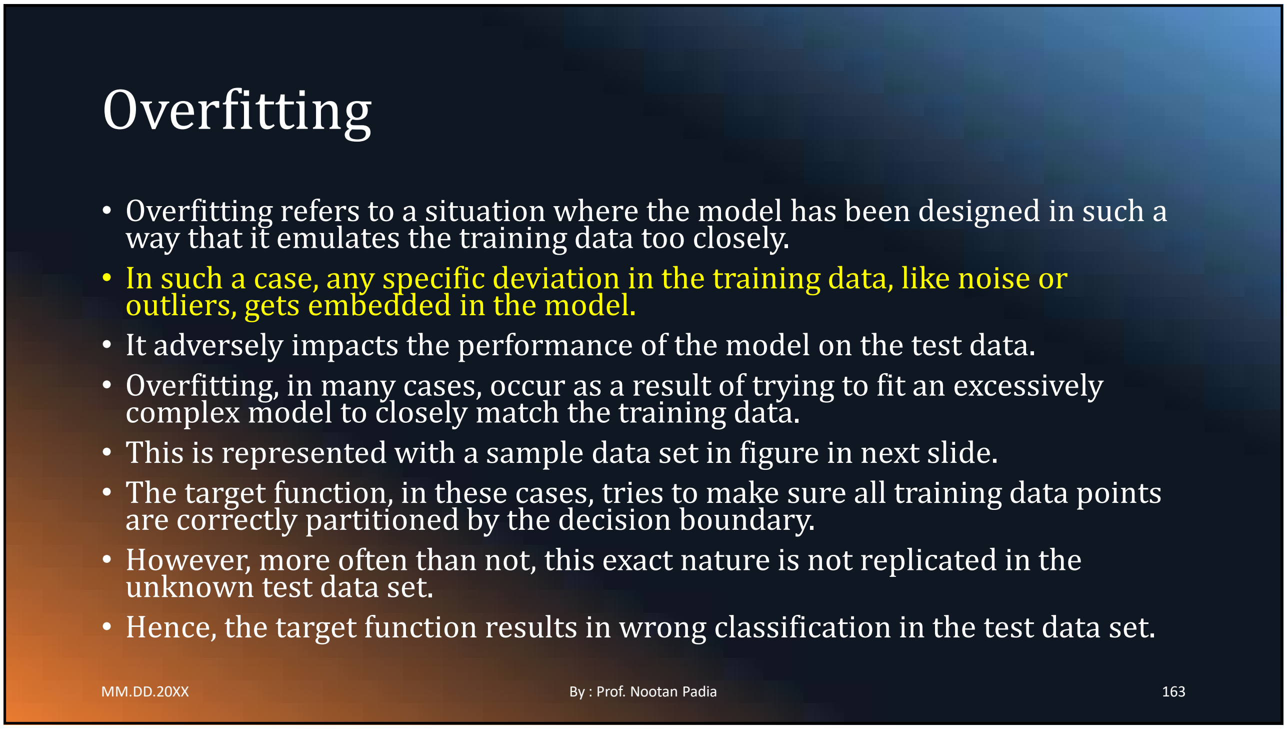
MODEL REPRESENTATION AND INTERPRETABILITY

- The goal of supervised machine learning is to learn or derive a target function which can best determine the target variable from the set of input variables.
- A key consideration in learning the target function from the training data is the extent of generalization.
- This is because the input data is just a limited, specific view and the new, unknown data in the test data set may be differing quite a bit from the training data.
- Fitness of a target function approximated by a learning algorithm determines how correctly it is able to classify a set of data it has never seen.



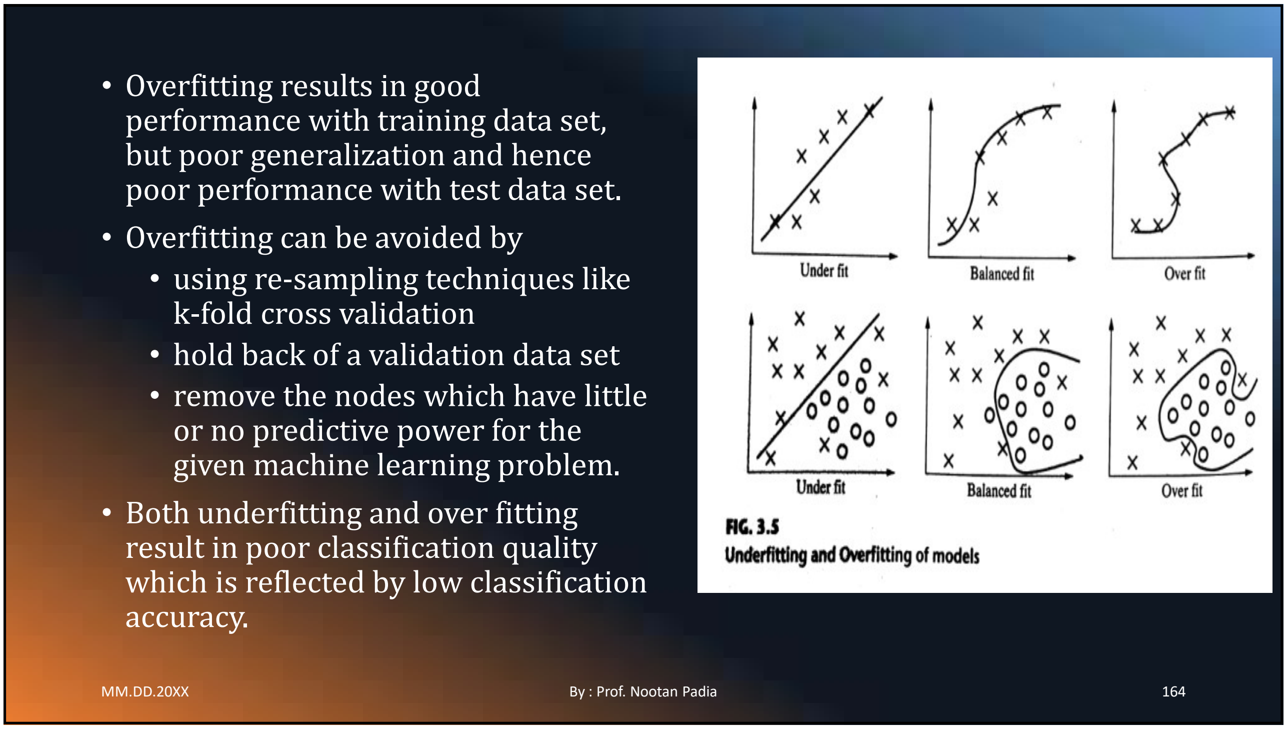
Underfitting

- If the target function is kept too simple, it may not be able to capture the essential nuances (hints) and represent the underlying data well.
- A typical case of underfitting may occur when trying to represent a non-linear data with a linear model as demonstrated by both cases of underfitting shown in figure in next slide.
- Many times underfitting happens due to unavailability of sufficient training data.
- Underfitting results in both poor performances with training data as well as poor generalization to test data.
- Underfitting can be avoided by
  - using more training data
  - reducing features by effective feature selection



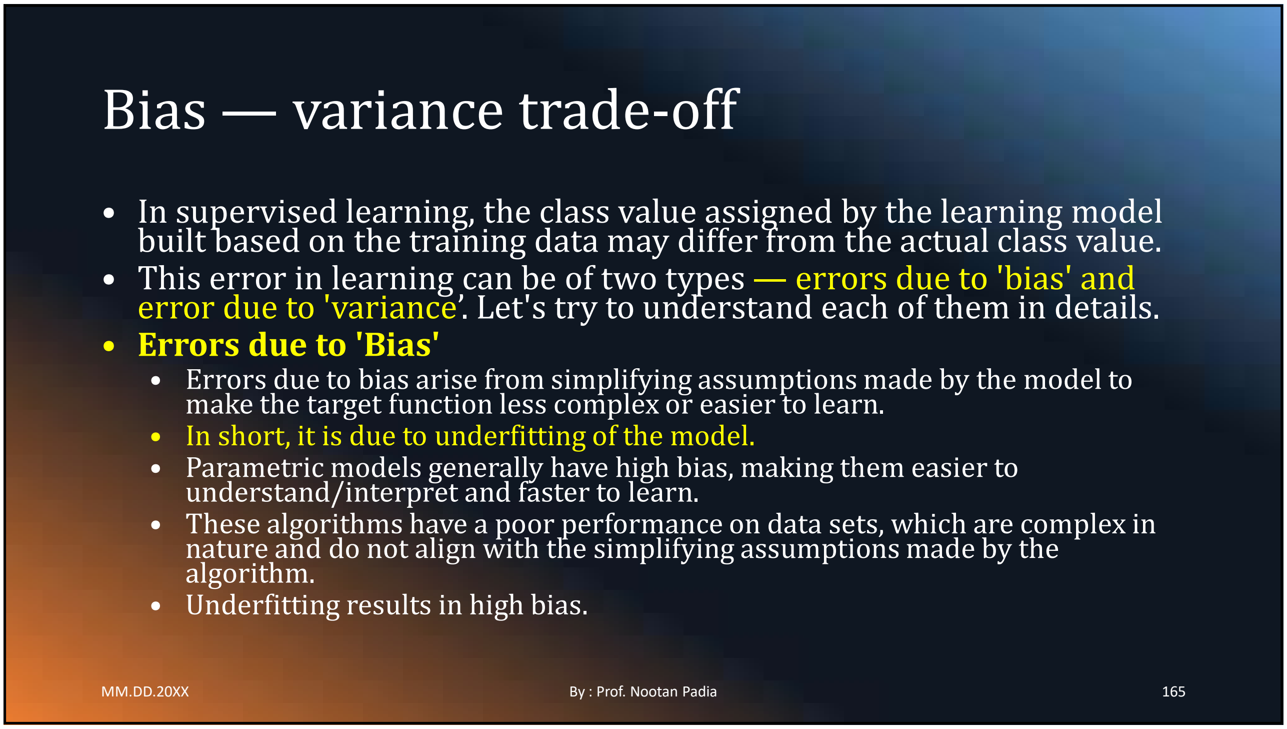
Overfitting

- Overfitting refers to a situation where the model has been designed in such a way that it emulates the training data too closely.
- In such a case, any specific deviation in the training data, like noise or outliers, gets embedded in the model.
- It adversely impacts the performance of the model on the test data.
- Overfitting, in many cases, occur as a result of trying to fit an excessively complex model to closely match the training data.
- This is represented with a sample data set in figure in next slide.
- The target function, in these cases, tries to make sure all training data points are correctly partitioned by the decision boundary.
- However, more often than not, this exact nature is not replicated in the unknown test data set.
- Hence, the target function results in wrong classification in the test data set.



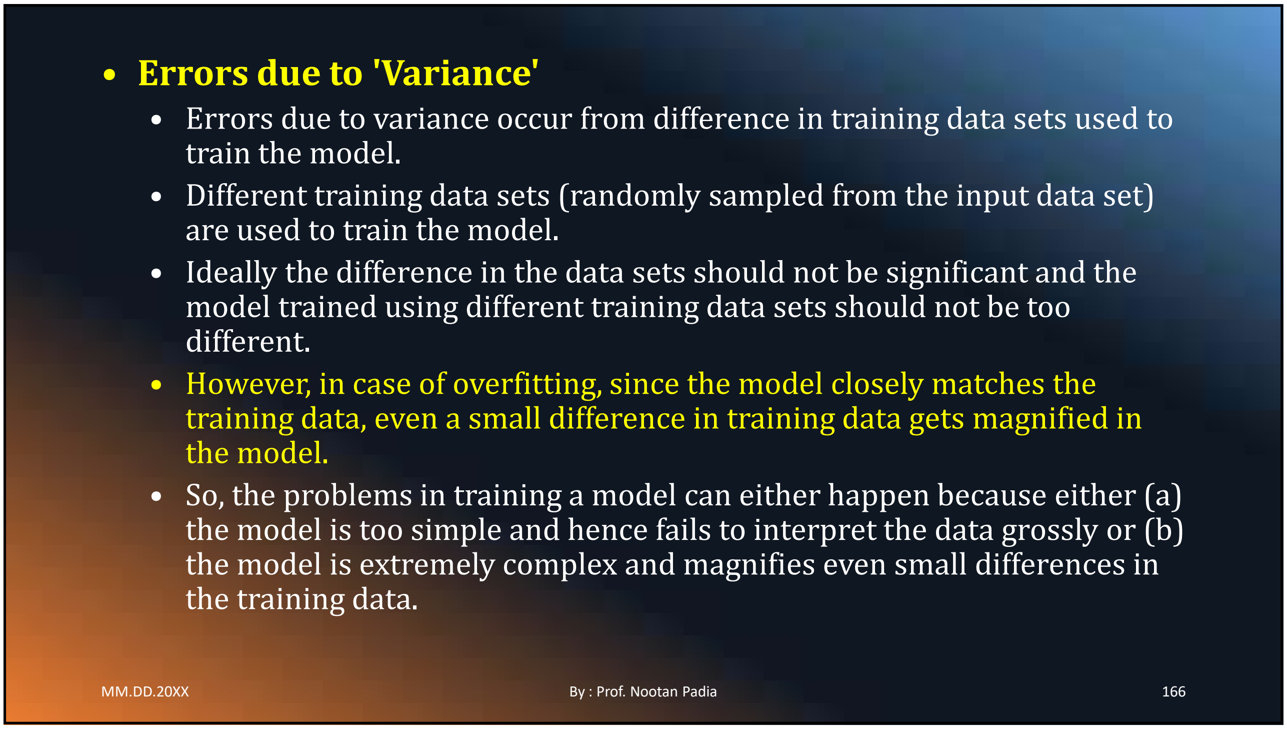
- Overfitting results in good performance with training data set, but poor generalization and hence poor performance with test data set.
- Overfitting can be avoided by
  - using re-sampling techniques like k-fold cross validation
  - hold back of a validation data set
  - remove the nodes which have little or no predictive power for the given machine learning problem.
- Both underfitting and over fitting result in poor classification quality which is reflected by low classification accuracy.

FIG. 3.5
Underfitting and Overfitting of models

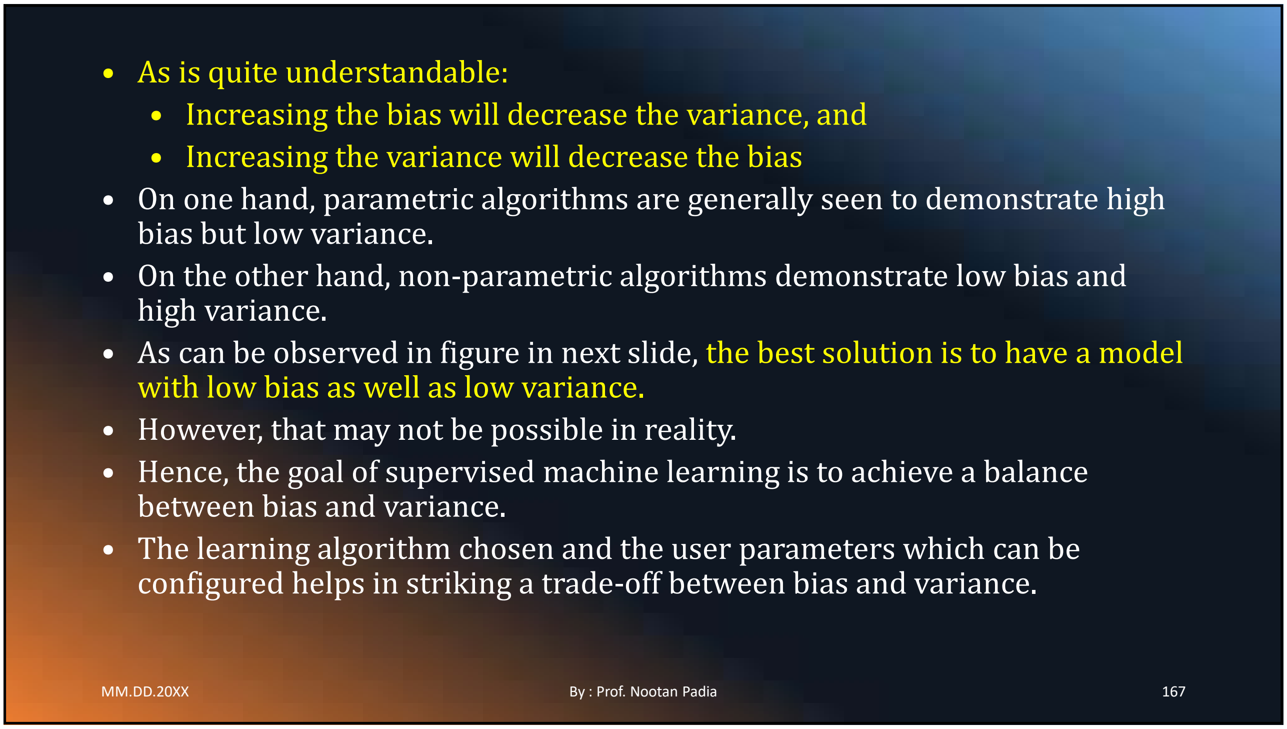


Bias — variance trade-off

- In supervised learning, the class value assigned by the learning model built based on the training data may differ from the actual class value.
- This error in learning can be of two types — errors due to 'bias' and error due to 'variance’. Let's try to understand each of them in details.
- **Errors due to 'Bias'**
  - Errors due to bias arise from simplifying assumptions made by the model to make the target function less complex or easier to learn.
  - In short, it is due to underfitting of the model.
  - Parametric models generally have high bias, making them easier to understand/interpret and faster to learn.
  - These algorithms have a poor performance on data sets, which are complex in nature and do not align with the simplifying assumptions made by the algorithm.
  - Underfitting results in high bias.



- **Errors due to 'Variance'**
  - Errors due to variance occur from difference in training data sets used to train the model.
  - Different training data sets (randomly sampled from the input data set) are used to train the model.
  - Ideally the difference in the data sets should not be significant and the model trained using different training data sets should not be too different.
  - However, in case of overfitting, since the model closely matches the training data, even a small difference in training data gets magnified in the model.
  - So, the problems in training a model can either happen because either (a) the model is too simple and hence fails to interpret the data grossly or (b) the model is extremely complex and magnifies even small differences in the training data.



- As is quite understandable:
  - Increasing the bias will decrease the variance, and
  - Increasing the variance will decrease the bias
- On one hand, parametric algorithms are generally seen to demonstrate high bias but low variance.
- On the other hand, non-parametric algorithms demonstrate low bias and high variance.
- As can be observed in figure in next slide, the best solution is to have a model with low bias as well as low variance.
- However, that may not be possible in reality.
- Hence, the goal of supervised machine learning is to achieve a balance between bias and variance.
- The learning algorithm chosen and the user parameters which can be configured helps in striking a trade-off between bias and variance.

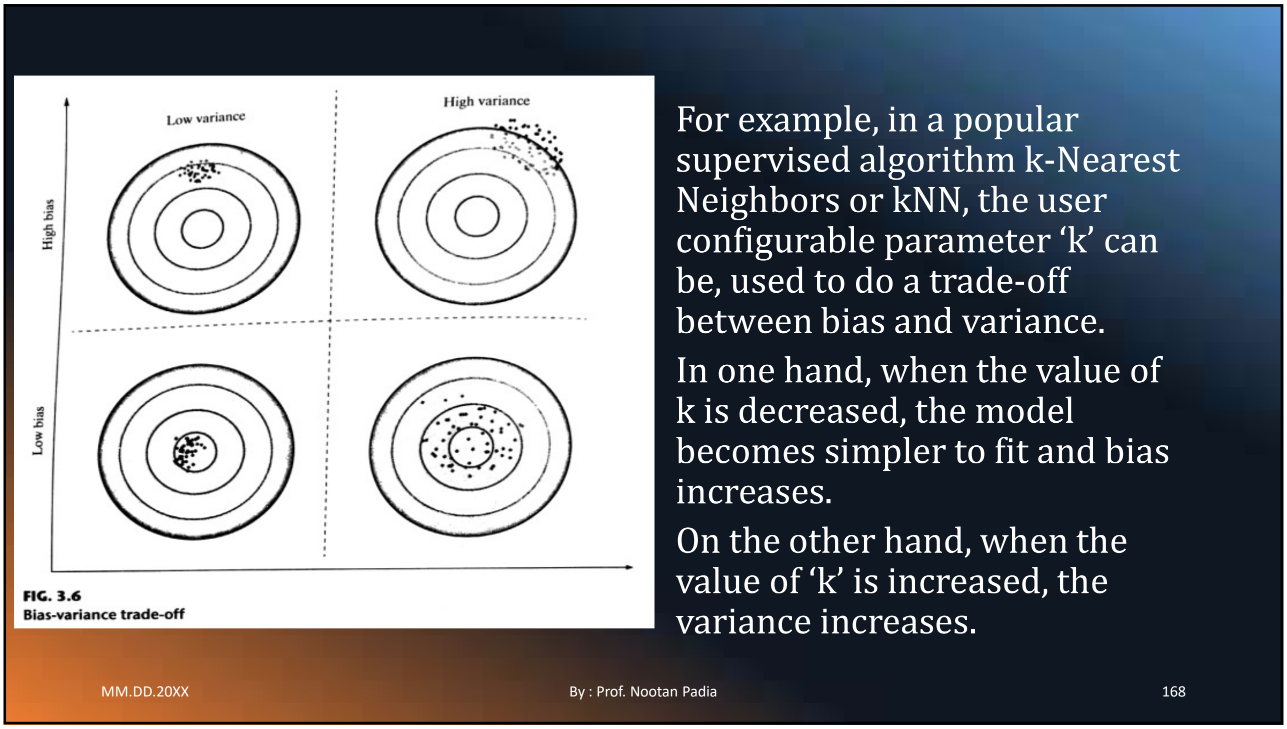
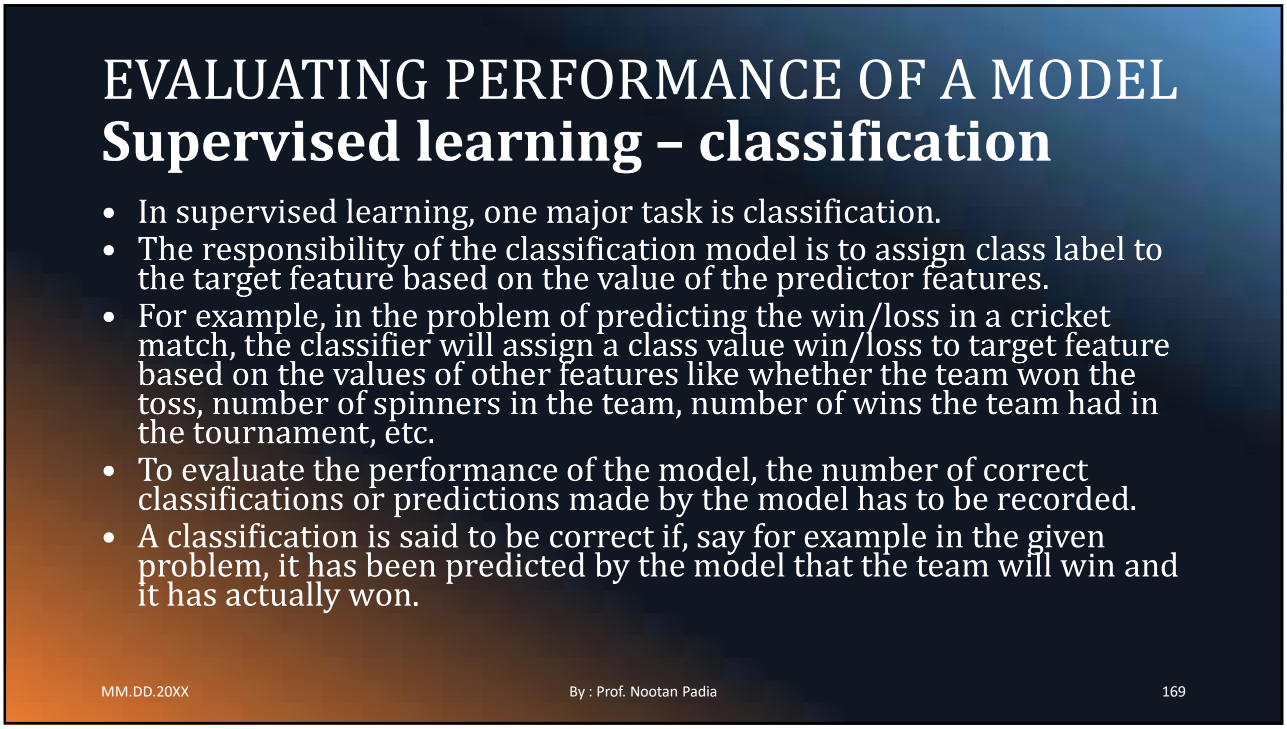


FIG. 3.6
Bias-variance trade-off

For example, in a popular supervised algorithm k-Nearest Neighbors or kNN, the user configurable parameter ‘k’ can be, used to do a trade-off between bias and variance.

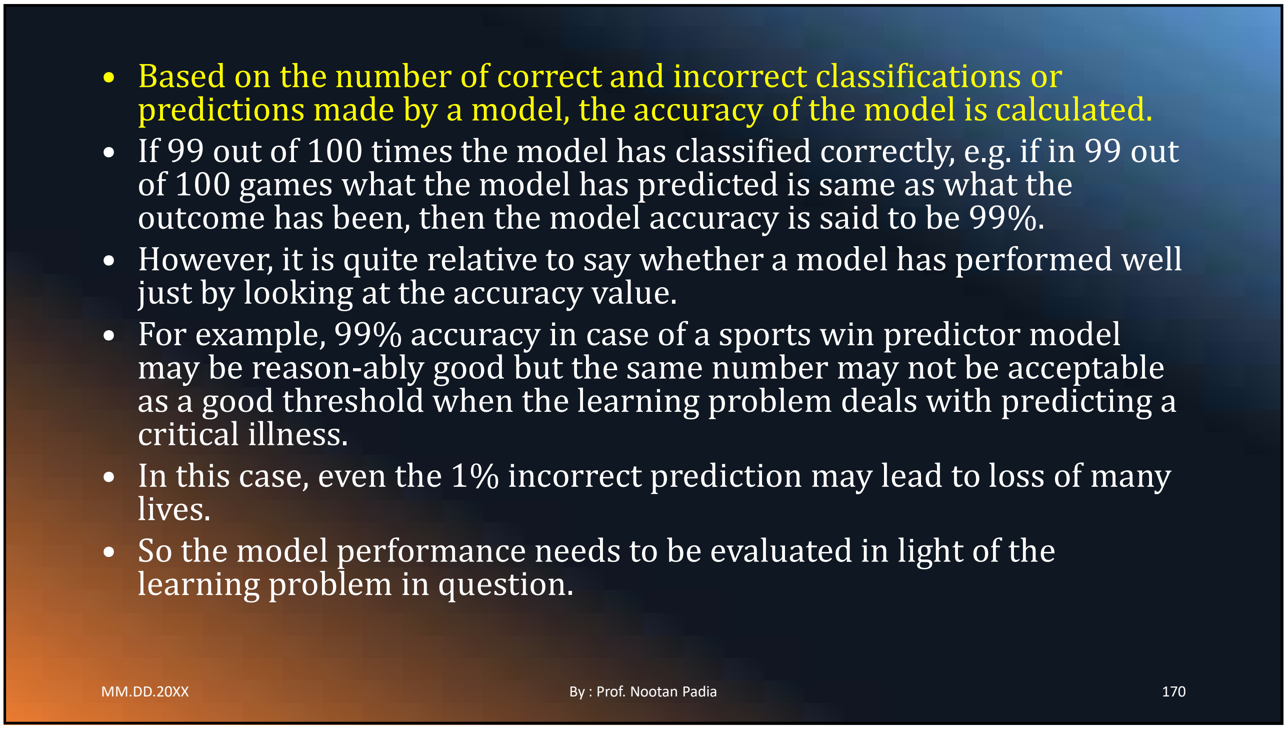
In one hand, when the value of k is decreased, the model becomes simpler to fit and bias increases.

On the other hand, when the value of ‘k’ is increased, the variance increases.

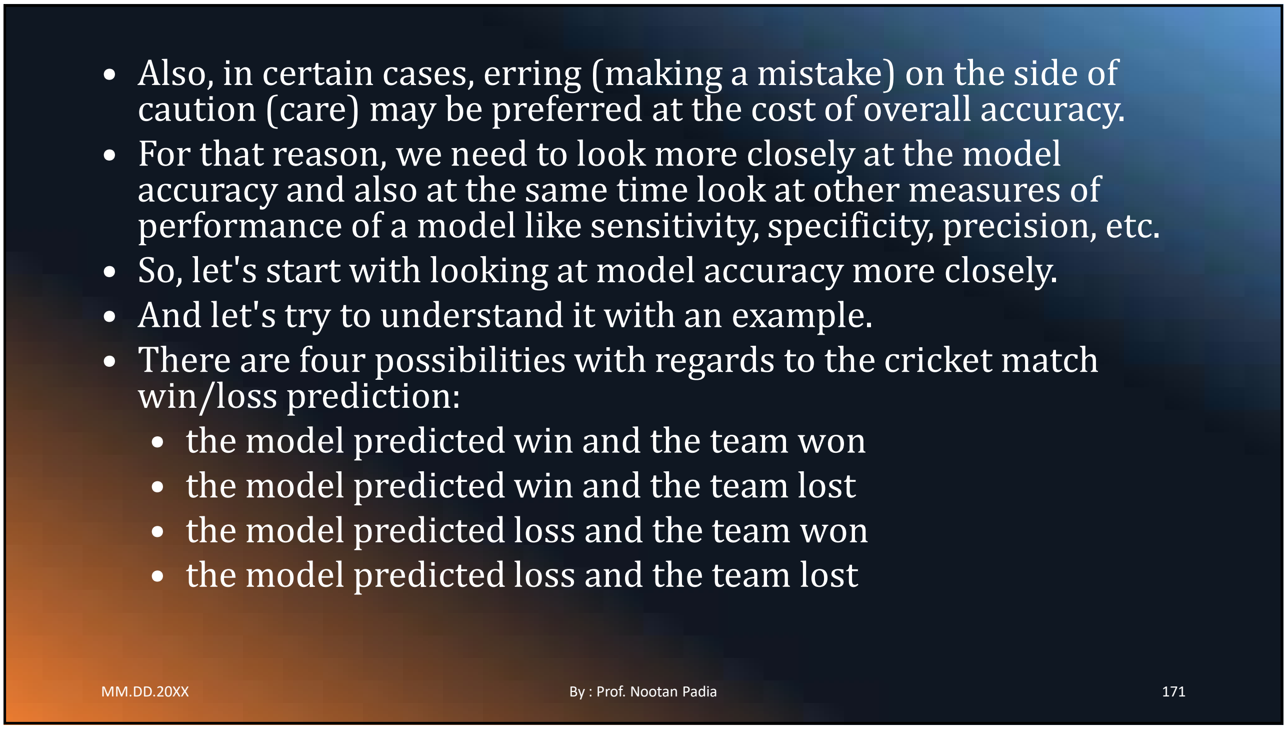


EVALUATING PERFORMANCE OF A MODEL
Supervised learning – classification

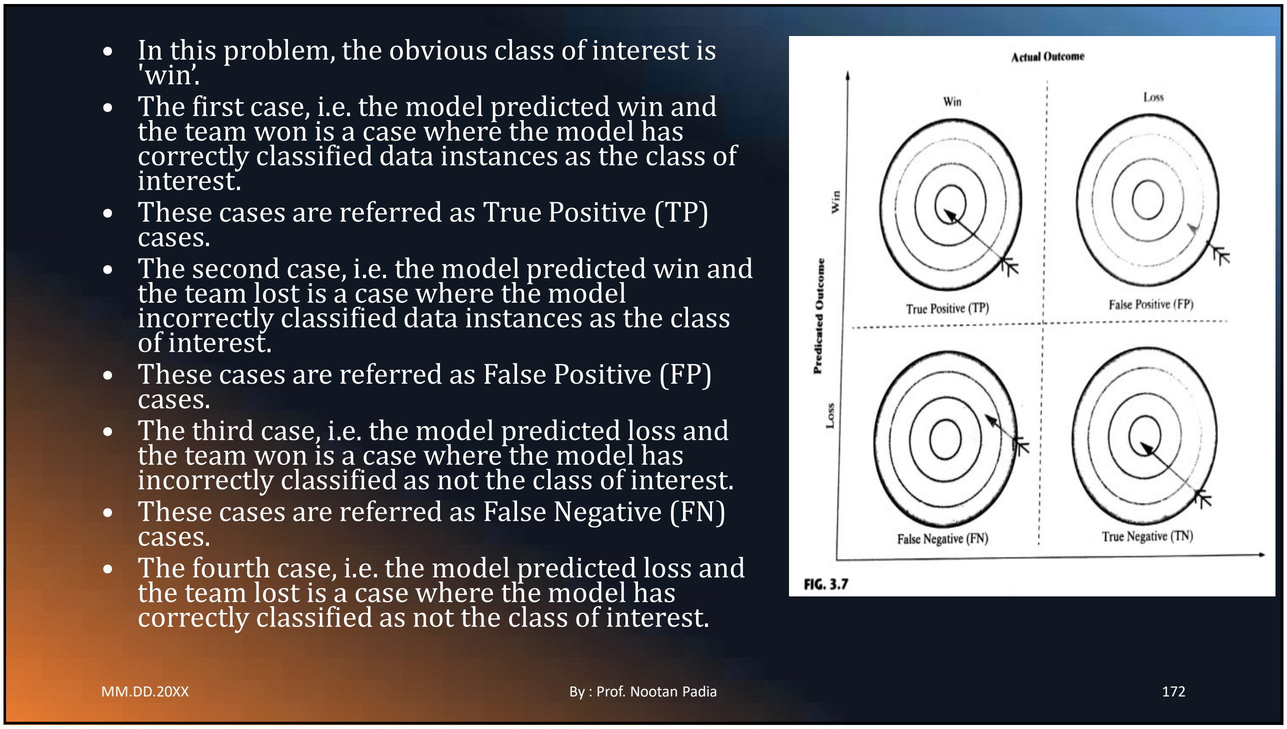
- In supervised learning, one major task is classification.
- The responsibility of the classification model is to assign class label to the target feature based on the value of the predictor features.
- For example, in the problem of predicting the win/loss in a cricket match, the classifier will assign a class value win/loss to target feature based on the values of other features like whether the team won the toss, number of spinners in the team, number of wins the team had in the tournament, etc.
- To evaluate the performance of the model, the number of correct classifications or predictions made by the model has to be recorded.
- A classification is said to be correct if, say for example in the given problem, it has been predicted by the model that the team will win and it has actually won.



- Based on the number of correct and incorrect classifications or predictions made by a model, the accuracy of the model is calculated.
- If 99 out of 100 times the model has classified correctly, e.g. if in 99 out of 100 games what the model has predicted is same as what the outcome has been, then the model accuracy is said to be 99%.
- However, it is quite relative to say whether a model has performed well just by looking at the accuracy value.
- For example, 99% accuracy in case of a sports win predictor model may be reason-ably good but the same number may not be acceptable as a good threshold when the learning problem deals with predicting a critical illness.
- In this case, even the 1% incorrect prediction may lead to loss of many lives.
- So the model performance needs to be evaluated in light of the learning problem in question.



- Also, in certain cases, erring (making a mistake) on the side of caution (care) may be preferred at the cost of overall accuracy.
- For that reason, we need to look more closely at the model accuracy and also at the same time look at other measures of performance of a model like sensitivity, specificity, precision, etc.
- So, let's start with looking at model accuracy more closely.
- And let's try to understand it with an example.
- There are four possibilities with regards to the cricket match win/loss prediction:
  - the model predicted win and the team won
  - the model predicted win and the team lost
  - the model predicted loss and the team won
  - the model predicted loss and the team lost

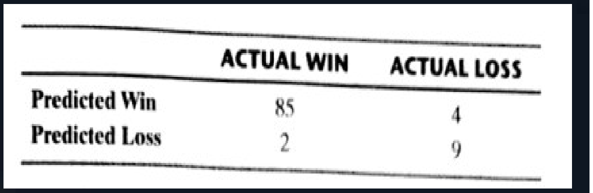


- In this problem, the obvious class of interest is 'win'.
- The first case, i.e. the model predicted win and the team won is a case where the model has correctly classified data instances as the class of interest.
- These cases are referred as True Positive (TP) cases.
- The second case, i.e. the model predicted win and the team lost is a case where the model incorrectly classified data instances as the class of interest.
- These cases are referred as False Positive (FP) cases.
- The third case, i.e. the model predicted loss and the team won is a case where the model has incorrectly classified as not the class of interest.
- These cases are referred as False Negative (FN) cases.
- The fourth case, i.e. the model predicted loss and the team lost is a case where the model has correctly classified as not the class of interest.

FIG. 3.7

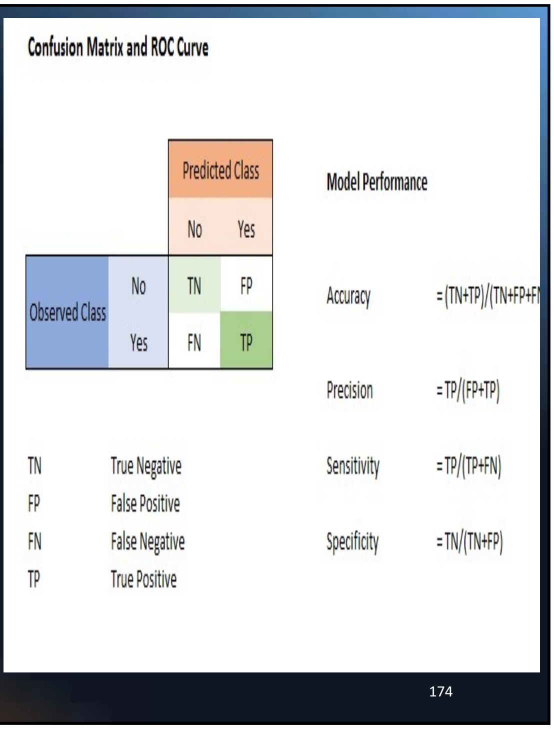
- These cases are referred as True Negative (TN) cases.
- All these four cases are depicted in Figure in previous slide.
- For any classification model, model accuracy is given by total number of correct classifications (either as the class of interest, i.e. True Positive or as not the class of interest, i.e. True Negative) divided by total number of classifications done.
- Model accuracy = (TP + TN)/ (TP + FP + FN + TN)
- A matrix containing correct and incorrect predictions in the form of TPs, FPs, FNs and TNs is known as confusion matrix.
- The win/loss prediction of cricket match has two classes of interest — win and loss.
- For that reason it will generate a 2 x 2 confusion matrix.
- For a classification problem involving three classes, the confusion matrix would be 3 x 3, etc.

- Calculate model accuracy, error rate, kappa value, Sensitivity, specificity, precision and recall for the following example :



	ACTUAL WIN	ACTUAL LOSS
Predicted Win	85	4
Predicted Loss	2	9

- A quick indicative interpretation of the predictive values from 0.5 to 1.0 is given below:
  - 0.5 — 0.6 → Almost no predictive ability
  - 0.6 — 0.7 → Weak predictive ability
  - 0.7 — 0.8 → Fair predictive ability
  - 0.8 — 0.9 → Good predictive ability
  - 0.9 — 1.0 → Excellent predictive ability



Confusion Matrix and ROC Curve

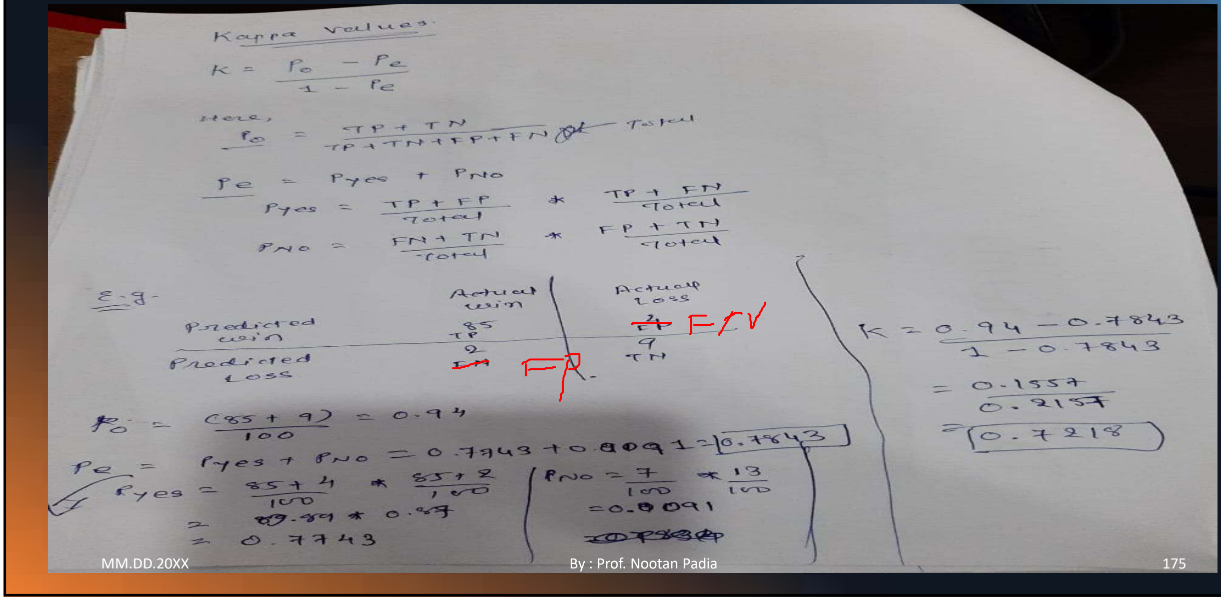
		Predicted Class	
		No	Yes
Observed Class	No	TN	FP
	Yes	FN	TP

Model Performance

Accuracy	=(TN+TP)/(TN+FP+F
Precision	=TP/(FP+TP)
Sensitivity	=TP/(TP+FN)
Specificity	=TN/(TN+FP)

TN	True Negative
FP	False Positive
FN	False Negative
TP	True Positive

Kappa Statistic compares the accuracy of the system to the accuracy of a random system.



Kappa values:

$K = \frac{P_o - P_e}{1 - P_e}$

Here,

$P_o = \frac{TP + TN}{TP + TN + FP + FN}$ or Total

$P_e = P_{yes} + P_{No}$

$P_{yes} = \frac{TP + FP}{Total} * \frac{TP + FN}{Total}$

$P_{No} = \frac{FN + TN}{Total} * \frac{FP + TN}{Total}$

E.g.

	Actual win	Actual loss
Predicted win	85 TP	4 ~~FP~~ FN ✓
Predicted loss	2 ~~FN~~ FP	9 TN

$P_o = \frac{(85 + 9)}{100} = 0.94$

$P_e = P_{yes} + P_{No} = 0.7743 + 0.0091 = \boxed{0.7843}$

$P_{yes} = \frac{85 + 4}{100} * \frac{85 + 2}{100}$

$= 0.89 * 0.87$

$= 0.7743$

$P_{No} = \frac{7}{100} * \frac{13}{100}$

$= 0.0091$

$K = \frac{0.94 - 0.7843}{1 - 0.7843}$

$= \frac{0.1557}{0.2157}$

$= \boxed{0.7218}$

Supervised learning - regression

- A well-fitted regression model churns out (mixes) predicted values close to actual values.
- Hence, a regression model which ensures that the difference between predicted and actual values is low can be considered as a good model.
- Figure in next slide represents a very simple problem of real estate value prediction solved using linear regression model.
- If 'area' is the predictor variable (say x) and 'value' is the target variable (say y), the linear regression model can be represented in the form:
- y = a + bx
- For a certain value of x, say x^, the value of y is predicted as y^ whereas the actual value of y is Y (say).
- The distance between the actual value and the fitted or predicted value, i.e. y^ is known as residual.

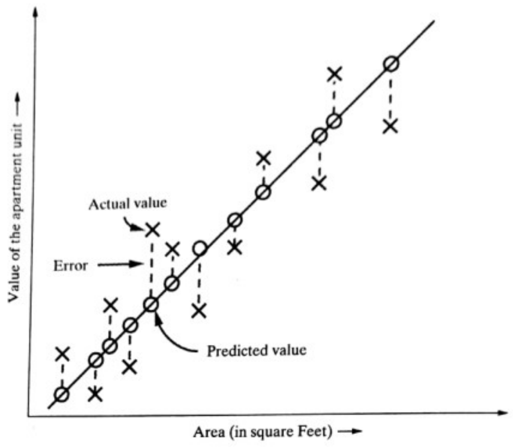


FIG. 3.9
Error – Predicted vs. actual value

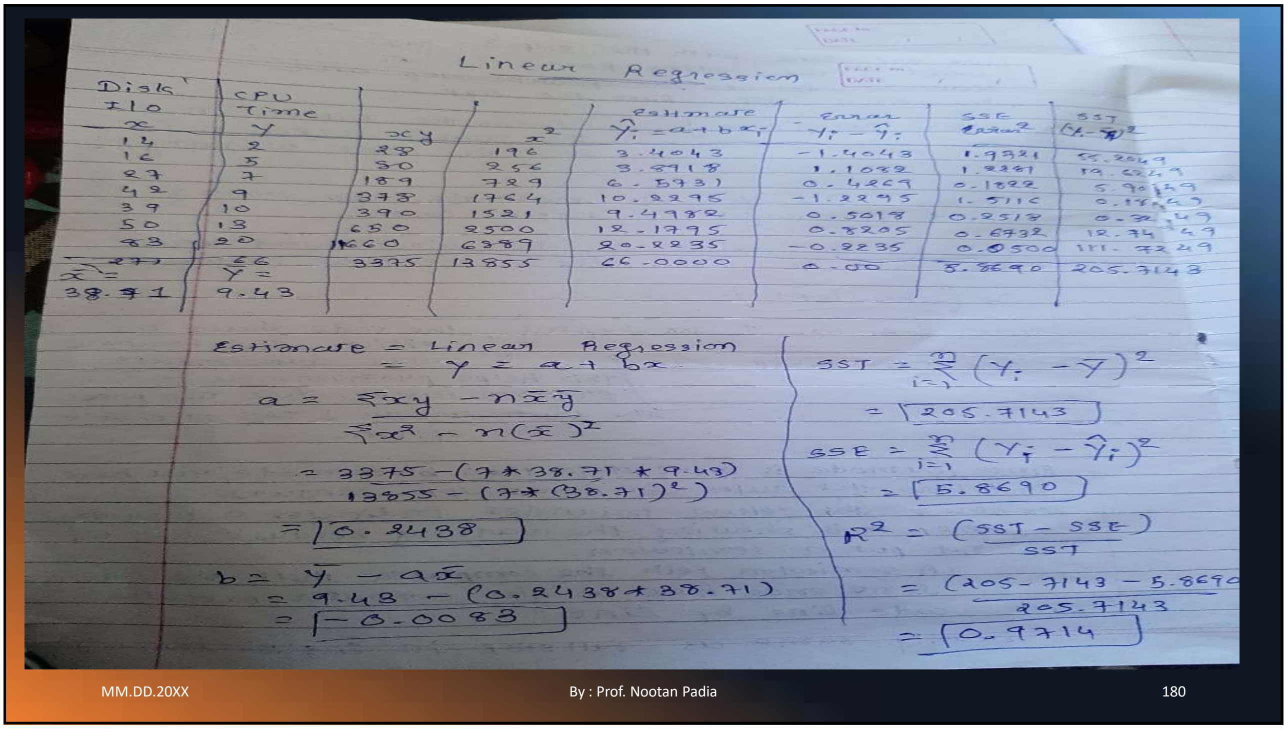
- The regression model can be considered to be fitted well if the difference between actual and predicted value, i.e. the residual value is less.
- R-squared is a good measure to evaluate the model fitness.
- It is also known as the coefficient of determination, or for multiple regression, the coefficient of multiple determination.
- The R-squared value lies between 0 to 1 (0%-100%) with a larger value representing a better fit.
- It is calculated as:

  $R^2 = (SST — SSE) / SST$
- Sum of Squares Total (SST) = squared differences of each observation from the overall mean = $\sum_{i=1}^{n}(y_i - \bar{y})^2$
- Here y is mean.

- Sum of Squared Errors (SSE) (of prediction) = sum of the squared residuals =

$$\sum_{i=1}^{n}(Y_i - \hat{y})^2$$

- Where y^ is the predicted value of y_i, and Y_i is the actual value of y_i.



Linear Regression

Disk I/O x	CPU Time y	xy	x^2	Estimate $\hat{Y}_i = a + bx_i$	Error $Y_i - \hat{Y}_i$	SSE $Error^2$	SST $(Y_i - \bar{Y})^2$
14	2	28	196	3.4043	-1.4043	1.9721	55.2049
16	5	80	256	3.8918	1.1082	1.2281	19.6249
27	7	189	729	6.5731	0.4269	0.1822	5.9049
42	9	378	1764	10.2295	-1.2295	1.5116	0.1849
39	10	390	1521	9.4982	0.5018	0.2518	0.3249
50	13	650	2500	12.1795	0.8205	0.6732	12.7449
83	20	1660	6889	20.2235	-0.2235	0.0500	111.7249
271	66	3375	13855	66.0000	0.00	5.8690	205.7143
$\bar{x}$ = 38.71	$\bar{Y}$ = 9.43						

Estimate = Linear Regression
= $y = a + bx$

$$a = \frac{\sum xy - n\bar{x}\bar{y}}{\sum x^2 - n(\bar{x})^2}$$

$$= \frac{3375 - (7 * 38.71 * 9.43)}{13855 - (7 * (38.71)^2)}$$

$$= \boxed{0.2438}$$

$$b = \bar{y} - a\bar{x}$$

$$= 9.43 - (0.2438 * 38.71)$$

$$= \boxed{-0.0083}$$

$$SST = \sum_{i=1}^{n}(Y_i - \bar{Y})^2$$

$$= \boxed{205.7143}$$

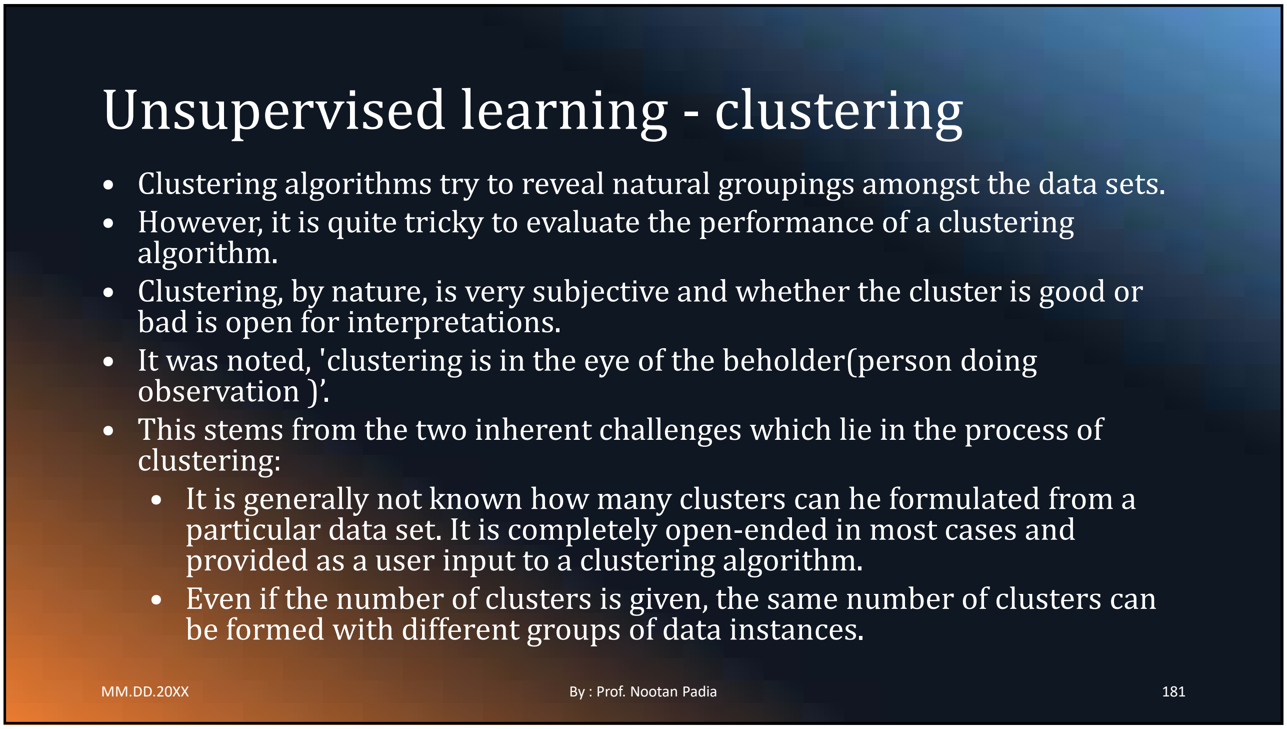
$$SSE = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$= \boxed{5.8690}$$

$$R^2 = \frac{(SST - SSE)}{SST}$$

$$= \frac{(205.7143 - 5.8690}{205.7143}$$

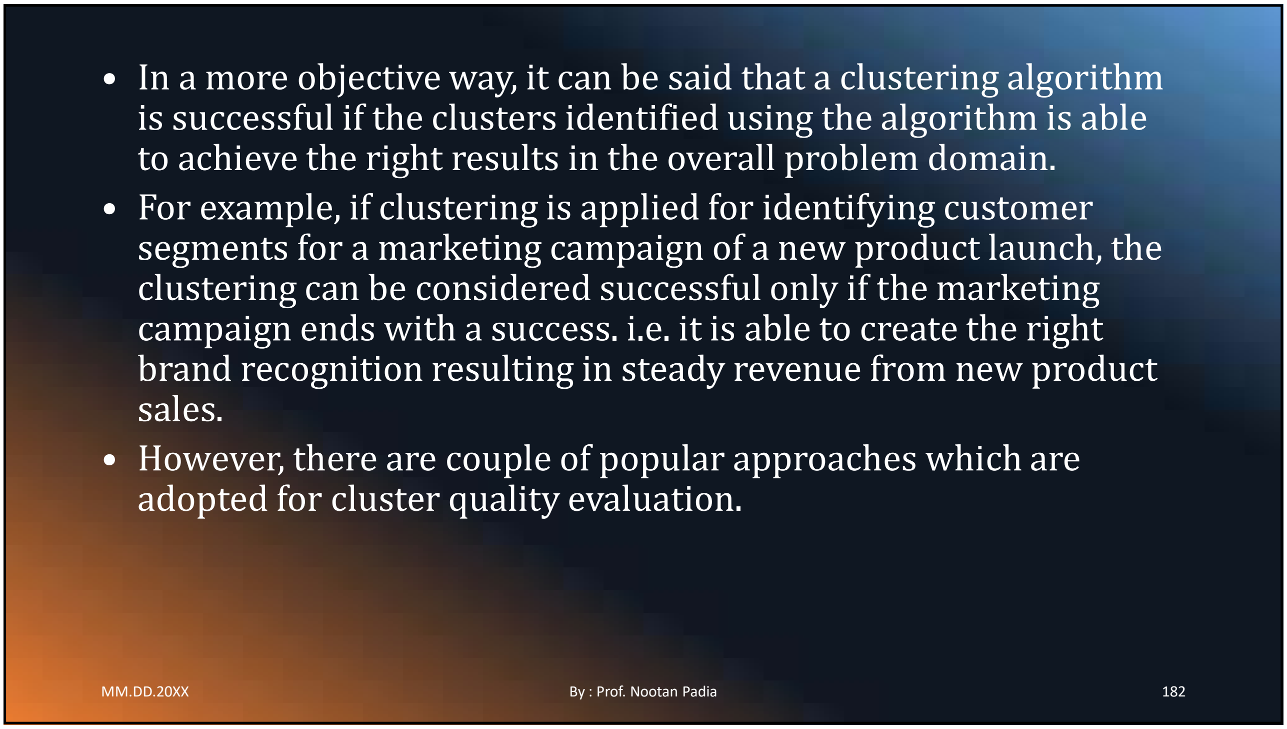
$$= \boxed{0.9714}$$



Unsupervised learning - clustering

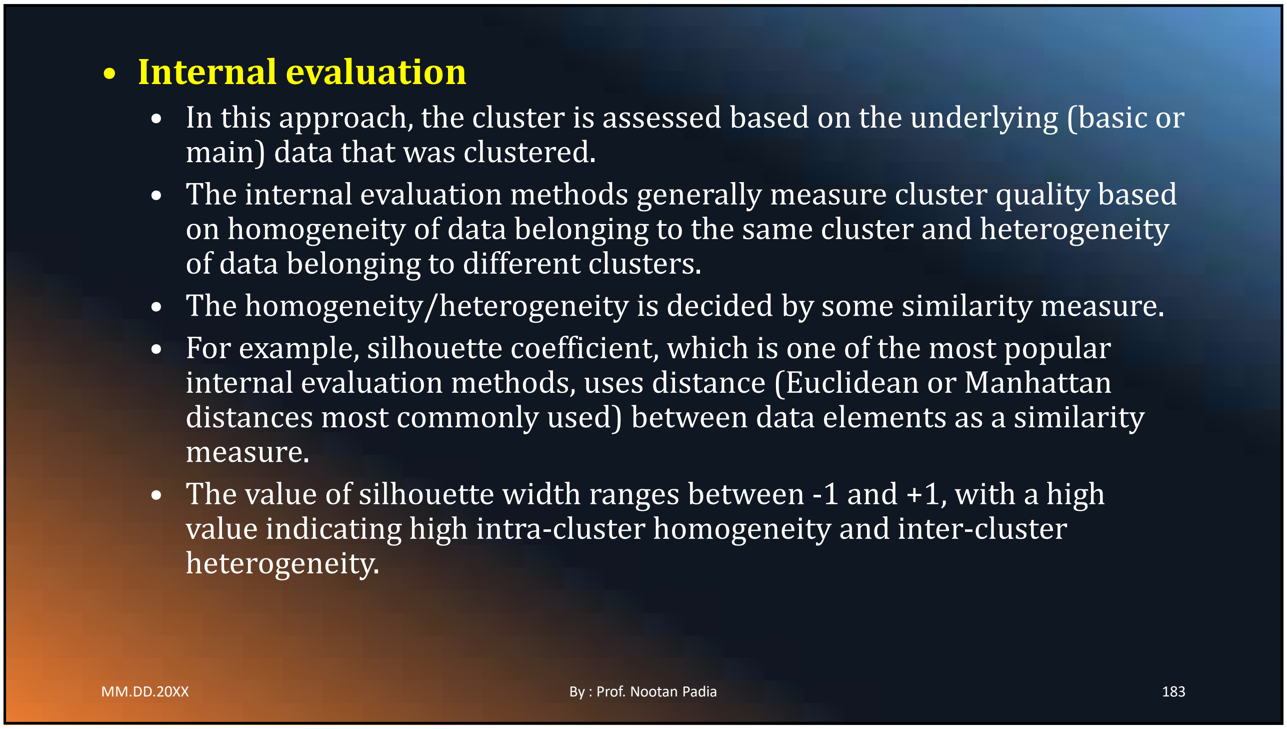
- Clustering algorithms try to reveal natural groupings amongst the data sets.
- However, it is quite tricky to evaluate the performance of a clustering algorithm.
- Clustering, by nature, is very subjective and whether the cluster is good or bad is open for interpretations.
- It was noted, 'clustering is in the eye of the beholder(person doing observation)'.
- This stems from the two inherent challenges which lie in the process of clustering:
  - It is generally not known how many clusters can he formulated from a particular data set. It is completely open-ended in most cases and provided as a user input to a clustering algorithm.
  - Even if the number of clusters is given, the same number of clusters can be formed with different groups of data instances.

MM.DD.20XX By : Prof. Nootan Padia 181

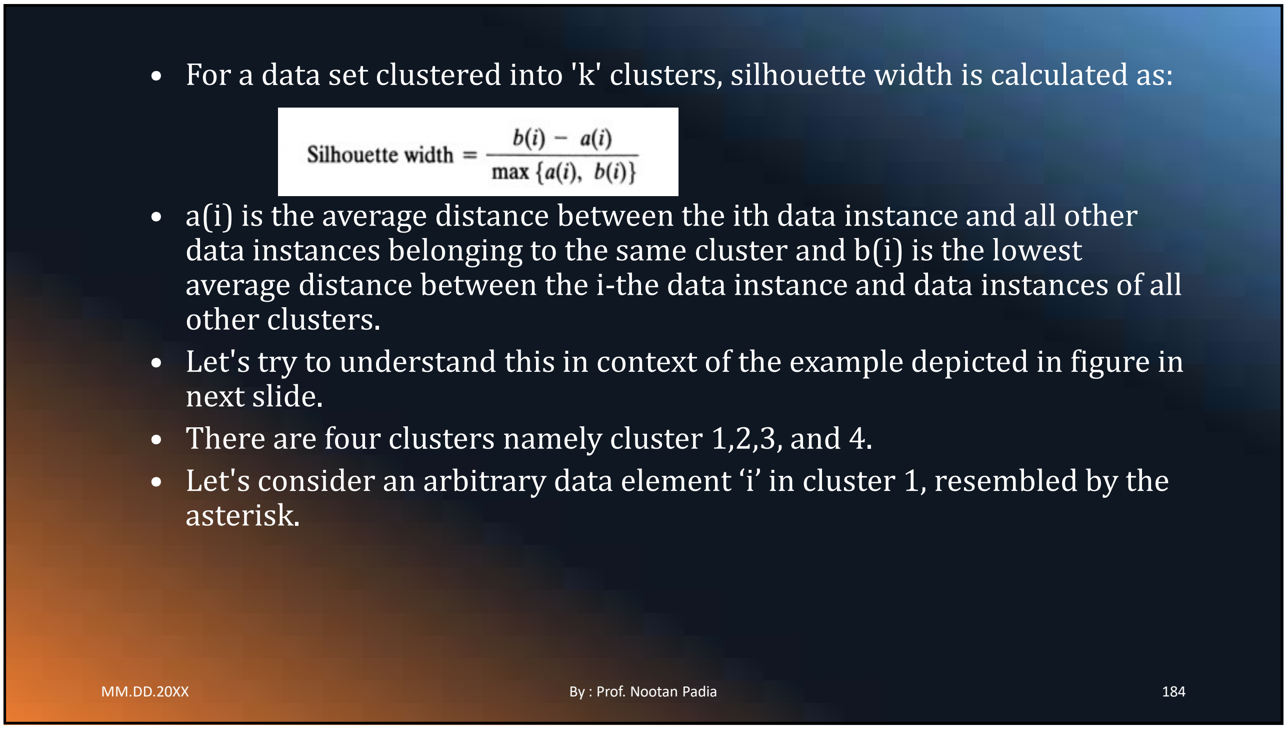


- In a more objective way, it can be said that a clustering algorithm is successful if the clusters identified using the algorithm is able to achieve the right results in the overall problem domain.
- For example, if clustering is applied for identifying customer segments for a marketing campaign of a new product launch, the clustering can be considered successful only if the marketing campaign ends with a success. i.e. it is able to create the right brand recognition resulting in steady revenue from new product sales.
- However, there are couple of popular approaches which are adopted for cluster quality evaluation.

MM.DD.20XX By : Prof. Nootan Padia 182



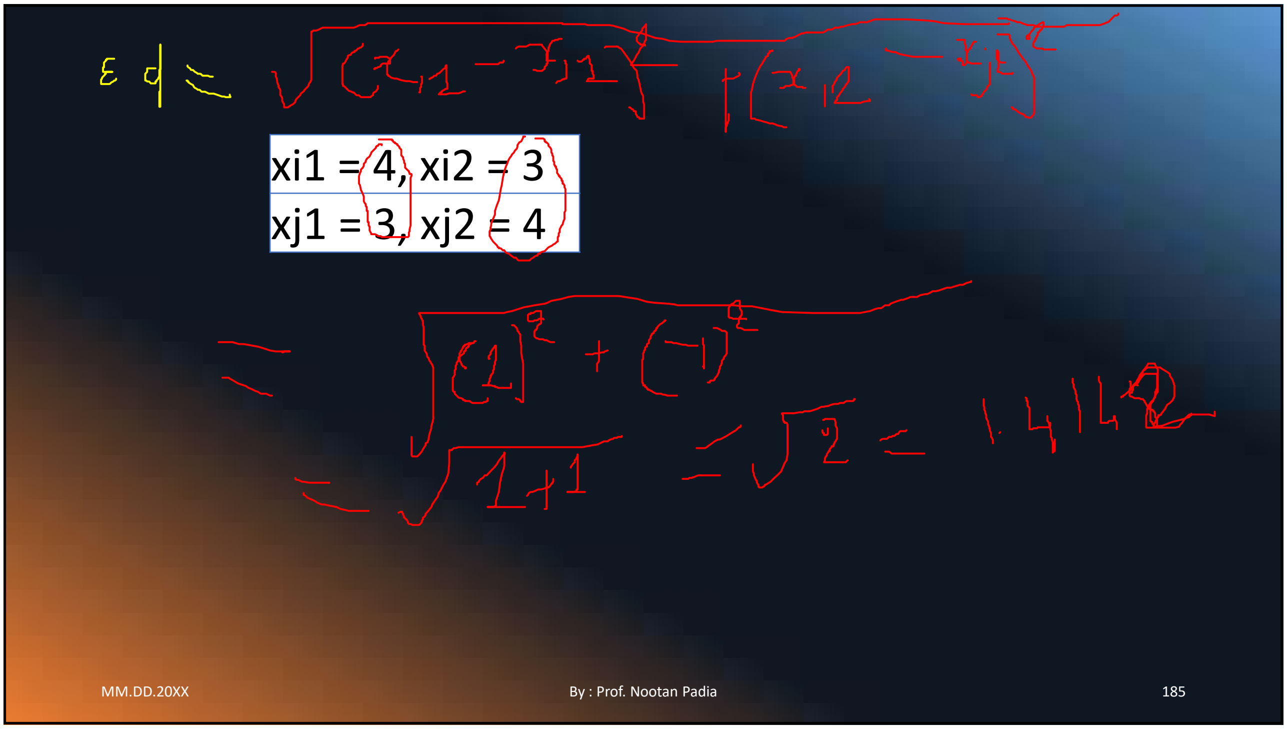
- **Internal evaluation**
  - In this approach, the cluster is assessed based on the underlying (basic or main) data that was clustered.
  - The internal evaluation methods generally measure cluster quality based on homogeneity of data belonging to the same cluster and heterogeneity of data belonging to different clusters.
  - The homogeneity/heterogeneity is decided by some similarity measure.
  - For example, silhouette coefficient, which is one of the most popular internal evaluation methods, uses distance (Euclidean or Manhattan distances most commonly used) between data elements as a similarity measure.
  - The value of silhouette width ranges between -1 and +1, with a high value indicating high intra-cluster homogeneity and inter-cluster heterogeneity.



- For a data set clustered into 'k' clusters, silhouette width is calculated as:

$$\text{Silhouette width} = \frac{b(i) - a(i)}{\max\{a(i),\ b(i)\}}$$

- a(i) is the average distance between the ith data instance and all other data instances belonging to the same cluster and b(i) is the lowest average distance between the i-the data instance and data instances of all other clusters.
- Let's try to understand this in context of the example depicted in figure in next slide.
- There are four clusters namely cluster 1,2,3, and 4.
- Let's consider an arbitrary data element ‘i’ in cluster 1, resembled by the asterisk.



$$Ed = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2}$$

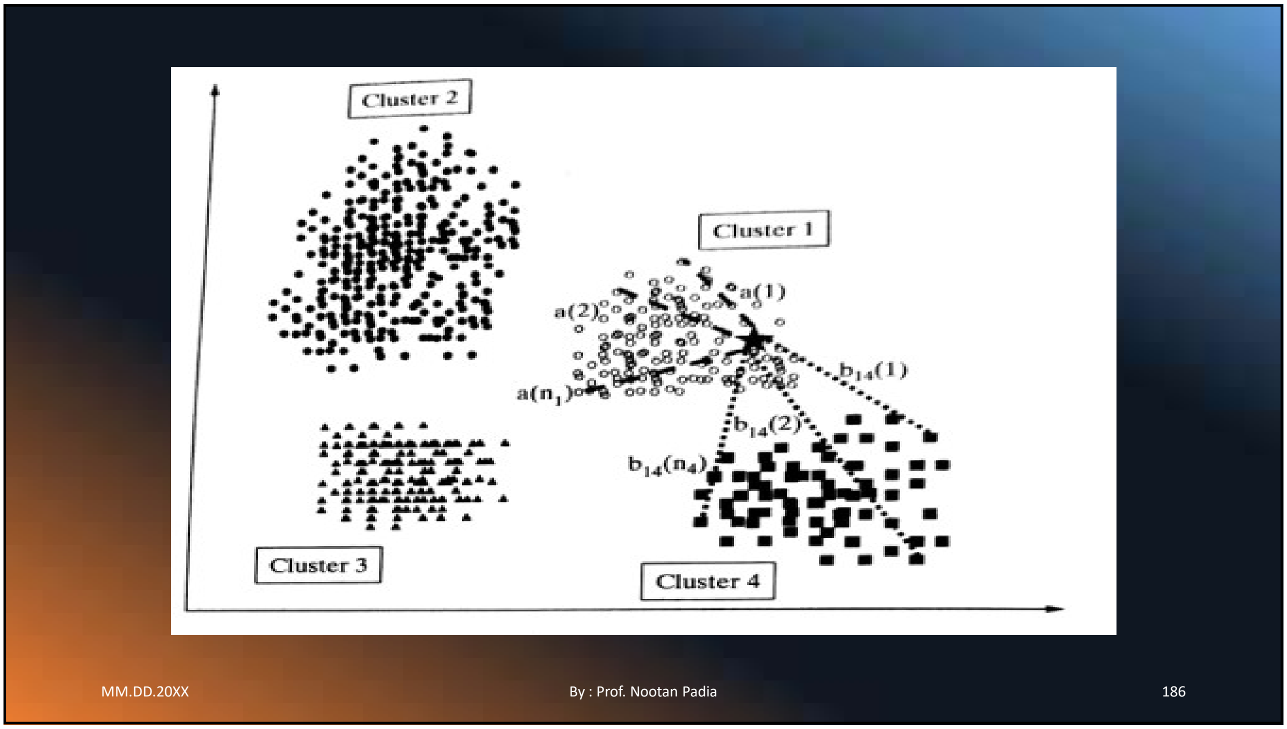
xi1 = 4, xi2 = 3

xj1 = 3, xj2 = 4

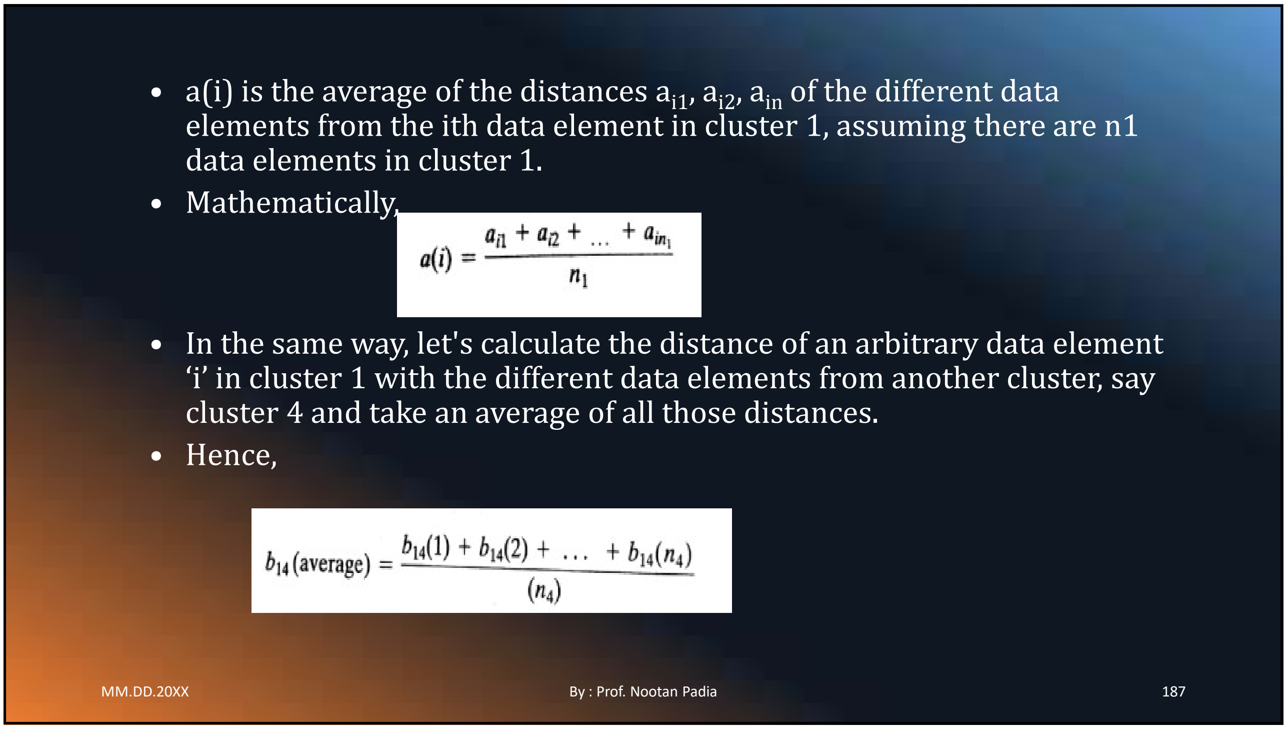
$$= \sqrt{(1)^2 + (-1)^2}$$

$$= \sqrt{1+1} = \sqrt{2} = 1.4142$$

MM.DD.20XX By : Prof. Nootan Padia 185



MM.DD.20XX By : Prof. Nootan Padia 186

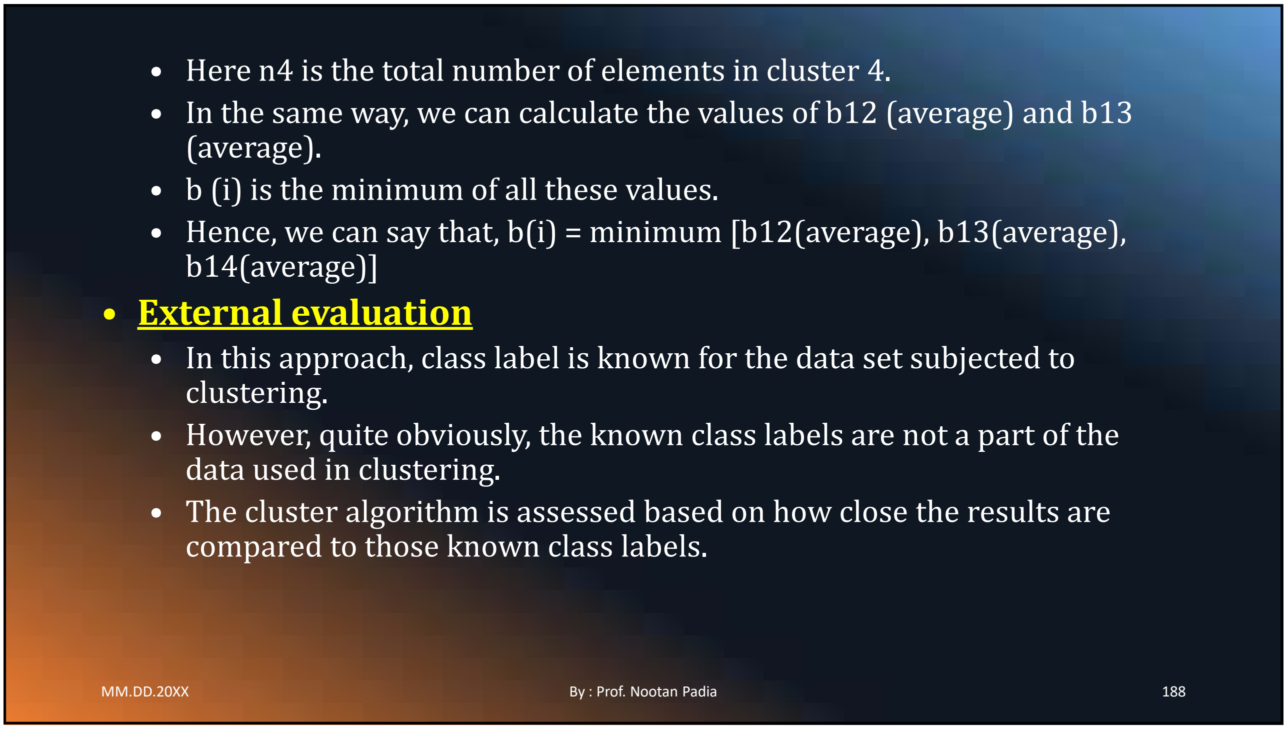


- a(i) is the average of the distances a_{i1}, a_{i2}, a_{in} of the different data elements from the ith data element in cluster 1, assuming there are n1 data elements in cluster 1.
- Mathematically,

$$a(i) = \frac{a_{i1} + a_{i2} + \ldots + a_{in_1}}{n_1}$$

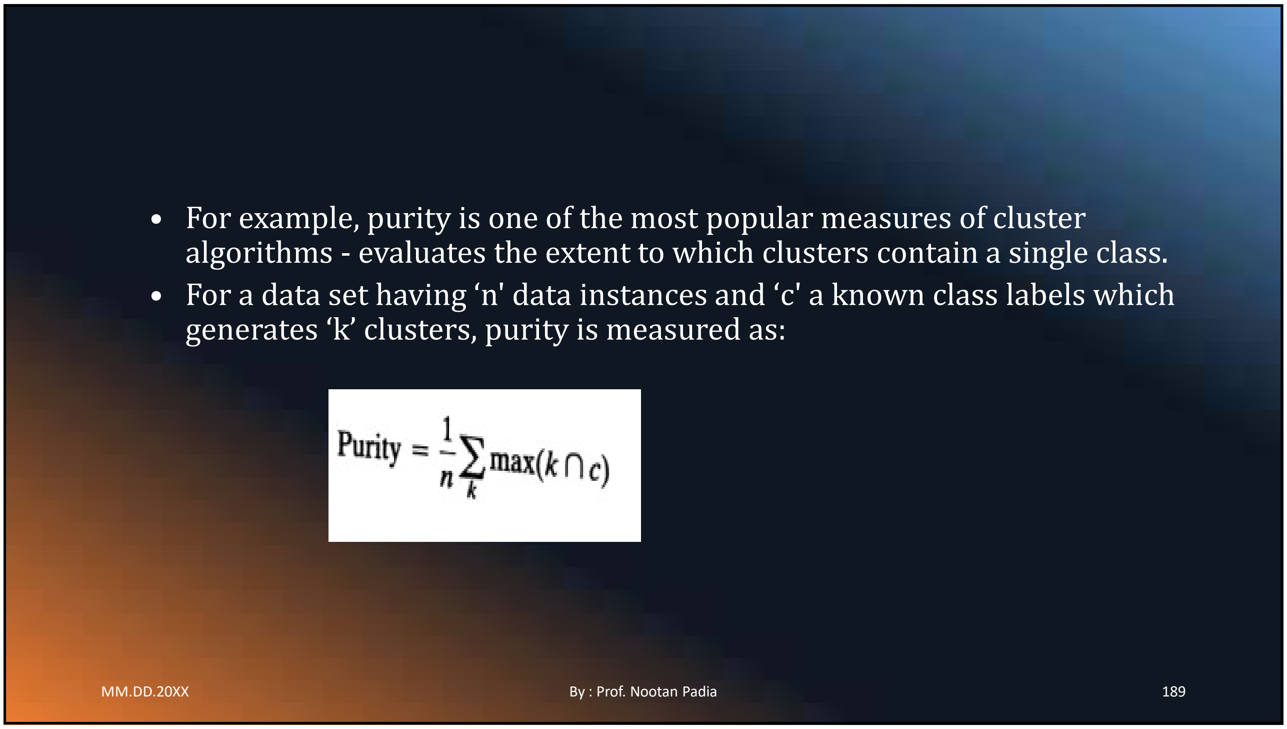
- In the same way, let's calculate the distance of an arbitrary data element 'i' in cluster 1 with the different data elements from another cluster, say cluster 4 and take an average of all those distances.
- Hence,

$$b_{14}(\text{average}) = \frac{b_{14}(1) + b_{14}(2) + \ldots + b_{14}(n_4)}{(n_4)}$$



- Here n4 is the total number of elements in cluster 4.
- In the same way, we can calculate the values of b12 (average) and b13 (average).
- b (i) is the minimum of all these values.
- Hence, we can say that, b(i) = minimum [b12(average), b13(average), b14(average)]

- **External evaluation**
  - In this approach, class label is known for the data set subjected to clustering.
  - However, quite obviously, the known class labels are not a part of the data used in clustering.
  - The cluster algorithm is assessed based on how close the results are compared to those known class labels.



- For example, purity is one of the most popular measures of cluster algorithms - evaluates the extent to which clusters contain a single class.
- For a data set having ‘n' data instances and ‘c' a known class labels which generates ‘k’ clusters, purity is measured as:

$$\text{Purity} = \frac{1}{n}\sum_{k}\max(k \cap c)$$