

Meeting now

26:01

Pop out

Chat

People574

Raise

React

View

More

Camera

Mic

Share

Leave

Teach Infinity II - Unnamed

group

1

2

3

ML —

(Power BI) → no code

⇒ data collection — understand of data — EDA — visualization

algorithms

↓

Build model — evaluate model — deployment

data Scientist

data Analyst

Uzma Sardar (Unverified)

Type here to search

27°C Partly sunny

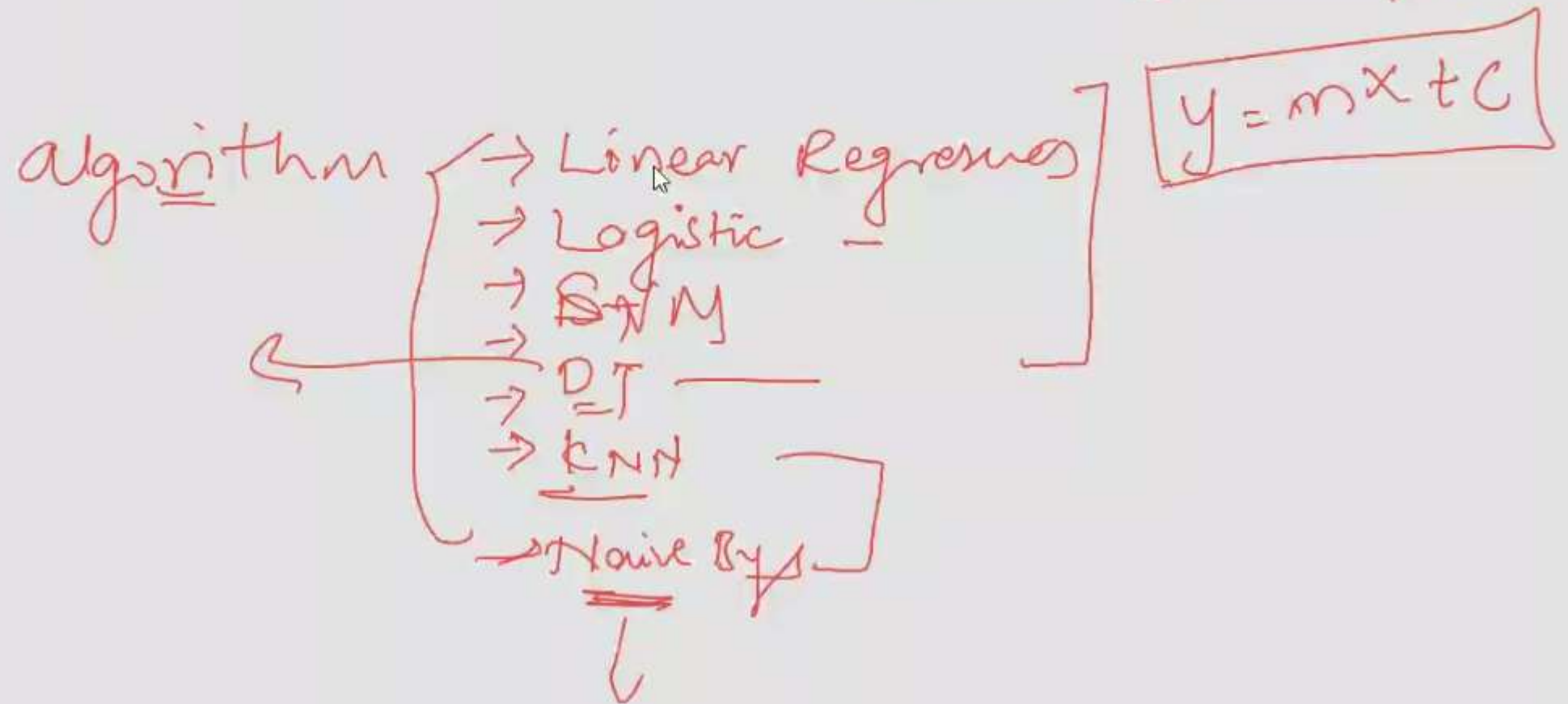
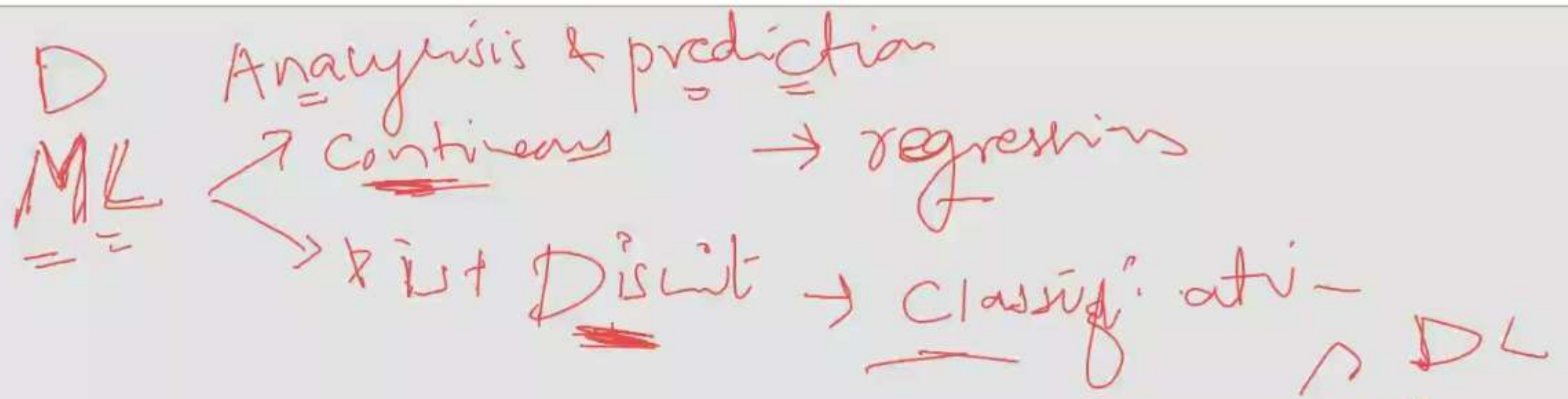
18:34

16-07-2024

Resources

250 GB

- ① ObjectStorage - Store
- ✓ ② WatsonStudio — data Analysis
- ③ Watson machine learning → algorithms





→ data Analyst

① data collection — scraping

② Understand → cloud  
→ python

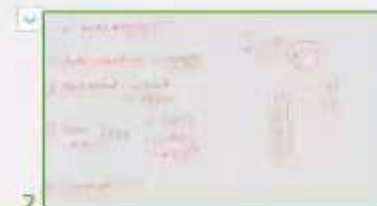
③ clean data } EDA  
→ Missing  
→ outlier  
→ duplicates

④ Visualization →

→ LG Why

① } ✓ ①  
② } ✓ ②  
③ }  
④ } →

group



Anshul A...



Pavan K...



Dishank



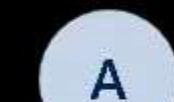
Sai Geet...



S SAI SA...



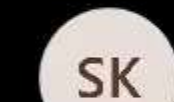
Piyush P...



Alan



Uzma Sa...



Sneha K...



Deepans...



Aditya S...



Ramnare...



Ananya ...



Satyam ...



Subrat G...



Dipan Ba...



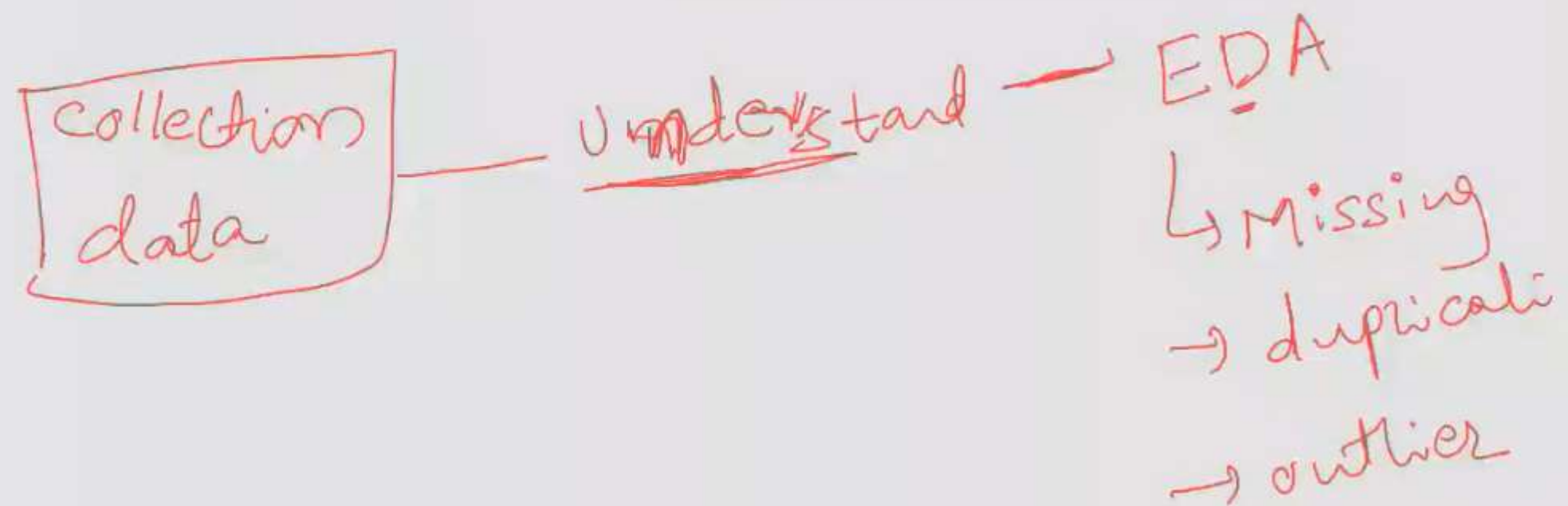
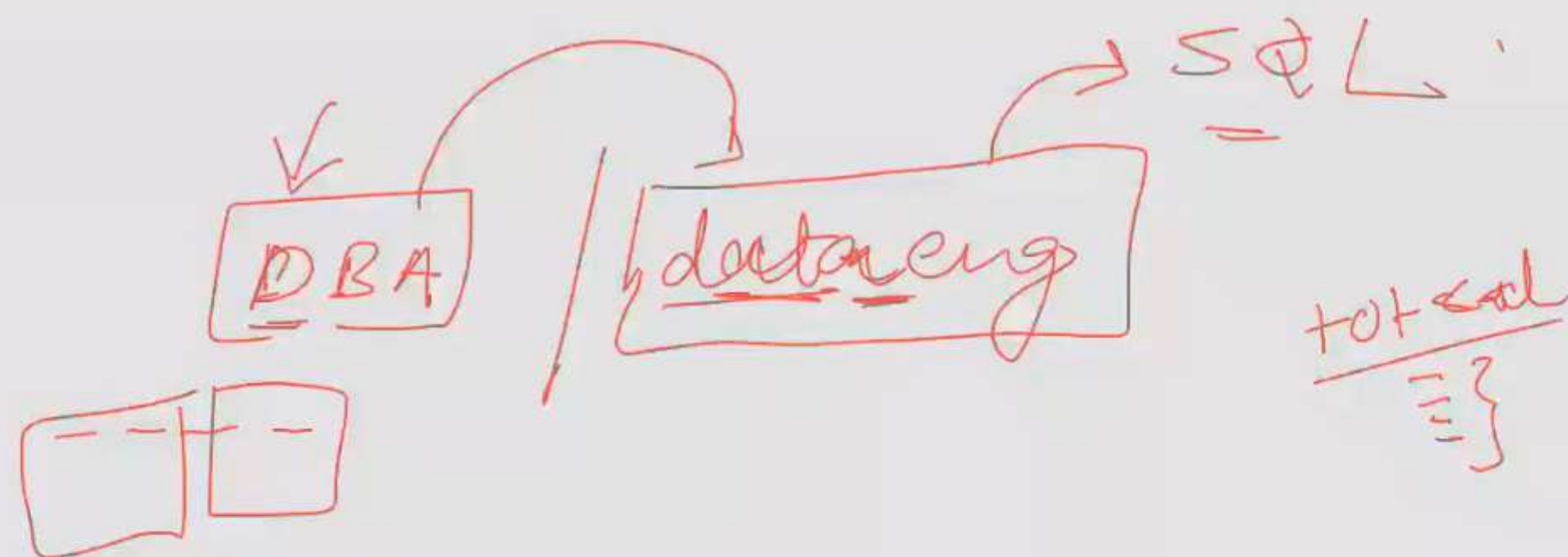
V



+579



ML eng  
DL eng  
data Eng





Teach Infinity II - Unnamed

Variable - columns  
Features

data collection =

handle

→ ① delete rows

② understanding →

③ EDA → missing value

Name	age	sales
a	25	250
b	26	350
c	29	400
d	NaN	NaN
e	29	800
f	<del>NaN</del>	

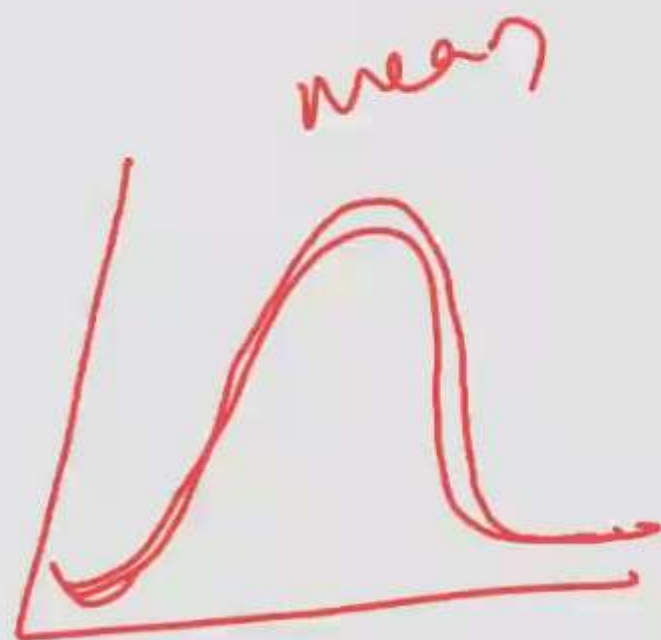
group



10,00 — 2%  
3%  
5% ]

② impute — fill

- ① mean / median / mode
- ② constant value
- ③ algorithm



age cordon

25  
27  
29  
30  
Na  
[29

group





# Model Building algorithm

→ LR

→ Logistic

→ SVM

→ DT

→ NB

→ KNN

$$\rightarrow y = mx + c$$

if-else

probability

distance

→ Euclidean

1

2

3

4

5



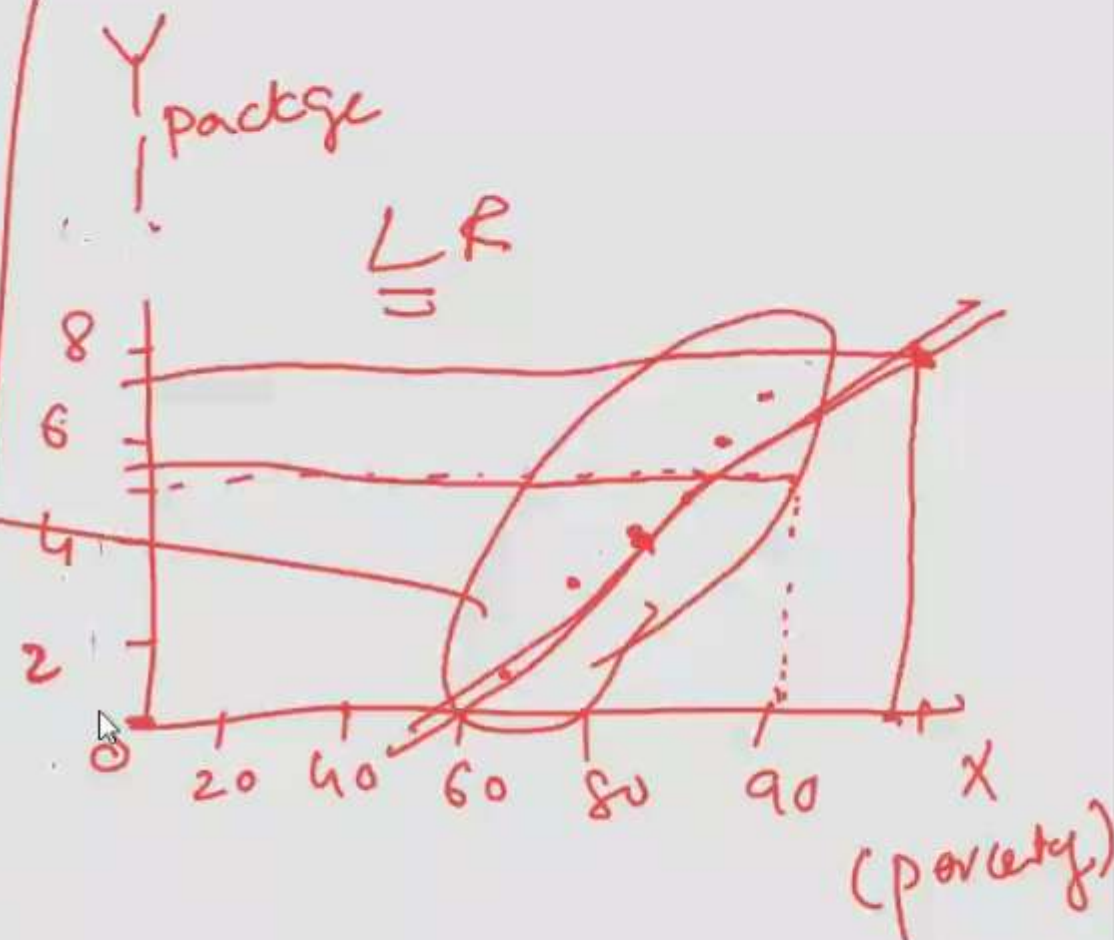
if  $ch =$   
 $\rightarrow DT$

91

Percentage	Package
80	3
89	3.5
90	4.5
95	5
97	5.5
75	2.5

9.8  
 $=$

6.5



2

3

4

5

6

Problem =

- Regression → O/P → Continuous
- Classification → Yes/No

Linear Regression - Regression  
Logistic Reg - Classification

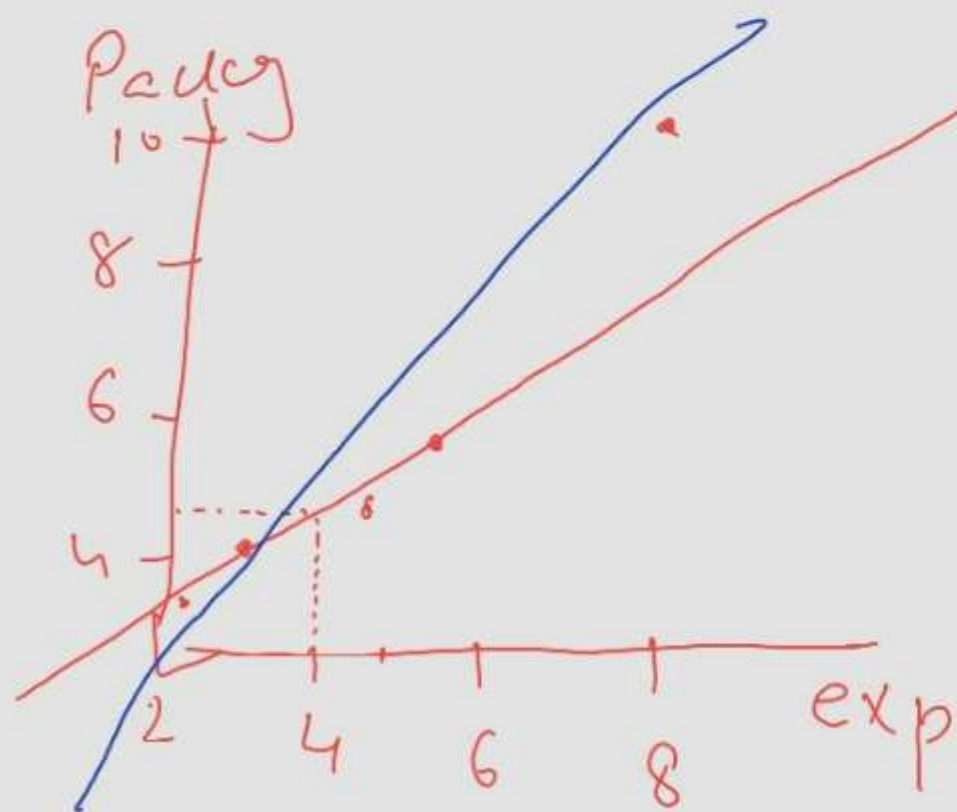
Skint learn =  
↓  
ML

3  
4  
5  
6  
7



$$y = \boxed{\textcircled{m}x + \textcircled{c}}$$

$\xrightarrow{\text{---} \times 4 \text{ ---}}$ 
 $\xrightarrow{\text{---} \text{---} \text{---}}$



$$y = mx + c$$

$$y = m \times 0 + c$$

$$y = c$$

exp	Package
2	3
3	4
5	4.5
6	5.5
8	10

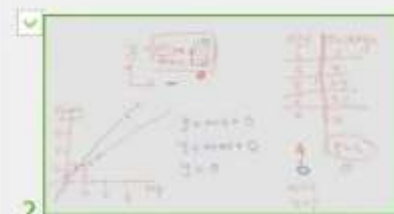
$$\begin{array}{r} 4 \\ - 0 \\ \hline \end{array}$$

$$x = 4$$

$$y = ?$$

$$\begin{array}{r} ? 4.2 \\ - 0 \\ \hline \end{array}$$

group



independent				Target / dependent
Weight	BP	BPI	insuli	<u>diabetes</u>

①

① ilocal [ - , - ]

drop

$$\chi = dF \cdot d\text{rop} \text{ (dielectric)}$$

~~X~~ =  $(x_1 + x_2 + x_3 + x_4)$

$$Y \sim \text{dF} ["di\_"]$$



→ 1000

	$x_1$	$x_2$	$x_3$	$y$
1				
2				
3				
4	-	-	-	1
5				0

Training - 70%  
80%

Testing - 30%  
20%

Pred

4	1
5	0

$x$	$y$
$x_{\text{-train}}$	$y_{\text{-train}}$
$x_{\text{-test}}$	$y_{\text{-test}}$

100%

evaluate

3

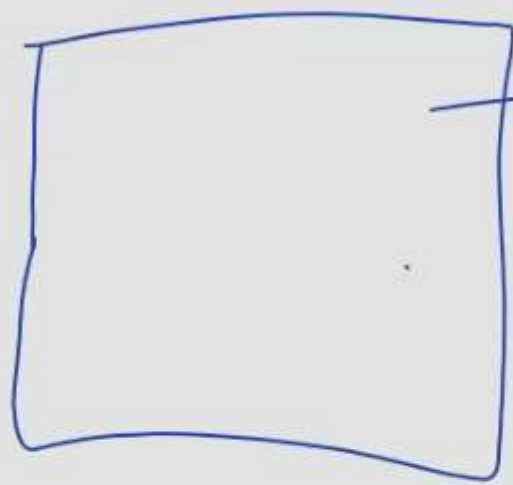
4

5

6

7

$y = f(x)$



① divide  $X - Y$  - 2 ✓

② training / testing - 2 ✓

③ apply algorithm - 2

④ predict the O/P - 1 ✓

⑤ evaluate - 4

→ fit(  
predict(

= accuracy ( $y$ -test,  $y$ -pred)

group

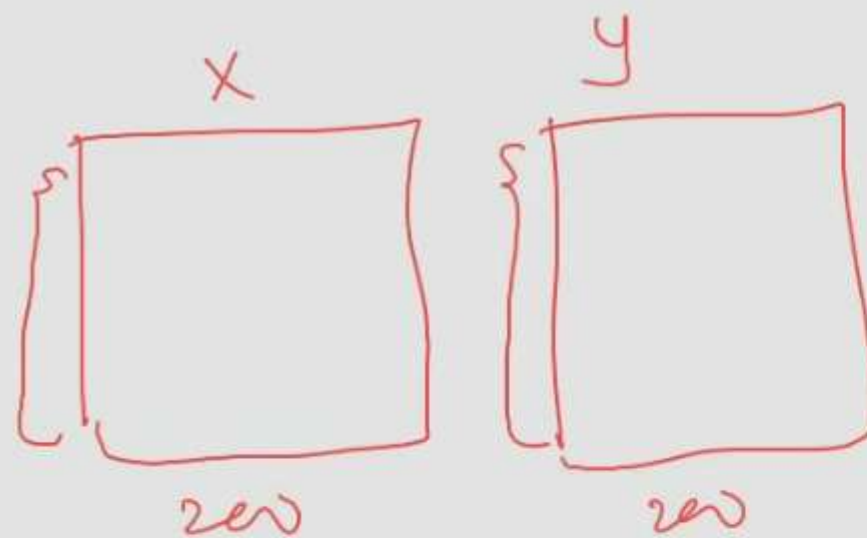
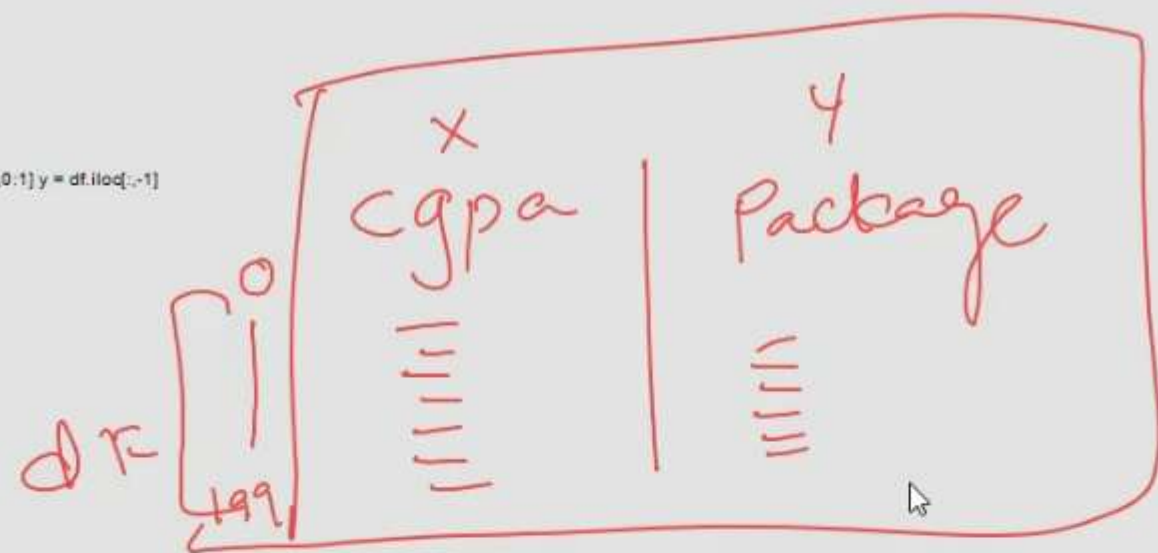
1

2

3

4



$x = df.iloc[:, 0:1] \quad y = df.iloc[:, 1:]$ 


$$x = df.iloc[\underline{0:200}, \underline{0:1}]$$

excluded

0 - 199

$$y = df.iloc[:, 0:1]$$

6

7

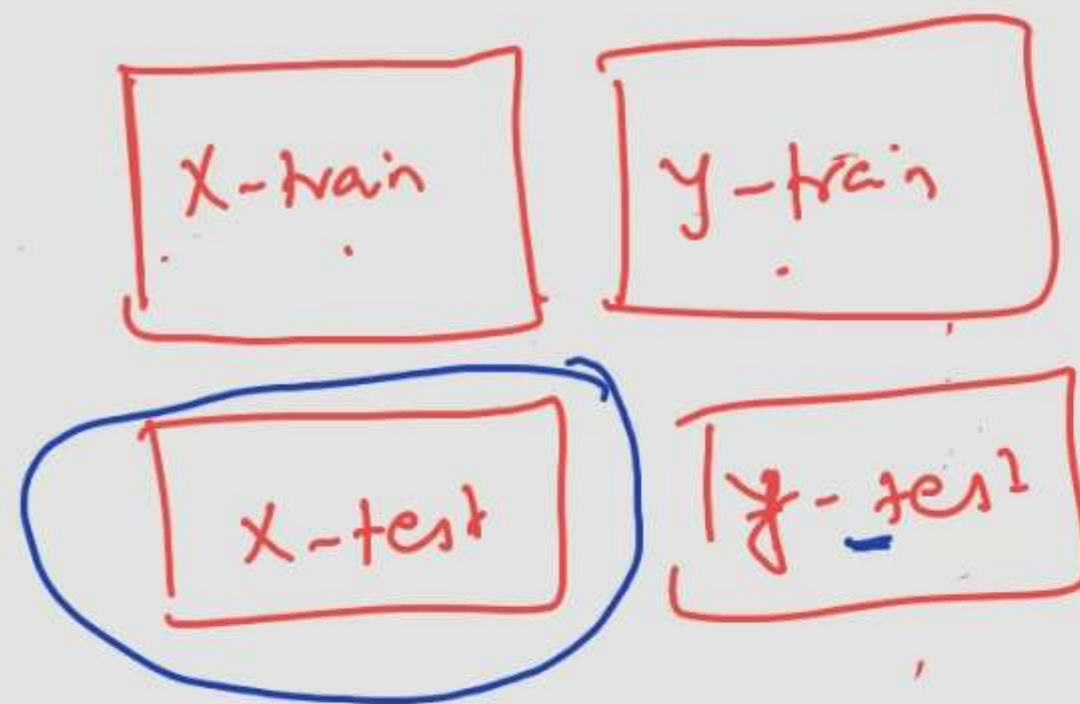
8

9

10

① fit ( )

✓ ② predict ( )



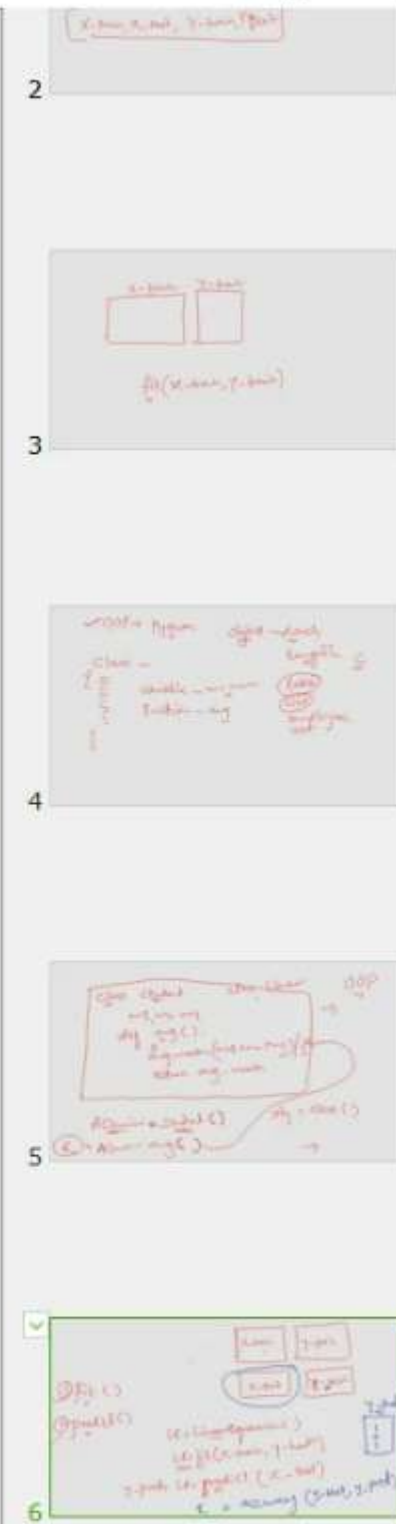
$y_{\text{-pred}}$

$LR = \text{LinearRegression}()$

$LR.\text{fit}(X_{\text{-train}}, y_{\text{-train}})$

$y_{\text{-pred}} = LR.\text{predict}(X_{\text{-test}})$

$R = \text{accuracy}(y_{\text{-test}}, y_{\text{-pred}})$





# Model Building — data scientist

- ✓ → divide the dataset into  $X, Y$  — iloc
- ✓ → divide  $X, Y$  into training and testing
  - apply algorithms → import — Sklearn
  - Predict the O/P
  - evaluate

group

Model Building — data scientist

- divide the dataset into  $X, Y$  — iloc
- divide  $X, Y$  into training and testing
- apply algorithms → import — Sklearn
- Predict the O/P
- evaluate

2

3

Model Building — data scientist

→ divide the dataset into  $X, Y$  — iloc

→ divide  $X, Y$  into training and testing

→ apply algorithms → import — Sklearn

→ Predict the O/P

→ evaluate

4

5

Model Building — data scientist

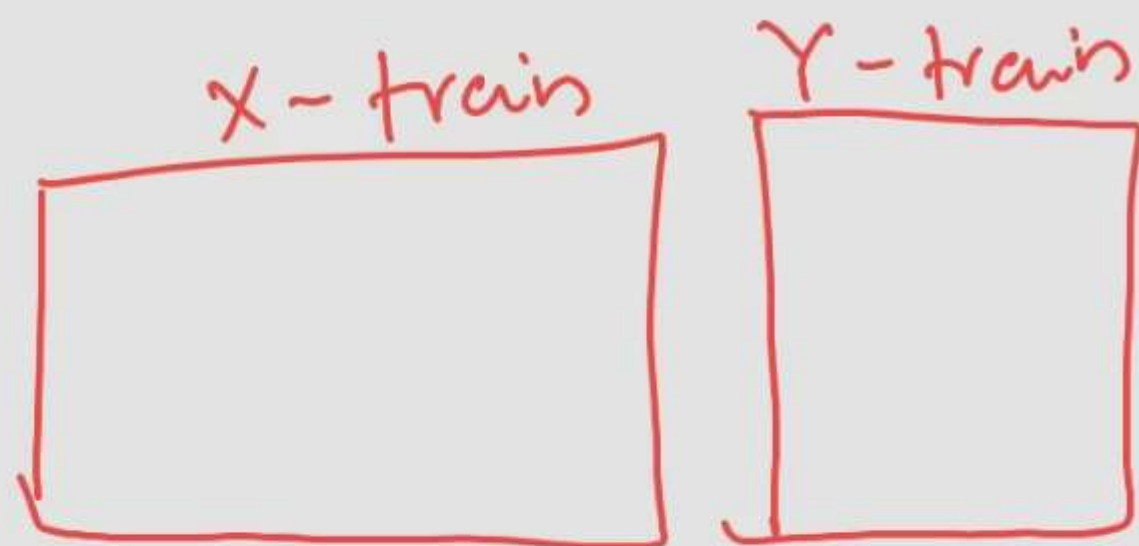
→ divide the dataset into  $X, Y$  — iloc

→ divide  $X, Y$  into training and testing

→ apply algorithms → import — Sklearn

→ Predict the O/P

→ evaluate



①  $\text{fit}(X\text{-train}, Y\text{-train})$   
=

② predict

apply - algorithm

group

Model building - data loaded  
→ divide the data into  $X, Y = \text{Data}$   
→ divide  $X, Y$  into training and testing  
→ applying algorithm → output (loss)  
→ predict the output  
→ evaluate

2

1. train, 2. test, 3. predict  
①  $\text{fit}(X\text{-train}, Y\text{-train})$  apply - algorithm  
② predict

4

Model building - data loaded  
→ divide the data into  $X, Y = \text{Data}$   
→ divide  $X, Y$  into training and testing  
→ applying algorithm → output (loss)  
→ predict the output  
→ evaluate



		cgpa	package
test		7	3
		8	4
		9	5
	{	7	3-5
		9	6

LR.fit (X-train, Y-train)

$$y_{pred} = LR.predict(x_{test})$$

50,000

y-pred

4
6 ✓

3

fit(x\_train, y\_train) apply algorithm  
predict

4

LR.fit(x\_train, y\_train)  
predict(x\_test)

5

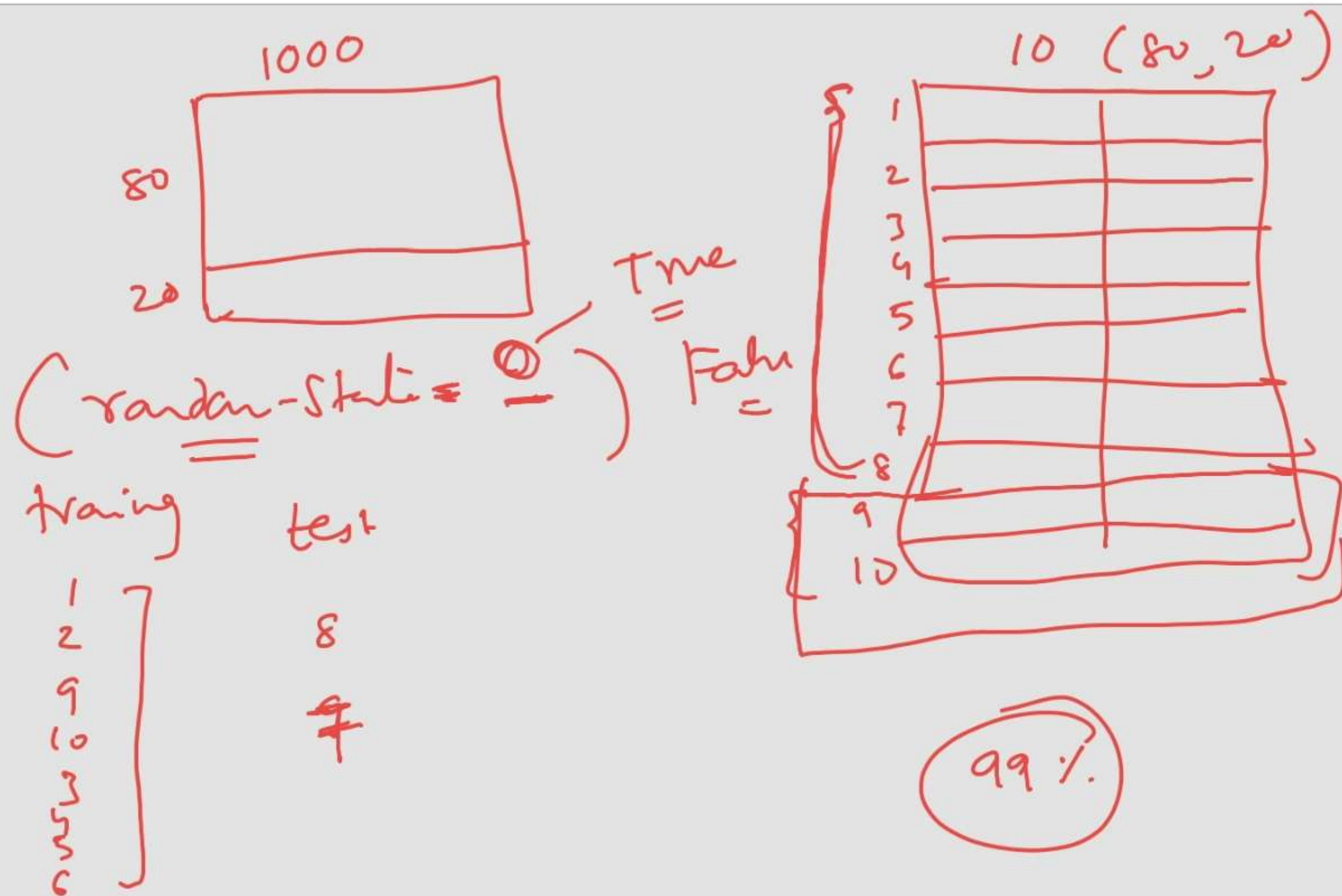
LR.fit(x\_train, y\_train)  
predict(x\_test)

6

LR.fit(x\_train, y\_train)  
predict(x\_test)

7

LR.fit(x\_train, y\_train)  
predict(x\_test)



Minimize

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100