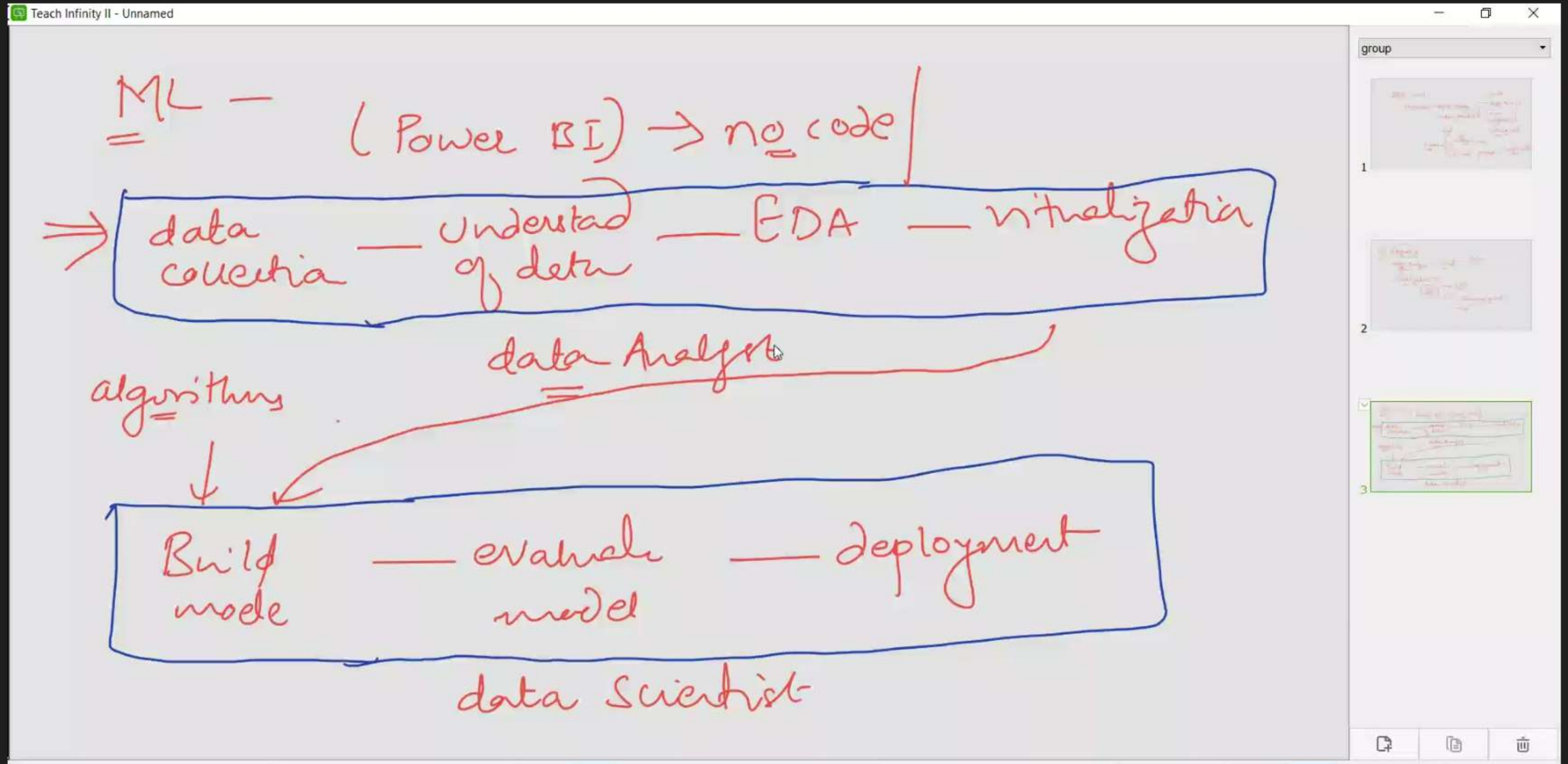
Teach Infinity II - Unnamed



group

1

2

3

Resources

250 GB

1. Object Storage – store
2. Watson Studio – data Analysis
3. Watson machine learning → algorithms

D Analysis & prediction

ML → Continuous → regression

→ Discrete → Classification

∧ DL

algorithm → Linear Regression
→ Logistic
→ SVM
→ DT
→ KNN
→ Naive Bayes

$y = mx + c$

Teach Infinity II - Unnamed

→ data Analyst

① data collection — kaggle

② Understand → cloud
→ python

③ clean data] EDA
→ Missing
→ outlier
→ duplicates

④ Visualization →

→ LG

→ Why

1 2 3 4 →

✓ 1

✓ 2

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... +579

Type here to search 30°C Mostly cloudy ENG 18:27 12-07-2024

ML eng

DL eng

data Eng

DBA / data eng → SQL

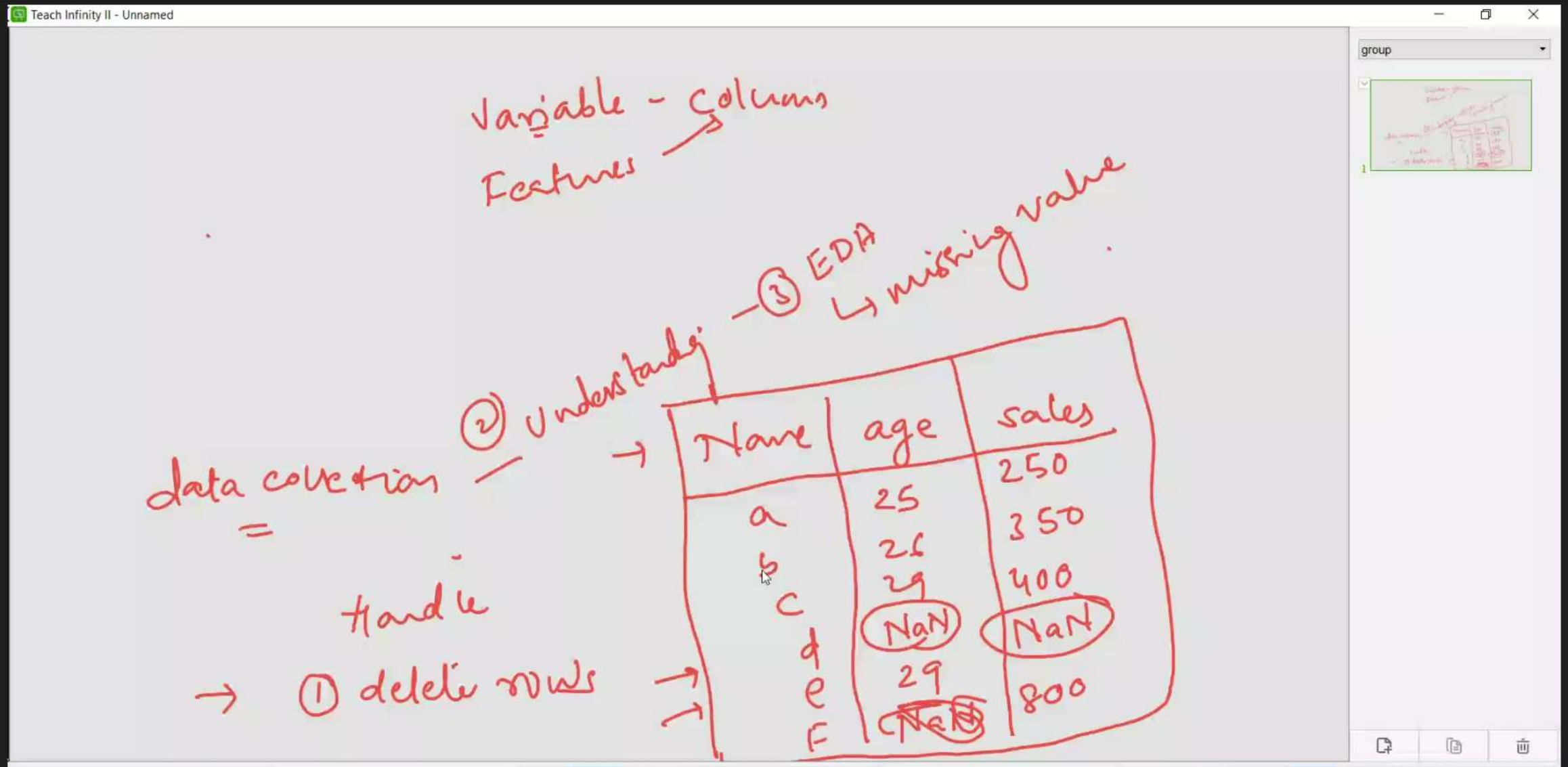
total cost

collection data — understand — EDA

- Missing
- duplicate
- outlier

Teach Infinity II - Unnamed

group

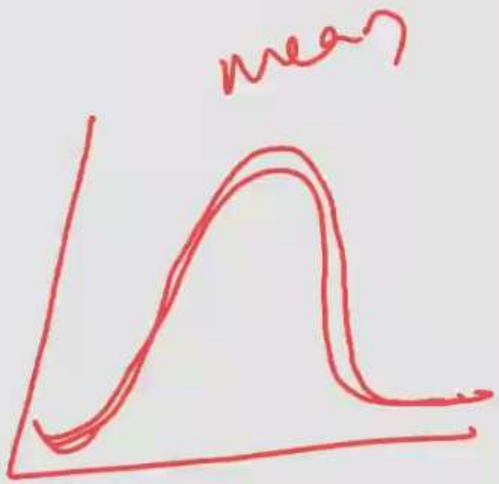


10,00 — 2%.
3%.
5%.

age Gender
25
24
29
30
Na
[29

② impute — fill

① mean | median | mode
② constant value
③ algorithms



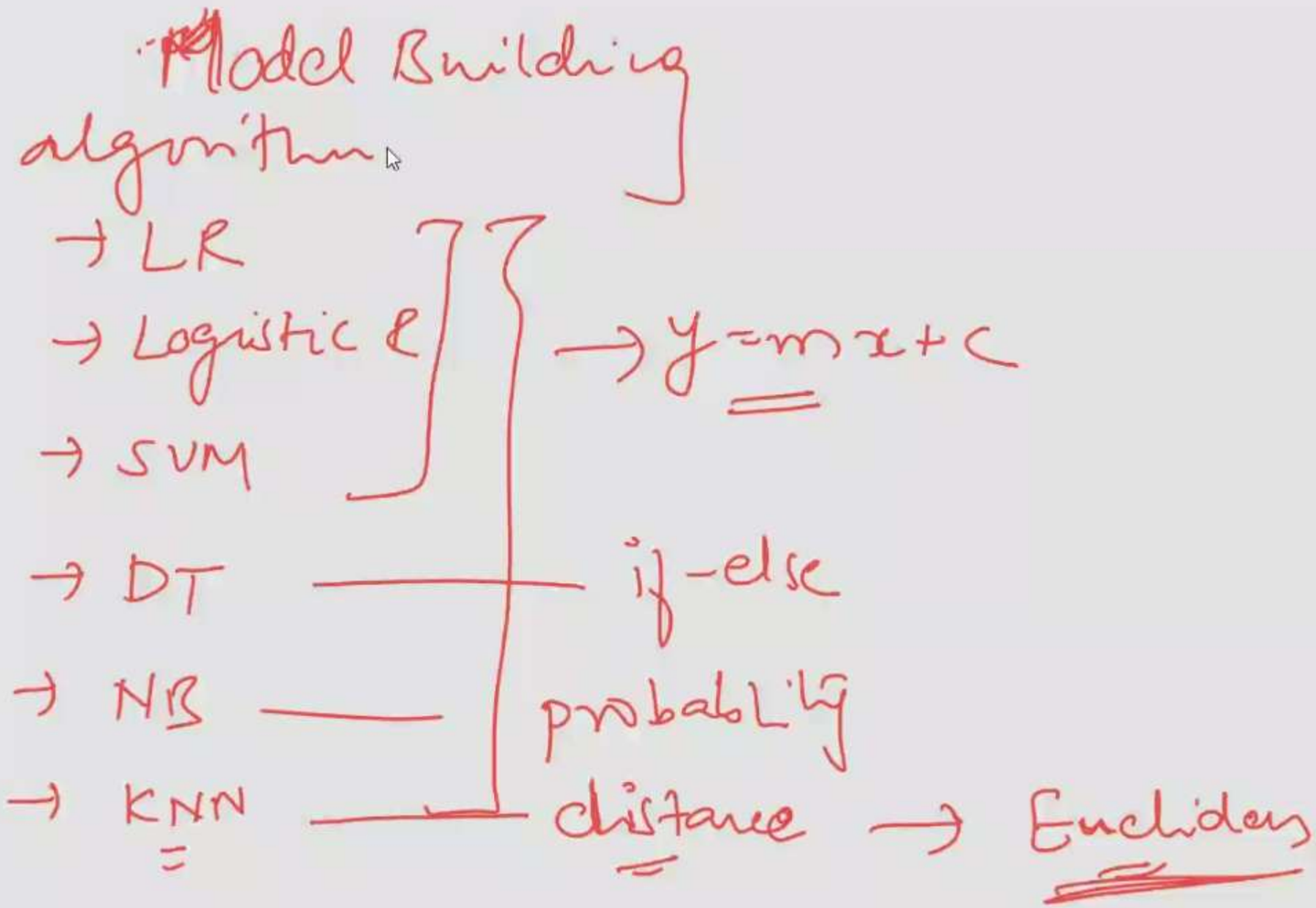
group

Model Building algorithm

- → LR
- → Logistic
- → SVM

→ $y = mx + c$

- → DT → if-else
- → NB → probability
- → KNN → distance → Euclidean



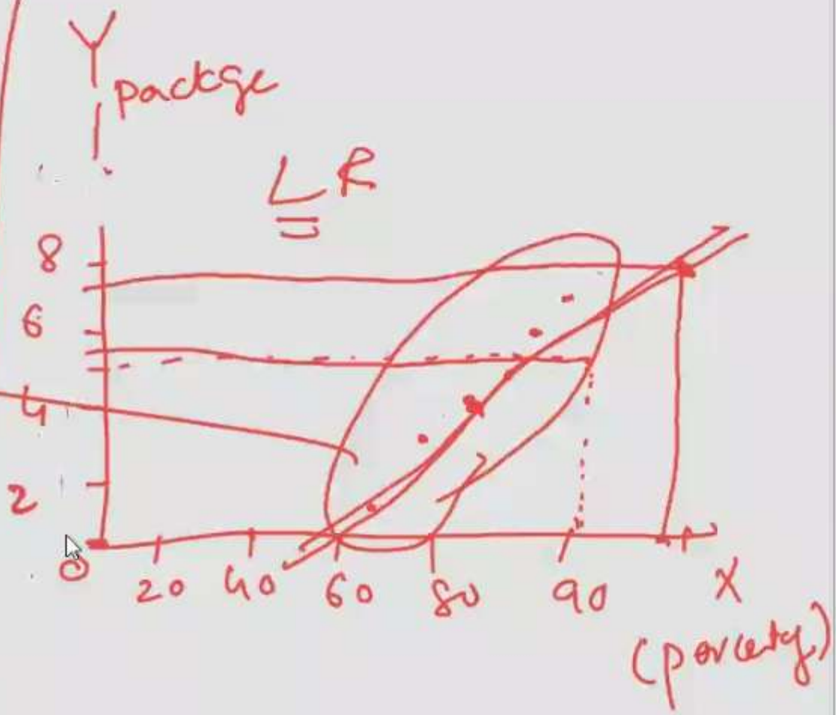
if-else
=
→ DT

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

91
=

98
=

6.5



..

Teach Infinity II - Unnamed

Problem

Regression → O/P → Continuous

classification → Yes / No

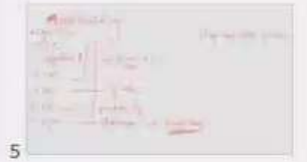
Linear Regression ≠ Regression

Logistic Reg = classification

Sklearn

ML

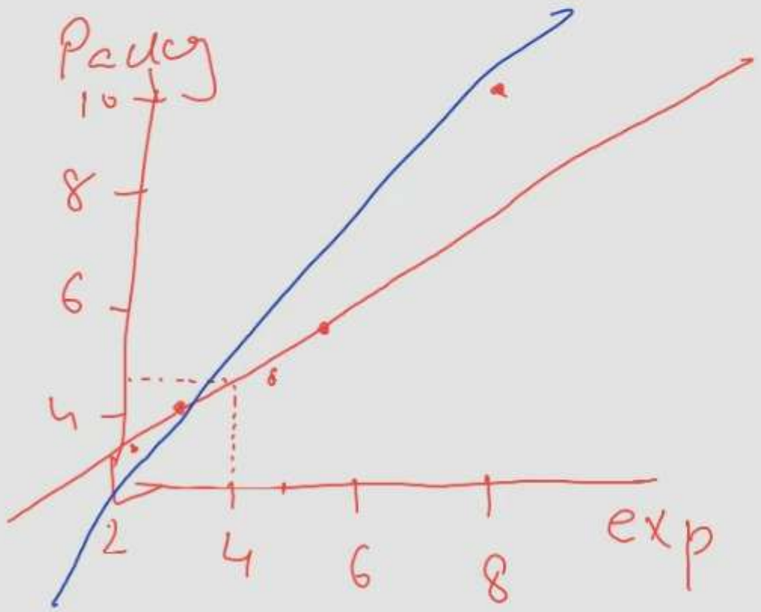




$y = mx + c$

$y = m * 4 + c$

$y = 0$



$y = mx + 0$

$y = m * 0 + 0$

$y = 0$

exp	Package
2	3
3	4
5	4.5
6	5.5
8	10
4	? 4.2

$x = 4$

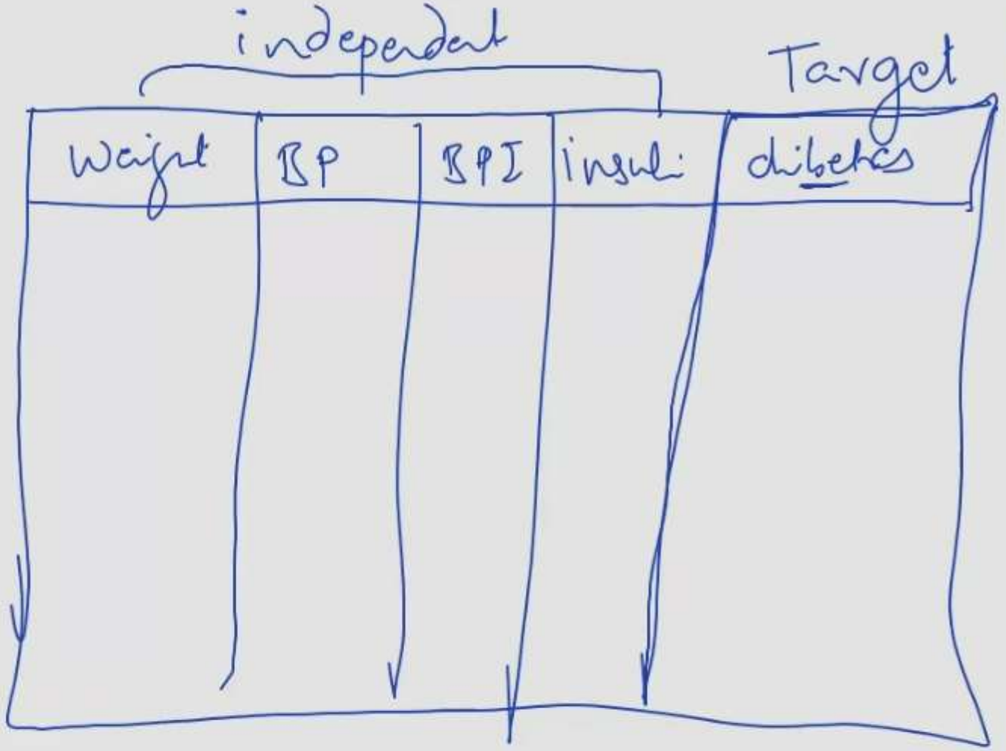
$y = ?$

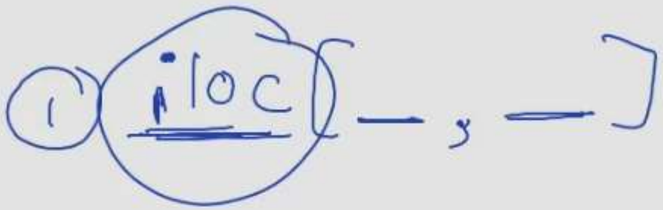
group

1

2

df





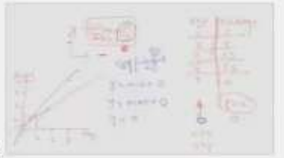
drop

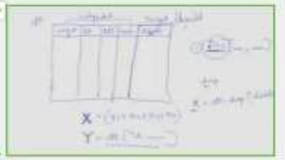
X = df.drop("diabetic

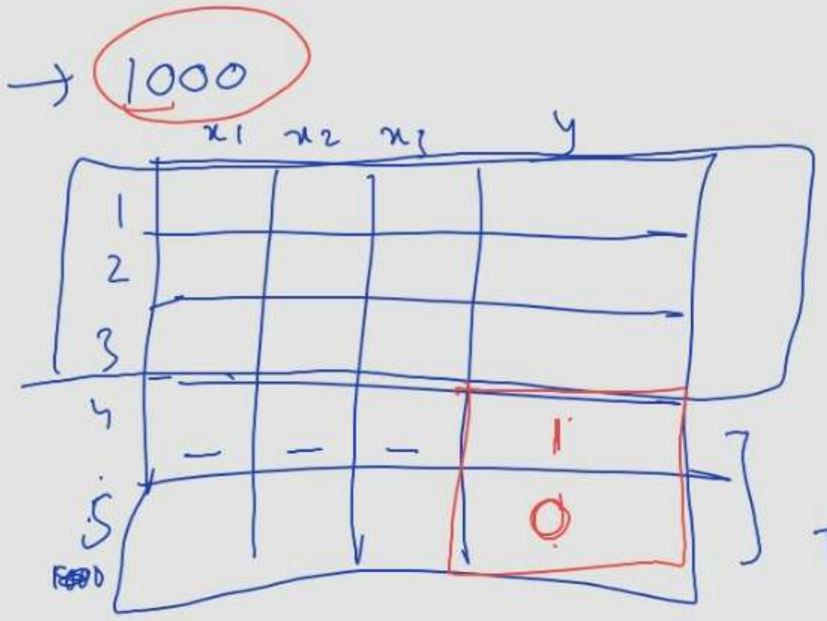
X = (x1 + x2 + x3 + x4)

Y = df["di

group

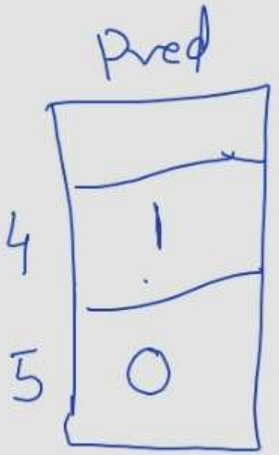


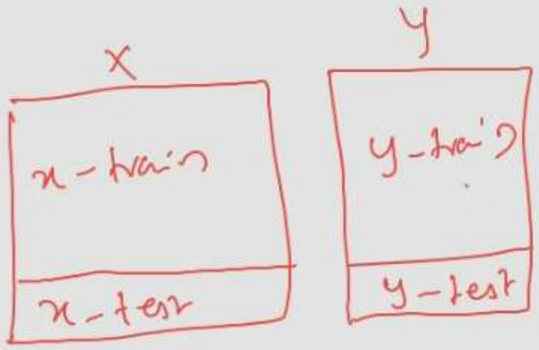




Traning – 70%
80%

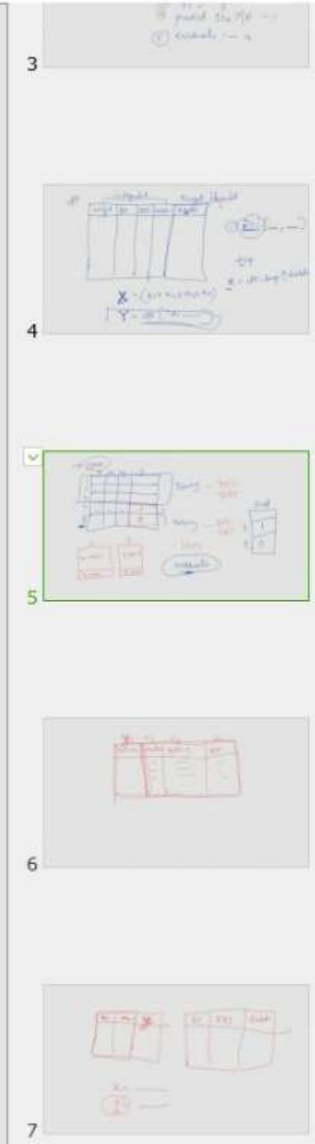
Testing – 30%
20%

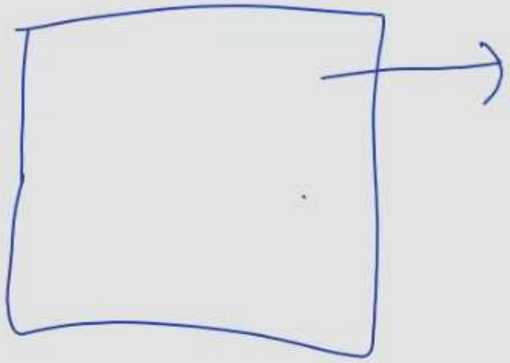




100%

evaluate





① divide X - y - 2

② traing | testing - 2

③ apply algorithm - 2

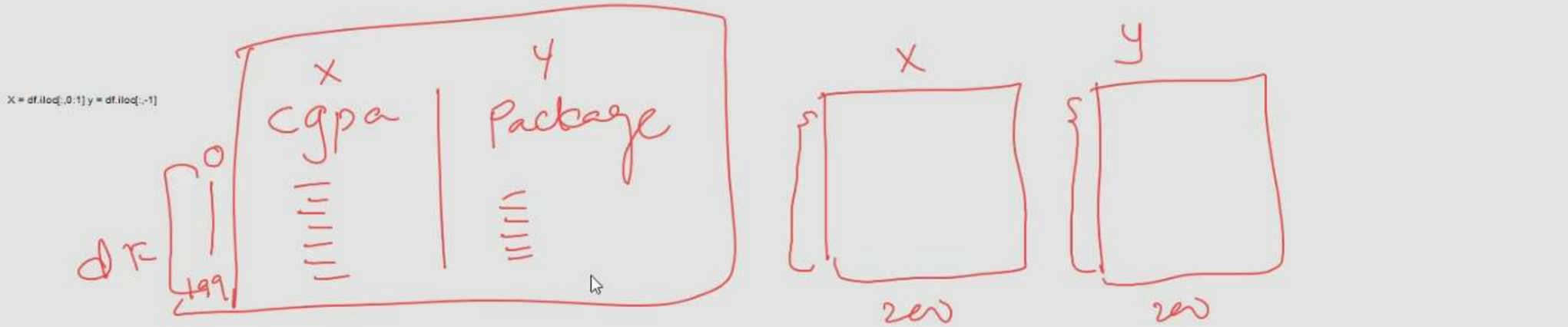
④ predict the O/P - 1

⑤ evaluate - 4

→ fit( )

predict( )

= accuracy(y-test, y-pred)



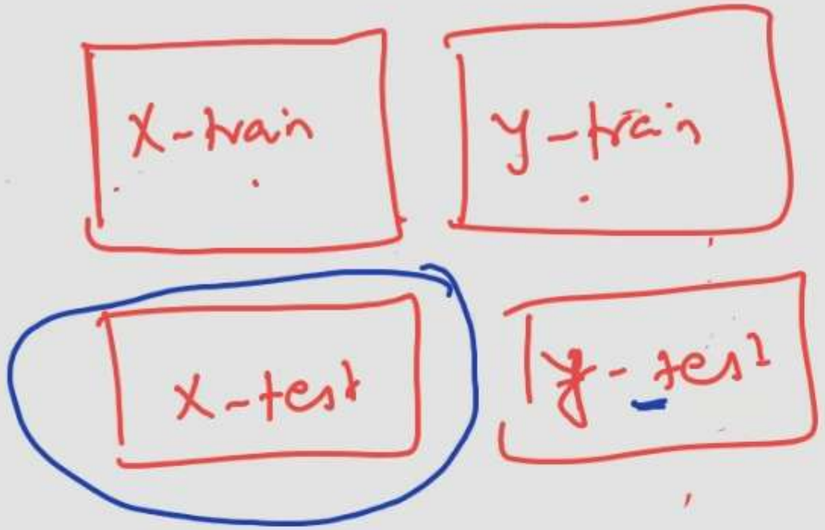
excluded

$X = df.iloc[0:200, 0:1]$ $X = df.iloc[:, 0:1]$

$y =$

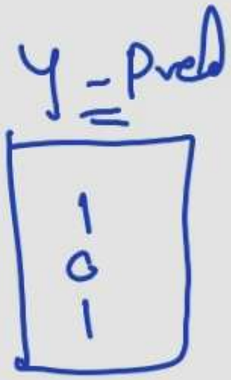
0 – 199

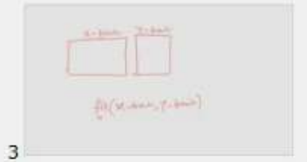
0



1. fit()
2. predict()

```
LR = LinearRegression()
LR.fit(X_train, y_train)
y_pred = LR.predict(X_test)
R = accuracy(y_test, y_pred)
```





Model Building — data scientist

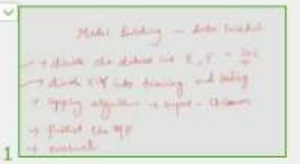
→ divide the dataset into X, Y — iloc

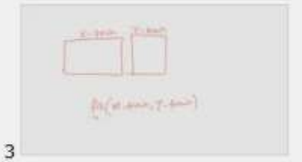
→ divide X-Y into training and testing

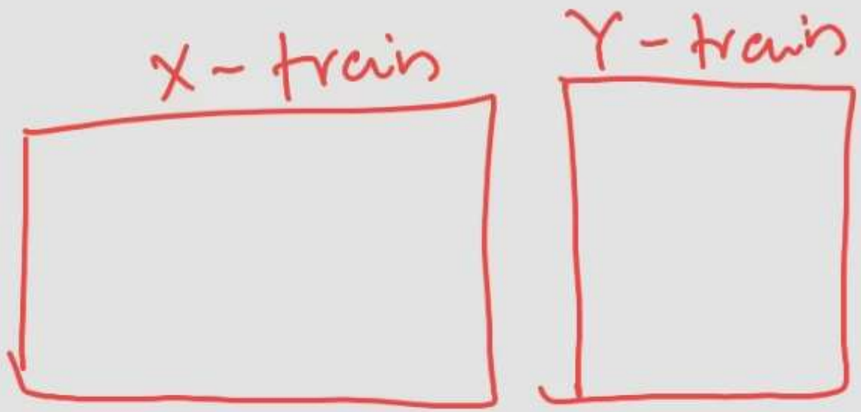
→ apply algorithms → import - Sklearn

→ Predict the O/P

→ evaluate

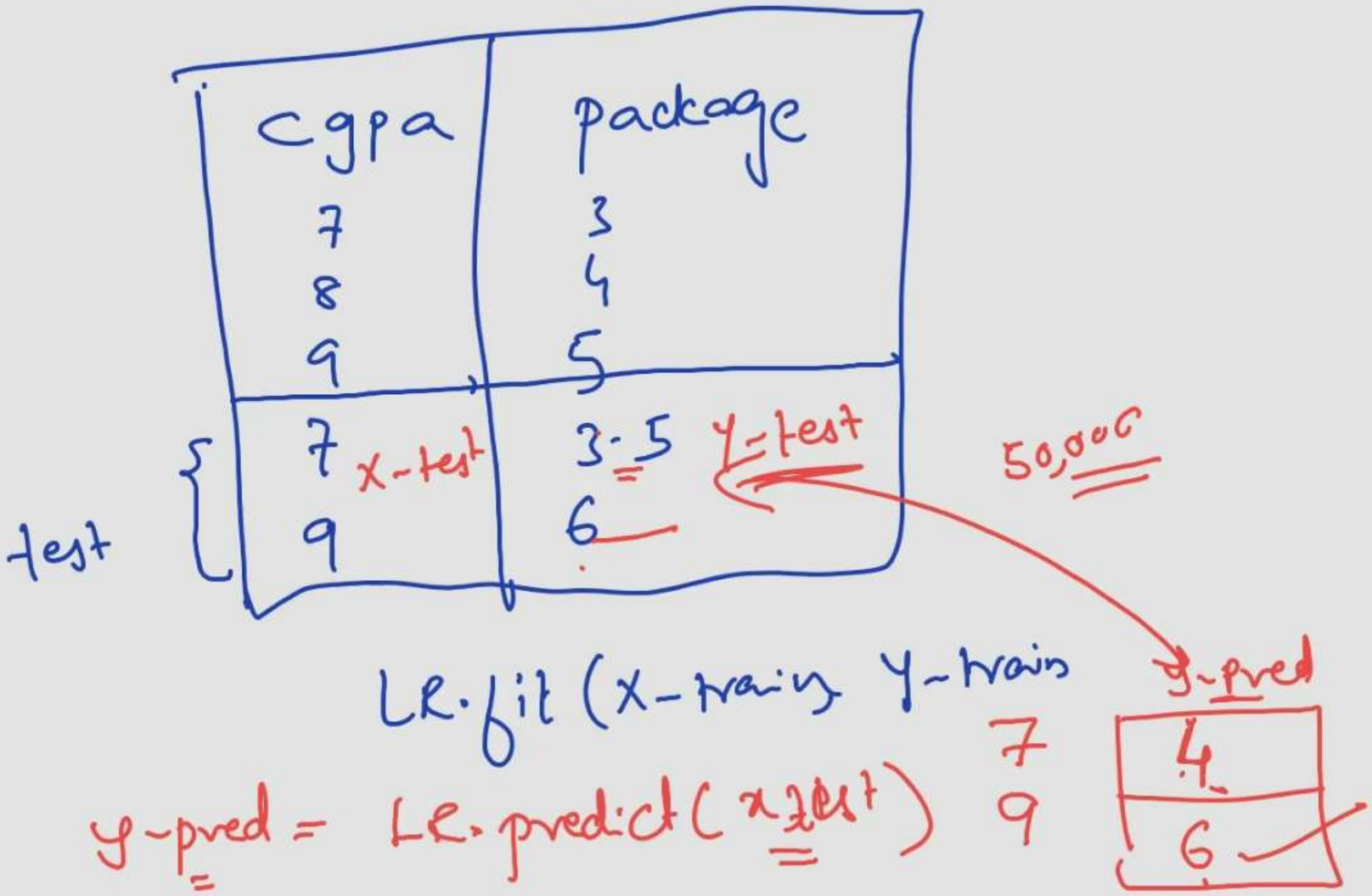






1. fit(X-train, Y-train)
2. predict

] apply - algorithm

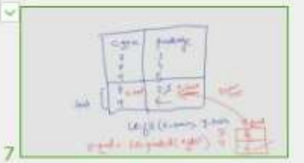

cgpa
package
7
8
9
3
4
5
test
7
9
x-test
3.5
6
y-test
50,000
LR.fit(X-train, Y-train)
y-pred = LR.predict(X-test)
y-pred
4
6
7
9

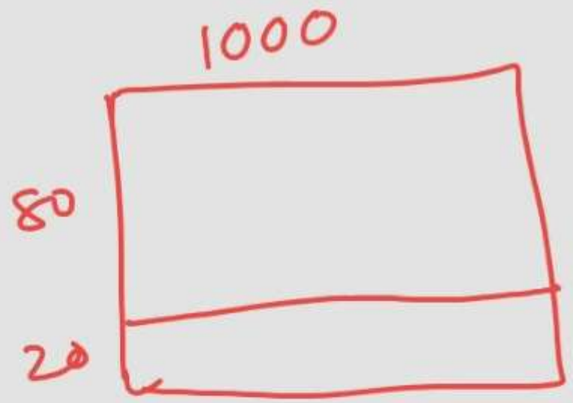

3


4


5


6

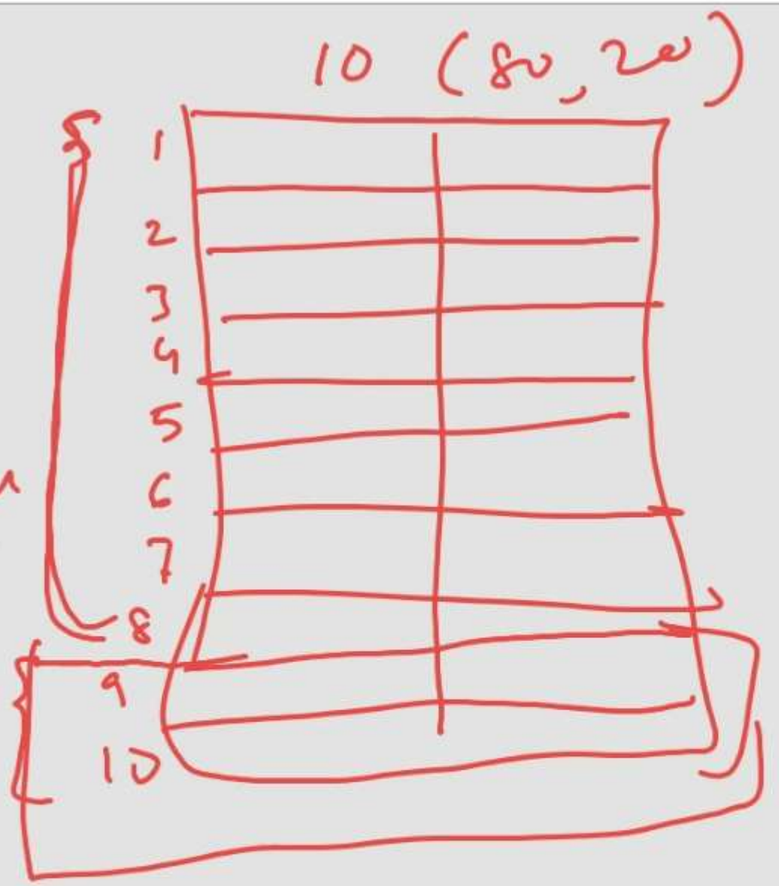

7



(random-state = 0)

True =

False =



training	test
1	8
2	~~9~~ 7
9	
10	
3	
5	
6	

99 %.