Machine learning:-

→ Problem
- Regression → Continuous
- Classification → Discrete

Examples
- Features
- Labels

Collect Data

Clean Data

m → examples
n → features

m labels for m examples

m x n

n x 1

example

→ Split

Train

Train Algorithm

train_ex
train_labels

Test

test_ex
test_labels

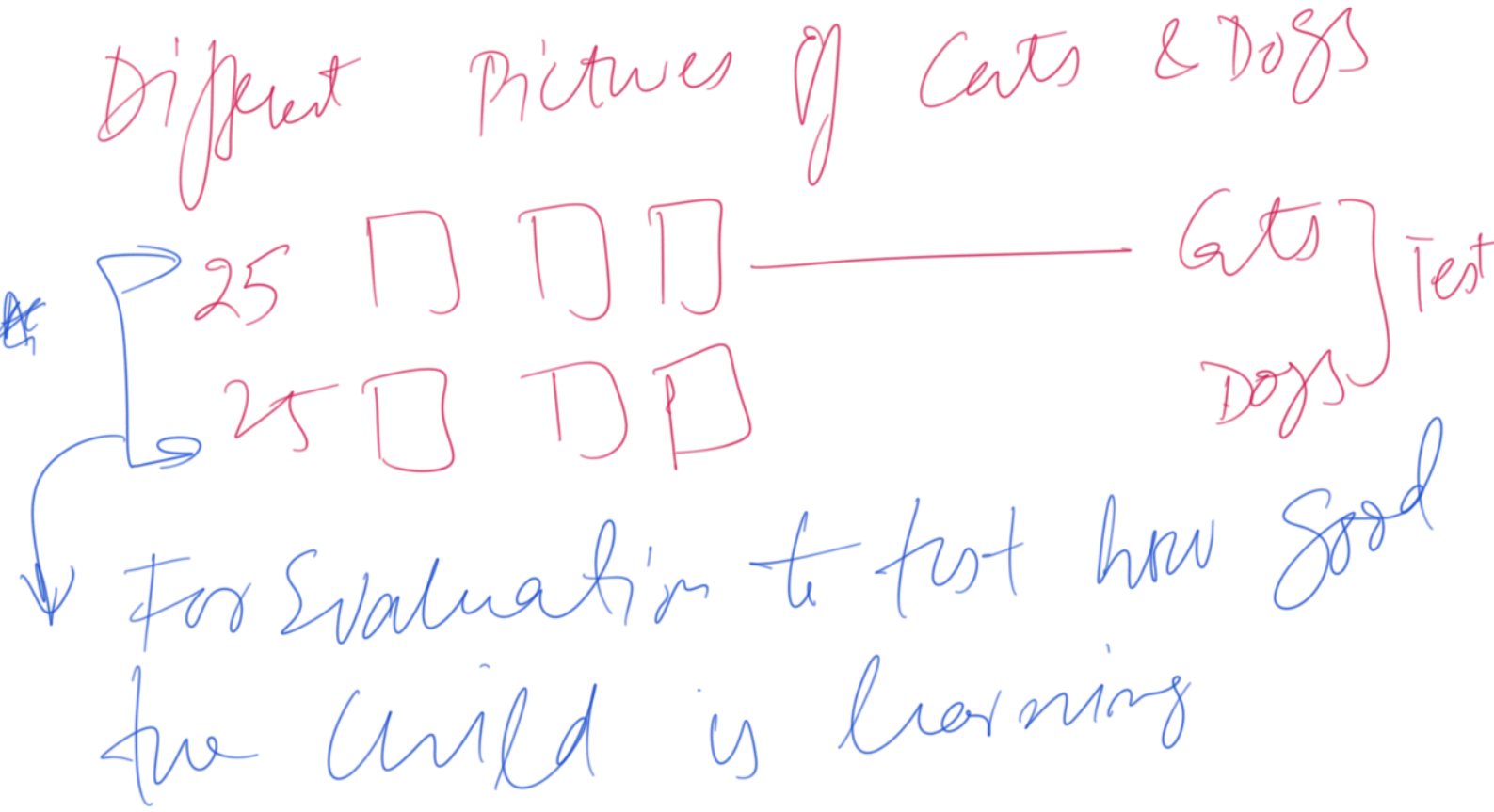
Evaluate Test

⇒ Goal: To teach a child to identify cats & dogs.

→ Collect Pictures of cats & Dogs

Expectation: Child will learn visual features ( ) to differentiate b/w cats & dogs.

This is a one time process to learn the child to identify b/w Cats & Dogs.


Different Pictures of Cats & Dogs
25
25
Cats
Dogs
Test
For Evaluation to test how Good the child is learning

You use this test set to measure child learning capability (via accuracy)

test metric

30% Correct Answers

Evaluation metric

Test set You will

100%

Unseen set)

stop

→ Q: What happens if I use the same training set for evaluation?

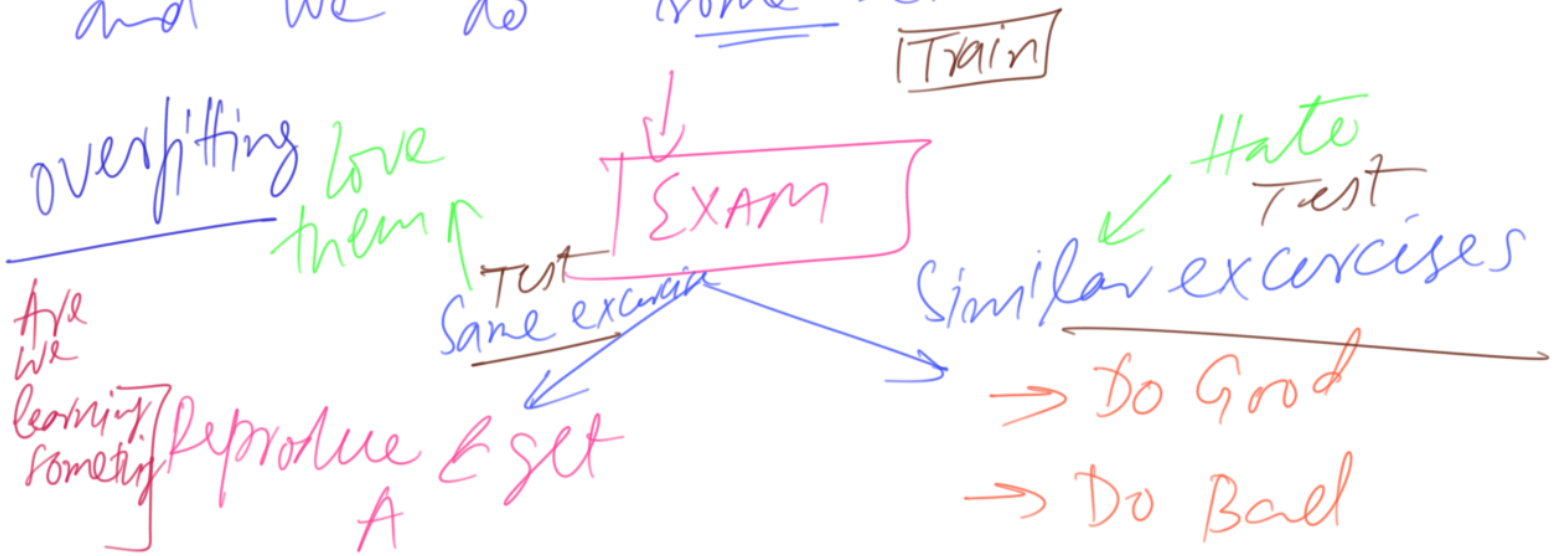
↙ Accuracy will be 100%.

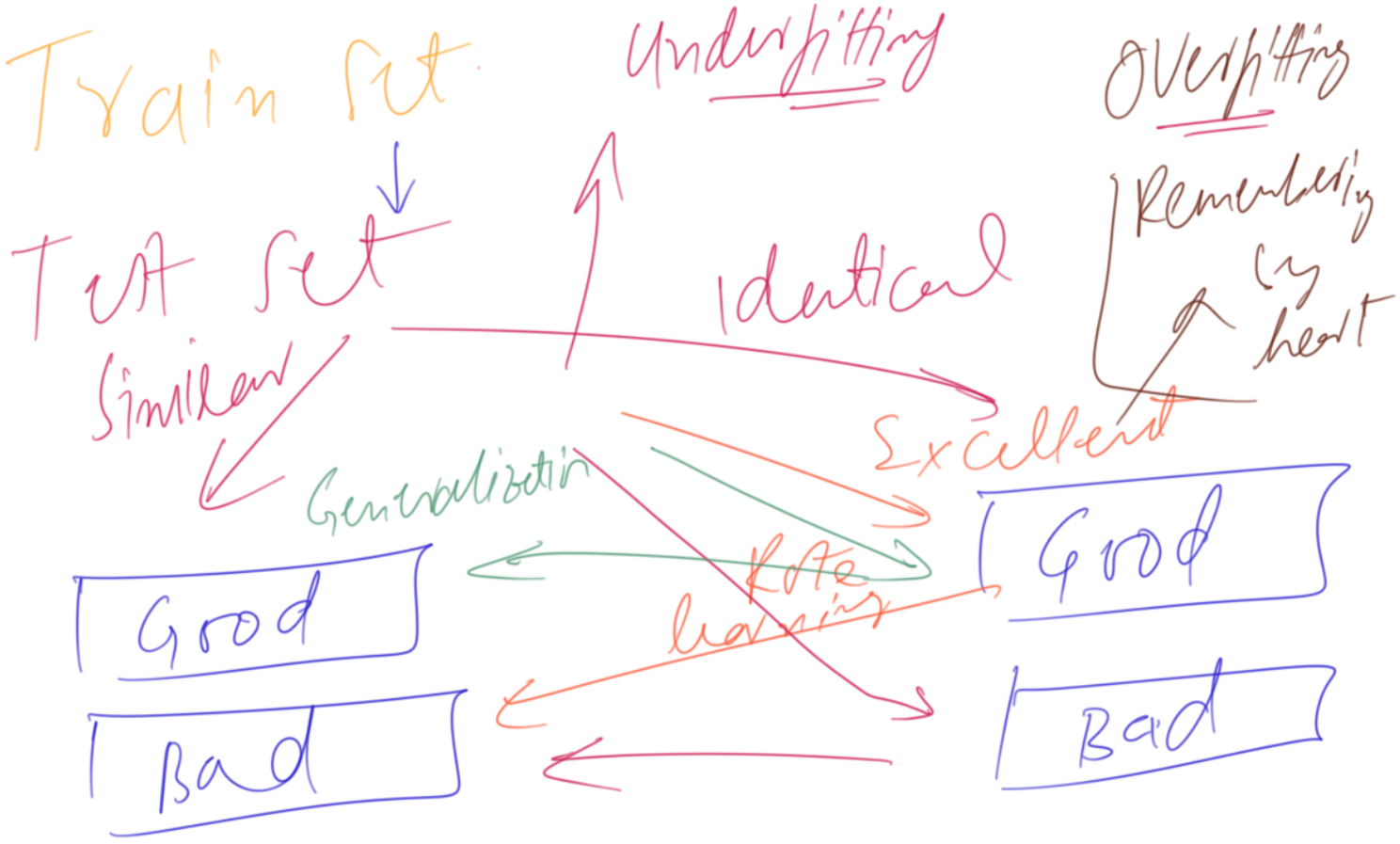
overfitting

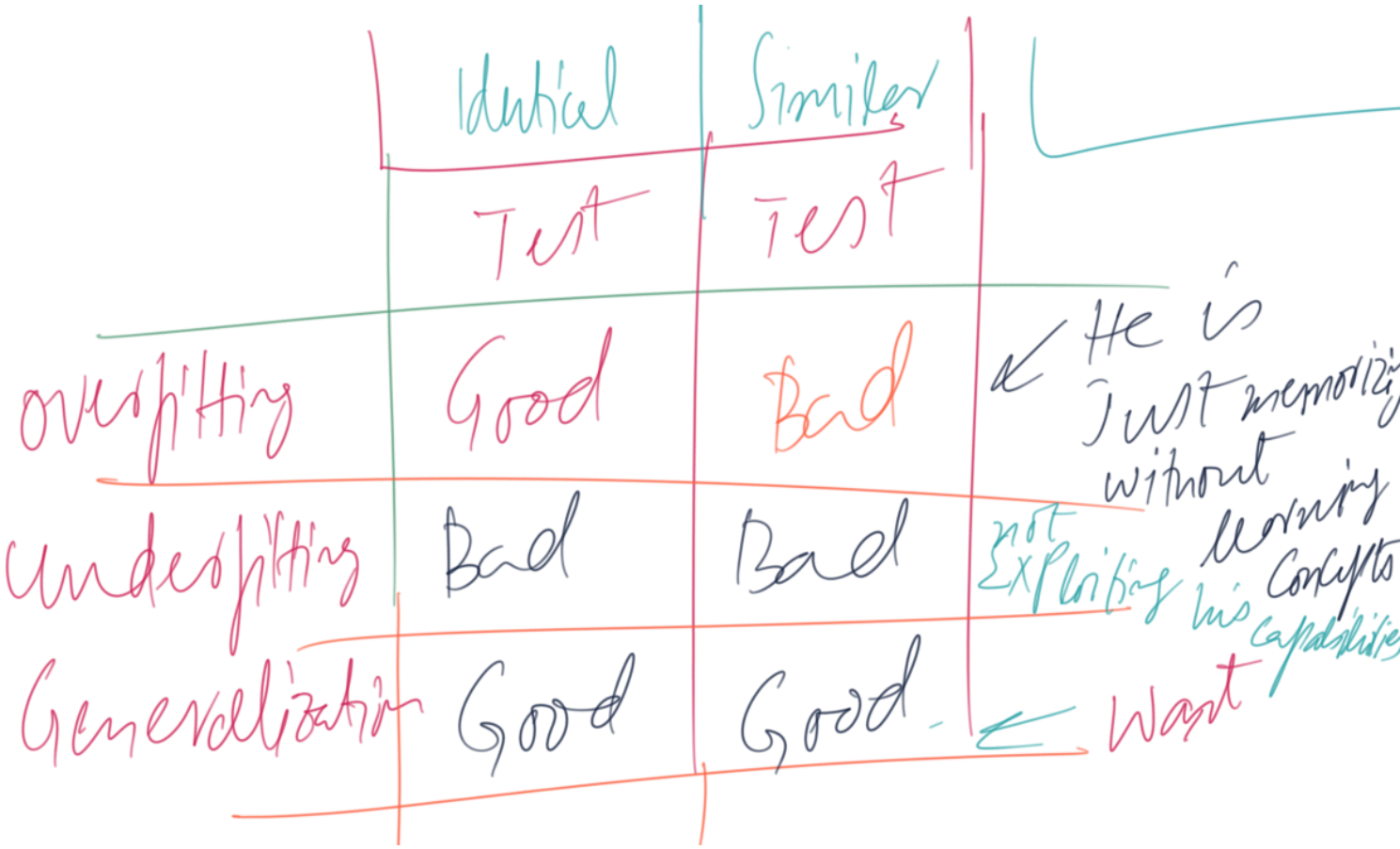
Rote learning]

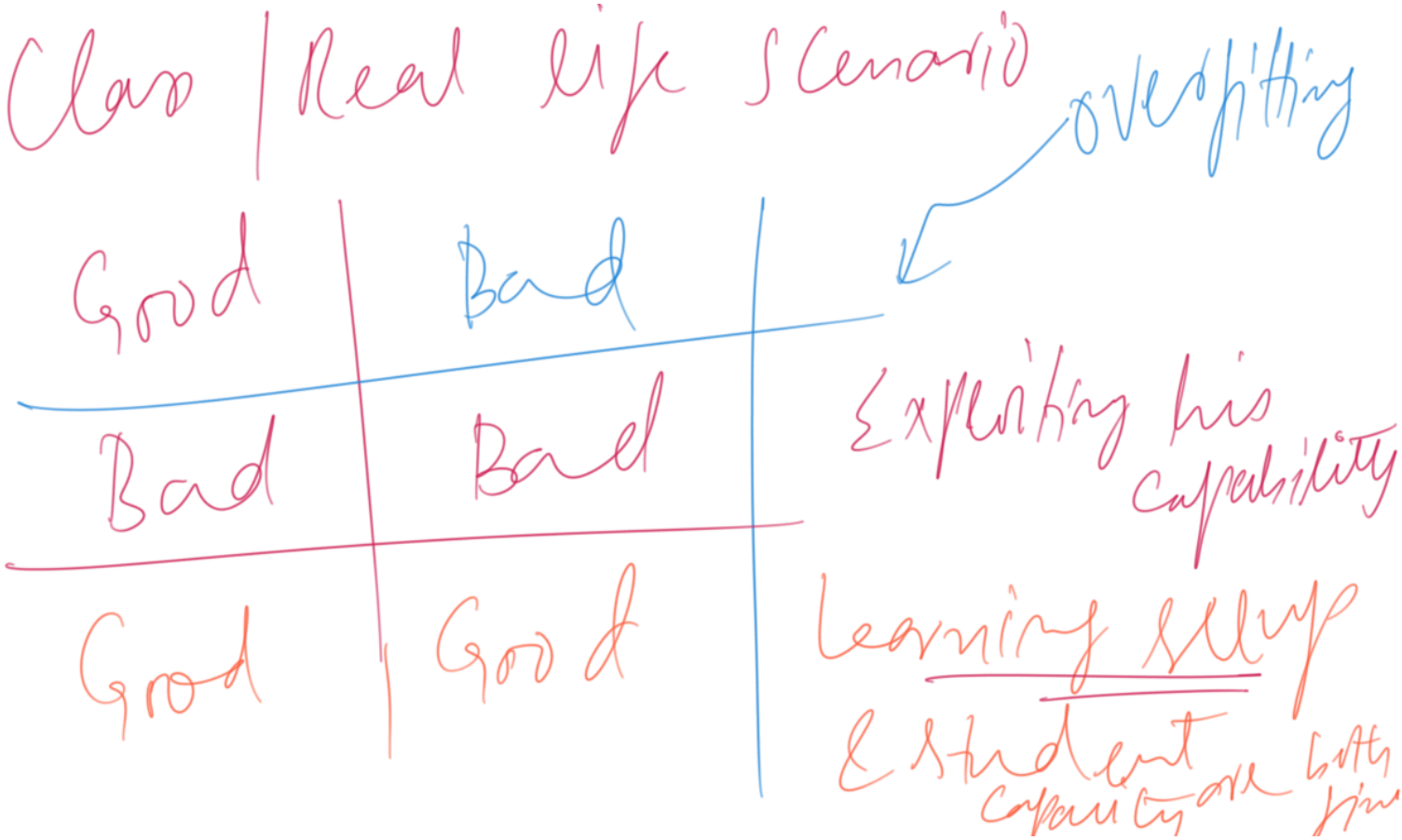
→ Class Scenario

Teach some Concepts in the class and we do some exercises on this.




Train set
Test set
Similar
Underfitting
Identical
Overfitting
Remembering by heart
Excellent
Generalization
Good
Bad
Rote learning
Good
Bad


Identical
Similar
Test
Test
overfitting
Good
Bad
Underfitting
Bad
Bad
Generalization
Good
Good
He is just memorizing without learning
not exploiting his capabilities
Want


Class
Real life Scenario
overfitting
Good
Bad
Bad
Bad
Exploiting his capability
Good
Good
learning steep
& student

→ Right learning capacity

→ Right learning setup

Right setup

Right algorithm

Simple algo for Complex

Complex algo for simple task

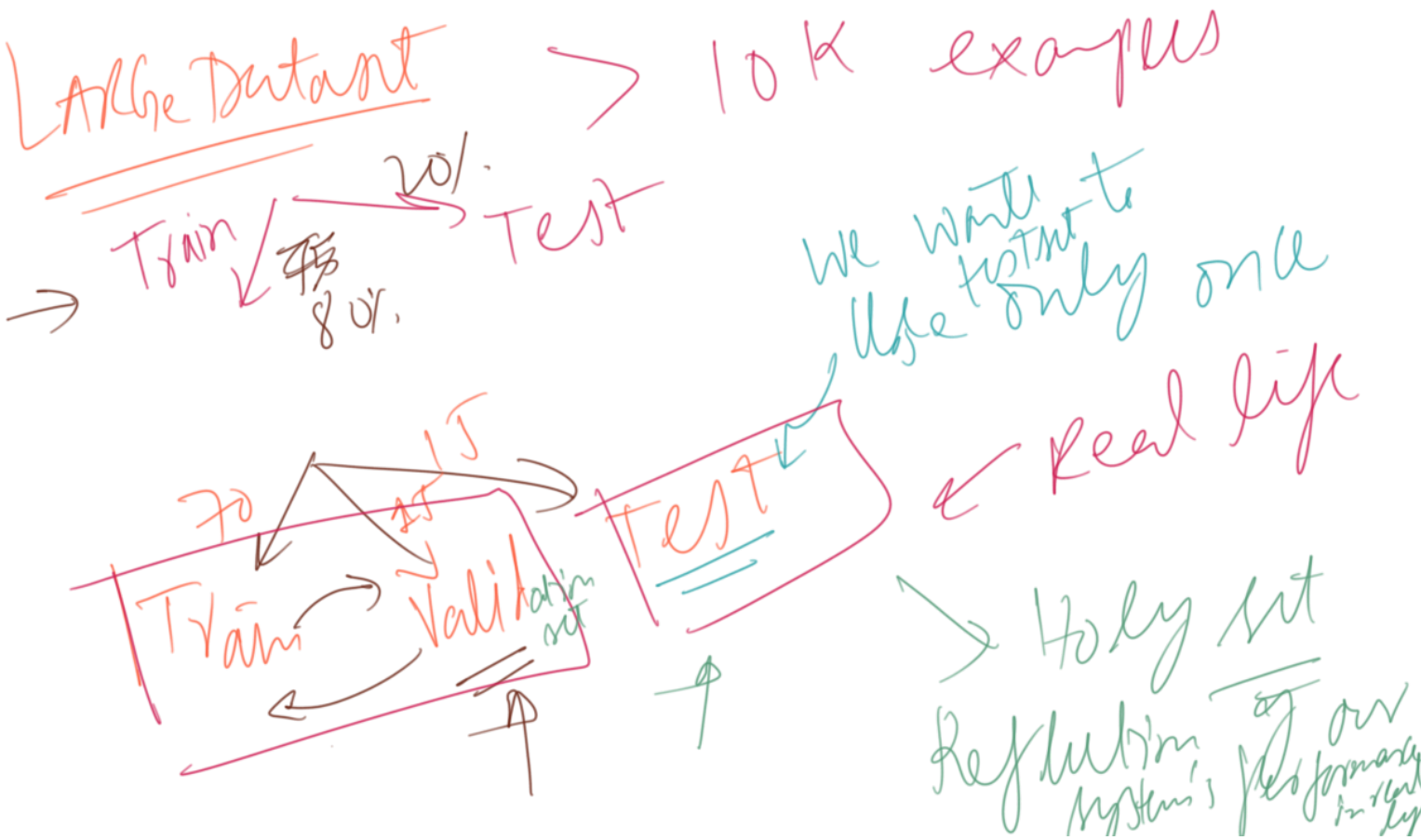
overfit

underfitting

→ Generalization — Good on training & test

→ Underfitting — Bad on both

→ Overfitting — Good on training / Bad on test


Large Dataset
> 10K examples
Train
20%
Test
80%
We want to use testset only once
Real life
70
15
15
Train
Validation
Test
Hold out
Reflection of our system's performance in real life

Small Dataset ≤ 10K

150 Examples

There is not enough data to split into train/valid/test set

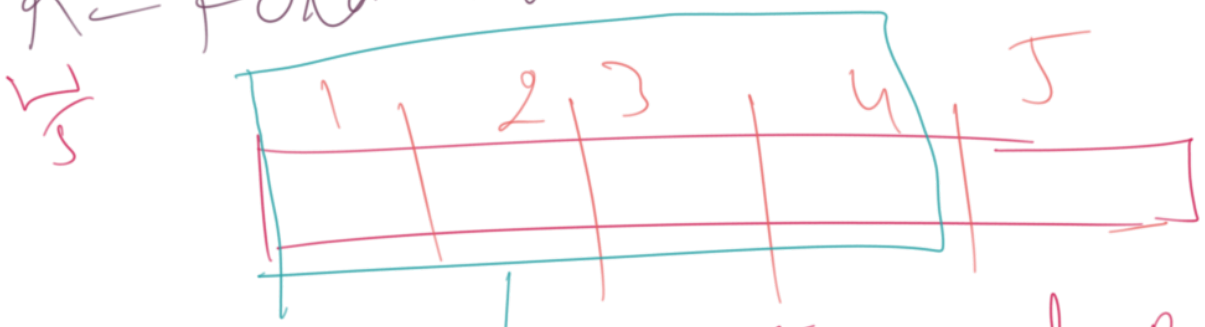
setup learning:

→ How to do train test split

→ large dataset

→ Small dataset

K-Fold Cross validation



1/5

① Split Data into 5 equal parts. 100 Examples

Train ← F[1:4]

F[5] → Test.

I1: F1 — F4 → Train, F5 - Test → Accuray

I2: F1, F2, F3, F5 → Train, F4 → Test →

I3: F1, F2, F4, F5 - Train; F3 - Test →

I4: F1, F3, F4, F5 → Train, F2 - Test →

I5: F2 — F5 → Train; F1 - Test →

Learning setup

↳ large
small

Check for Generalization; or overfitting
& Underfitting

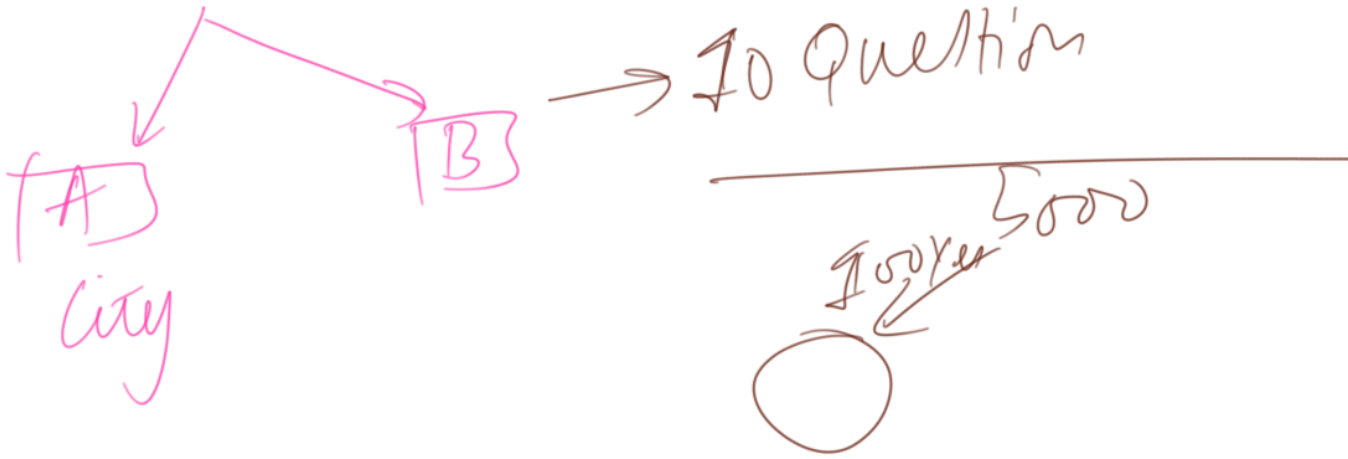
Evaluation metrics for classification
Algorithm

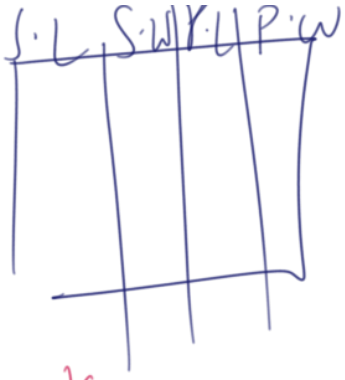
K- Nearest Neighbours

Decision Trees | Bagging | Random Forest

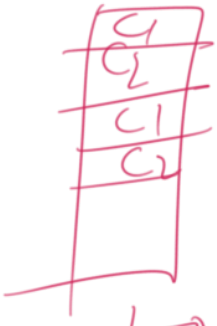
What a decision tree :-

Guess a city Game :-





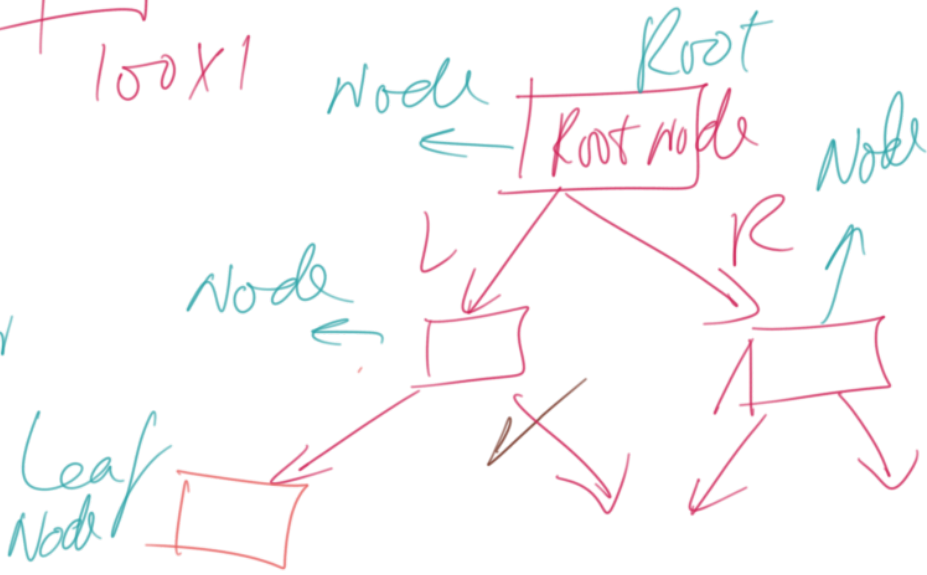
100 X 4



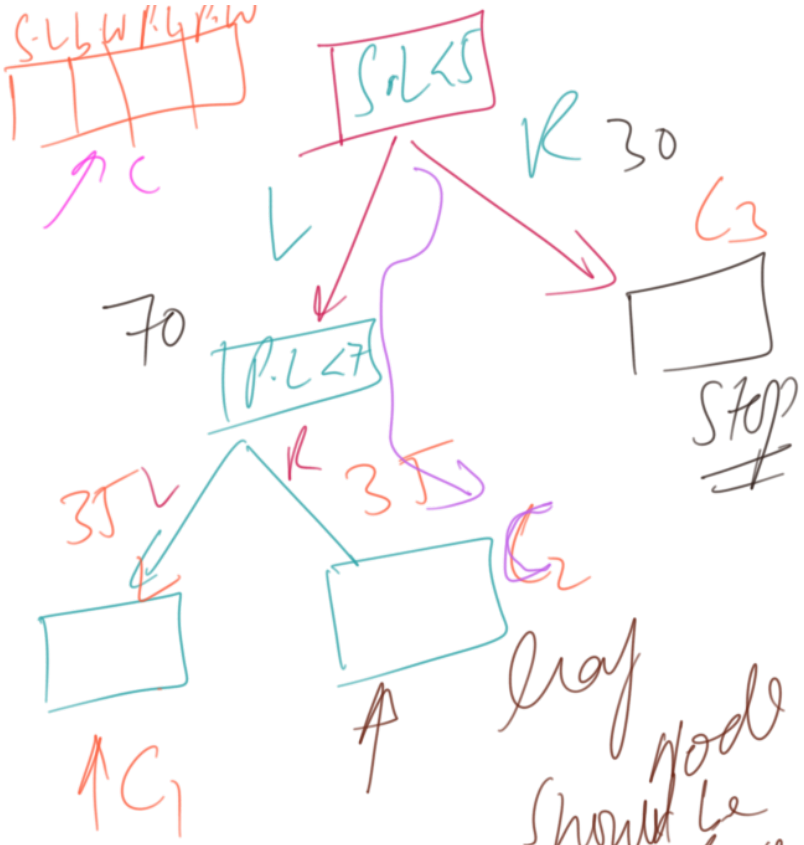
100 X 1

It will build a tree

Goal: Given features Predict type of flower



Pure means belonging to same Class.

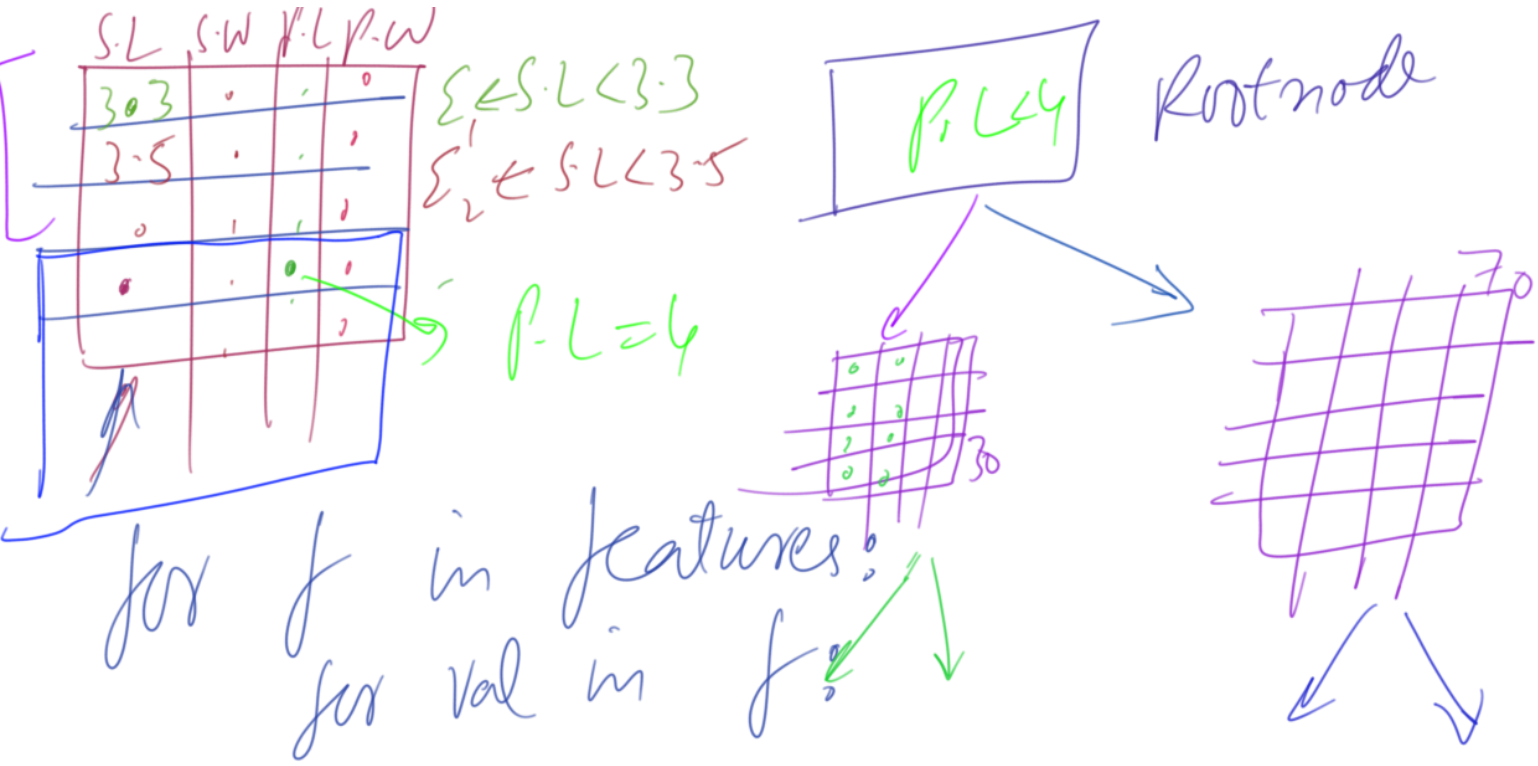


Leaf node should be pure

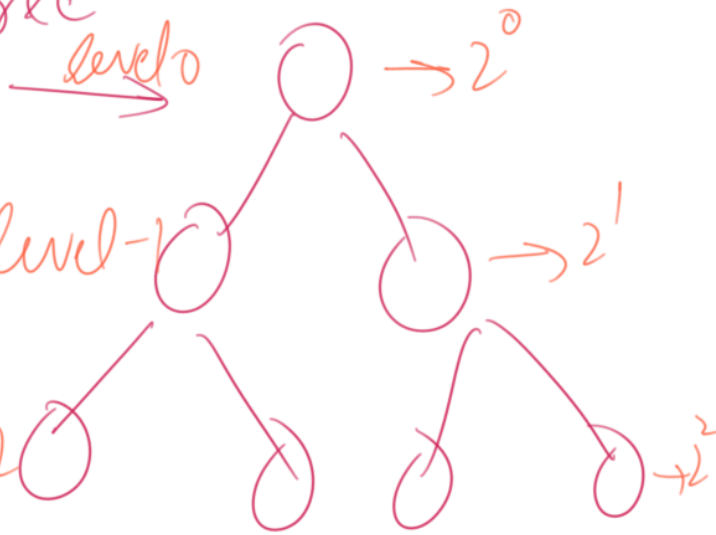
How to build Decision tree.

Joker is an agent of Entropy:-

Entropy:- measure of impurity in a set


S.L
S.W
P.L
P.W
3.3
3.5
P.L=4
P.L<4
Rootnode
30
70
for f in features:
for val in f:

⇒ Depth of a tree



depth-2 has 4 nodes

depth-8 has 64 nodes

depth 5 → 32 level-8 ⇒ 2^8 → 64 spots

How to use decision tree in scikit

from sklearn.tree import DecisionTreeClassifier

- fit
- predict

height

$h = 3 \rightarrow 2^3 = 8$

depth / Height & overfitting

→ In real life we don't build too deep trees.

How to choose tree depth?

for d in [4, 5, 6, 7, 8, 9, 10]:

tree = DecisionTreeClassifier(max_depth=d)

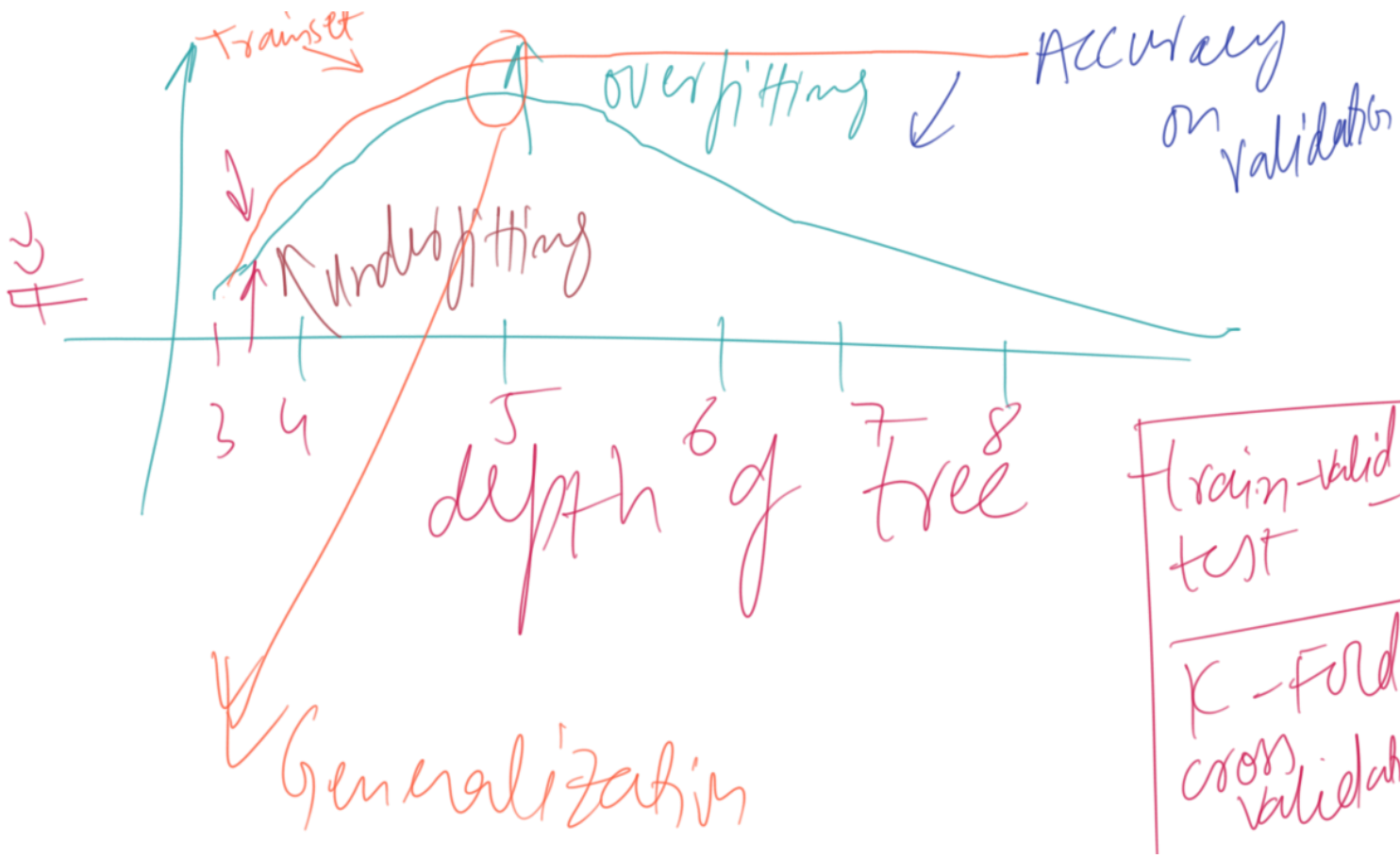
tree.fit(train_ex, train_labels)

train_Pred = tree.Predict(train_ex)

acc_train = accuracy_score(train_labels, train_Pred)

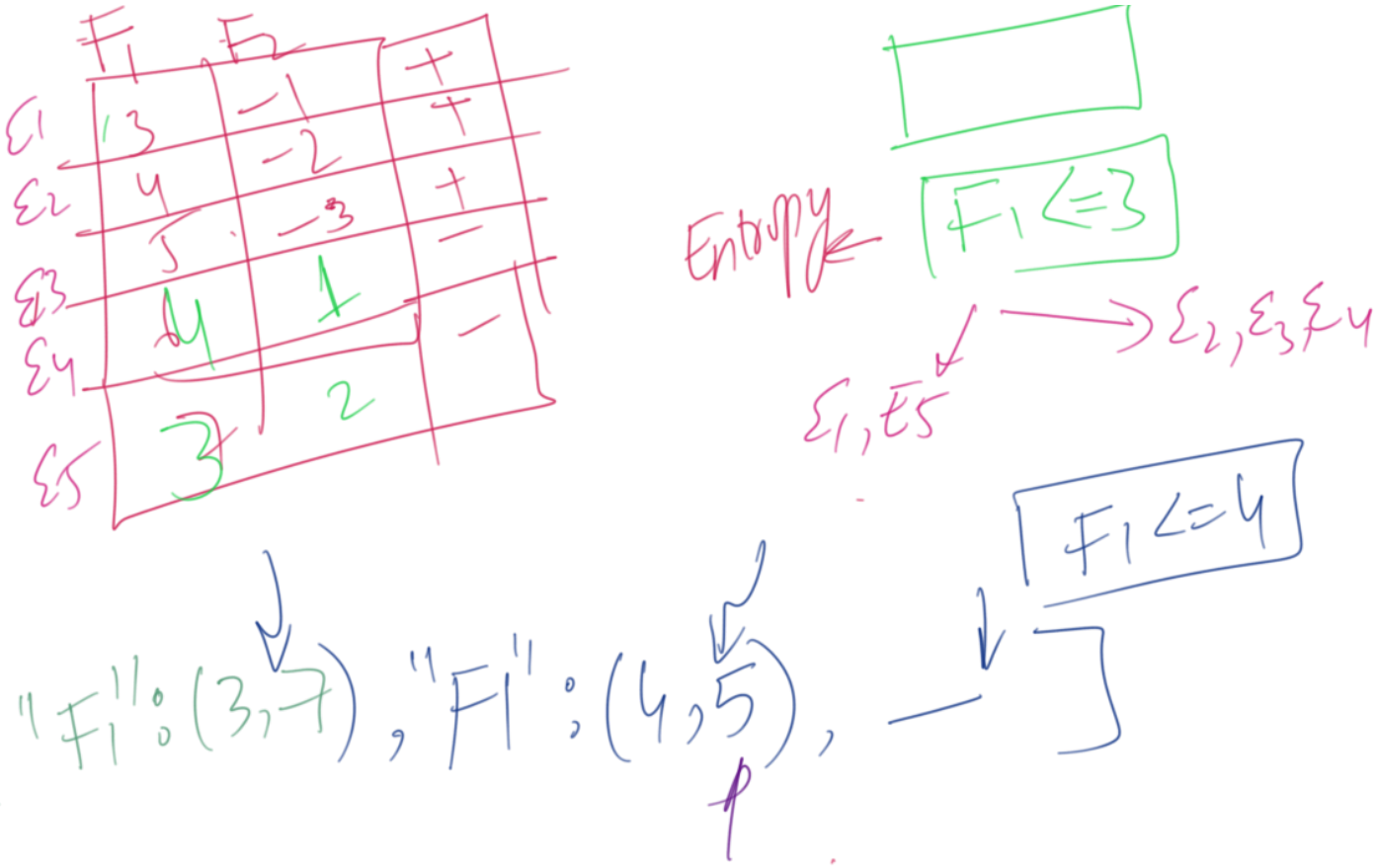
Pred = tree.Predict(test_ex)

acc = accuracy_score(test_ex, test_labels)



train-valid
test

K-Fold
cross validation


F1
F2
ε1
ε2
ε3
ε4
ε5
Entropy
F1<=3
ε1, ε5
ε2, ε3, ε4
F1<=4
["F1":(3,7), "F1":(4,5), —]

→ Entropy :-

$$E(S) = \sum_{\text{labels}} -P_{\text{label}} \log P_{\text{label}}$$

↑ 2 labels

↑ 2 labels

↑ 2 labels

$P(+) = 3/5$

$P(-) = 2/5$

$\rightarrow = -3/5 \log 3/5 - 2/5 \log 2/5$

$=$

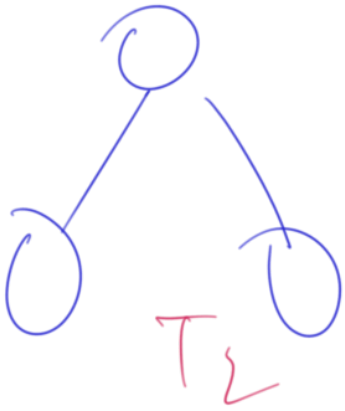
150 : 3

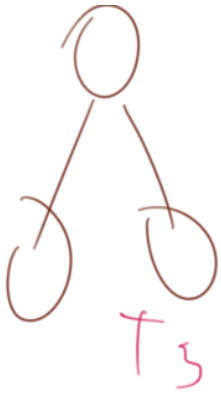
Random Forest ← many trees

Ensemble

"A committe of trees is always better than a single tree".

Instead of a sing tree we build many trees then take the majority vote.

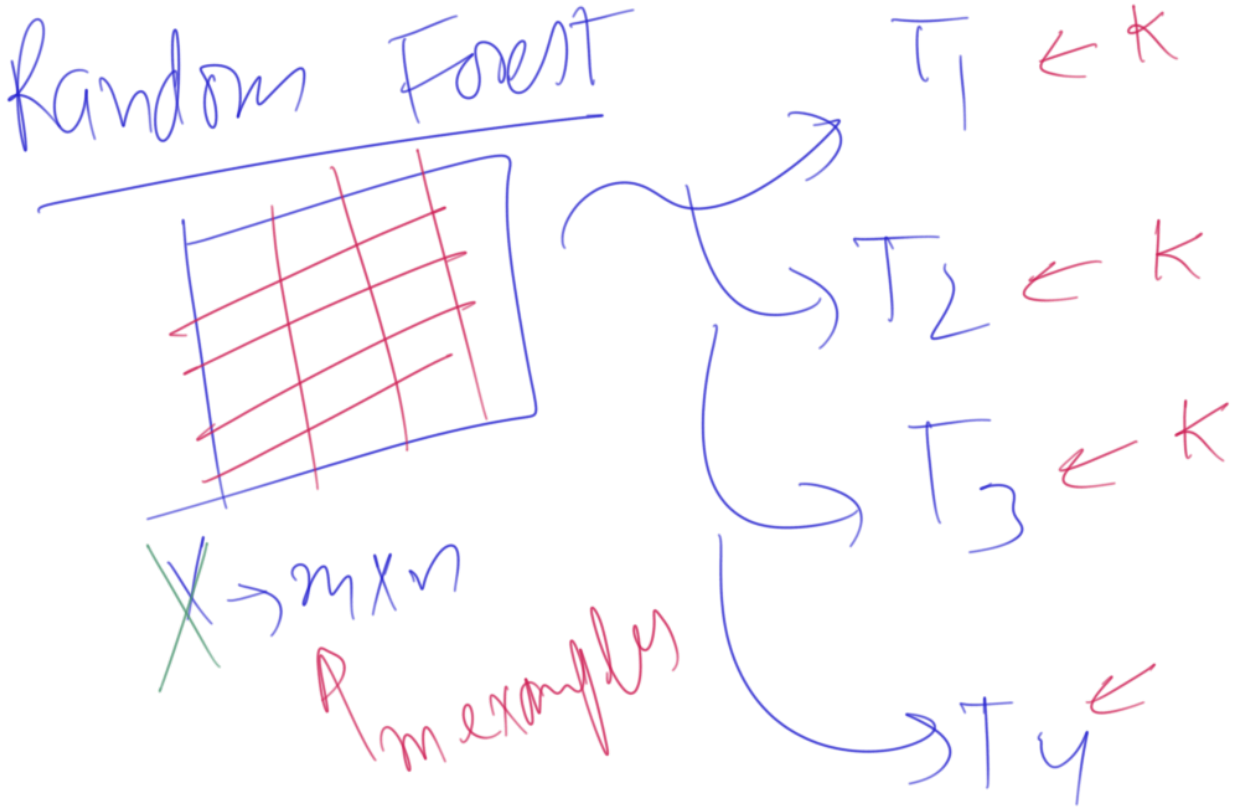




C1

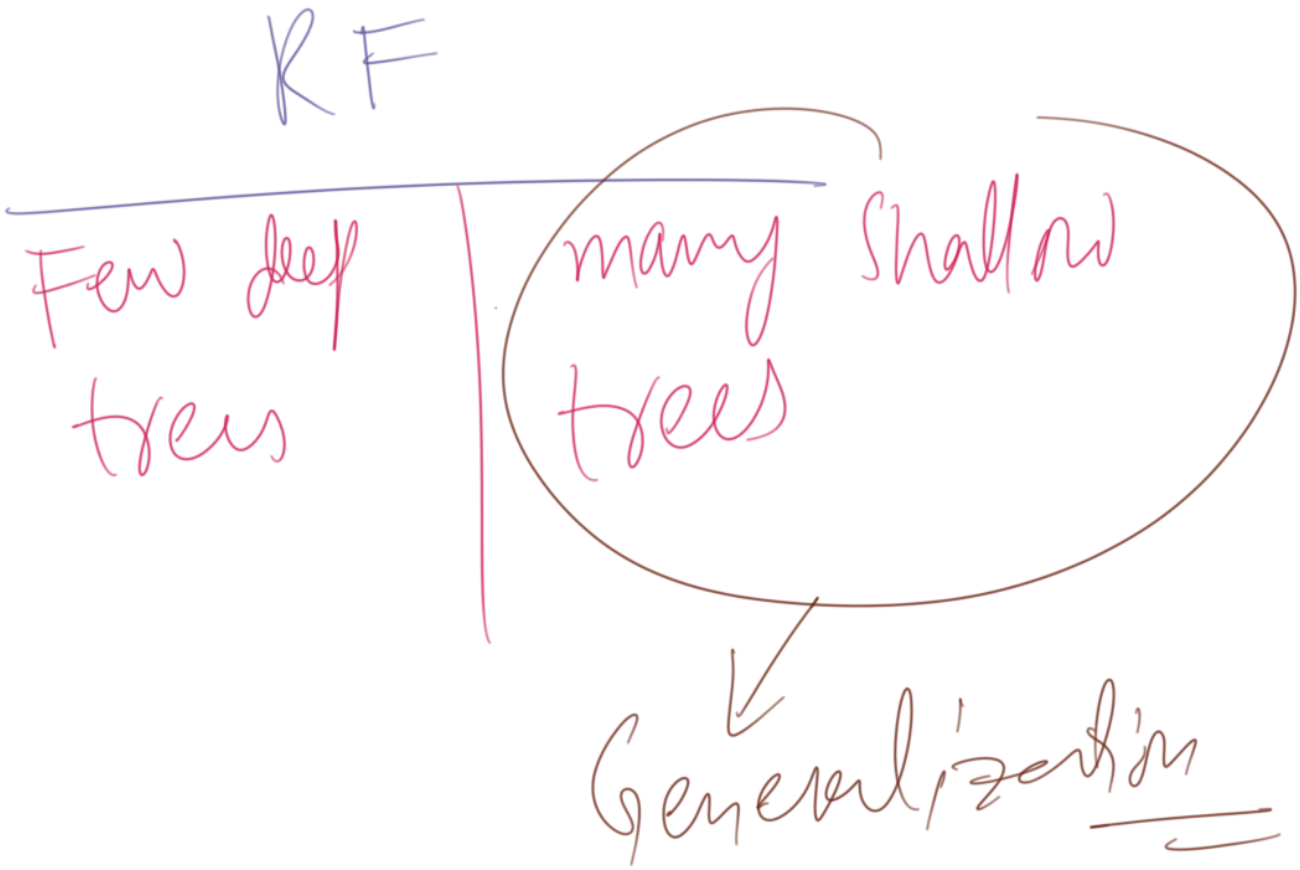
C2

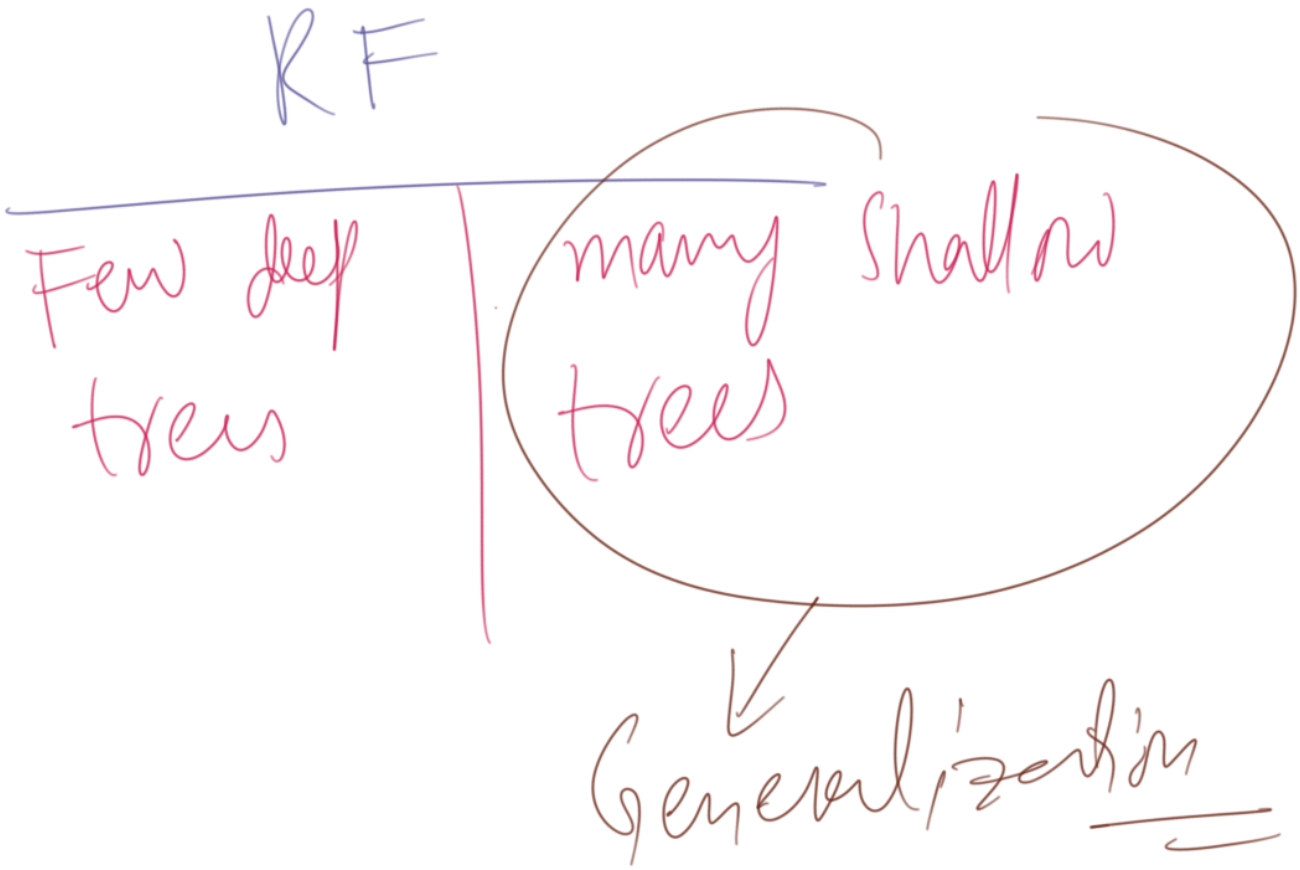
C1


Random Forest
$T_1 \leftarrow k$
$T_2 \leftarrow k$
$T_3 \leftarrow k$
$T_4 \leftarrow$
$X \rightarrow m \times n$
m examples

→ BIAS → Underfitting

→ VARIANCE → overfitting


RF
Few deep trees
many Shallow trees
Generalization


RF
Few deep trees
many Shallow trees
Generalization