Lesson Topics

- How PCA works
- PCA with sklearn
- Interpreting PCA results

Clustering

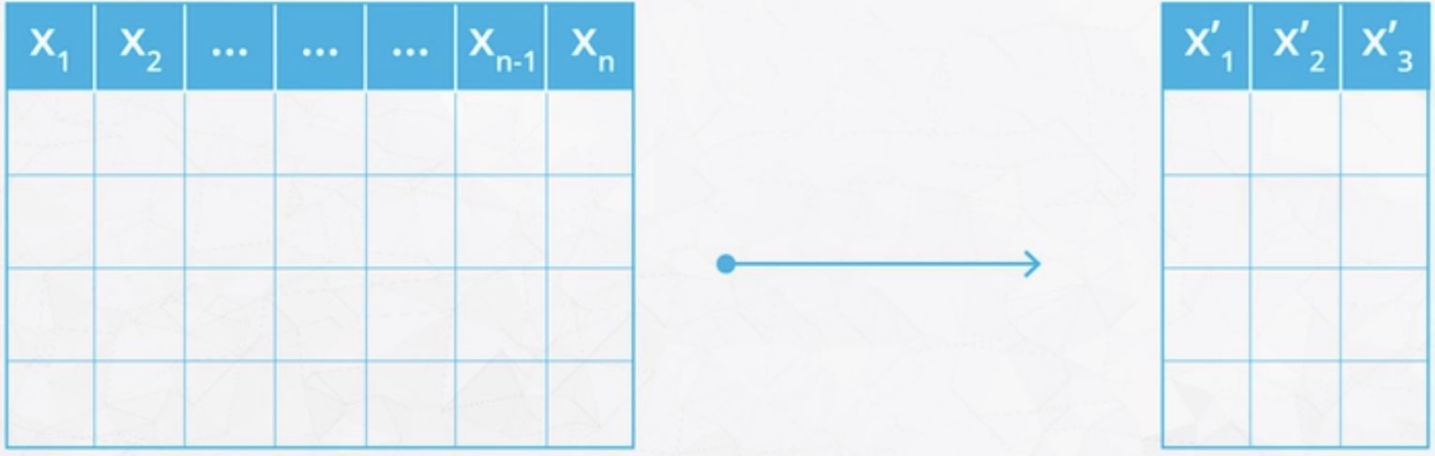
Dimensionality Reduction

So, I was running late, and I was going to grab a Nutrigrain Bar, but then noticed I was all out. So, I stopped by the coffee shop, and I had a coffee and then I was a little hungry, so I also grabbed a pastry.

"I had a coffee and a pastry."

" So, I was running late and...
I stopped by the coffee shop, and
I had a coffee, and then...
I grabbed a pastry. "

"I had a coffee and a pastry."



Size of House

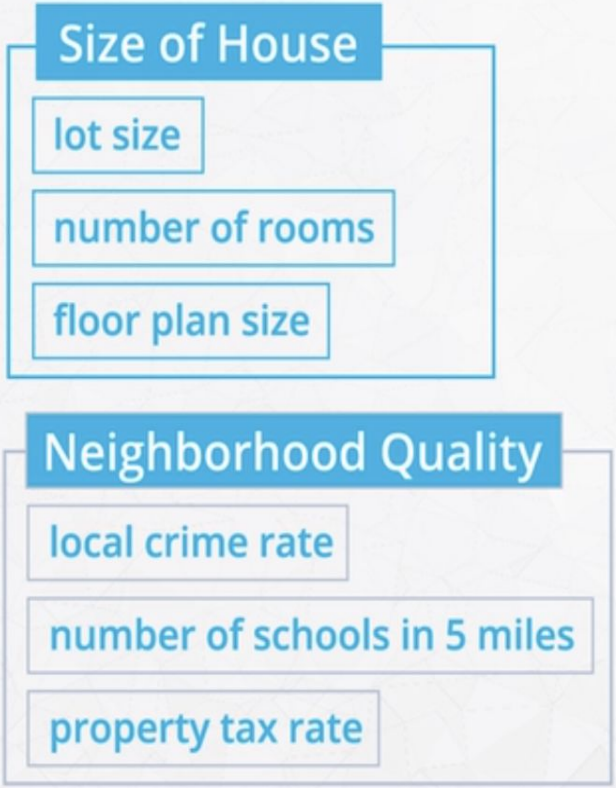
- lot size
- number of rooms
- floor plan size
- size of garage
- number of bedrooms
- number of bathrooms

Neighborhood Quality

- local crime rate
- number of schools in 5 miles
- property tax rate
- local median income
- average air quality
- distance to highway


Size of House
lot size
number of rooms
floor plan size
Neighborhood Quality
local crime rate
number of schools in 5 miles
property tax rate
floor plan size
FEATURE 2

Latent Features



Latent Feature

Feature that is not directly observed, but underlies other features that are observed

I feel that my manager is supportive of my growth within the company.

I feel confident that I can contribute to the company goals.

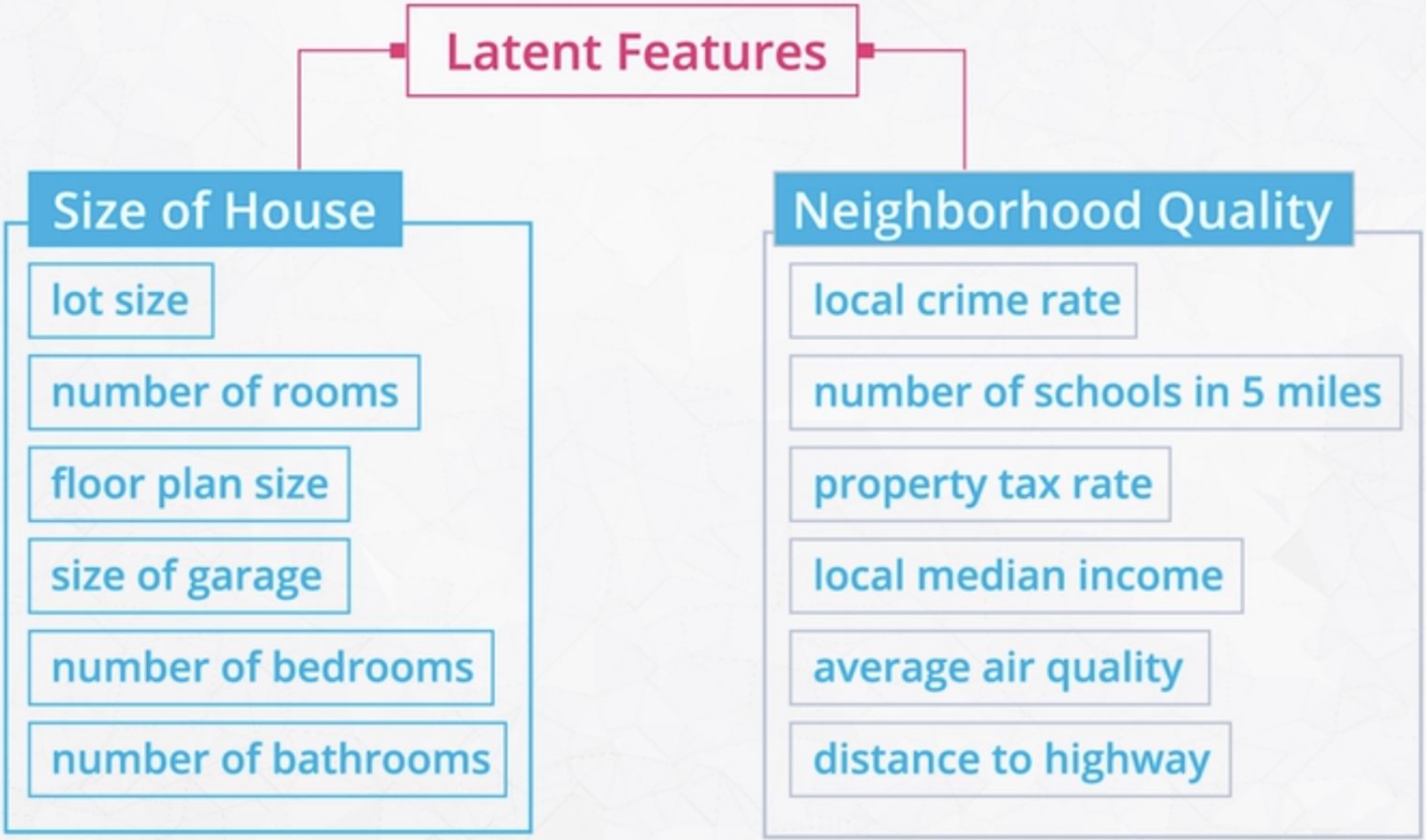
I feel that my work contributes positively to the world.

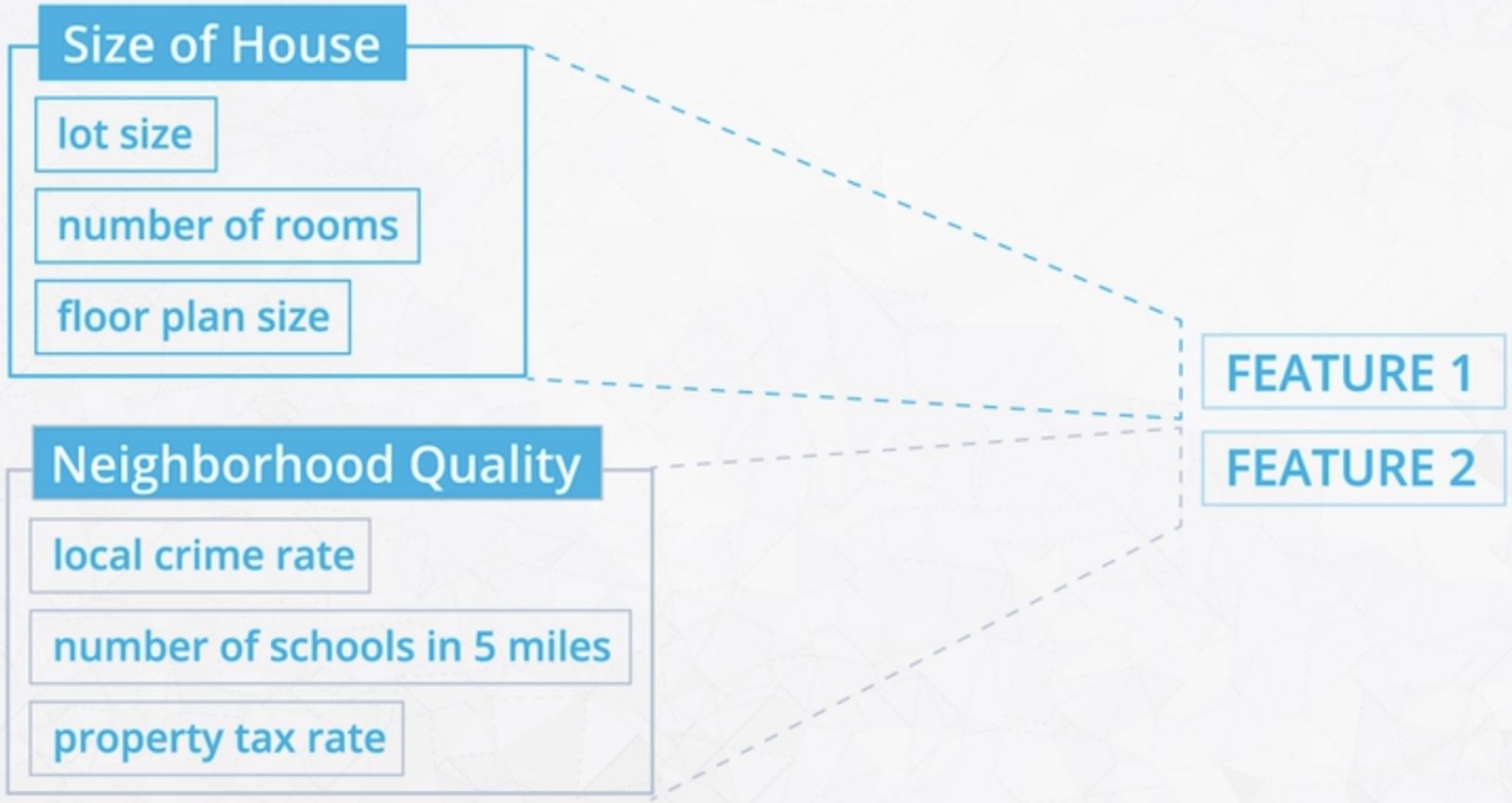
I feel that my opinions are heard when making decisions about company decisions.

My peers are supportive of my contributions.

How to Reduce Features


Size of House
lot size
number of rooms
floor plan size
size of garage
number of bedrooms
number of bathrooms
Neighborhood Quality
local crime rate
number of schools in 5 miles
property tax rate
local median income
average air quality
distance to highway

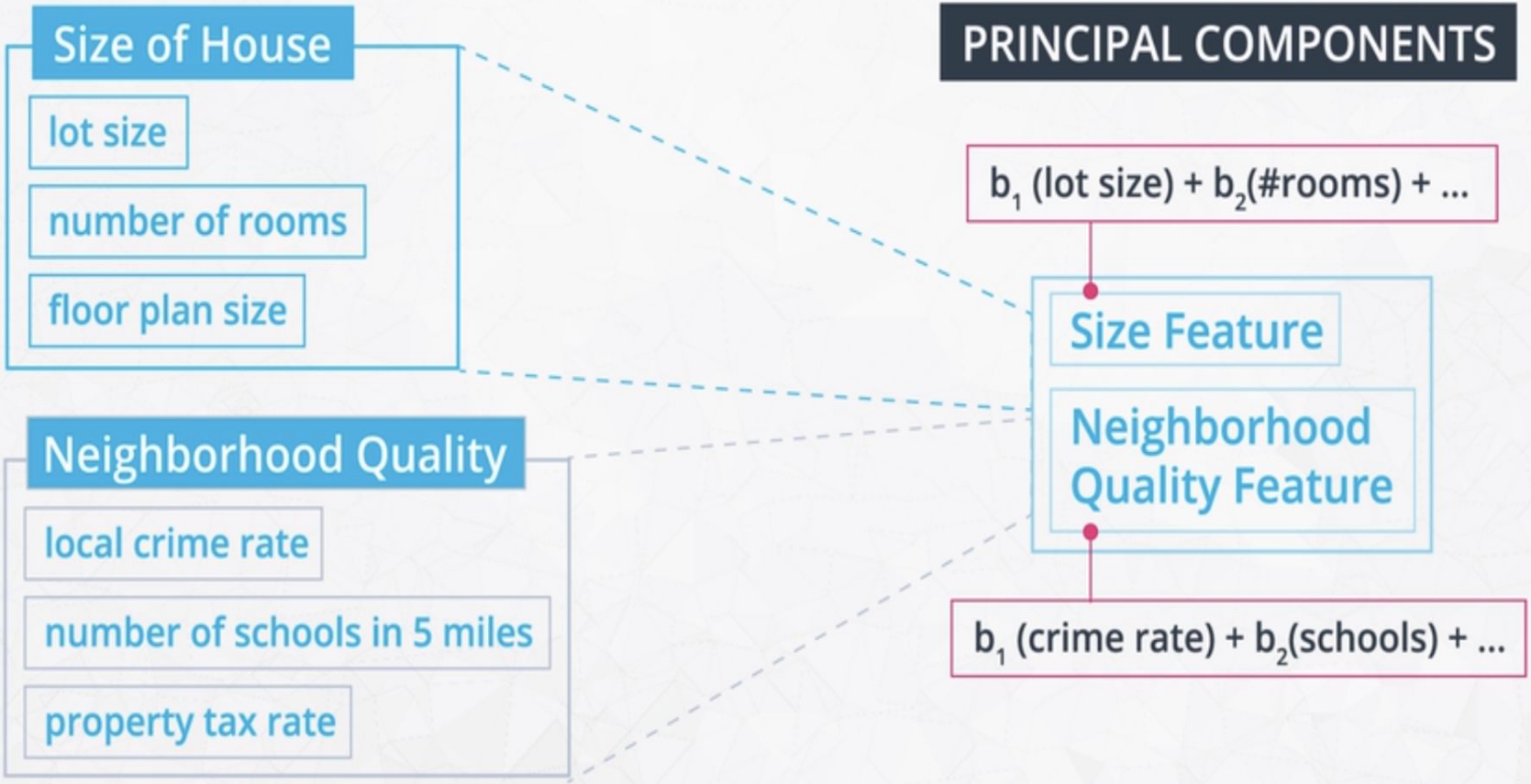

Latent Features
Size of House
lot size
number of rooms
floor plan size
size of garage
number of bedrooms
number of bathrooms
Neighborhood Quality
local crime rate
number of schools in 5 miles
property tax rate
local median income
average air quality
distance to highway


Size of House
lot size
number of rooms
floor plan size
Neighborhood Quality
local crime rate
number of schools in 5 miles
property tax rate
FEATURE 1
FEATURE 2

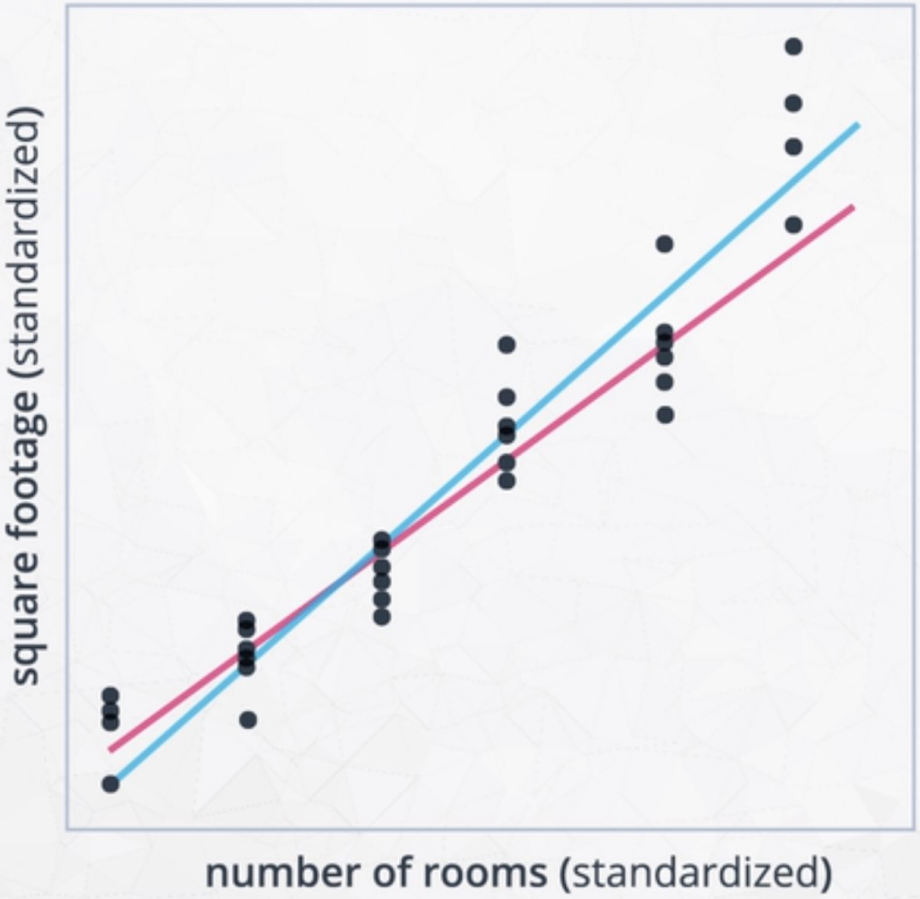

Size of House
lot size
number of rooms
floor plan size
Neighborhood Quality
local crime rate
number of schools in 5 miles
property tax rate
floor plan size
local crime rate

Dimensionality Reduction


Size of House
lot size
number of rooms
floor plan size
Neighborhood Quality
local crime rate
number of schools in 5 miles
property tax rate
b1 (lot size) + b2(#rooms) + ...
Size Feature
Neighborhood Quality Feature
b1 (crime rate) + b2(schools) + ...


Size of House
lot size
number of rooms
floor plan size
Neighborhood Quality
local crime rate
number of schools in 5 miles
property tax rate
PRINCIPAL COMPONENTS
b1 (lot size) + b2(#rooms) + ...
Size Feature
Neighborhood Quality Feature
b1 (crime rate) + b2(schools) + ...

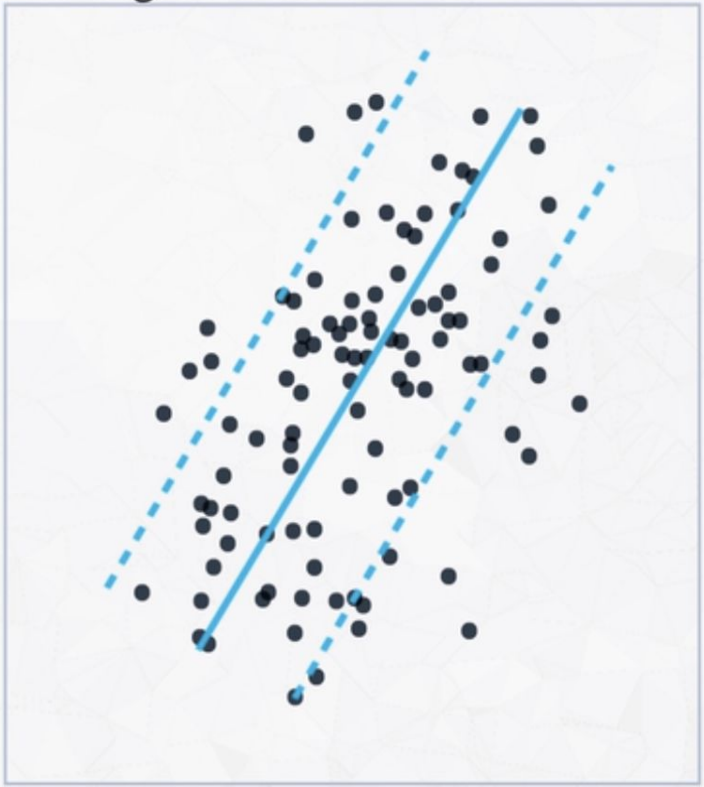
PCA Properties

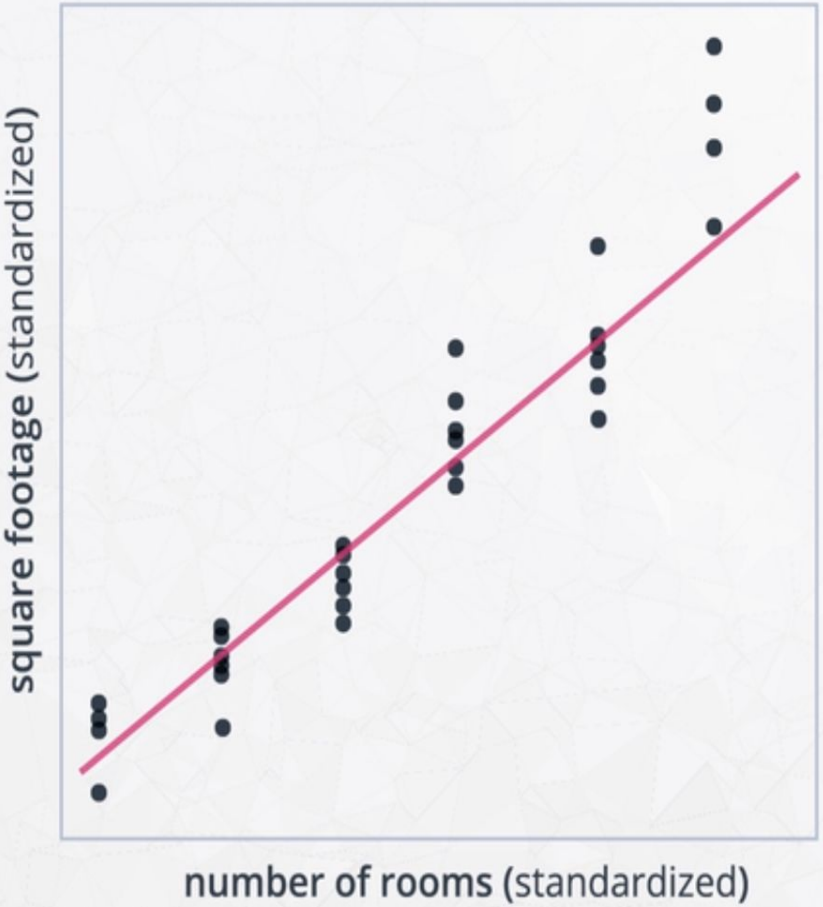

square footage (standardized)
number of rooms (standardized)

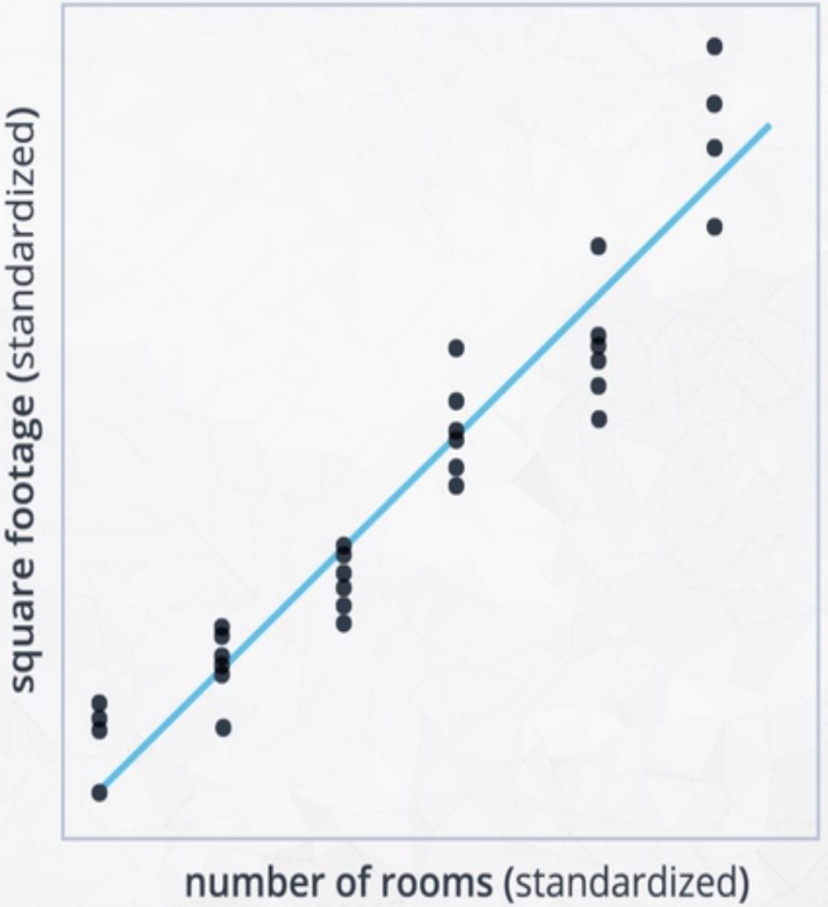
Properties of Principal Components

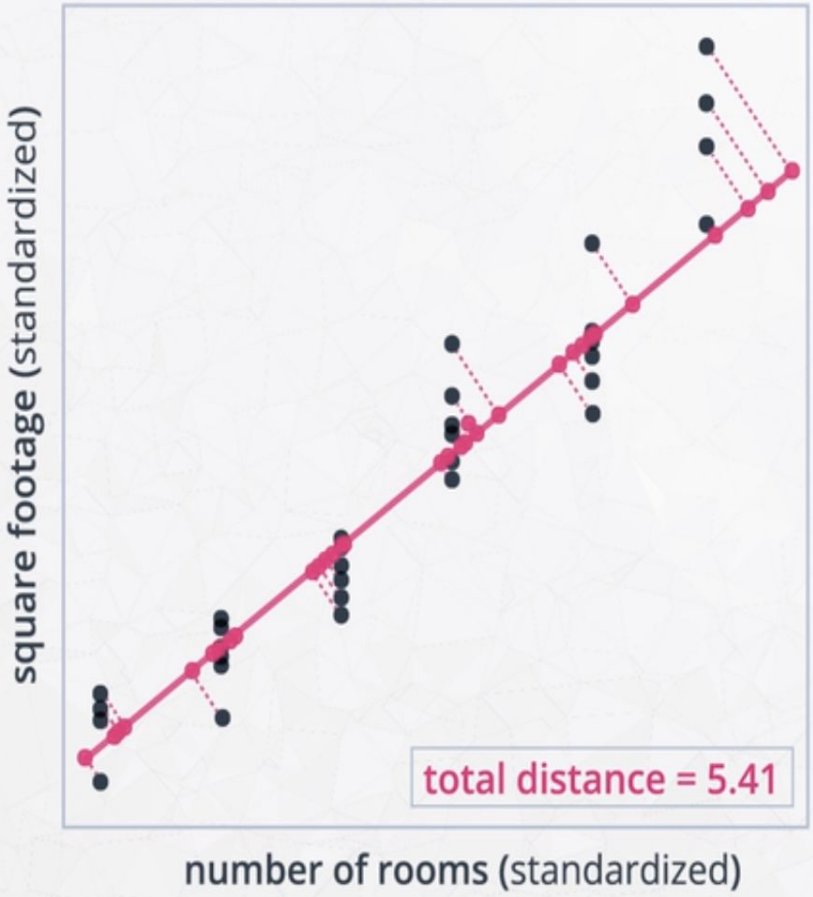
- Each component captures the largest amount of variance left in the data

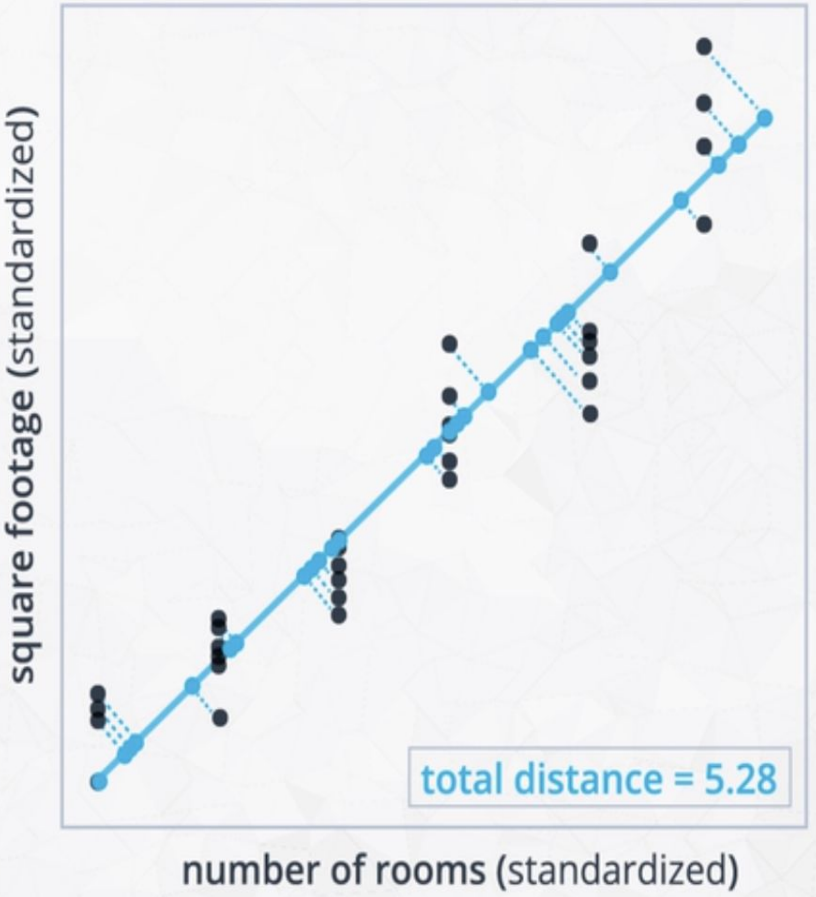
The largest variance




square footage (standardized)
number of rooms (standardized)

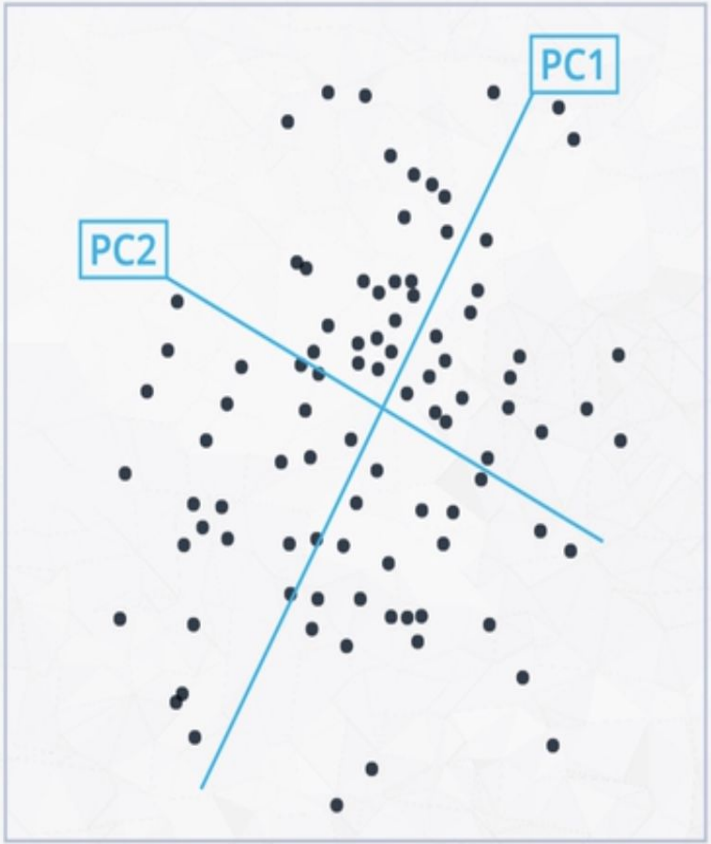

square footage (standardized)
number of rooms (standardized)


square footage (standardized)
total distance = 5.41
number of rooms (standardized)

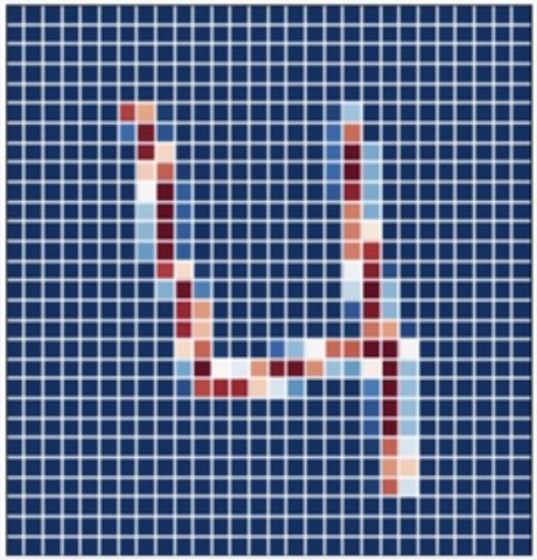

square footage (standardized)
total distance = 5.28
number of rooms (standardized)

Properties of Principal Components

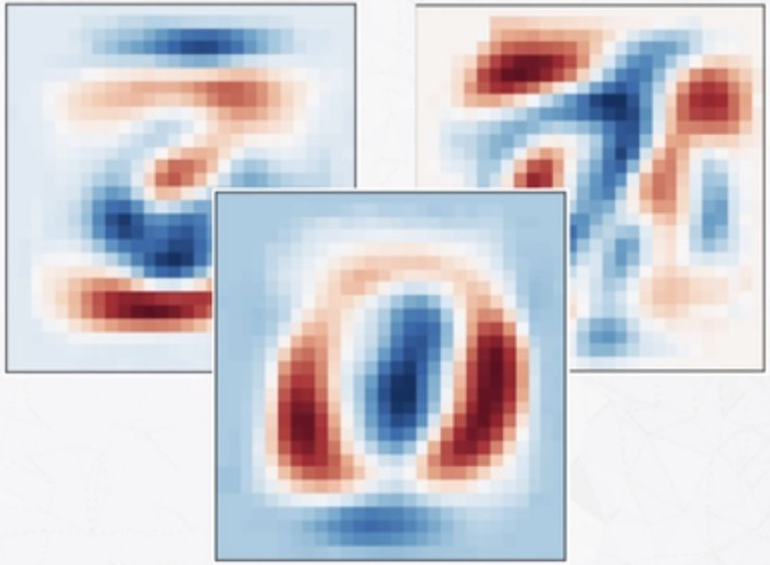
- Each component captures the largest amount of variance left in the data
- Components are orthogonal to one another



When to use PCA?



PCA



4

4

PCA Recap

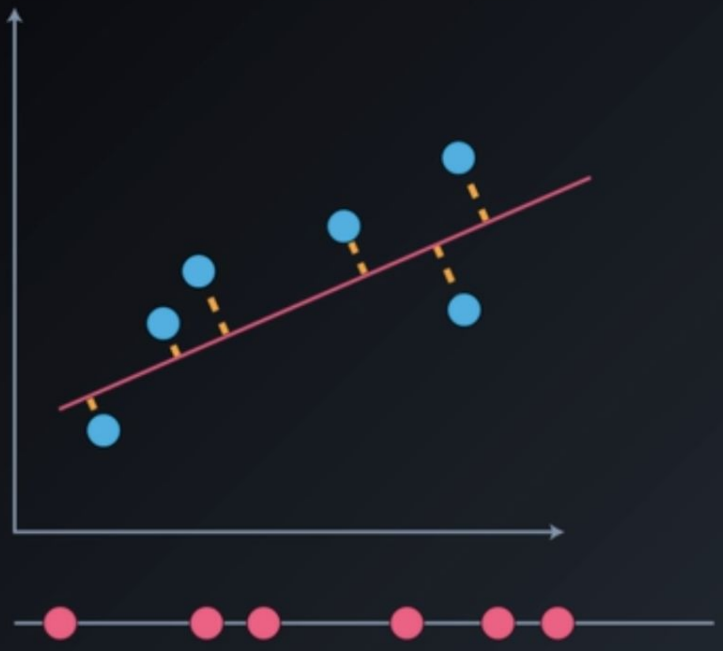
Principal Component Analysis (PCA)

- Dimensionality reduction technique
- Finds the directions of maximal variance - principal components

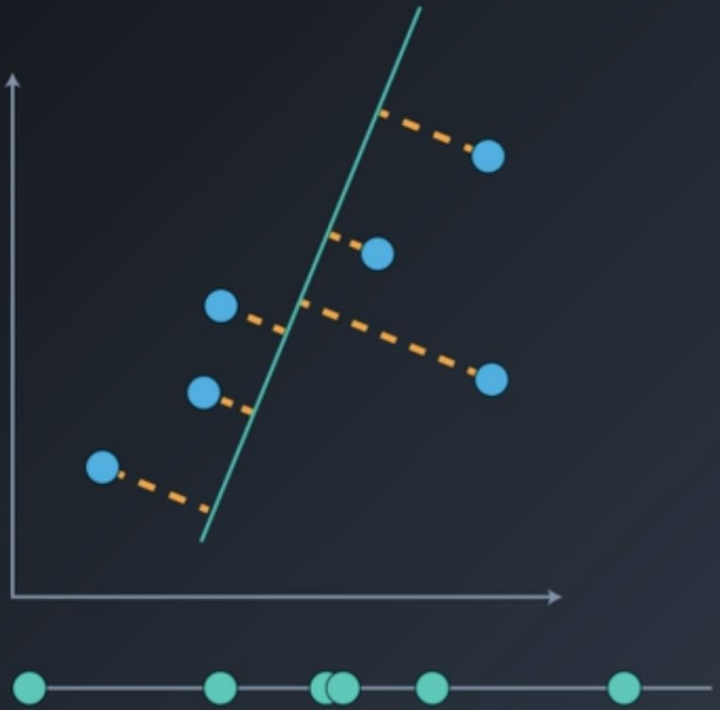
RANDOM PROJECTION

RANDOM PROJECTION

PCA



RANDOM PROJECTION



RANDOM PROJECTION

from d dimensions to k dimensions

$$X^{RP}_{k\text{x}N} = R_{k\text{x}d} X_{d\text{x}N}$$

$$\begin{bmatrix} X^{RP}_{11} & X^{RP}_{12} & \dots & X^{RP}_{1n} \\ X^{RP}_{21} & X^{RP}_{22} & \dots & X^{RP}_{2n} \\ \dots & \dots & \dots & \dots \\ X^{RP}_{k1} & X^{RP}_{k2} & \dots & X^{RP}_{kn} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1d} \\ r_{21} & r_{22} & \dots & r_{2d} \\ \dots & \dots & \dots & \dots \\ r_{k1} & r_{k2} & \dots & r_{kd} \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1n} \\ X_{21} & X_{22} & \dots & X_{2n} \\ X_{31} & X_{32} & \dots & X_{3n} \\ X_{41} & X_{42} & \dots & X_{4n} \\ \dots & \dots & \dots & \dots \\ X_{d1} & X_{d2} & \dots & X_{dn} \end{bmatrix}$$

RANDOM PROJECTION

from d dimensions to k dimensions

12,000

1,500

docID	a2i	aaa	aaai	aapo	aat	aazhang	abandonment	abbott	...	zorn	zornetzer	zoubin	zucker	zue	zur	zurich	zwislocki
1	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1496	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
1497	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
1498	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
1499	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
1500	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0

RANDOM PROJECTION

6,268

1,500

	0	1	2	3	4	...	6263	6264	6265	6266	6267
0	2.131532	0.133221	0.266441	0.598649	0.266441	...	0.932545	-1.198987	0.266441	0.666104	-1.065766
1	0.532883	-1.998311	0.266441	-0.532883	-0.932545	...	2.797635	0.000000	0.266441	0.000000	0.266441
2	-0.932545	0.000000	0.133221	0.666104	0.133221	...	0.133221	-0.532883	0.133221	0.399662	0.000000
3	0.133221	-2.131532	0.399662	0.532883	-0.399662	...	1.065766	-0.133221	0.799324	1.865090	-0.133221
4	0.399662	-2.131532	0.000000	-0.532883	0.399662	...	1.065766	-1.332207	1.332207	0.000000	1.065766
...	...	...	...	...	...	...	...	...	...	...	...
1495	5.728491	-0.266441	0.133221	-0.266441	-1.332207	...	0.266441	-0266441	0.266441	-1.598649	-1.198987
1496	-0.266441	-1.598649	0.266441	-1.332207	-0.133221	...	2.264752	0.133221	0.666104	-0.399662	0.666104
1497	-1.198987	-1.332207	0.133221	-0.532883	0.532883	...	0.932545	-0.133221	0.266441	-1.198987	0.000000
1498	2.131532	-0.932545	0.399662	-0.666104	0.532883	...	1.198987	-1.332207	0.133221	-1.198987	-0.799324
1499	0.133221	-0.399662	0.666104	-0.266441	0.399662	...	0.000000	-0.133221	-0.133221	-0.133221	0.266441

RANDOM PROJECTION

from d dimensions to k dimensions

Johnson-Lindenstrauss lemma

A dataset of N points in high-dimensional Euclideanspace can be mapped down to a space in much lower dimension in a way that preserves the distance between the points to a large degree.

12,000

N

docID	a2i	aaa	aaai	aapo	aat	aazhang	abandonment	abbott	...	zorn	zornetzer	zoubin	zucker	zue	zur	zurich	zwislocki
1	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0

x

$$\begin{bmatrix} r_{11} & \dots & r_{1d} \\ \dots & \dots & \dots \\ r_{k1} & \dots & r_{kd} \end{bmatrix}$$

6,268

N

	0	1	2	3	4	...	6263
0	2.131532	0.133221	0.266441	0.598649	0.266441	...	0.932545
1	0.532883	-1.998311	0.266441	-0.532883	-0.932545	...	2.797635
2	-0.932545	0.000000	0.133221	0.666104	0.133221	...	0.133221
3	0.133221	-2.131532	0.399662	0.532883	-0.399662	...	1.065766
4	0.399662	-2.131532	0.000000	-0.532883	0.399662	...	1.065766

RANDOM PROJECTION

from d dimensions to k dimensions

Johnson-Lindenstrauss lemma

A dataset of N points in high-dimensional Euclideanspace can be mapped down to a space in much lower dimension in a way that preserves the distance between the points to a large degree.

12,000

N

docID	a2i	aaa	aaai	aapo	aat	aazhang	abandonment	abbott	...	zorn	zornetzer	zoubin	zucker	zue	zur	zurich	zwislocki
1																	
2																	
3	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0

x

$$\begin{bmatrix} r_{11} & \dots & r_{1d} \\ \dots & \dots & \dots \\ r_{k1} & \dots & r_{kd} \end{bmatrix}$$

6,268

N

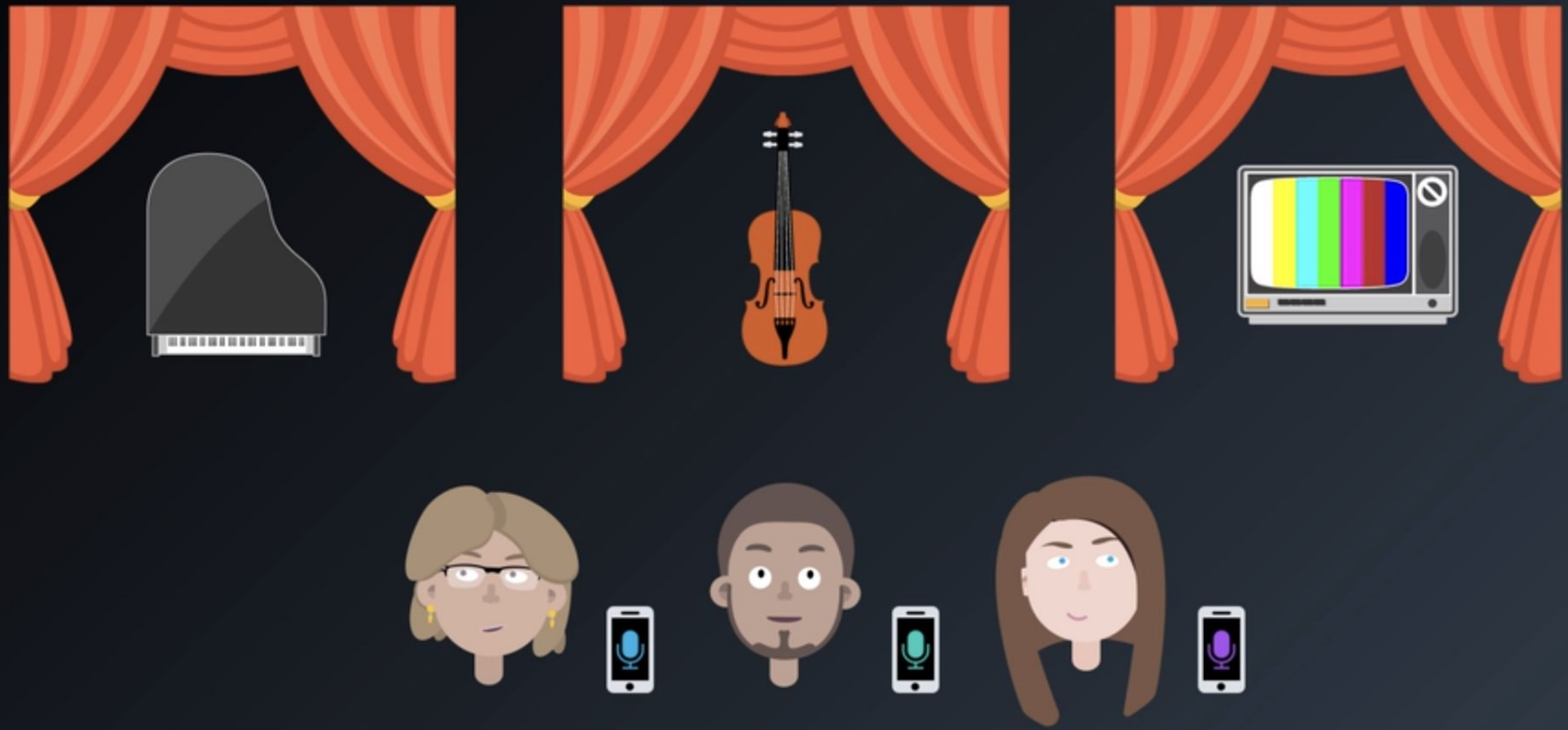
	0	1	2	3	4	...	6263
0							
1							
2	-0.932545	0.000000	0.133221	0.666104	0.133221	...	0.133221
3	0.133221	-2.131532	0.399662	0.532883	-0.399662	...	1.065766
4	0.399662	-2.131532	0.000000	-0.532883	0.399662	...	1.065766

$$\underset{0.1}{(1-eps)} \|u-v\|^2 < \underset{125.8}{\|p(u)-p(v)\|^2} < \underset{125.6}{(1+eps)} \|u-v\|^2$$

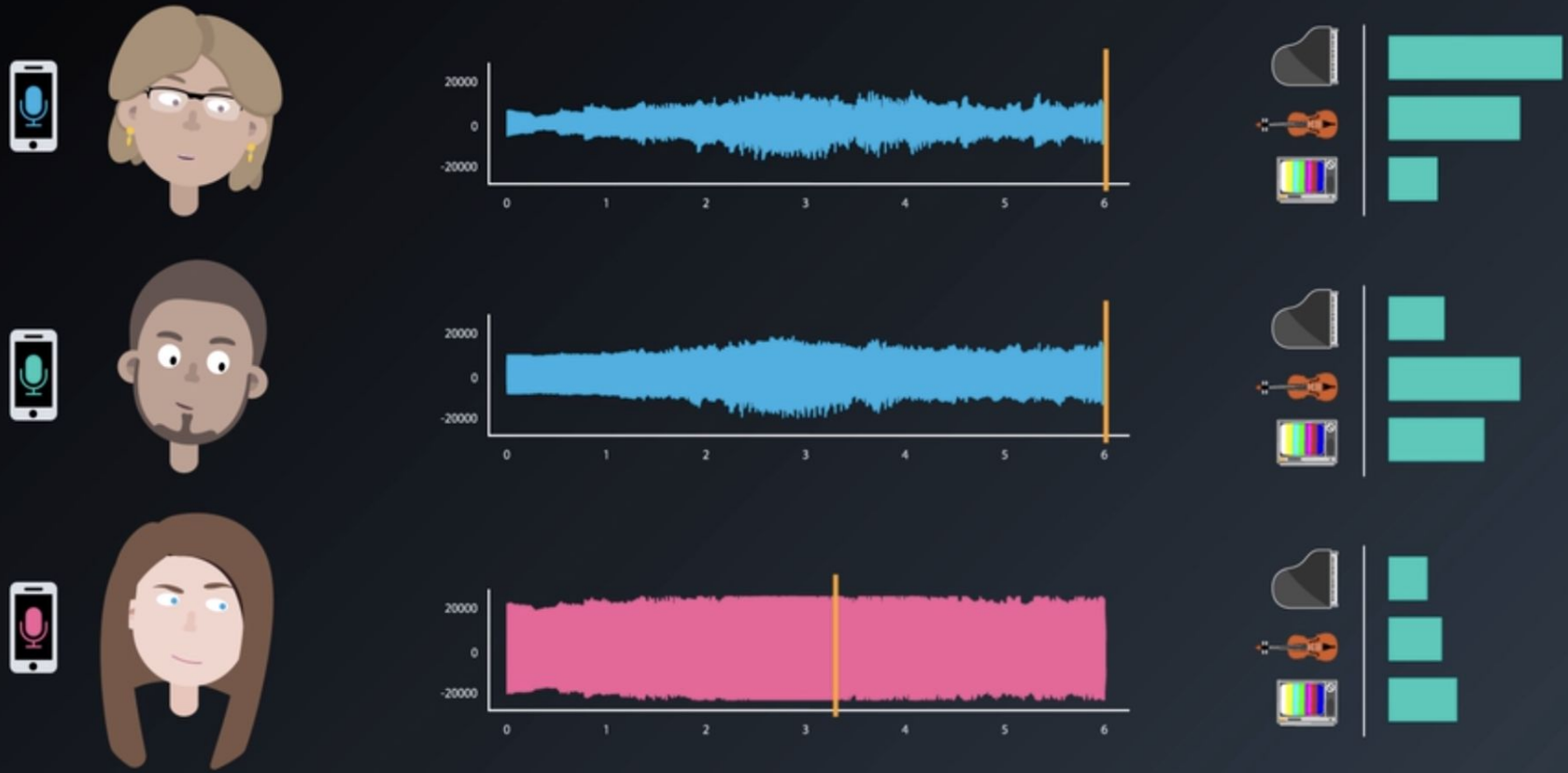
$$0.9 \times (125.6)^2 \qquad (125.8)^2 \qquad 1.1 \times (125.6)^2$$

INDEPENDENT COMPONENT ANALYSIS | MOTIVATION

INDEPENDENT COMPONENT ANALYSIS | MOTIVATION



INDEPENDENT COMPONENT ANALYSIS | MOTIVATION



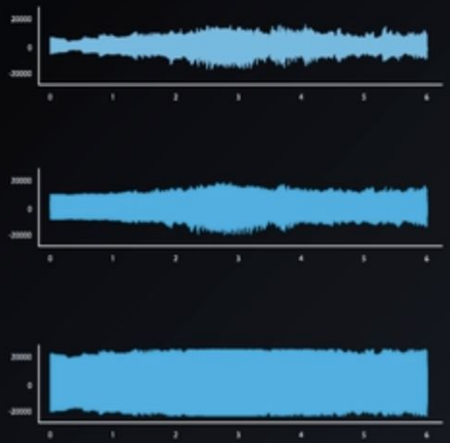
INDEPENDENT COMPONENT ANALYSIS | MOTIVATION

Is there a way to retrieve the original signals/datasets?

Yes, by using:

ICA

INDEPENDENT COMPONENT ANALYSIS | MOTIVATION

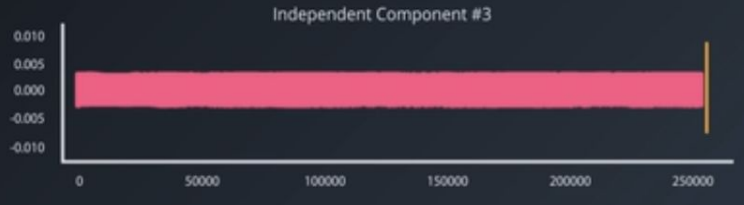


Input

ICA

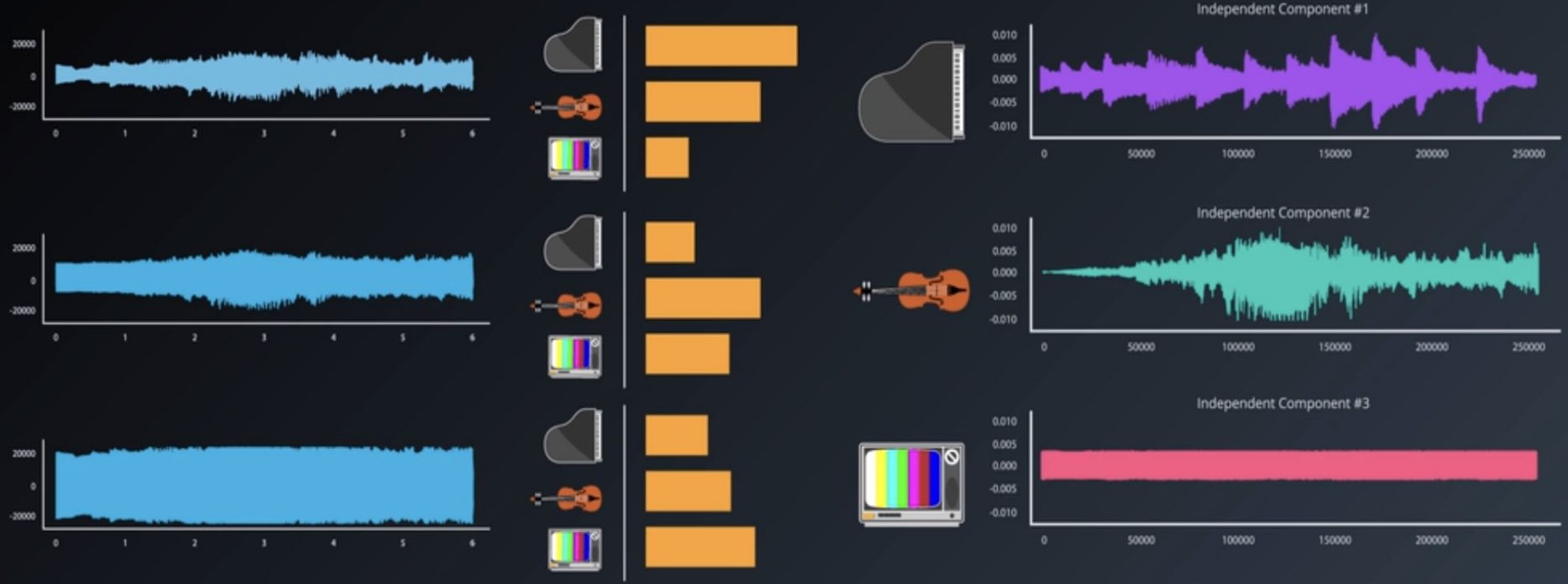
n_components = 3

Output



INDEPENDENT COMPONENT ANALYSIS | ALGORITHM

$$X = AS$$



INDEPENDENT COMPONENT ANALYSIS | ALGORITHM

X = AS

S = WX

$$W = A^{-1}$$

S = W X

INDEPENDENT COMPONENT ANALYSIS | ALGORITHM

Independent Component Analysis: Algorithms and Applications

Aapo Hyvärinen and Erkki Oja
Neural Networks Research Centre
Helsinki University of Technology
P.O. Box 5400, FIN-02015 HUT, Finland
Neural Networks, 13(4-5):411-430, 2000

$$w^+ = E\left\{xg(w^T x)\right\} - E\left\{g'(w^T x)\right\} w$$

$$g1(u) = tanh(a_1 u),$$

$$\text{Let } W = (WW^T)^{-1/2} W$$

FastICA Algorithm:

#1: CENTER, WHITEN X

#2: CHOOSE INITIAL RANDOM WEIGHT MATRIX W_1, W_2,...W_n

#3: ESTIMATE W, CONTAINING VECTORS

#4: DECORRELATE W

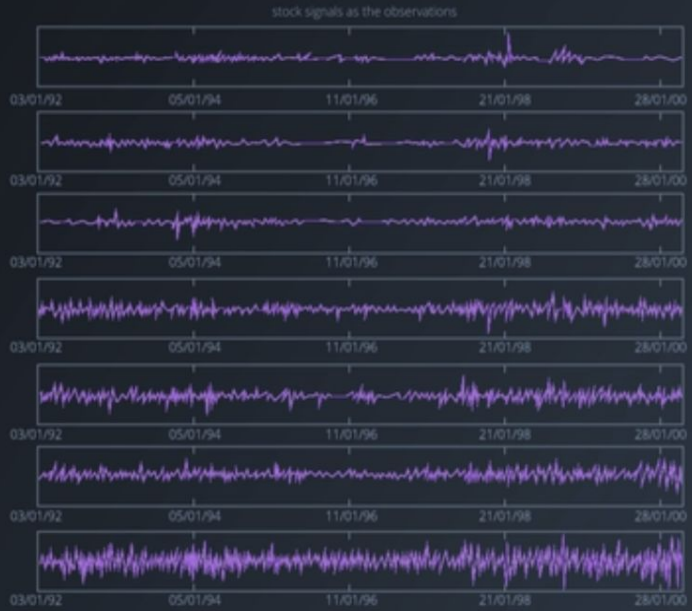
#5: REPEAT FROM STEP #3 UNTIL CONVERGED

INDEPENDENT COMPONENT ANALYSIS | APPLICATIONS

Makeig, Scott, et al.
"Independent component analysis of electroencephalographic data."
Advances in neural information processing systems. 1996.

INDEPENDENT COMPONENT ANALYSIS | APPLICATIONS

Cha, Siu-Ming, and Lai-Wan Chan.
"Applying independent component analysis to factor model in finance."
Intelligent Data Engineering and Automated Learning - IDEAL 2000.
Data Mining, Financial Engineering, and Intelligent Agents (2000); 161-173.



INDEPENDENT COMPONENT ANALYSIS | APPLICATIONS

Kiviluoto, Kimmo, and Erkki Oja.
"Independent Component Analysis for Parallel Financial Time Series"
ICONIP. Vol. 2. 1998.

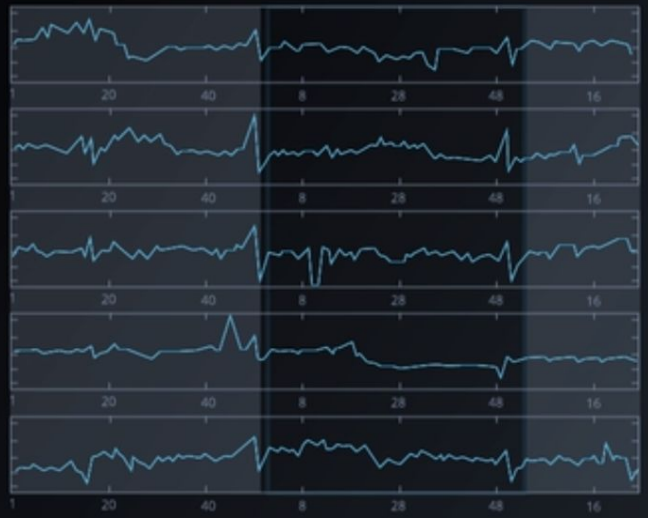


Figure 1. Five samples of the original cashflow time series. (mean removed, normalized to unit standard deviation).

Figure 2. Four fundamental factors found with ICA.v

ENSEMBLE METHODS

Weak Learners

Strong Learner

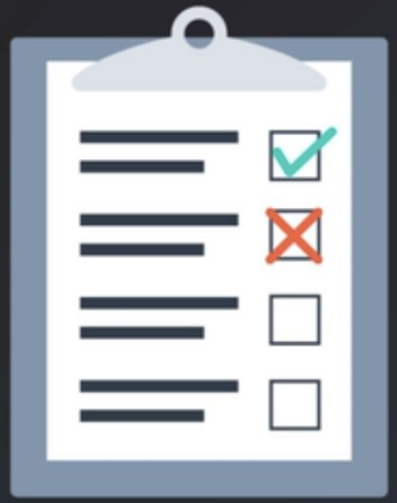
ENSEMBLE METHODS

Bagging

Boosting

BAGGING

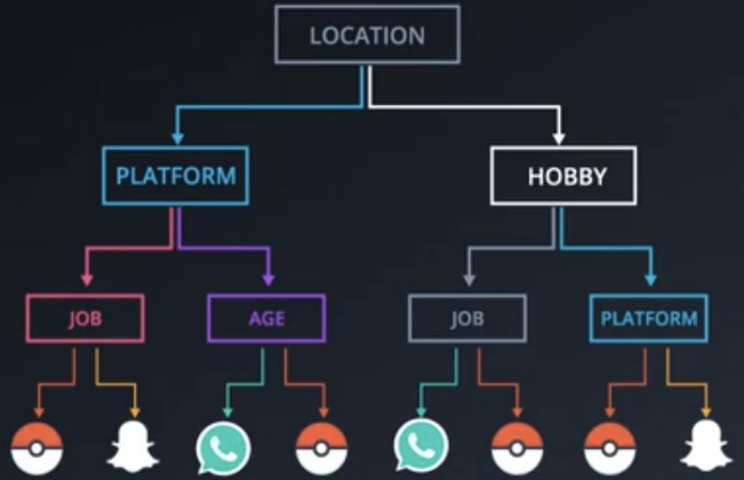
BOOSTING



BOOSTING

Large Tables

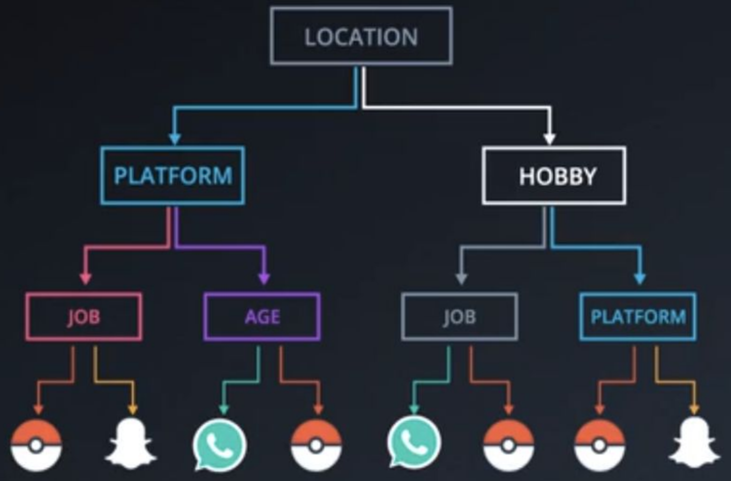
Gender	Age	Location	Platform	Job	Hobby	App
F	15	US	iOS	School	Videogames	
F	25	France	Android	Work	Tennis	
M	32	Chile	iOS	Temp	Tennis	
F	40	China	iOS	Retired	Chess	
M	12	US	Android	School	Tennis	
M	14	Australia	Android	School	Videogames	



If client is male, between 15 and 25, in the US, on Android, in school, likes tennis, pizza, but does not like long walks on the beach, then they are likely to download Pokemon Go.

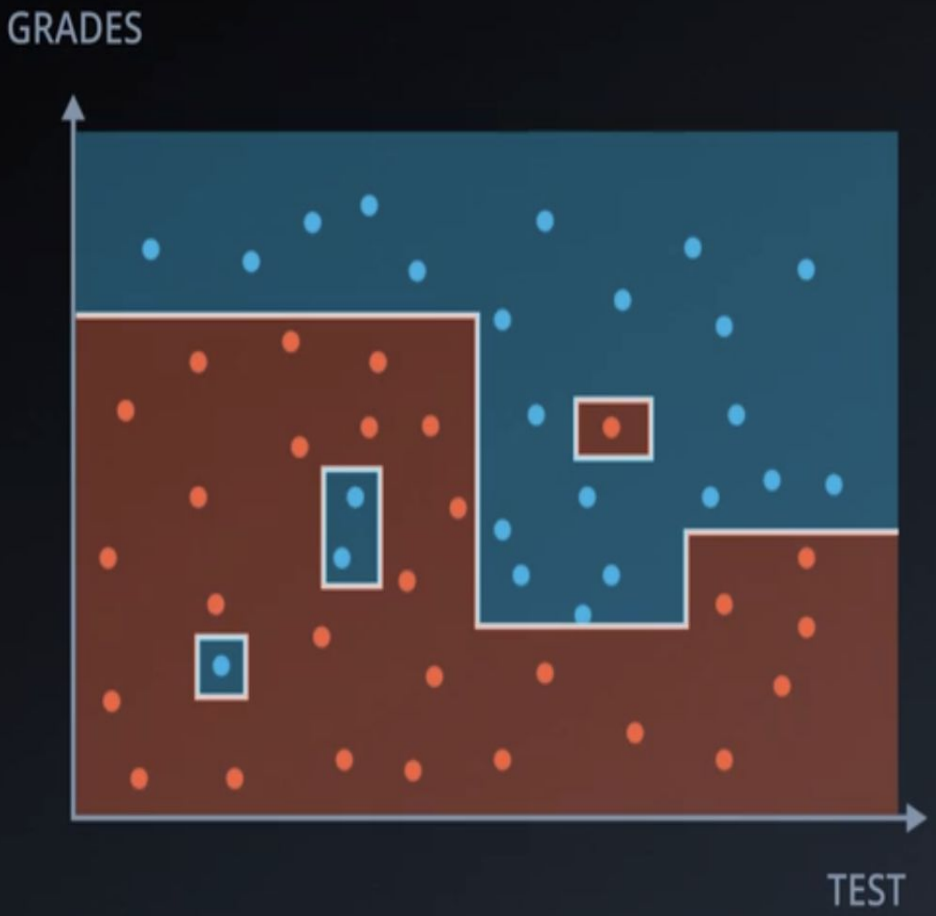
Large Tables

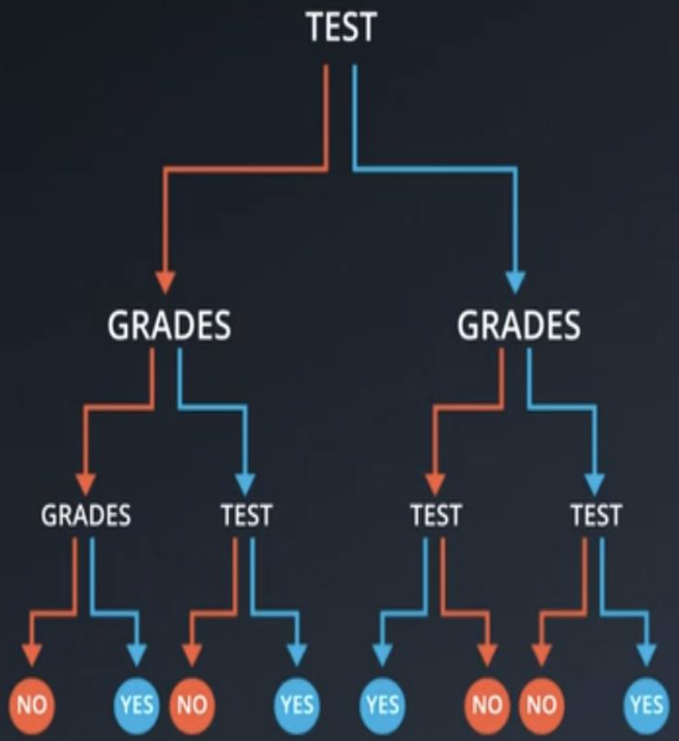
Gender	Age	Location	Platform	Job	Hobby	App
F	15	US	iOS	School	Videogames	
F	25	France	Android	Work	Tennis	
M	32	Chile	iOS	Temp	Tennis	
F	40	China	iOS	Retired	Chess	
M	12	US	Android	School	Tennis	
M	14	Australia	Android	School	Videogames	

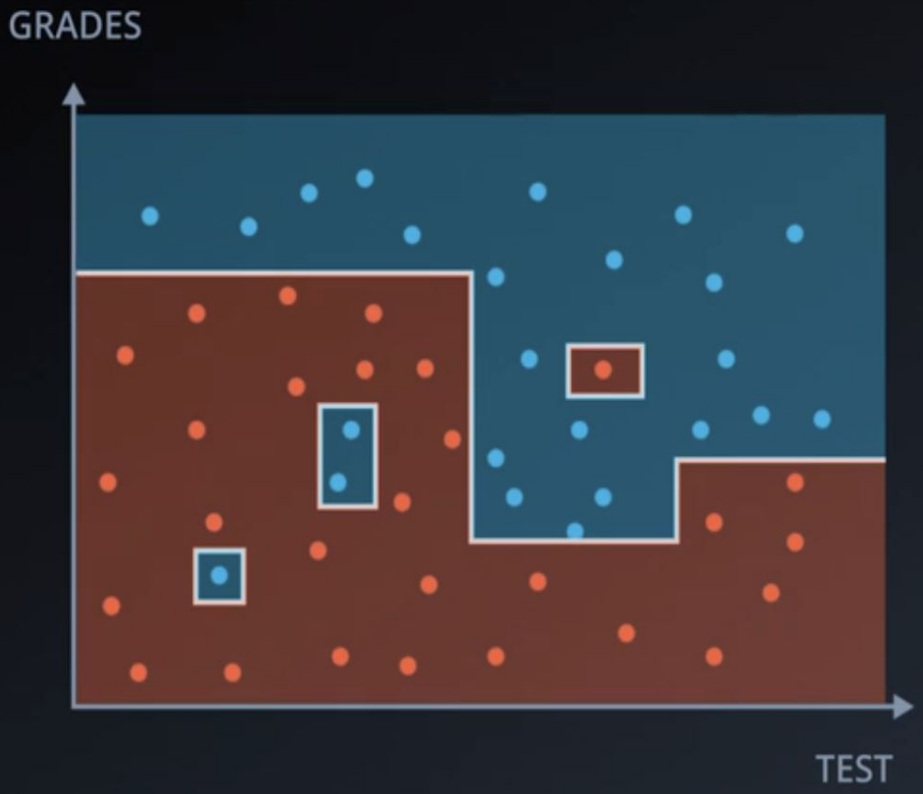


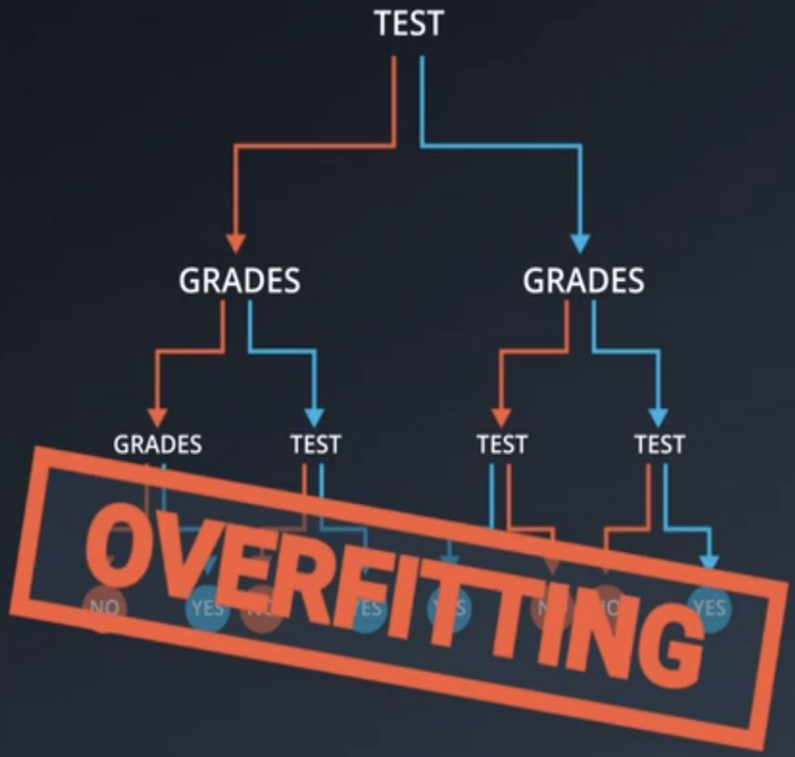
If client is male, between 15 and 25 in the US, on Android, in school, likes tennis, pizza, but does not like long walks on the beach, then they are likely to download Pokemon Go.

OVERFITTING


GRADES
TEST

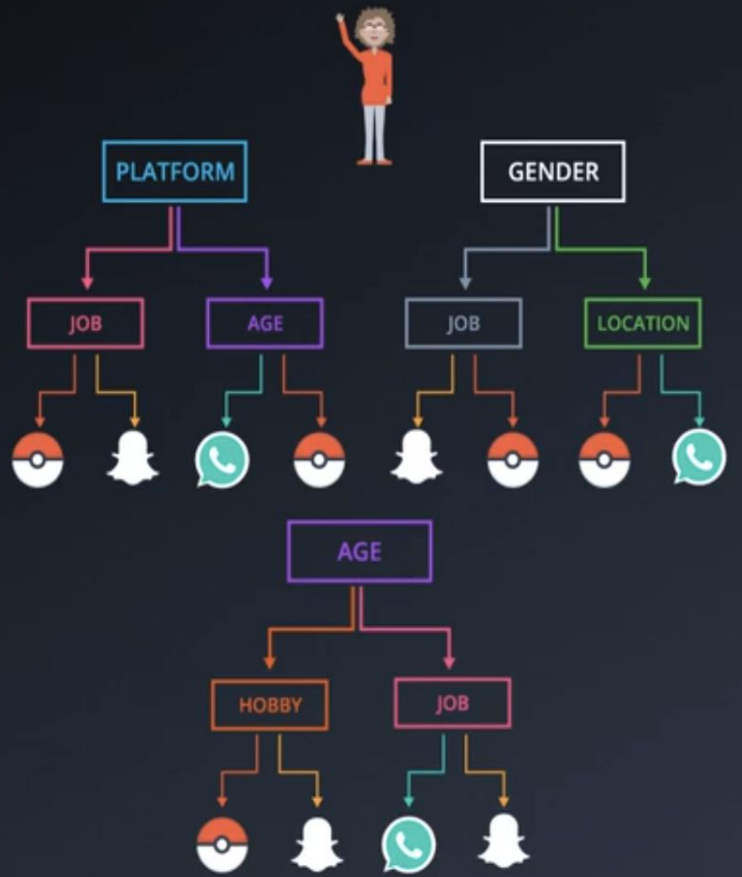

TEST
GRADES
GRADES
GRADES
TEST
TEST
TEST
NO
YES
NO
YES
YES
NO
NO
YES


GRADES
TEST

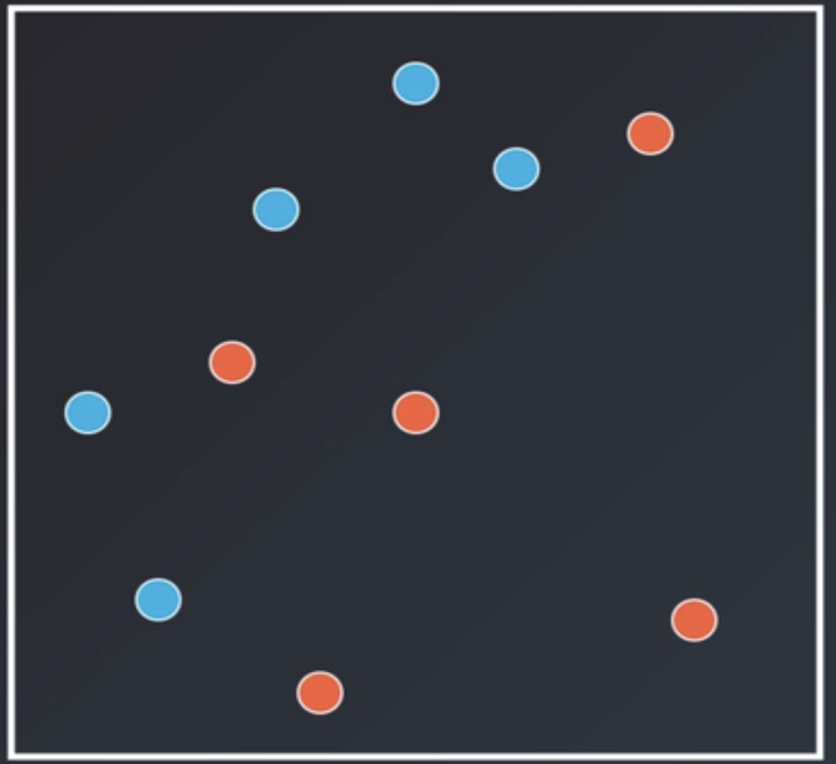

TEST
GRADES
GRADES
GRADES
TEST
TEST
TEST
OVERFITTING
NO
YES
YES
YES
YES

Random Forests

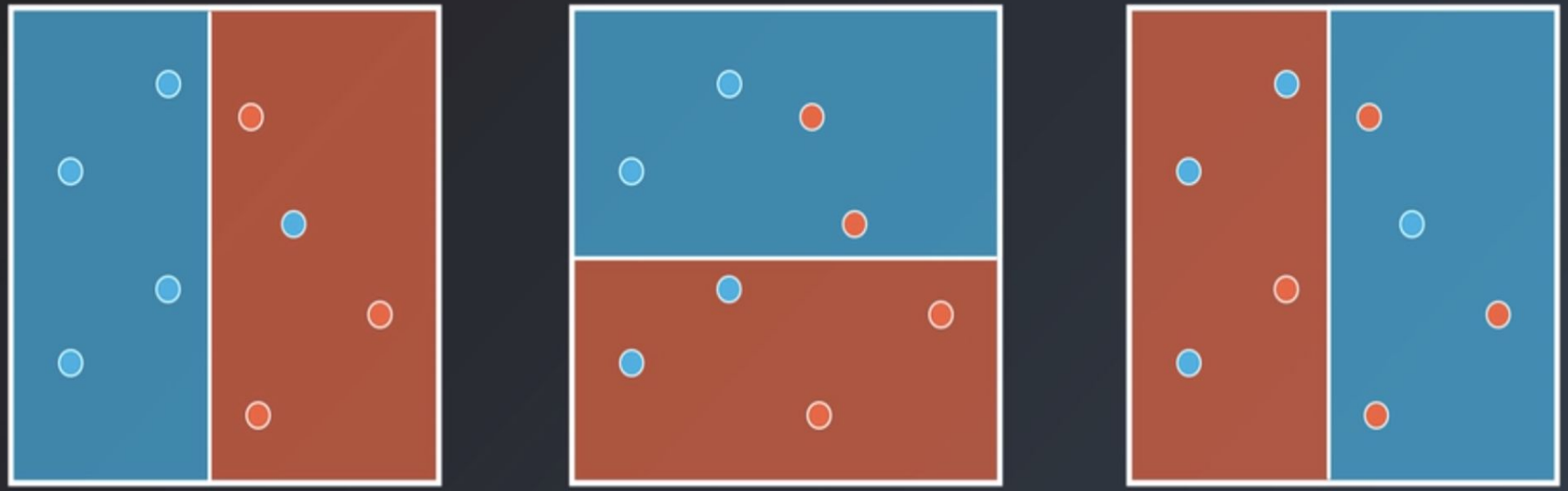
Gender	Age	Location	Platform	Job	Hobby	App
F	15	US	iOS	School	Videogames	
F	25	France	Android	Work	Tennis	
M	32	Chile	iOS	Temp	Tennis	
F	40	China	iOS	Retired	Chess	
M	12	US	Android	School	Tennis	
M	14	Australia	Android	School	Videogames	



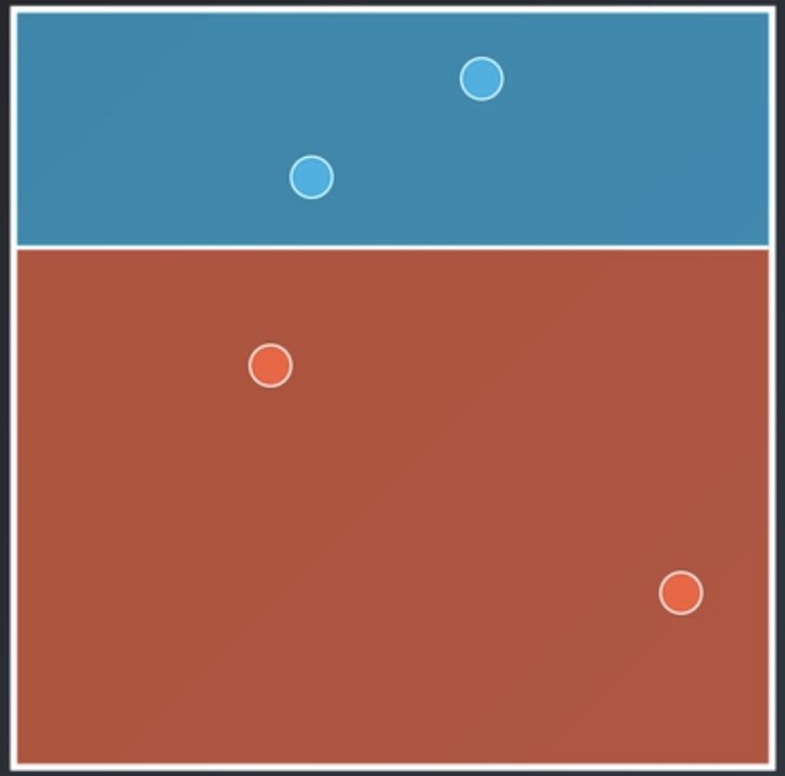
DATA



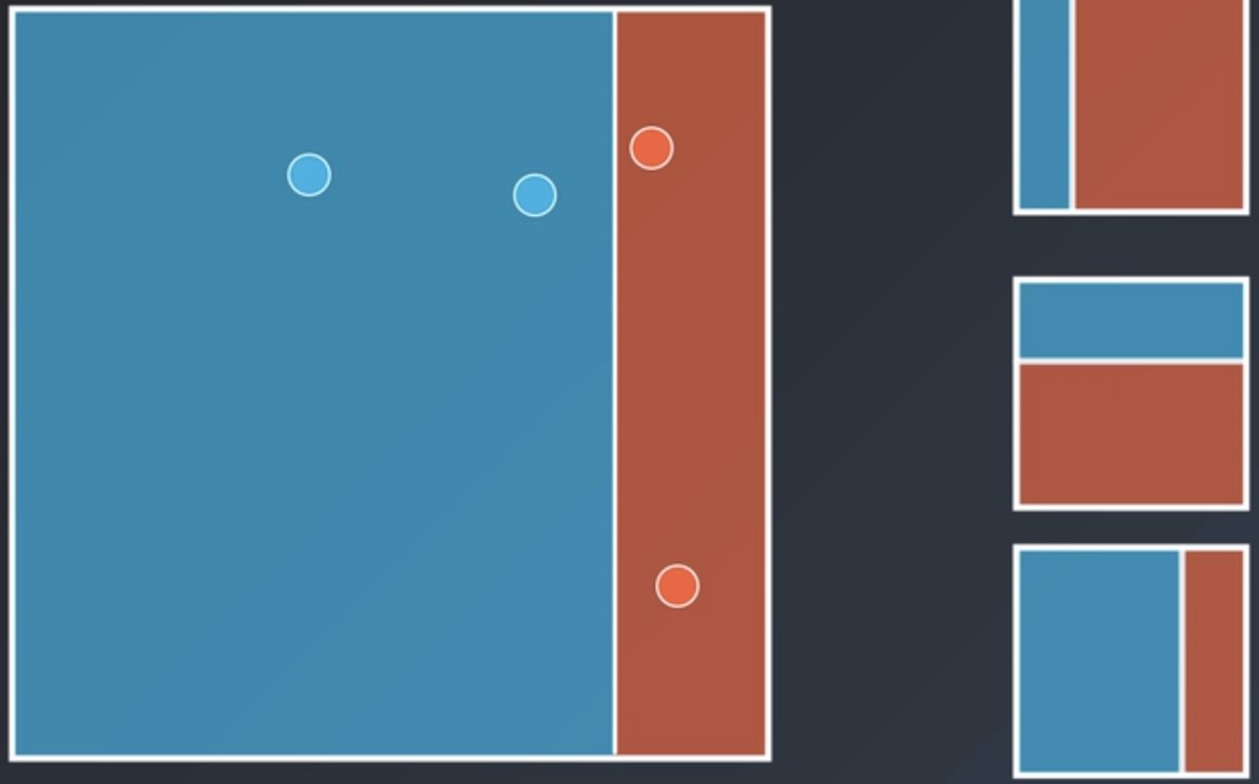
WEAK LEARNERS: ONE-NODE DECISION TREES



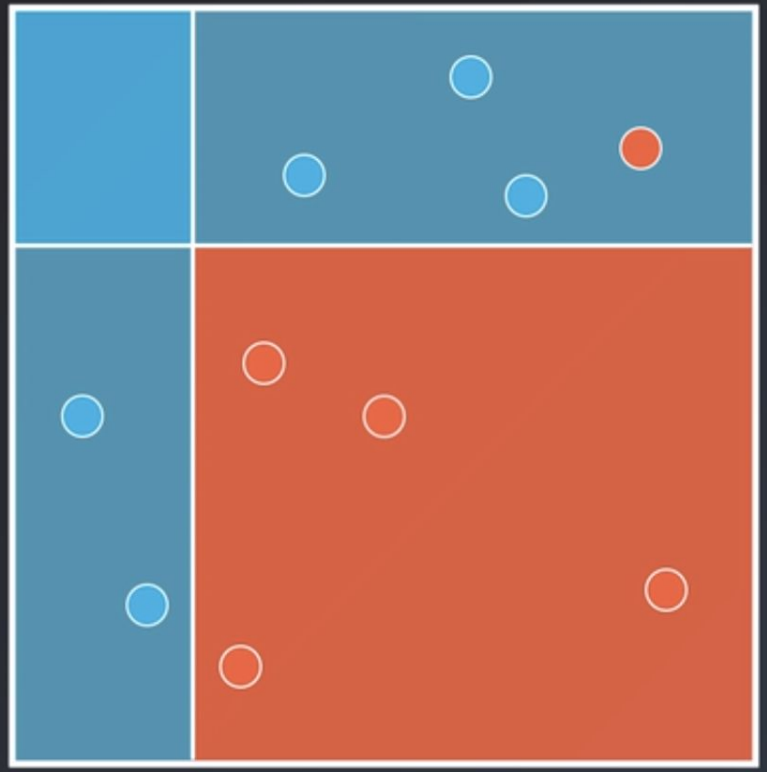
BAGGING



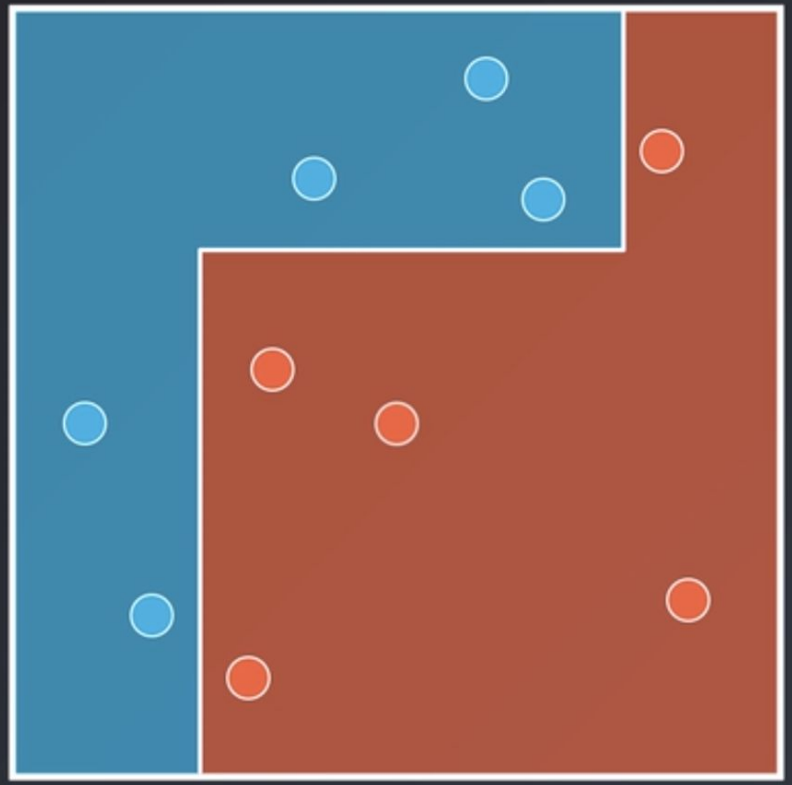
BAGGING



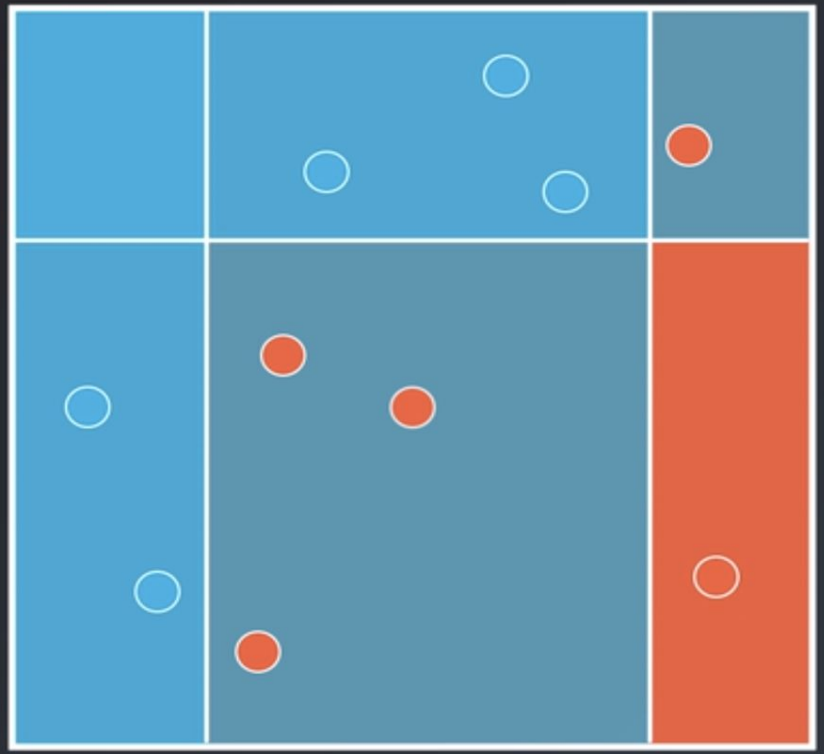
BAGGING



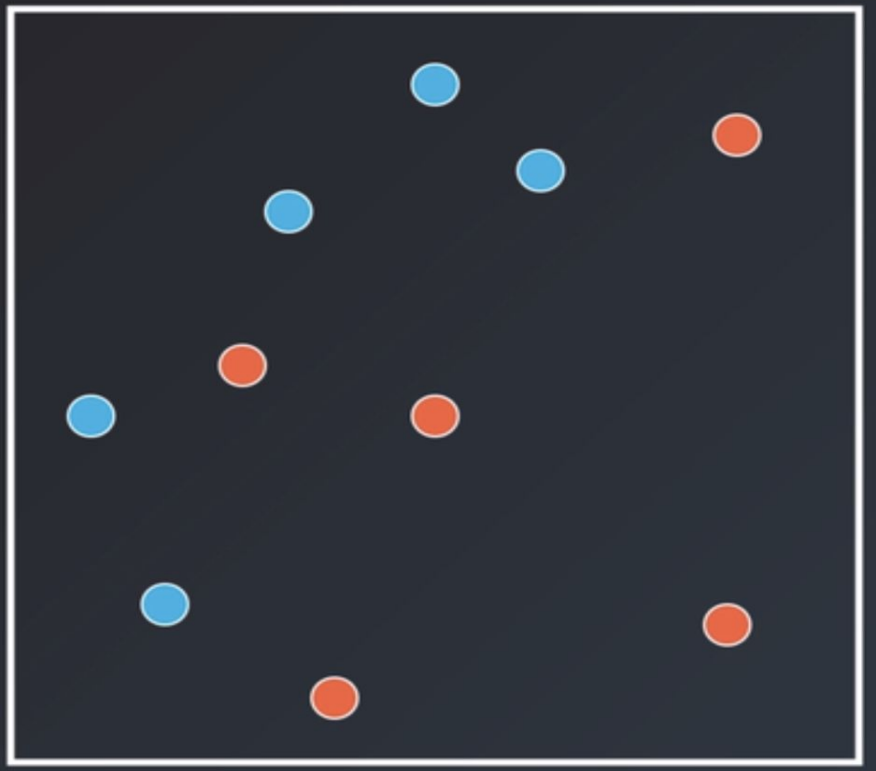
BAGGING



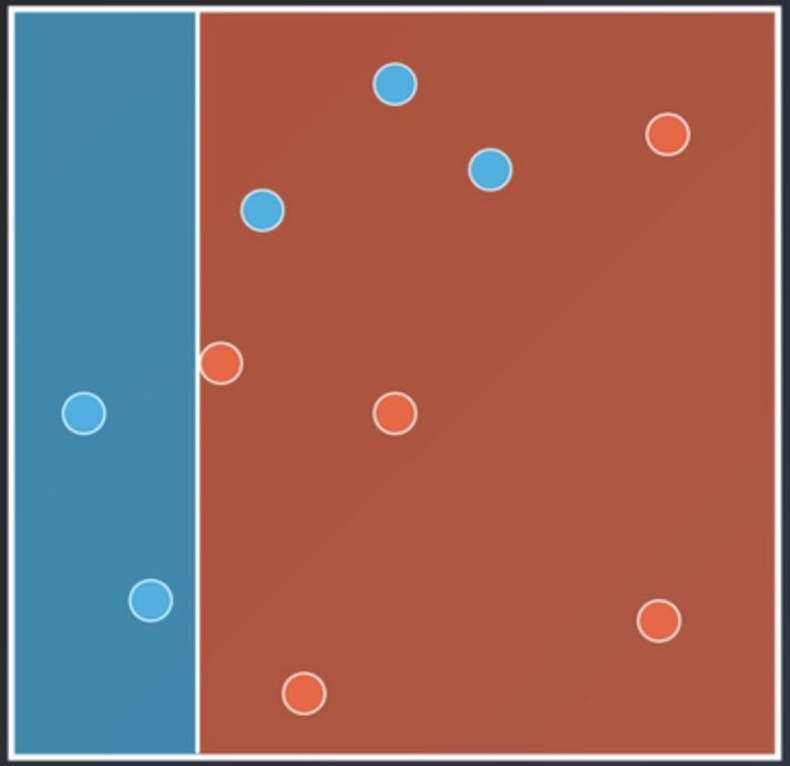
BAGGING



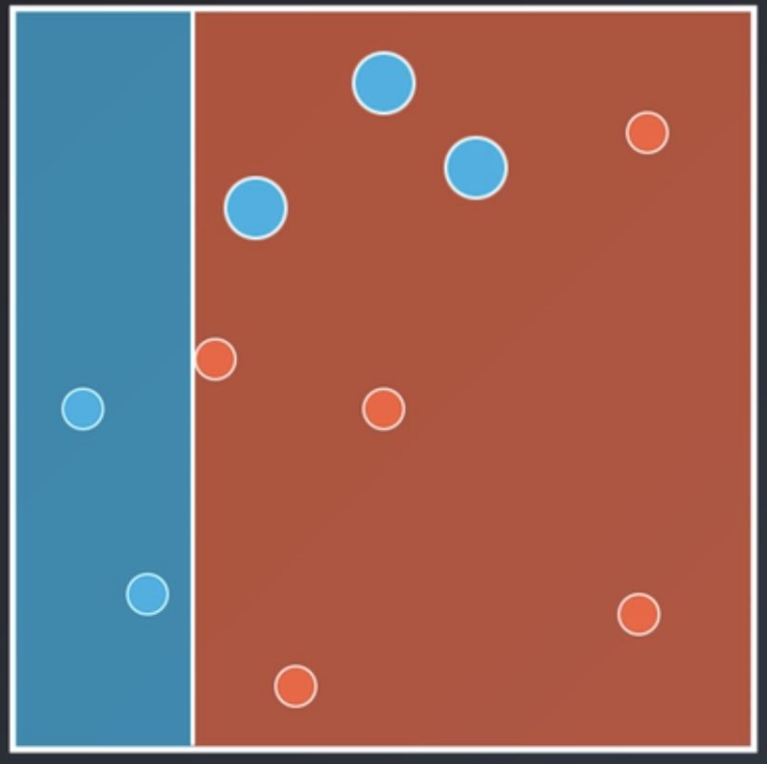
ADABOOST



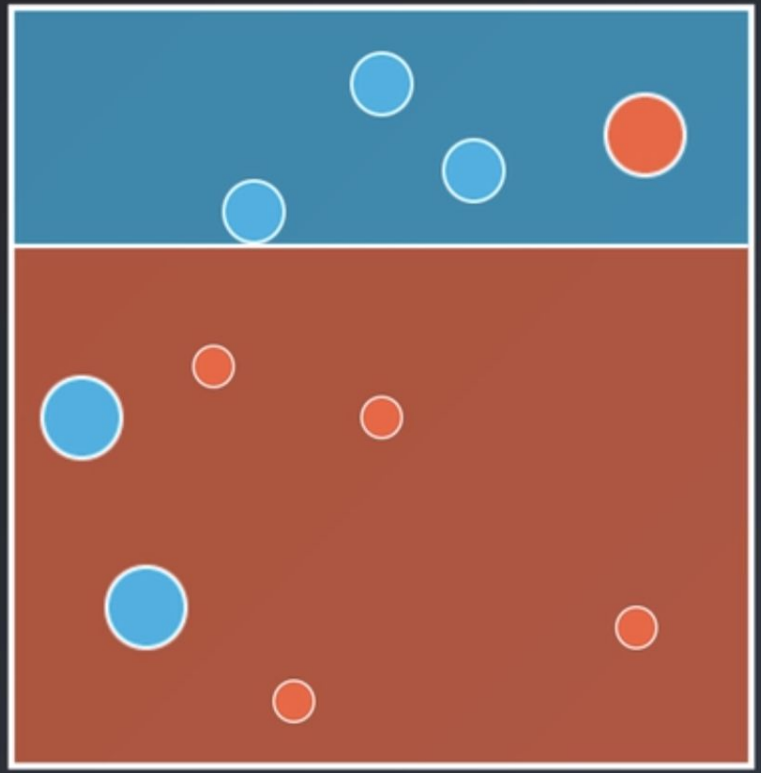
ADABOOST



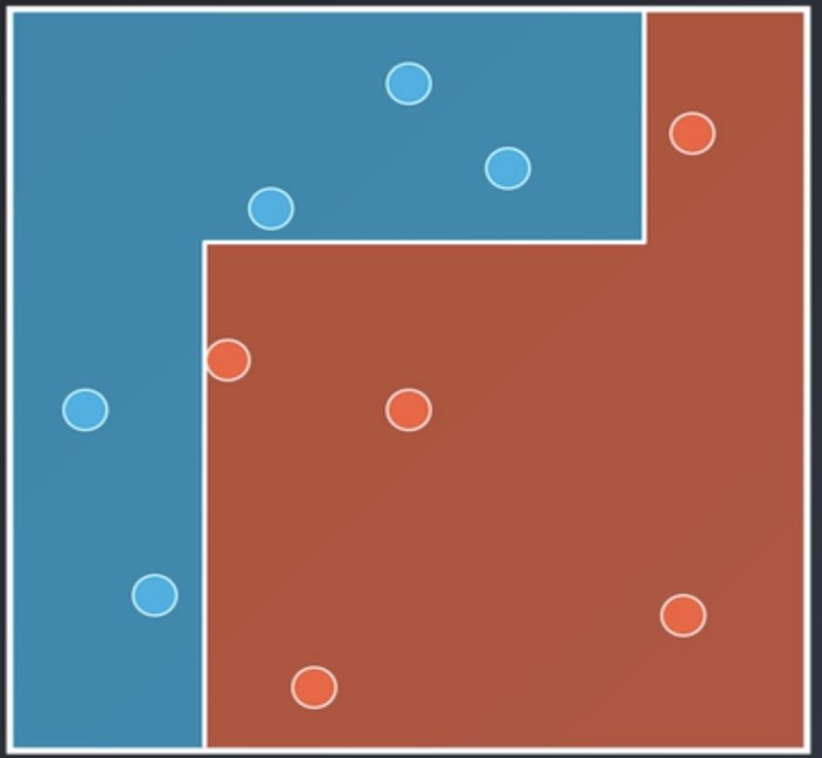
ADABOOST



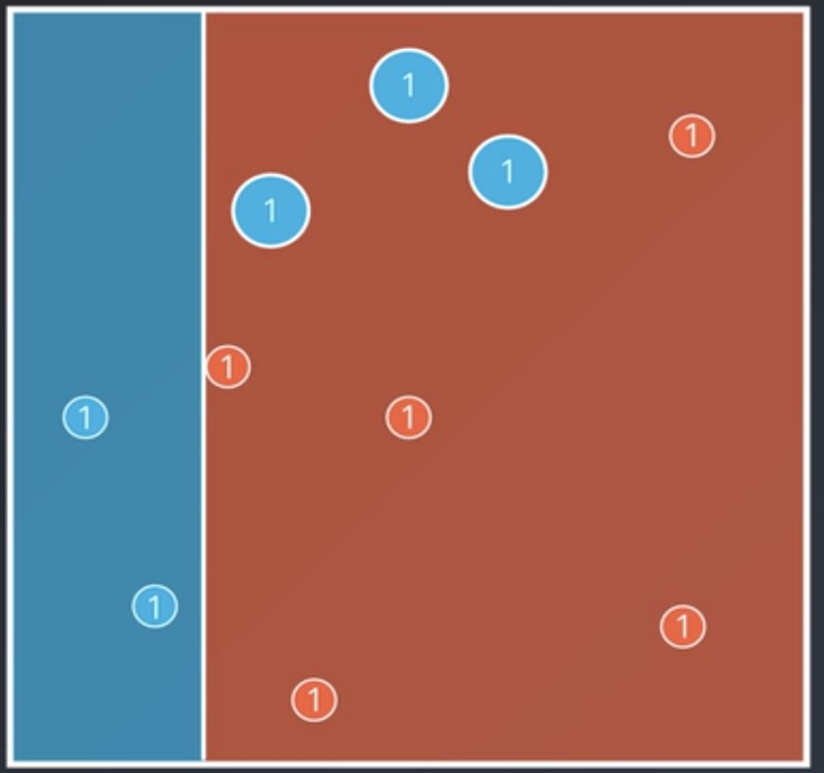
ADABOOST



ADABOOST

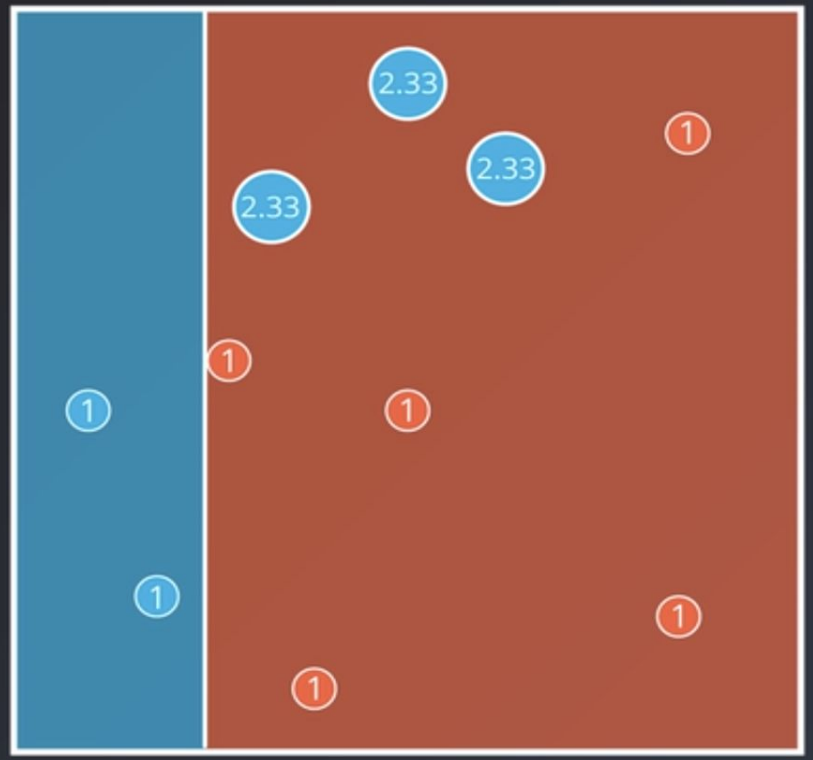


ADABOOST



Correct: 7
Incorrect: 3

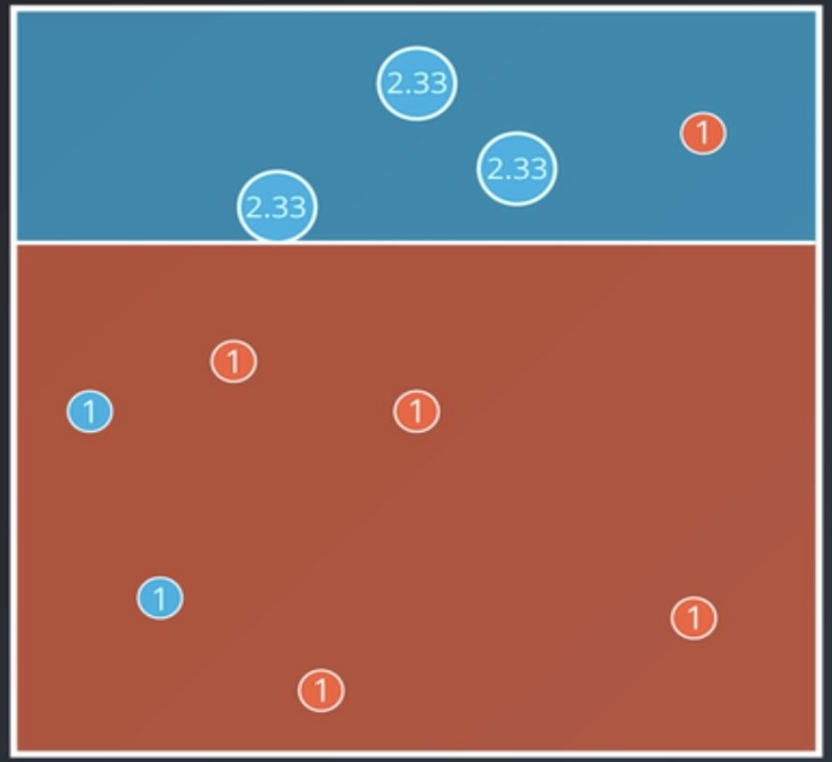
ADABOOST



Correct: 7
Incorrect: 3

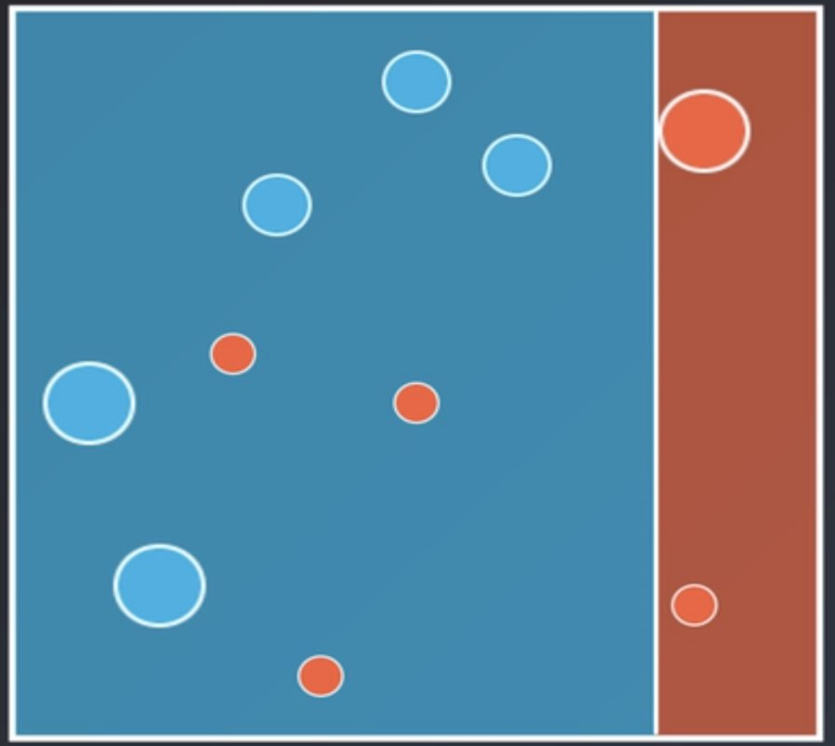
Correct: 7
Incorrect: 7

ADABOOST

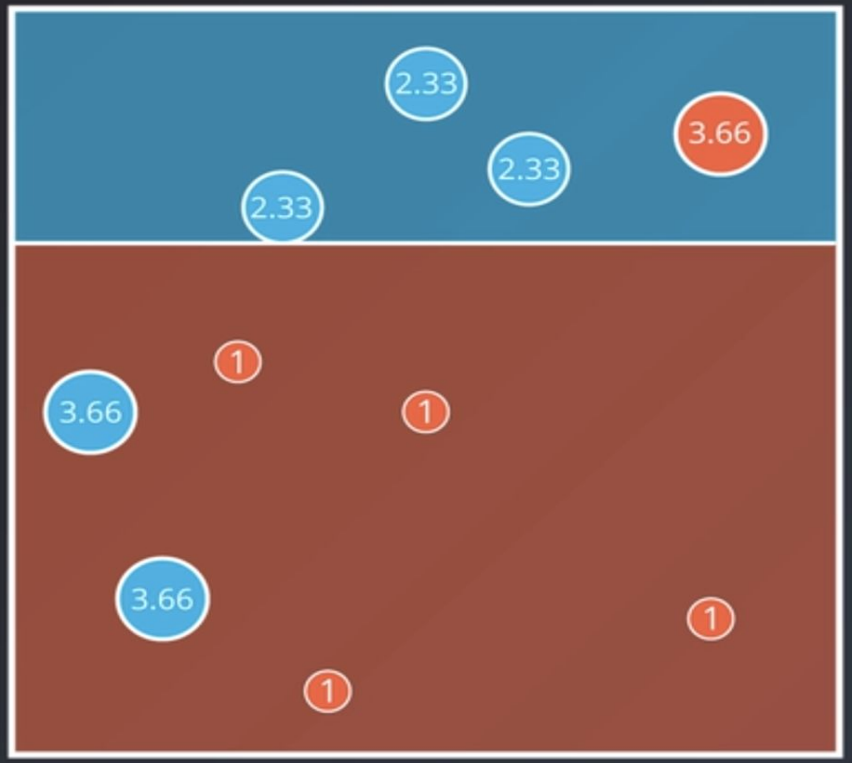


Correct: 11
Incorrect: 3

ADABOOST



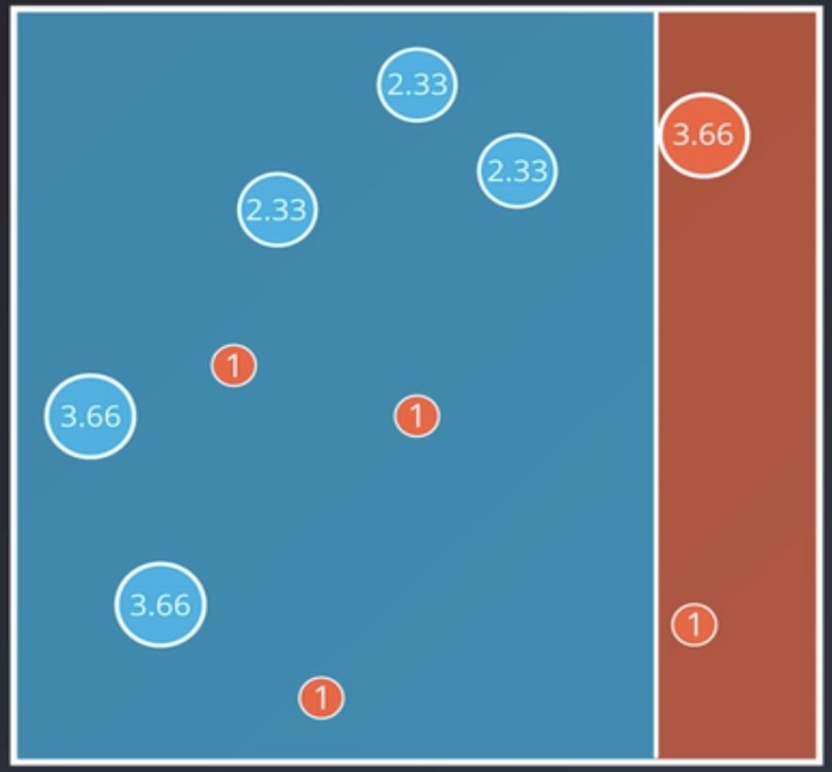
ADABOOST



Correct: 11
Incorrect: 3

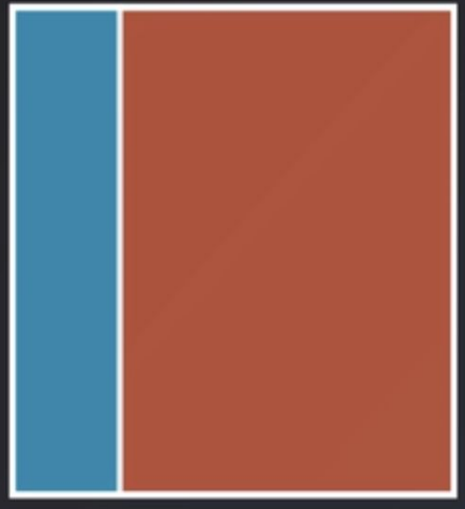
Correct: 11
Incorrect: 11

ADABOOST

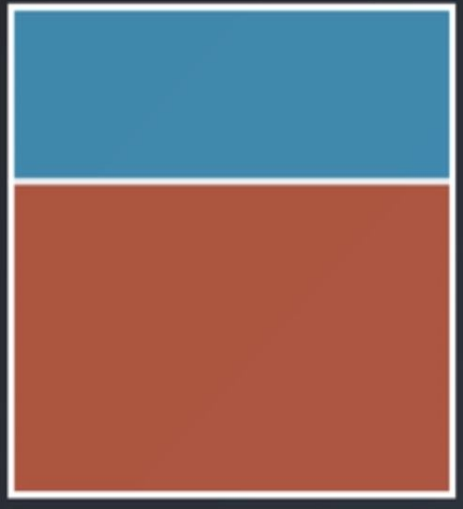


Correct: 19
Incorrect: 3

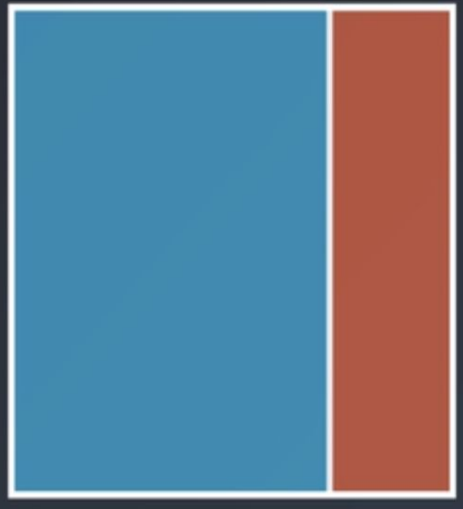
ADABOOST



Model 1



Model 2



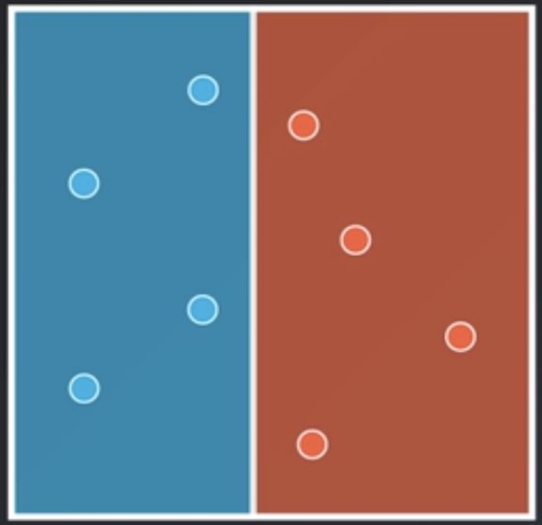
Model 3

ADABOOST

$$weight = \ln\left(\frac{accuracy}{1 - accuracy}\right)$$

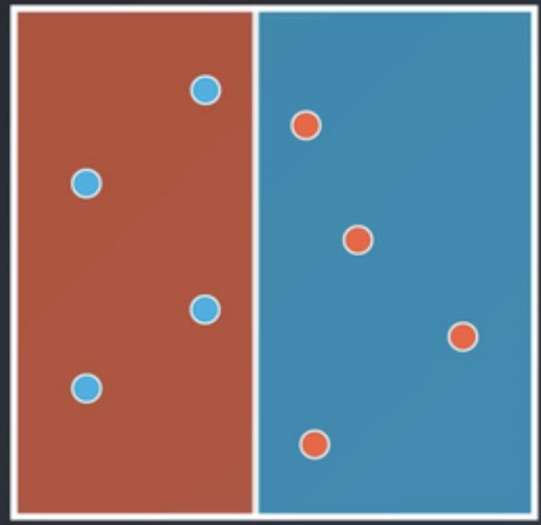
$$weight = \ln\left(\frac{\#correct}{\#incorrect}\right)$$

ADABOOST



$$\ln\left(\frac{8}{0}\right) ?$$

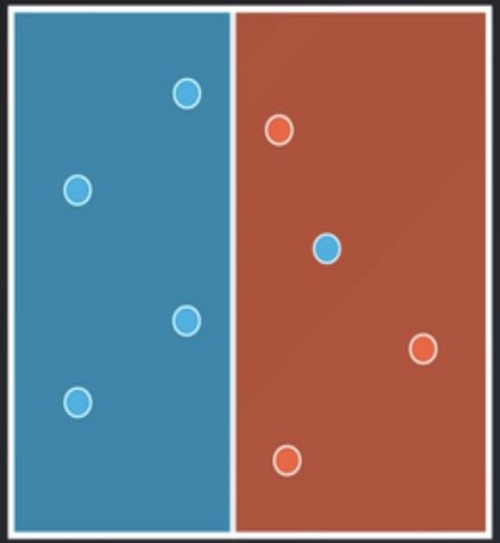
$$\infty$$

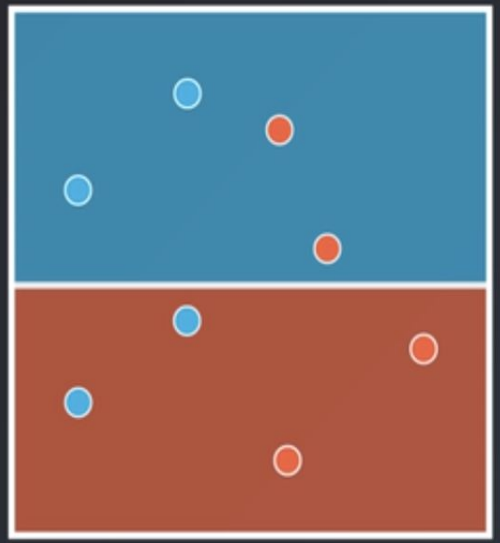


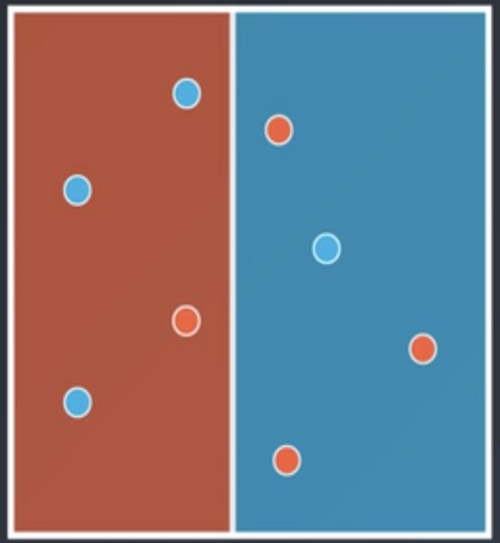
$$\ln\left(\frac{0}{8}\right) ?$$

$$-\infty$$

ADABOOST





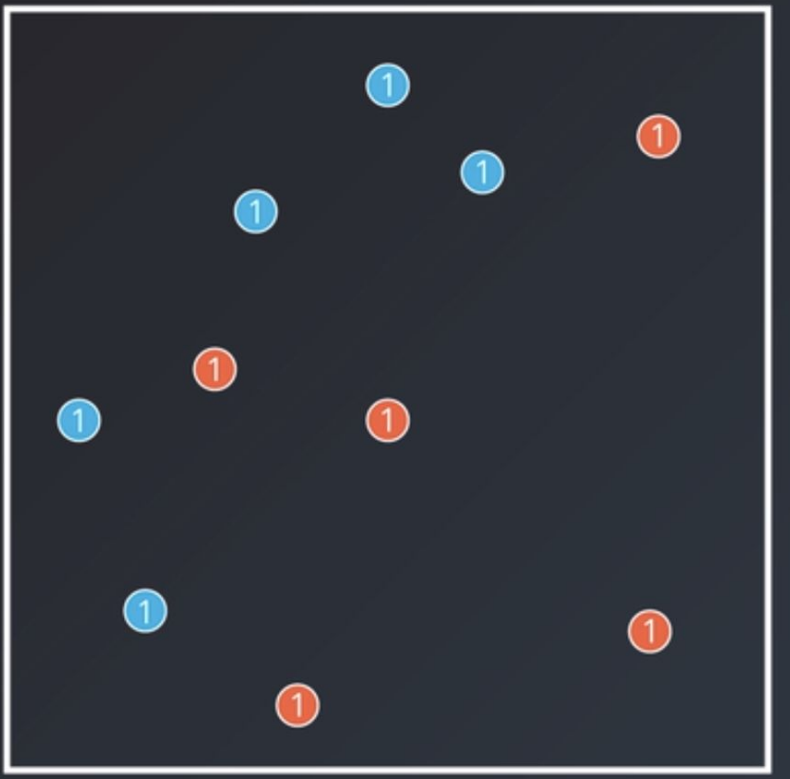


$$\ln\left(\frac{7}{1}\right) = 1.95$$

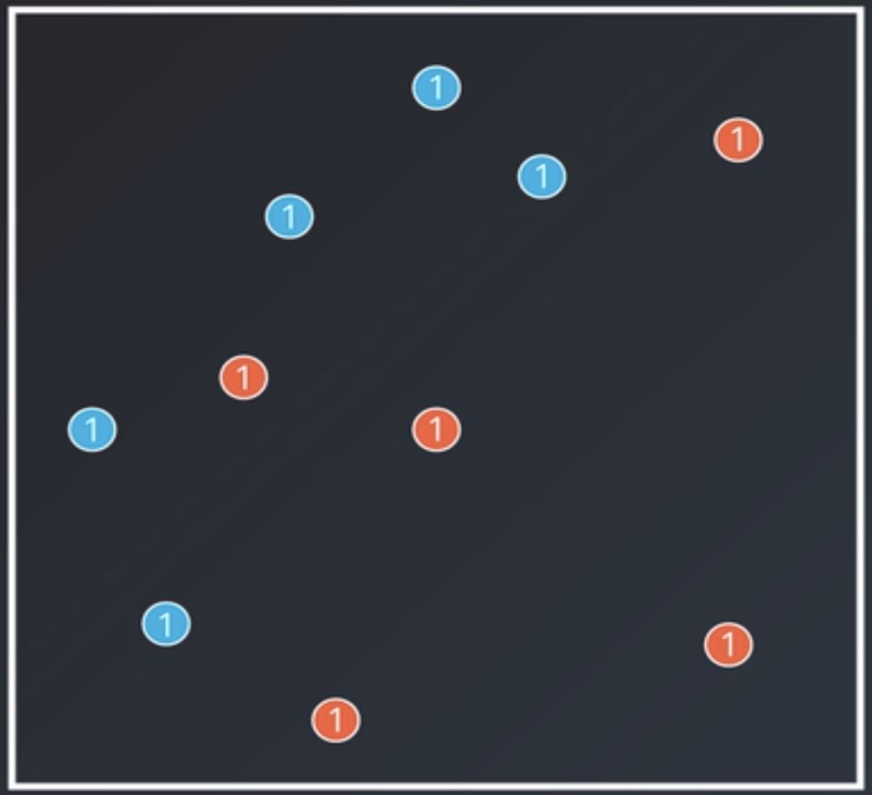
$$\ln\left(\frac{4}{4}\right) = 0$$

$$\ln\left(\frac{2}{6}\right) = -1.099$$

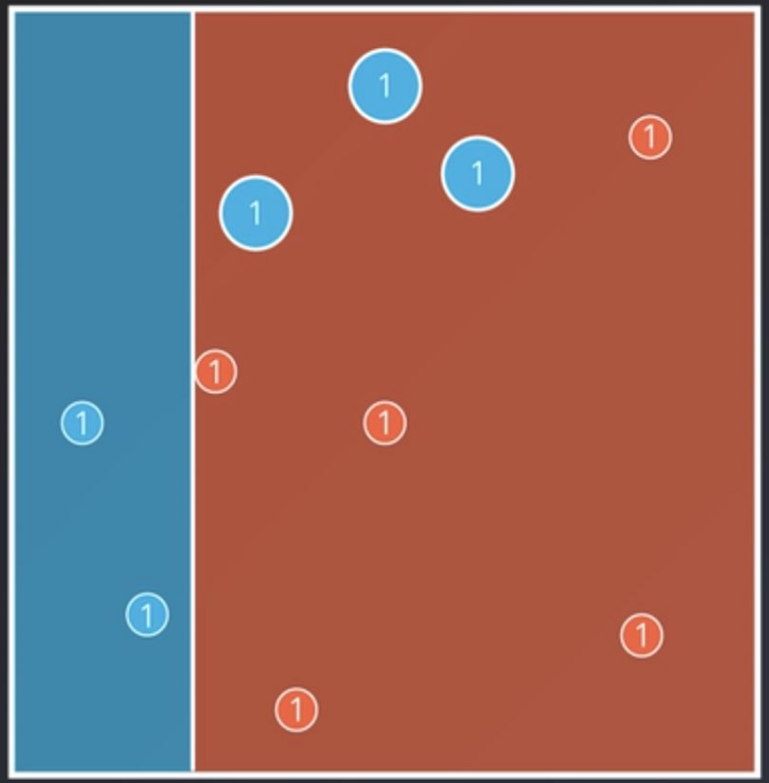
ADABOOST



ADABOOST



ADABOOST

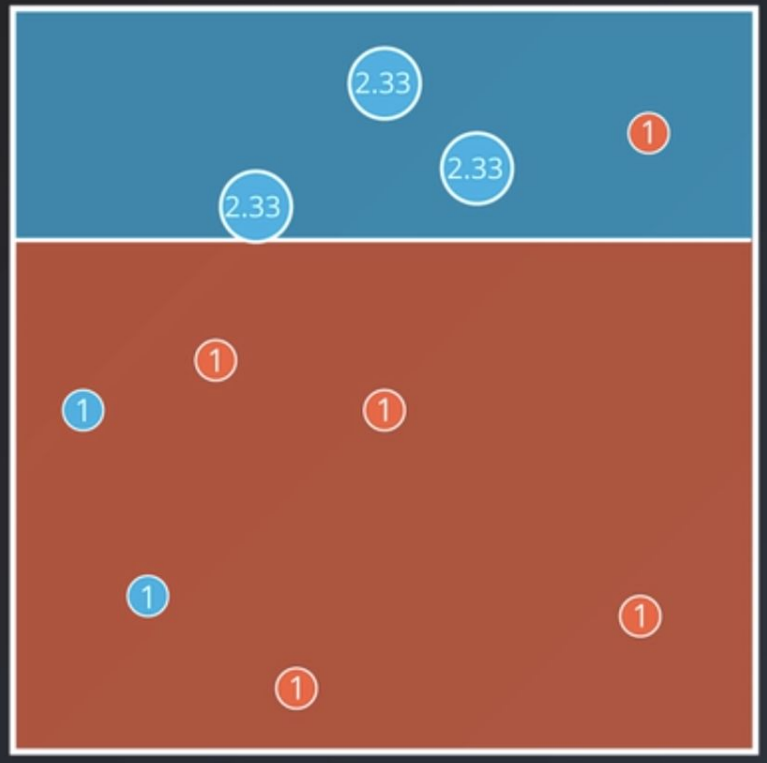


Correct: 7

Incorrect: 3

$$weight = \ln\left(\frac{7}{3}\right) = 0.84$$

ADABOOST



Correct: 11
Incorrect: 3

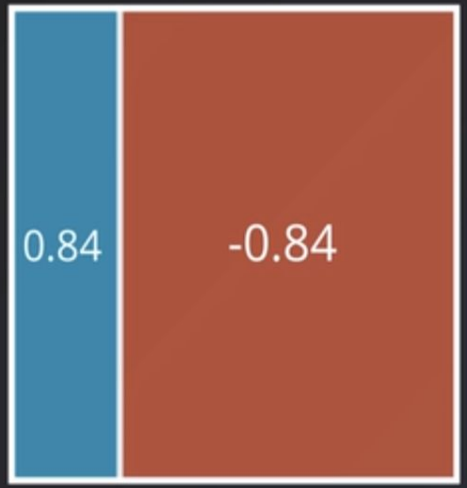
$$weight = \ln\left(\frac{11}{3}\right) = 1.3$$

ADABOOST

Correct: 19
Incorrect: 3

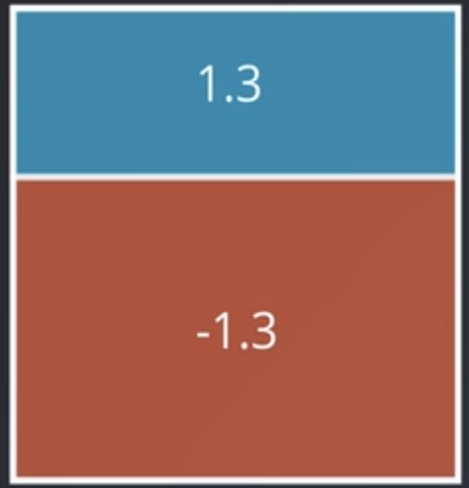
$$weight = \ln\left(\frac{19}{3}\right) = 1.84$$

ADABOOST



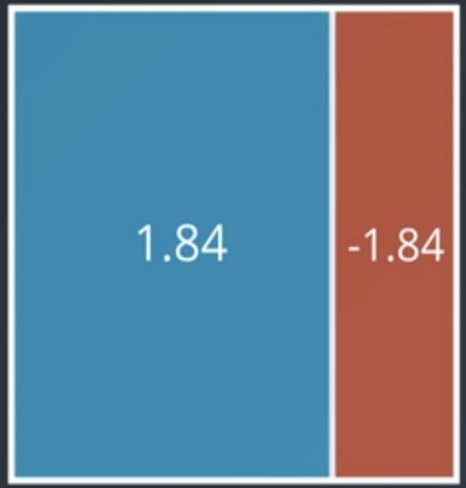
Model 1

Weight = 0.84



Model 2

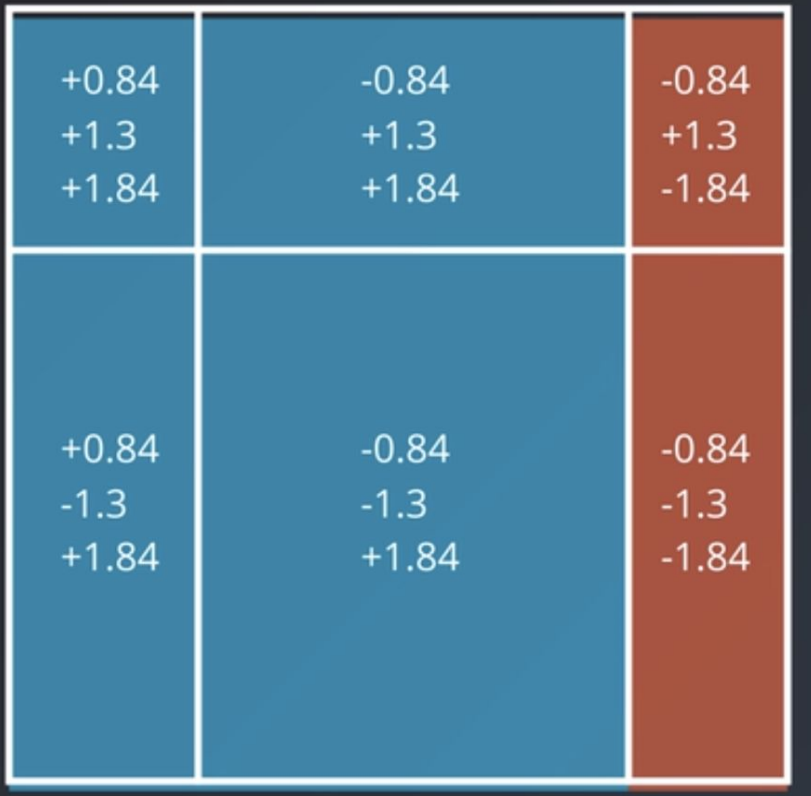
Weight = 1.3



Model 3

Weight = 1.84

ADABOOST

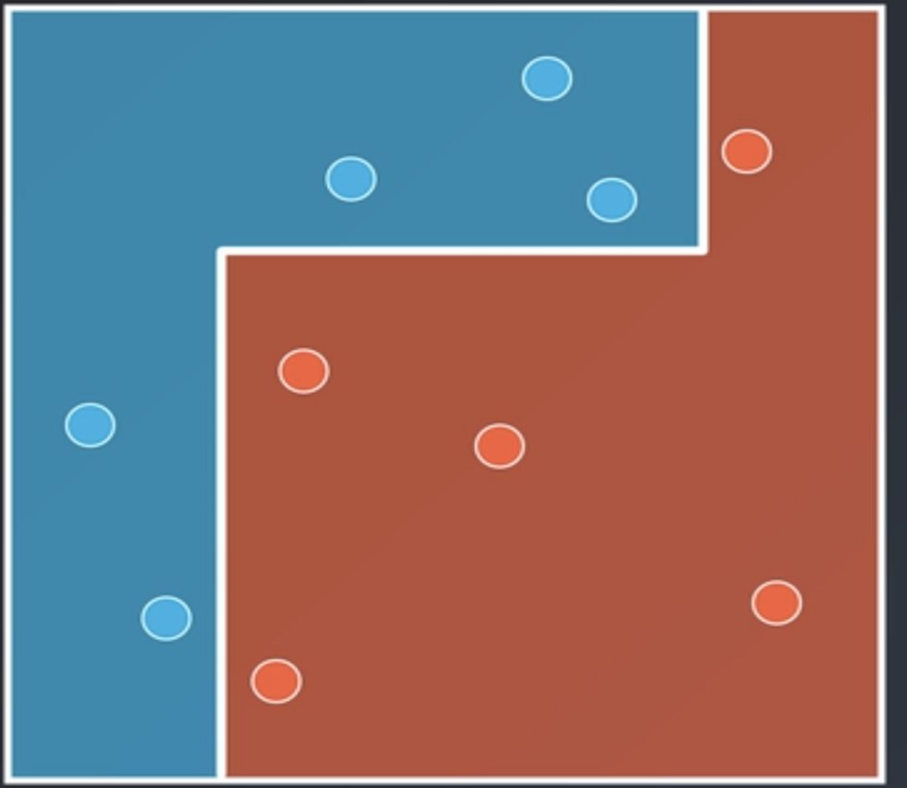


ADABOOST

3.98	2.3	-1.38
1.38	-0.3	-3.98

ADABOOST

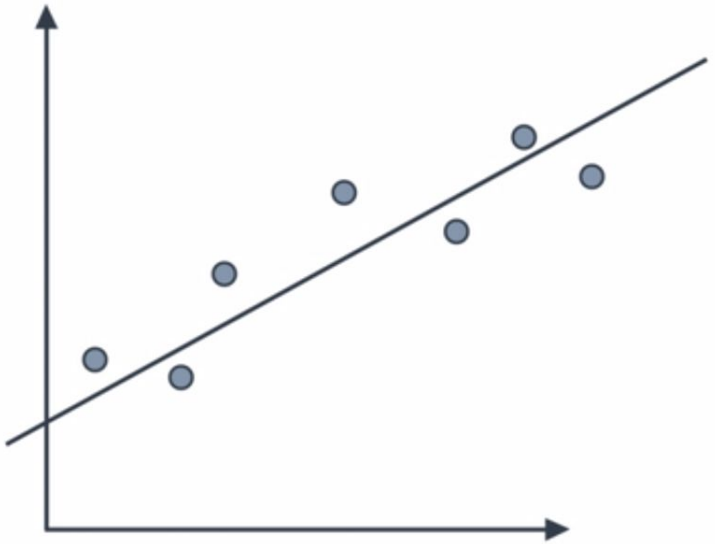
ADABOOST



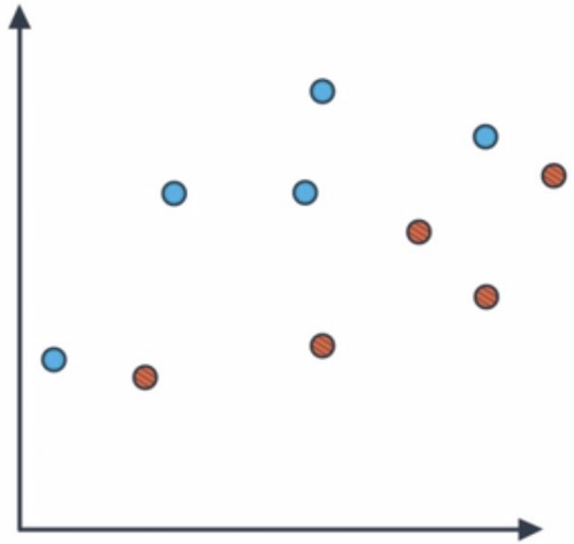
TESTING

HOW WELL IS MY MODEL DOING?

REGRESSION AND CLASSIFICATION



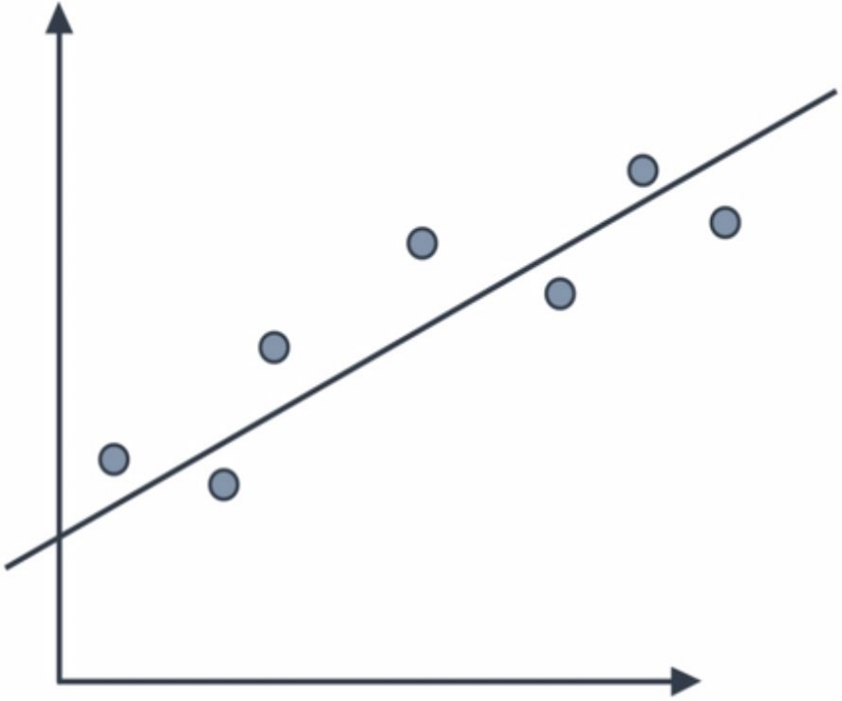
Regression



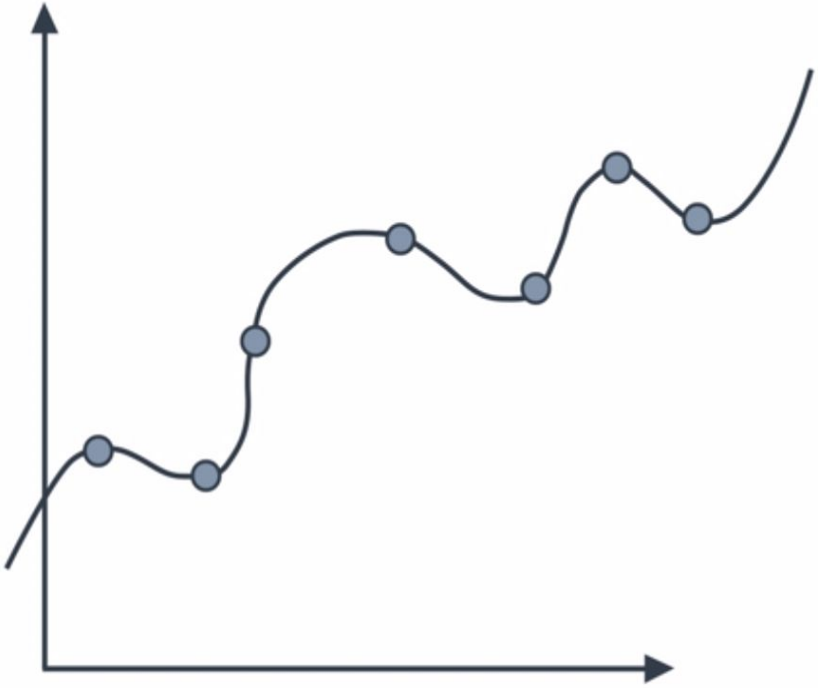
Classification

WHICH MODEL IS BETTER?

Does not fit the data

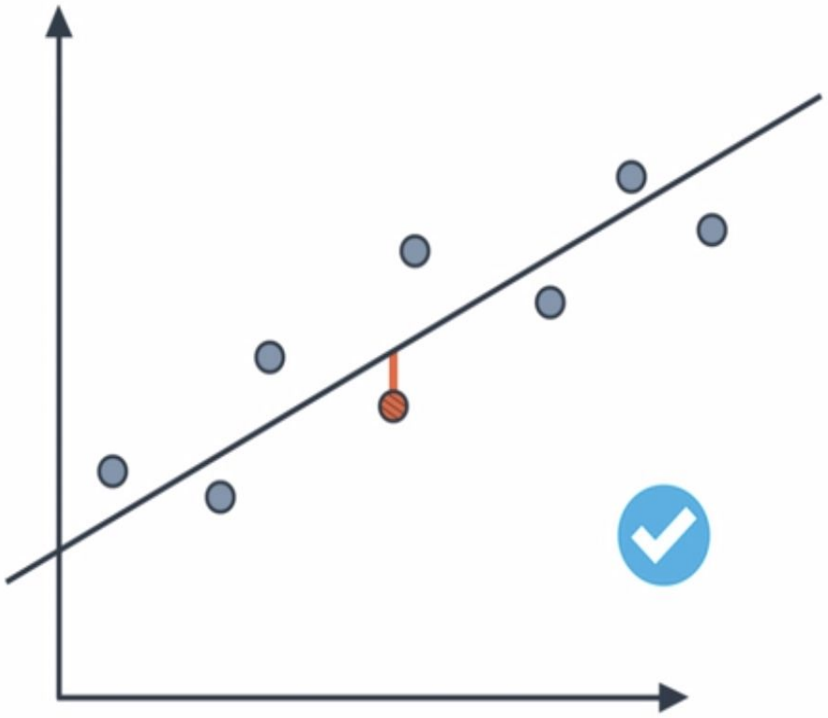


Fits the data

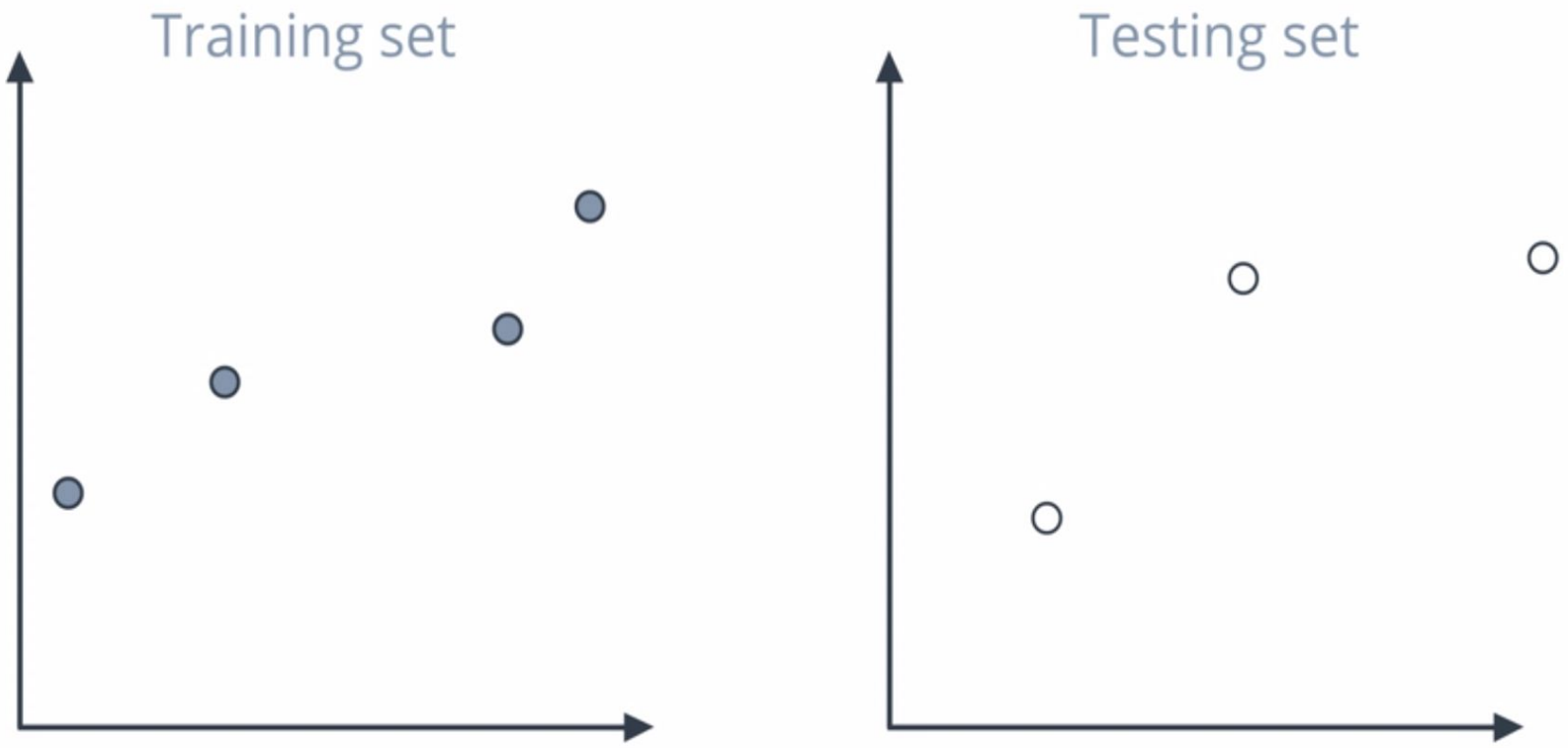


WHICH MODEL IS BETTER?

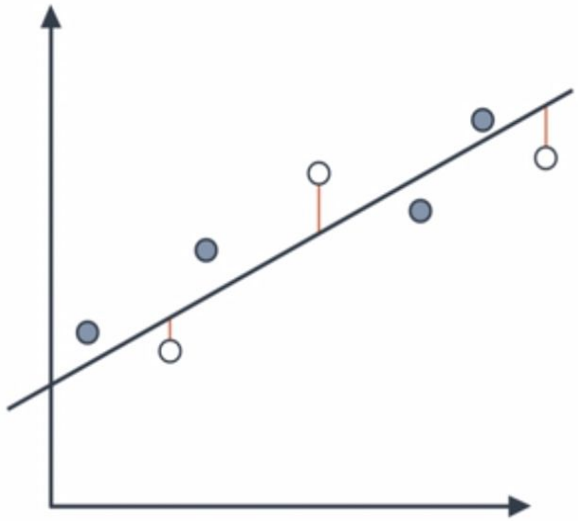
This model generalizes better



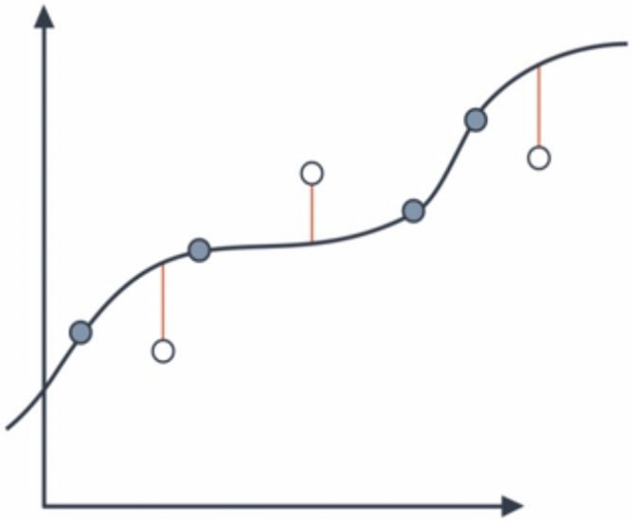
TESTING



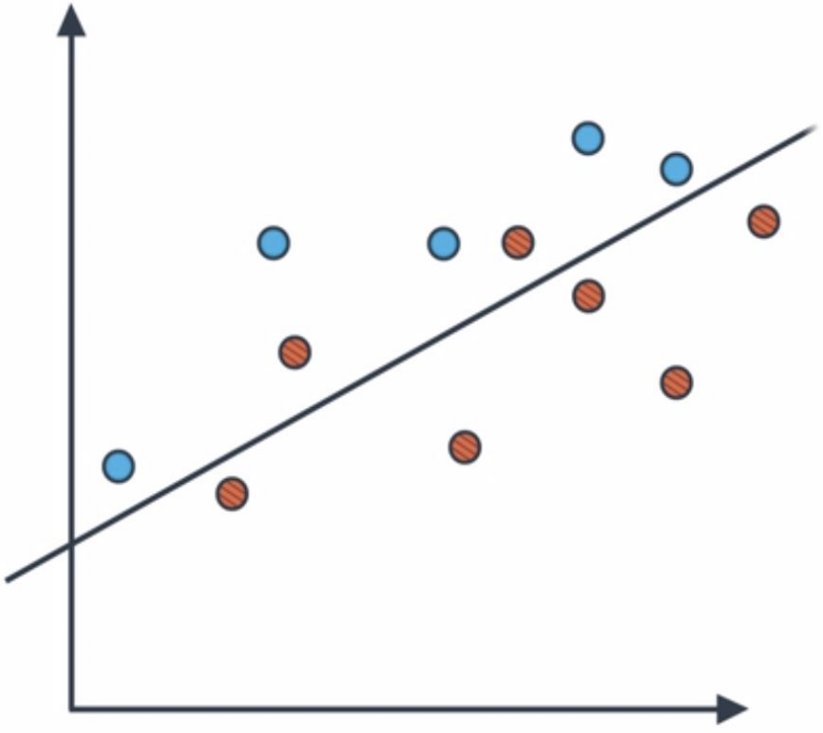
TESTING



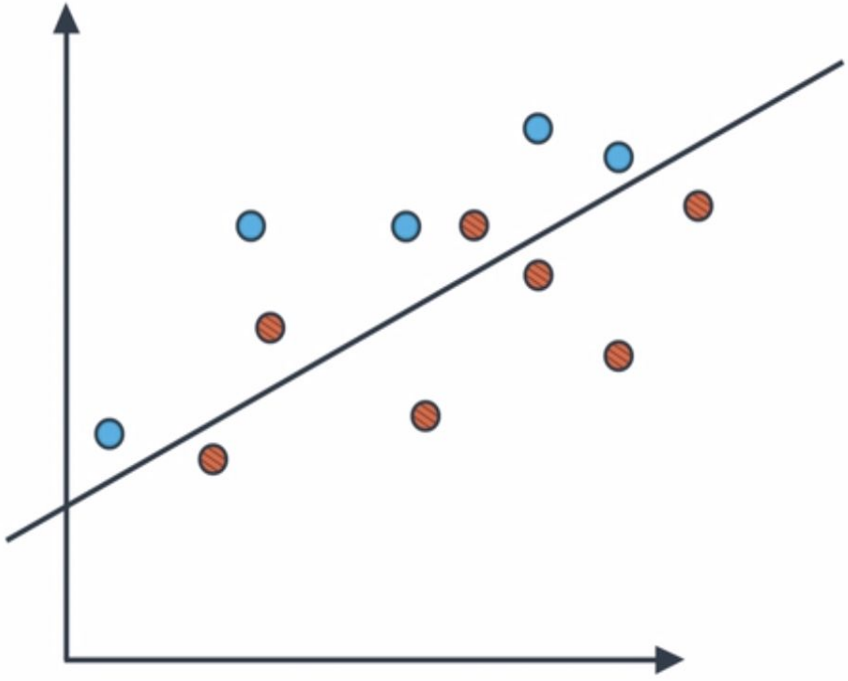
Errors are smaller

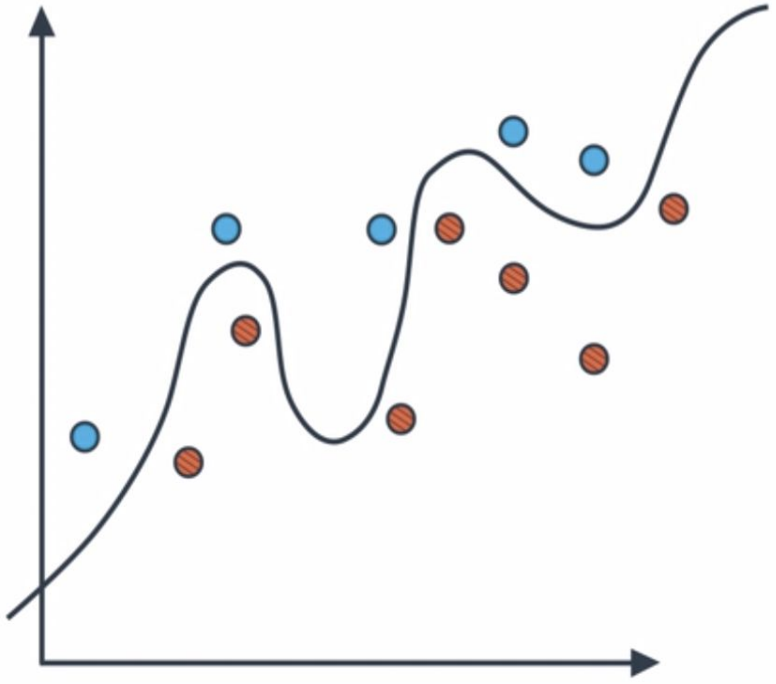


WHICH MODEL IS BETTER?

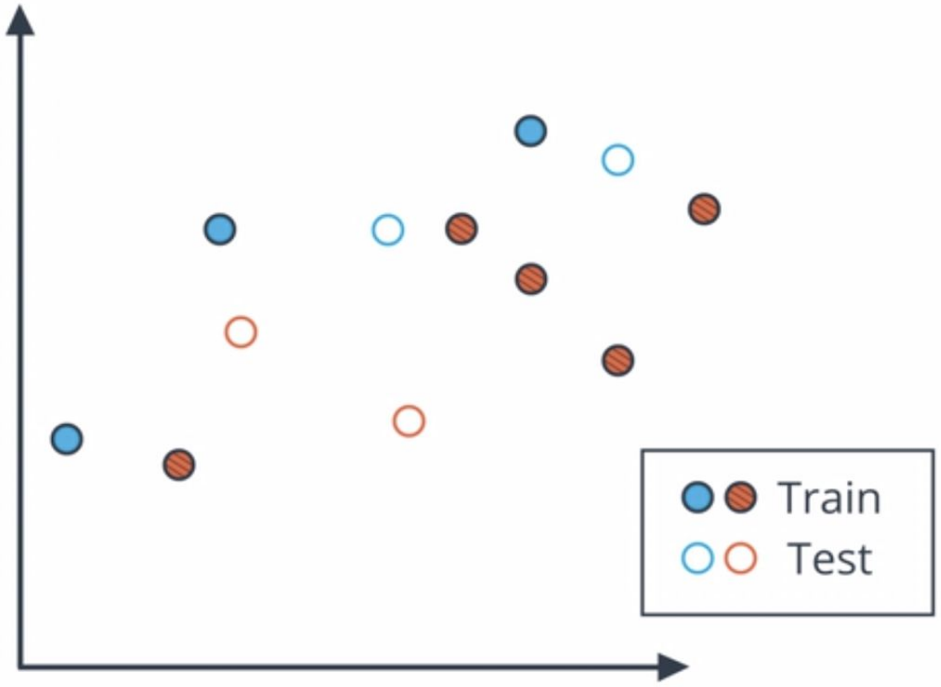


WHICH MODEL IS BETTER?

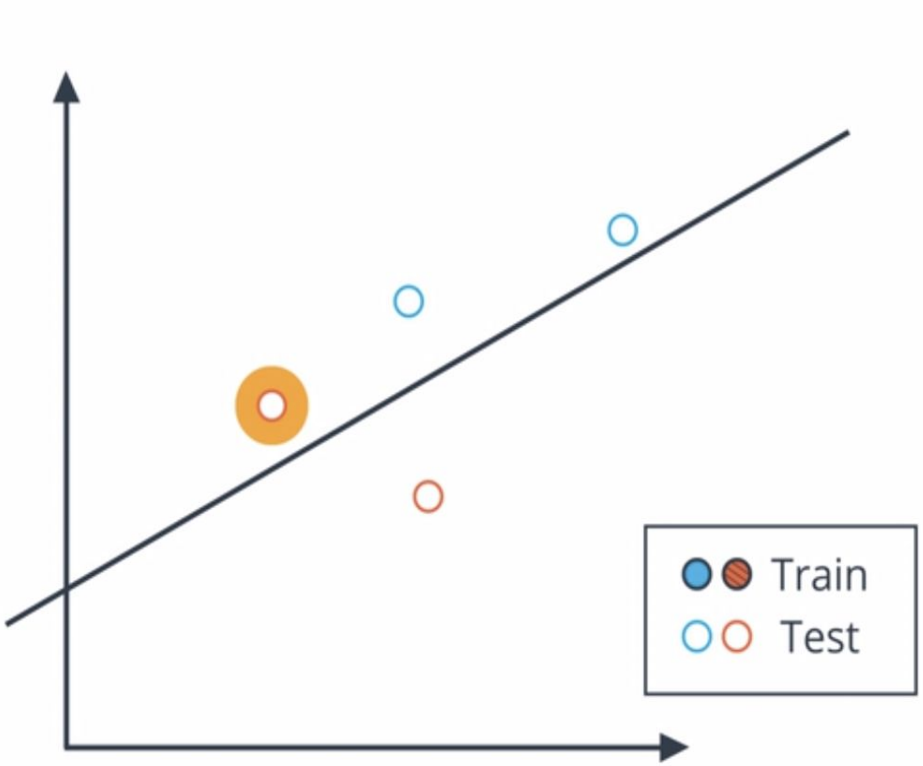


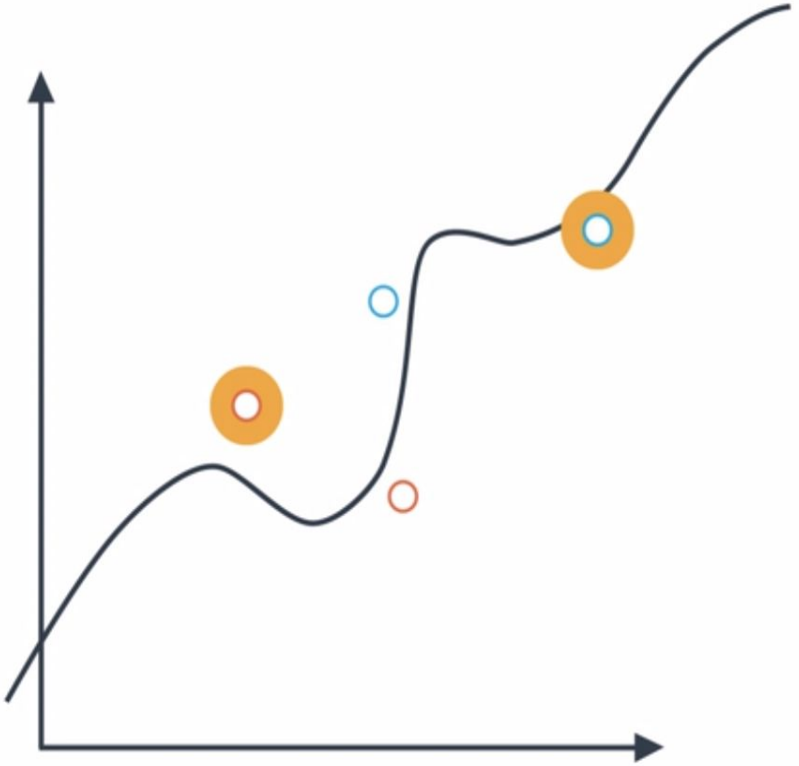


WHY TESTING?



WHY TESTING?



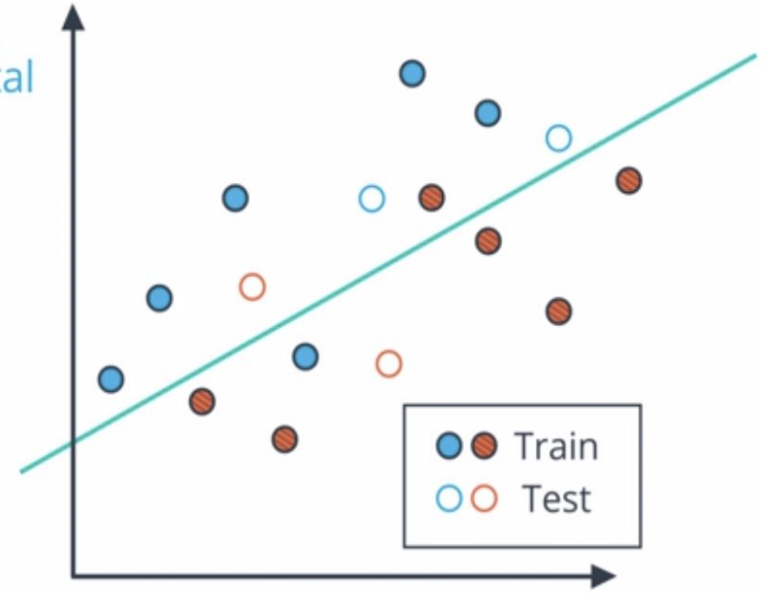


TESTING IN SKLEARN

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test =
train_test_split(X,
                 y,
                 test_size = 0.25)
```

4 test
16 total



THOU SHALT NEVER
USE YOUR TESTING DATA
FOR TRAINING

SPAM CLASSIFIER MODEL

NOT SPAM

SPAM

EVALUATION MATRIX

How well is my model doing?

MEDICAL MODEL

HEALTHY

SICK

CONFUSION MATRIX

10, 000 PATIENTS

DIAGNOSIS

PATIENTS

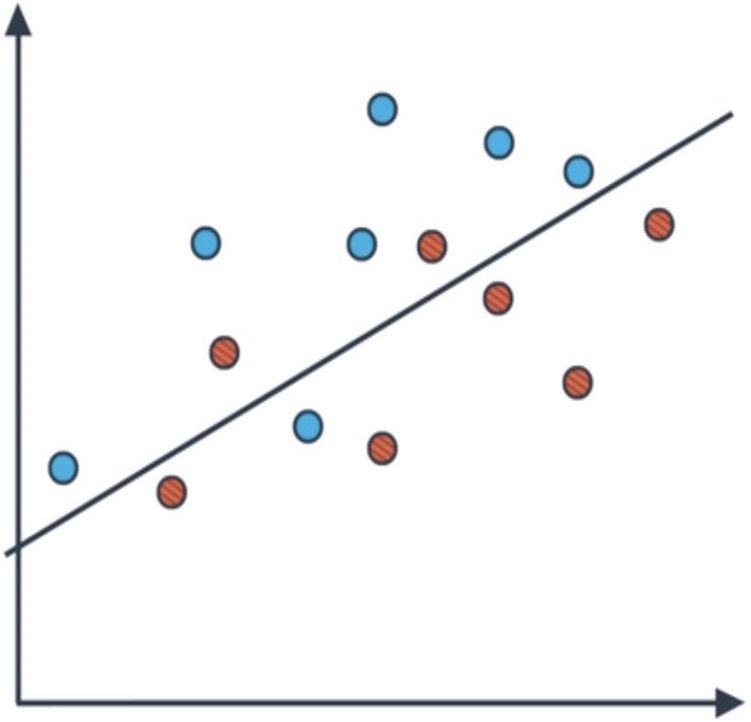
	Diagnosed Sick	Diagnosed Healthy
Sick	1000 True Positives	200 False Negatives
Healthy	800 False Positives	8000 True Negatives

CONFUSION MATRIX

1000 EMAILS

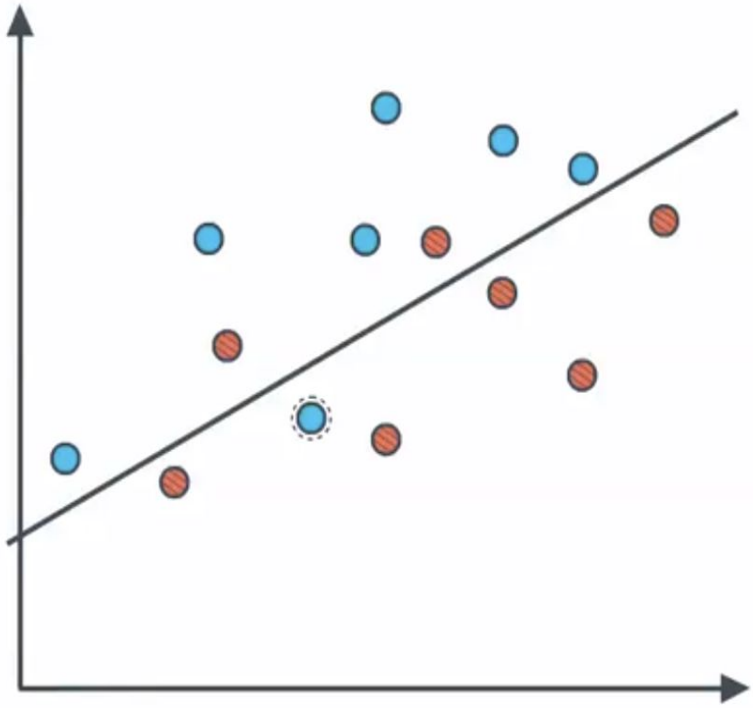
EMAIL \ SPAM	Spam Folder	Inbox
Spam	100 True Positives	170 False Negatives
Not Spam	30 False Positives	700 True Negatives

CONFUSION MATRIX



	Guessed Positive	Guessed Negative
Positive	True Positives	False Negatives
Negative	False Positives	True Negatives

CONFUSION MATRIX



	Guessed Positive	Guessed Negative
Positive	6 True Positives	1 False Negatives
Negative	2 False Positives	5 True Negatives

ACCURACY

One of the ways to measure how good a model is.

ACCURACY

Out of all the patients, how many did we classify correctly?

ACCURACY

Out of all the patients, how many did we classify correctly?

	Diagnosed sick	Diagnosed healthy
Sick	1,000	200
Healthy	800	8,000

$$\text{Accuracy} = \frac{1{,}000 + 8{,}000}{10{,}000} = 90\%$$

ACCURACY

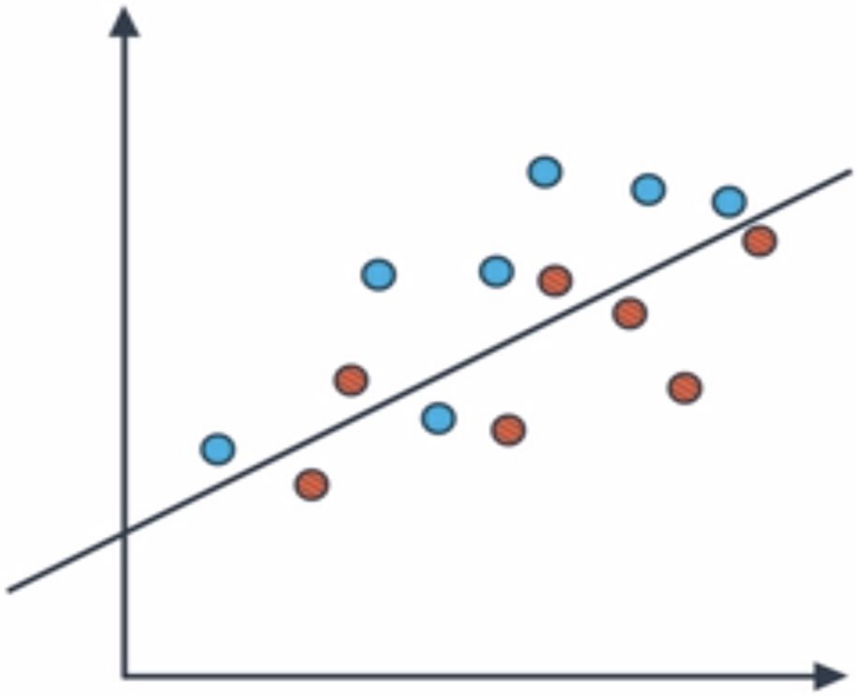
Out of all the emails, how many did we classify correctly?

	Spam folder	Inbox
Spam	100	170
Not spam	30	700

$$\text{Accuracy} = \frac{100 + 700}{1{,}000} = 80\%$$

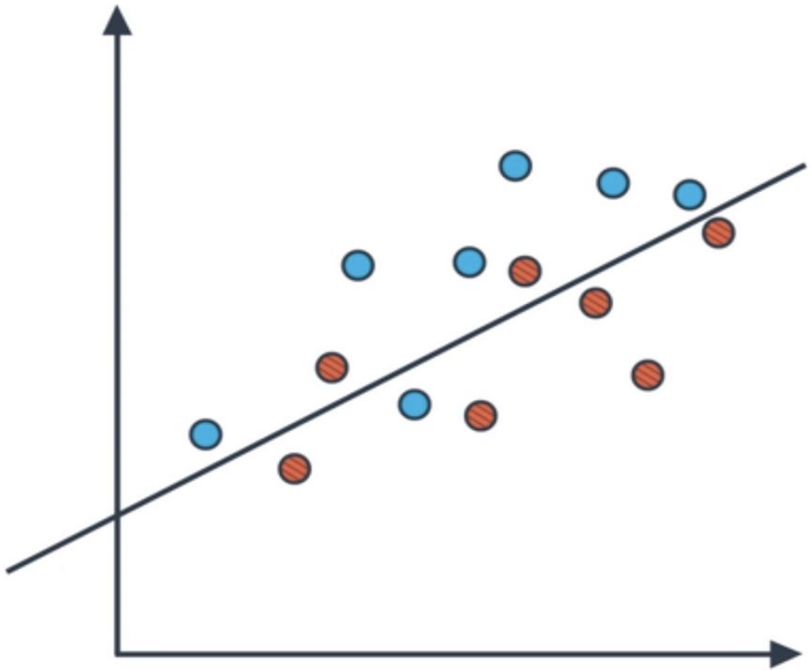
ACCURACY

Out of all the data, how many points did we classify correctly?



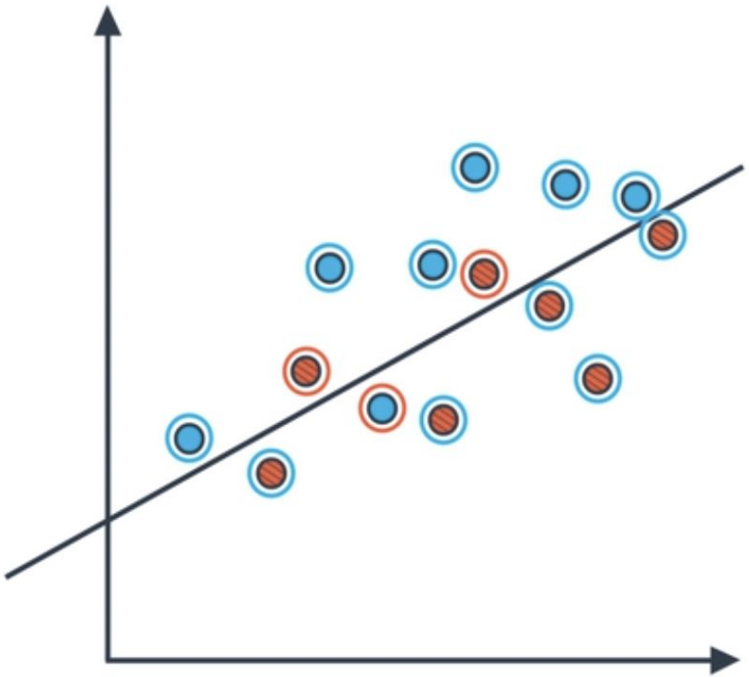
ACCURACY

Out of all the data, how many points did we classify correctly?



ACCURACY

Out of all the data, how many points did we classify correctly?



$$\text{Accuracy} = \frac{\text{Correctly classified points}}{\text{All points}}$$

$$= \frac{11}{11 + 3}$$

$$= \frac{11}{14}$$

$$= 78.57\%$$

◦ CREDIT CARD FRAUD

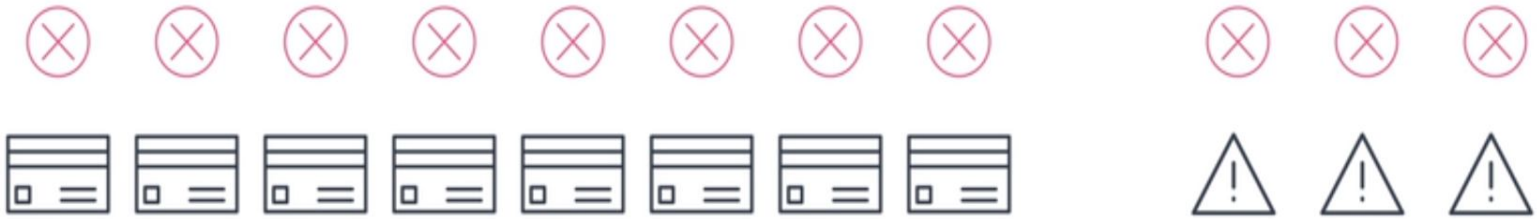
284,335 472

MODEL: ALL TRANSACTIONS ARE GOOD.

$$\text{ACCURACY} = \frac{284{,}335}{284{,}887} = 99.83\%$$

PROBLEM: I'M NOT CATCHING ANY OF THE BAD ONES.

◦ CREDIT CARD FRAUD



284,335 472

MODEL: ALL TRANSACTIONS ARE FRAUDULENT.

GREAT! NOW I'M CATCHING ALL OF THE FRAUDULENT TRANSACTIONS!

PROBLEM: I'M ACCIDENTALLY CATCHING ALL OF THE GOOD ONES!

◦ SOLUTION: FALSE POSITIVES AND NEGATIVES

Medical Model

FALSE POSITIVES OK

FALSE NEGATIVES NOT OK

OK IF NOT ALL ARE SICK
FIND ALL THE SICK PEOPLE

Spam Detector

FALSE POSITIVES NOT OK

FALSE NEGATIVES OK

DON'T NECESSARILY NEED
TO FIND ALL THE SPAM

◦ SOLUTION: FALSE POSITIVES AND NEGATIVES

Medical Model

FALSE POSITIVES OK

FALSE NEGATIVES NOT OK

OK IF NOT ALL ARE SICK
FIND ALL THE SICK PEOPLE

HIGH RECALL

Spam Detector

FALSE POSITIVES NOT OK

FALSE NEGATIVES OK

DON'T NECESSARILY NEED
TO FIND ALL THE SPAM
BETTER BE SPAM

HIGH PRECISION

◦ SOLUTION: FALSE POSITIVES AND NEGATIVES

Medical Model

FALSE POSITIVES OK

FALSE NEGATIVES NOT OK

OK IF NOT ALL ARE SICK
FIND ALL THE SICK PEOPLE

HIGH RECALL

Spam Detector

FALSE POSITIVES NOT OK

FALSE NEGATIVES OK

DON'T NECESSARILY NEED
TO FIND ALL THE SPAM
BETTER BE SPAM

HIGH PRECISION

◦ PRECISION

DIAGNOSIS

PATIENTS

	Diagnosed Sick	Diagnosed Healthy
Sick	1000	200 ⊗
Healthy	800	9000

PRECISION: OUT OF THE PATIENTS WE DIAGNOSED WITH AN ILLNESS, HOW MANY DID WE CLASSIFY CORRECTLY?

$$\text{PRECISION} = \frac{1{,}000}{1{,}000 + 800} = 55.6\%$$

◦ PRECISION

FOLDER

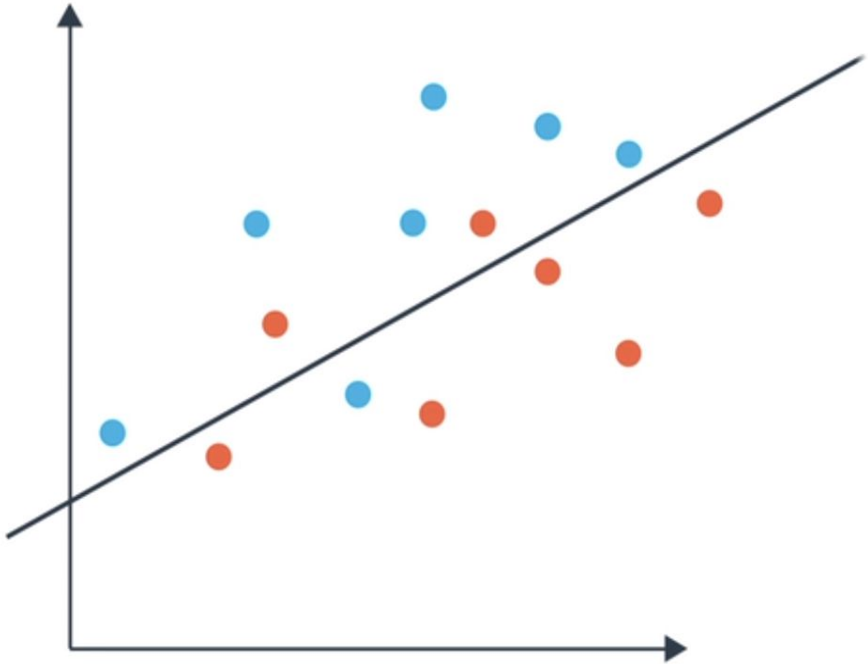
EMAIL

	Sent to Spam Folder	Sent to Inbox
Spam	100	170
Not Spam	30	700

OUT OF ALL THE E-MAILS
SENT TO THE SPAM FOLDER,
HOW MANY WERE ACTUALLY SPAM?

$$\text{PRECISION} = \frac{100}{100 + 30} = 76.9\%$$

◦ PRECISION



OUT OF THE POINTS WE HAVE PREDICTED TO BE POSITIVE, HOW MANY ARE CORRECT?

◦ RECALL

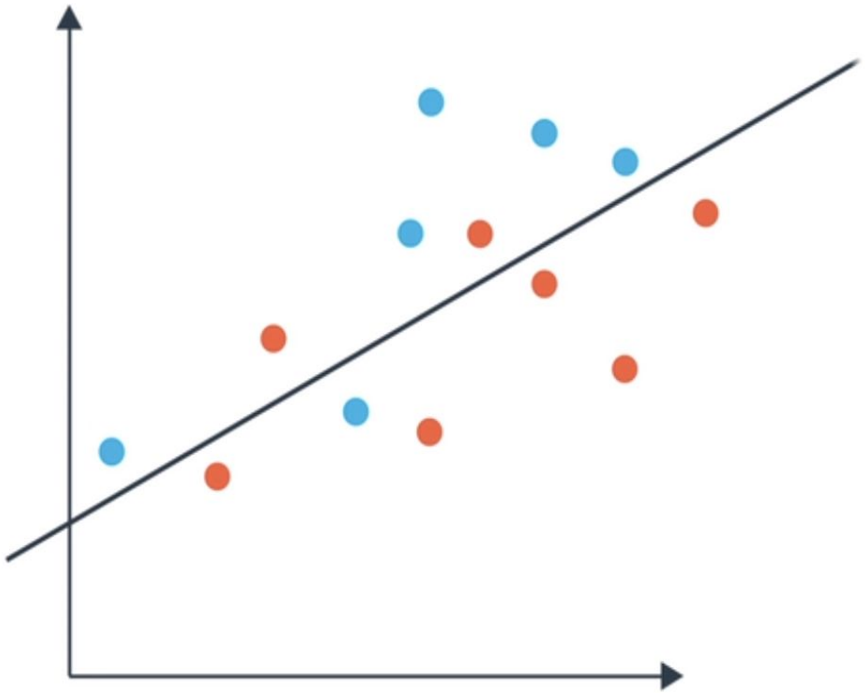
DIAGNOSIS

PATIENTS	Diagnosed Sick	Diagnosed Healthy
Sick	1000	200 ⊗
Healthy	800	8000

OUT OF THE SICK PATIENTS, HOW MANY DID WE CORRECTLY DIAGNOSE AS SICK?

$$\text{RECALL} = \frac{1{,}000}{1{,}000 + 200} = 83.3\%$$

- RECALL



OUT OF THE POINTS LABELLED "POSTIIVE," HOW MANY DID WE CORRECTLY PREDICT?

- F_1 SCORE

ONE SCORE?

MEDICAL MODEL

PRECISION: 55.7%

RECALL: 83.3%

AVERAGE = 69.5%

SPAM DETECTOR

PRECISION: 76.9%

RECALL: 37%

AVERAGE = 56.95%

CREDIT CARD FRAUD

284,335

472

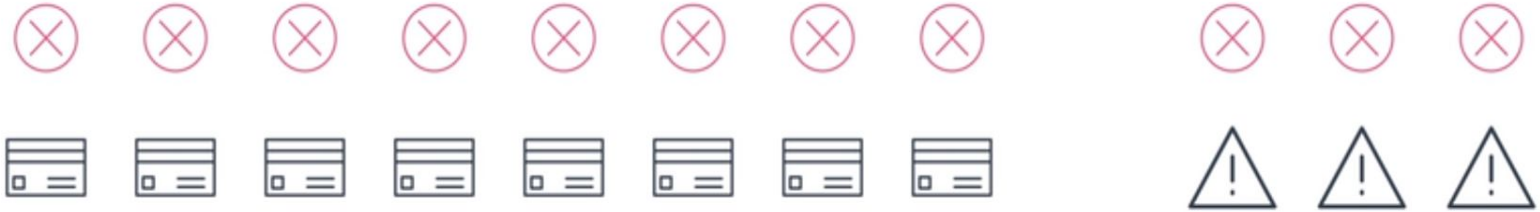
MODEL: ALL TRANSACTIONS ARE GOOD.

PRECISION = 100%

AVERAGE = 50%

RECALL = 0%

◦ CREDIT CARD FRAUD



284,335 472

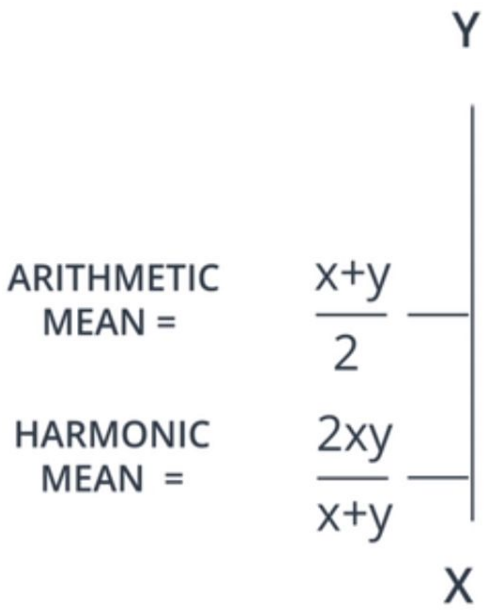
MODEL: ALL TRANSACTIONS ARE FRAUDULENT.

PRECISION = 472/284,807 = 0.16%

RECALL = 472/472 = 100%

AVERAGE = 50.08%

◦ HARMONIC MEAN



PRECISION = 1
RECALL = 0
AVERAGE = 0.5
HARMONIC MEAN = 0

PRECISION = 0.2
RECALL = 0.8
AVERAGE = 0.5
HARMONIC MEAN = 0.32

~~ARITHMETIC MEAN (PRECISION, RECALL)~~

F_1 SCORE = HARMONIC MEAN (PRECISION, RECALL)

CREDIT CARD FRAUD

284,335 472

MODEL: ALL TRANSACTIONS ARE GOOD.

PRECISION = 100% F_1 SCORE = 0 RECALL = 0%

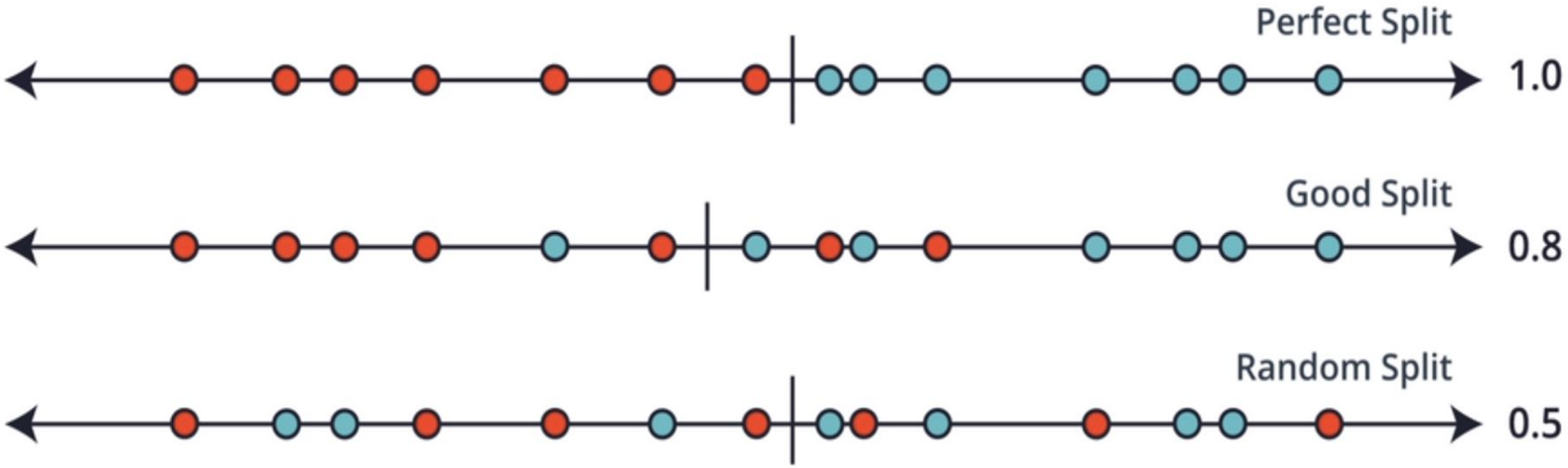
◦ QUIZ: F_β SCORE

$$F_1 \text{ SCORE} = 2 \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$F_\beta \text{ SCORE} = (1+\beta^2) \frac{\text{Precision} * \text{Recall}}{\beta^2 * \text{Precision} + \text{Recall}}$$

PRECISION $F_{0.5}$ SCORE F_1 SCORE F_2 SCORE RECALL

◦ RECEIVER OPERATING CHARACTERISTIC



ROC CURVE

$$\text{True Positive Rate} = \frac{\text{TRUE POSITIVES}}{\text{ALL POSITIVES}} = \frac{6}{7} =$$

$$\text{False Positive Rate} = \frac{\text{FALSE POSITIVES}}{\text{ALL NEGATIVES}} = \frac{2}{7} =$$

ROC CURVE

$$\text{True Positive Rate} = \frac{\text{TRUE POSITIVES}}{\text{ALL POSITIVES}} = \frac{7}{7} =$$

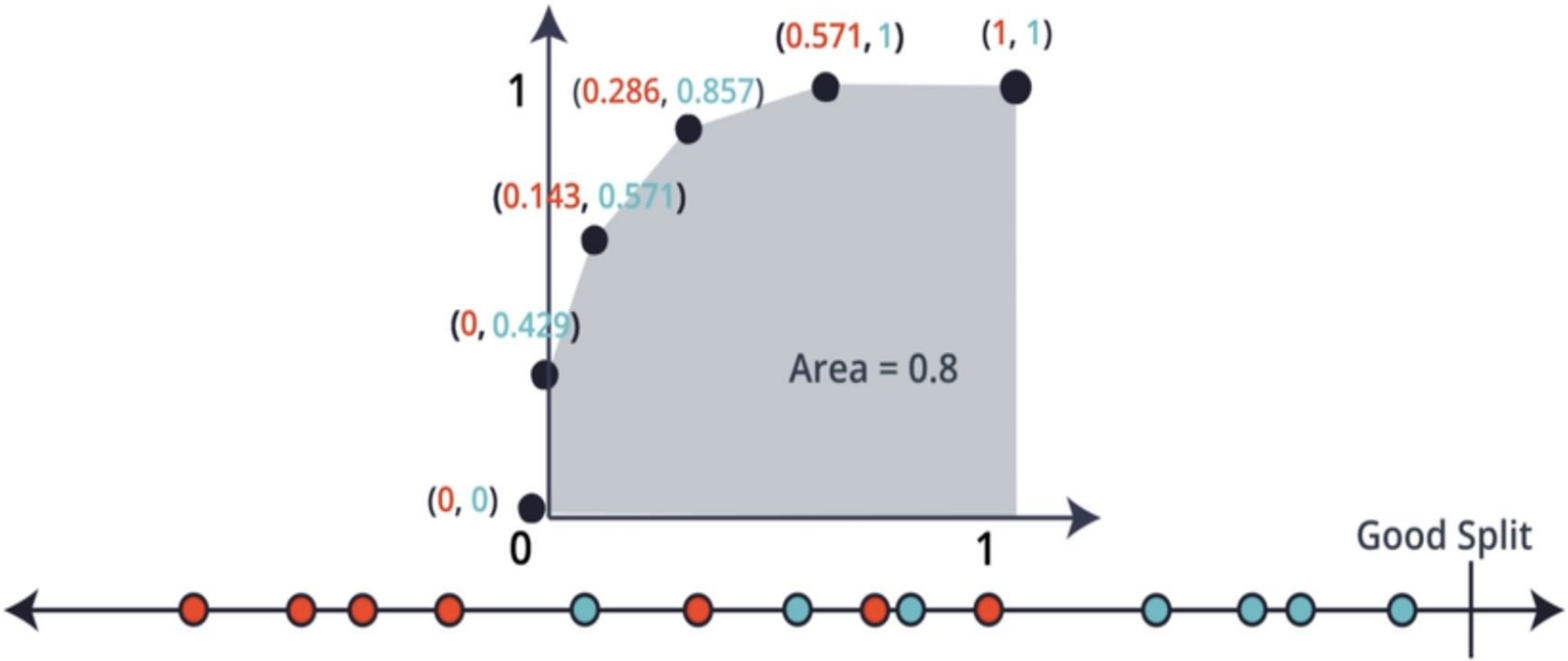
$$\text{False Positive Rate} = \frac{\text{FALSE POSITIVES}}{\text{ALL NEGATIVES}} = \frac{7}{7} =$$

ROC CURVE

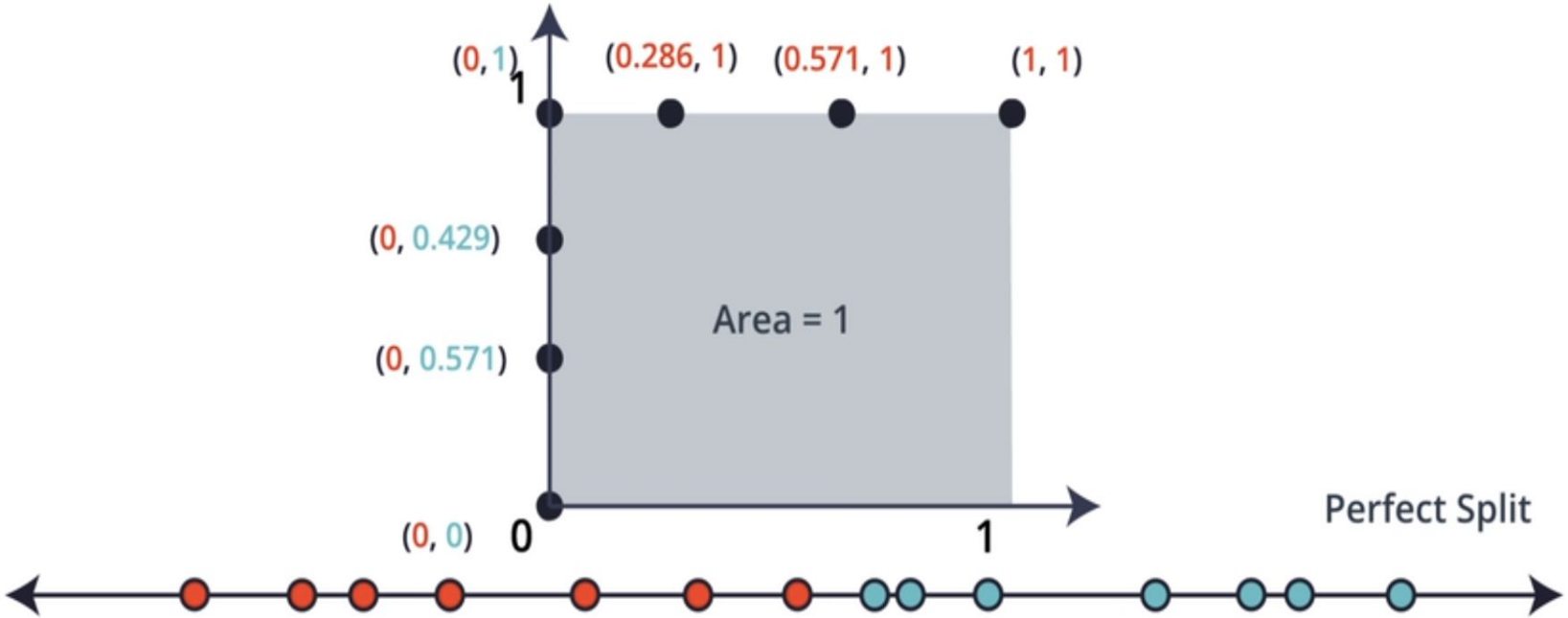
$$\text{True Positive Rate} = \frac{\text{TRUE POSITIVES}}{\text{ALL POSITIVES}} = \frac{0}{7} =$$

$$\text{False Positive Rate} = \frac{\text{FALSE POSITIVES}}{\text{ALL NEGATIVES}} = \frac{0}{7} =$$

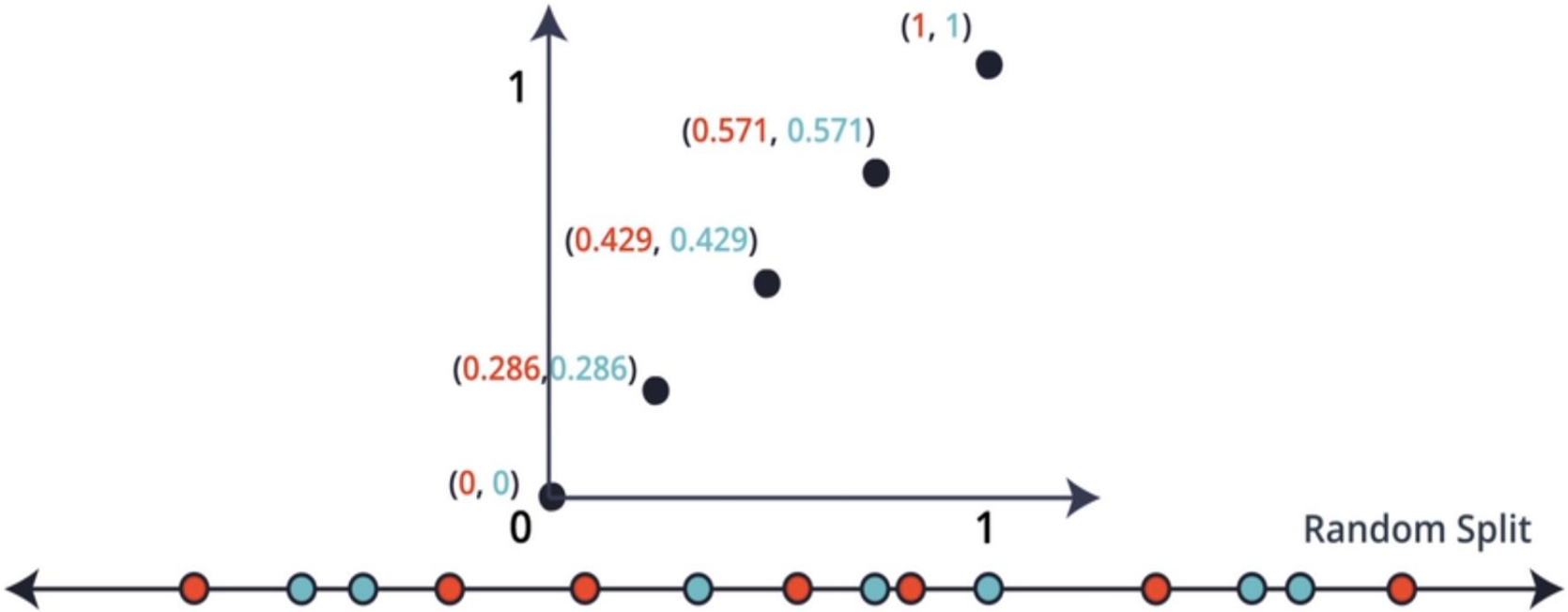
ROC CURVE



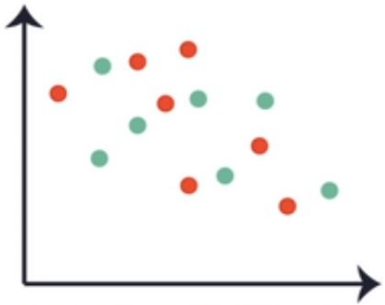
ROC CURVE



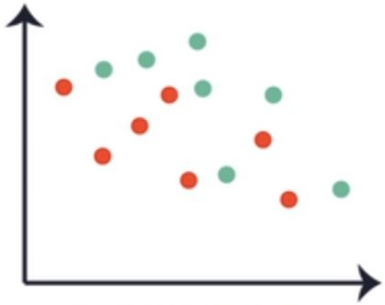
ROC CURVE



AREA UNDER A ROC CURVE

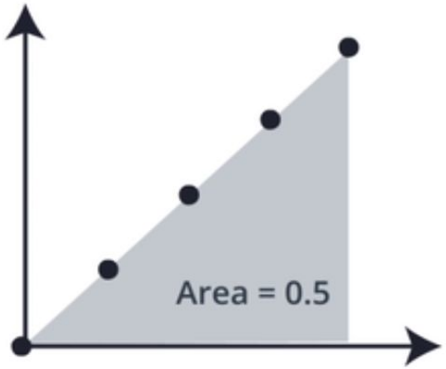


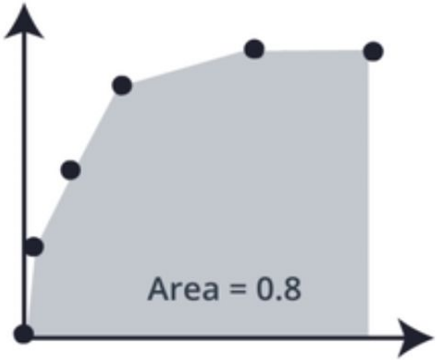
RANDOM SPLIT

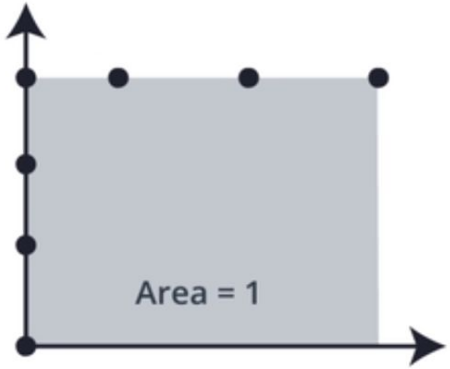


GOOD SPLIT

PERFECT SPLIT







Can the area under the ROC Curve be less than 0.5?

YES

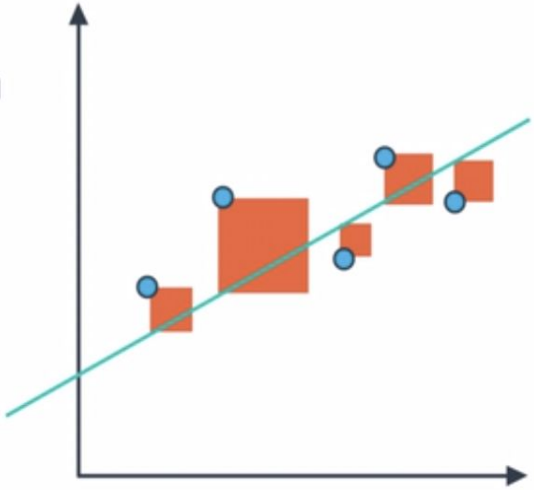
MEAN SQUARED ERROR IN SKLEARN

```
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression

classifier = LinearRegression()
classifier.fit(X,y)

guesses = classifier.predict(X)

error = mean_squared_error(y, guesses)
```



R2 SCORE

BAD MODEL

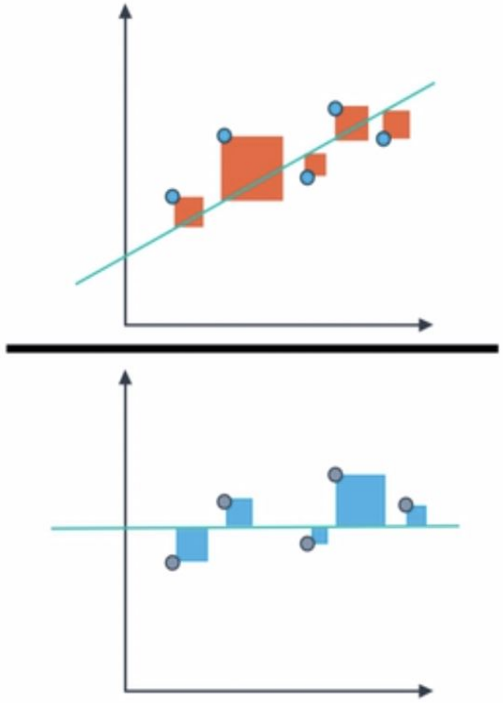
The errors should be similar.
R2 score should be close to 0.

GOOD MODEL

The mean squared error for the linear regression model should be a lot smaller than the mean squared error for the simple model.

R2 score should be close to 1.

$$R2 = 1 - \frac{}{}$$

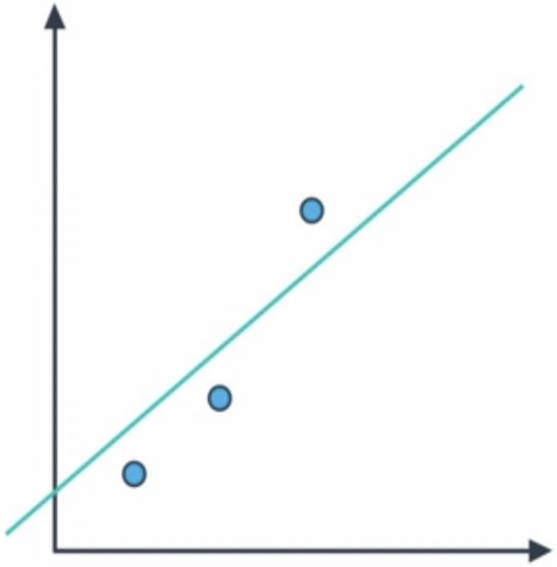


○ R2 SCORE IN SKLEARN

```
from sklearn.metrics import r2_score

y_true = [1, 2, 4]
y_pred = [1.3, 2.5, 3.7]

r2_score(y_true, y_pred)
```



◦ HOW TO SOLVE A PROBLEM

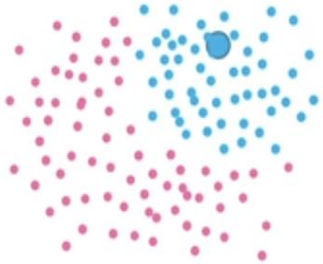
PROBLEM

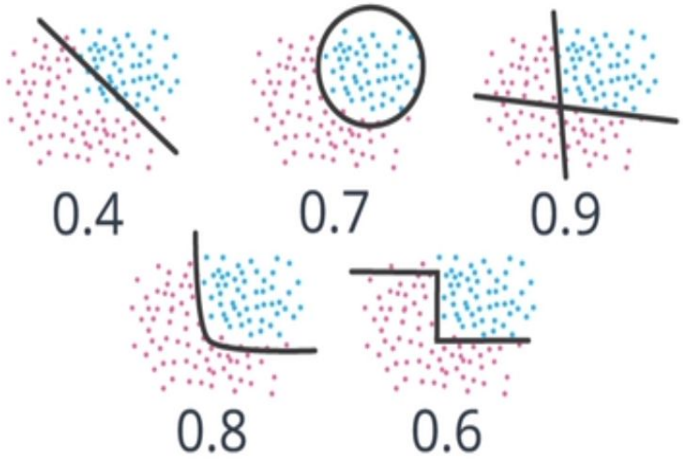
TOOLS

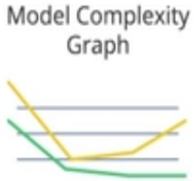
MEASUREMENT TOOLS

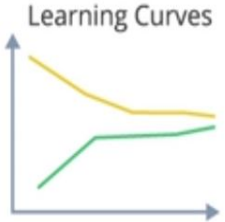
Measure each tool's performance & pick the best tool

◦ HOW TO USE MACHINE LEARNING









DATA

ALGORITHMS

METRICS