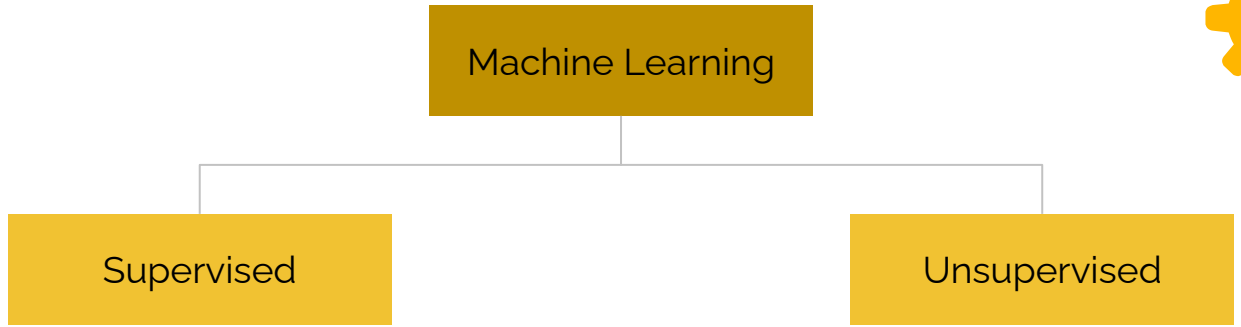
Machine Learning I

Random Forests

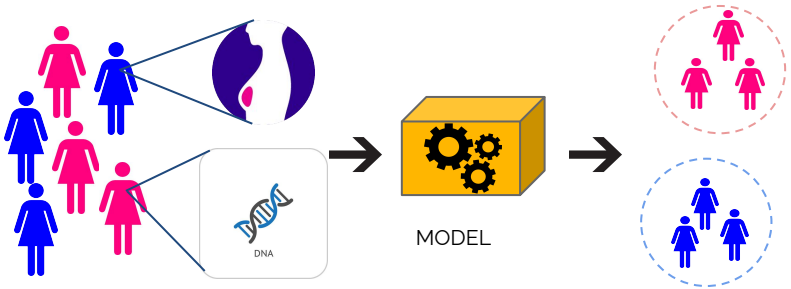
August 8th 2020


Machine Learning
Supervised
Unsupervised

Machine Learning

Supervised

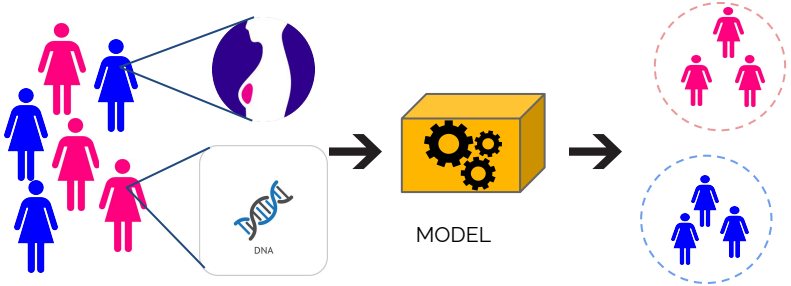
Unsupervised



Labelled Data

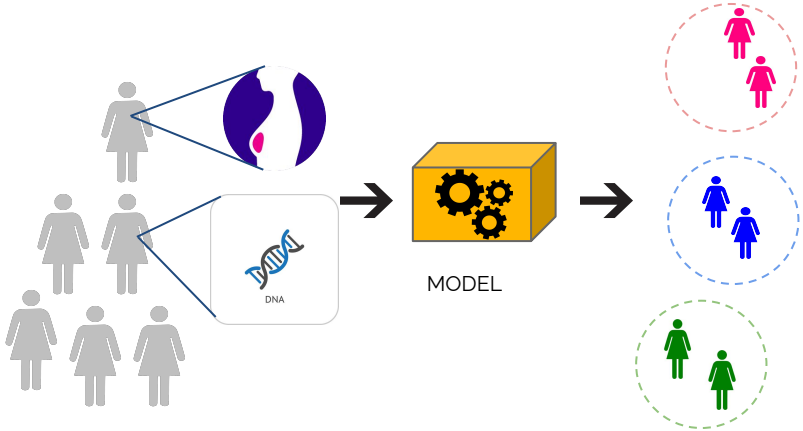
Machine Learning

Supervised



Labelled Data

Unsupervised

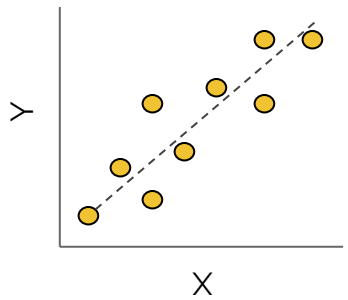


Unlabelled Data

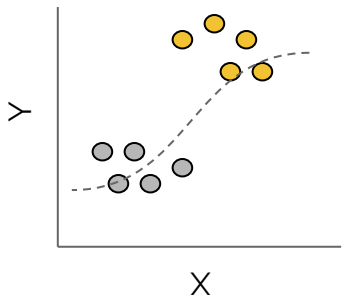
Machine Learning

Supervised

Regression

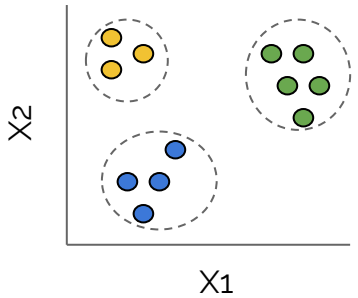


Classification



Unsupervised

Clustering



Type	Name
Linear	Linear Regression
	Logistic Regression
Survival	Cox Proportional Hazard
Tree-based	Random Forest
	Gradient Boosting
Neural Network	Neural Networks

Type	Name
Linear	Linear Regression
	Logistic Regression
Survival	Cox Proportional Hazard
Tree-based	Random Forest
	Gradient Boosting
Neural Network	Neural Networks

Let's Build an Orange Classifier!

ORANGE

GRAPEFRUIT

citrus?

sweet?

weight?

ORANGE

ORANGE

GRAPEFRUIT

ORANGE

GRAPEFRUIT

GRAPEFRUIT

Collect Data!

ORANGE

GRAPEFRUIT

fruit	citrus
1	yes
2	yes
3	yes
4	yes
5	yes
6	yes

Collect Data!

ORANGE

GRAPEFRUIT

fruit	citrus	sugar
1	yes	10
2	yes	11
3	yes	7
4	yes	10
5	yes	6
6	yes	5

Collect Data!

ORANGE

GRAPEFRUIT

fruit	citrus	sugar	weight
1	yes	10	130
2	yes	11	115
3	yes	7	120
4	yes	10	200
5	yes	6	190
6	yes	5	123

Collect Data!

ORANGE

GRAPEFRUIT

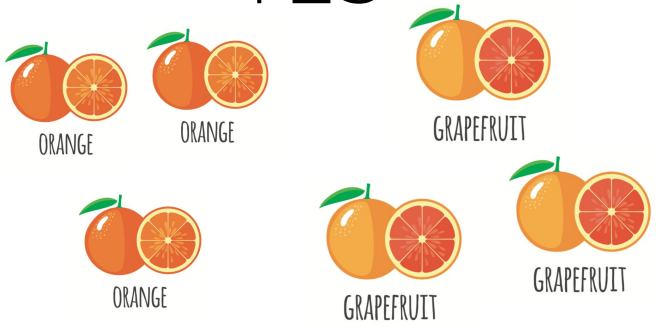
fruit	citrus	sugar	weight	orange
1	yes	10	130	1
2	yes	11	115	1
3	yes	7	120	1
4	yes	10	200	0
5	yes	6	190	0
6	yes	5	123	0

A Bad Question

Is it a citrus fruit?

YES

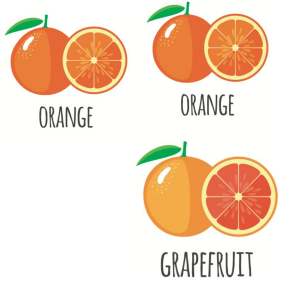
NO



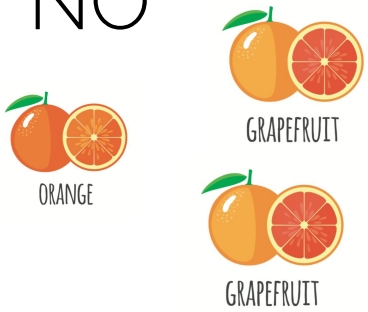
A Better Question

Is sugar >= 10g

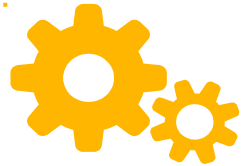
YES



NO

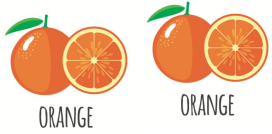


A Much Better Question



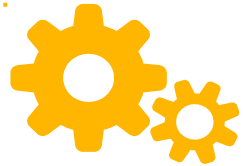
Is weight < 150g

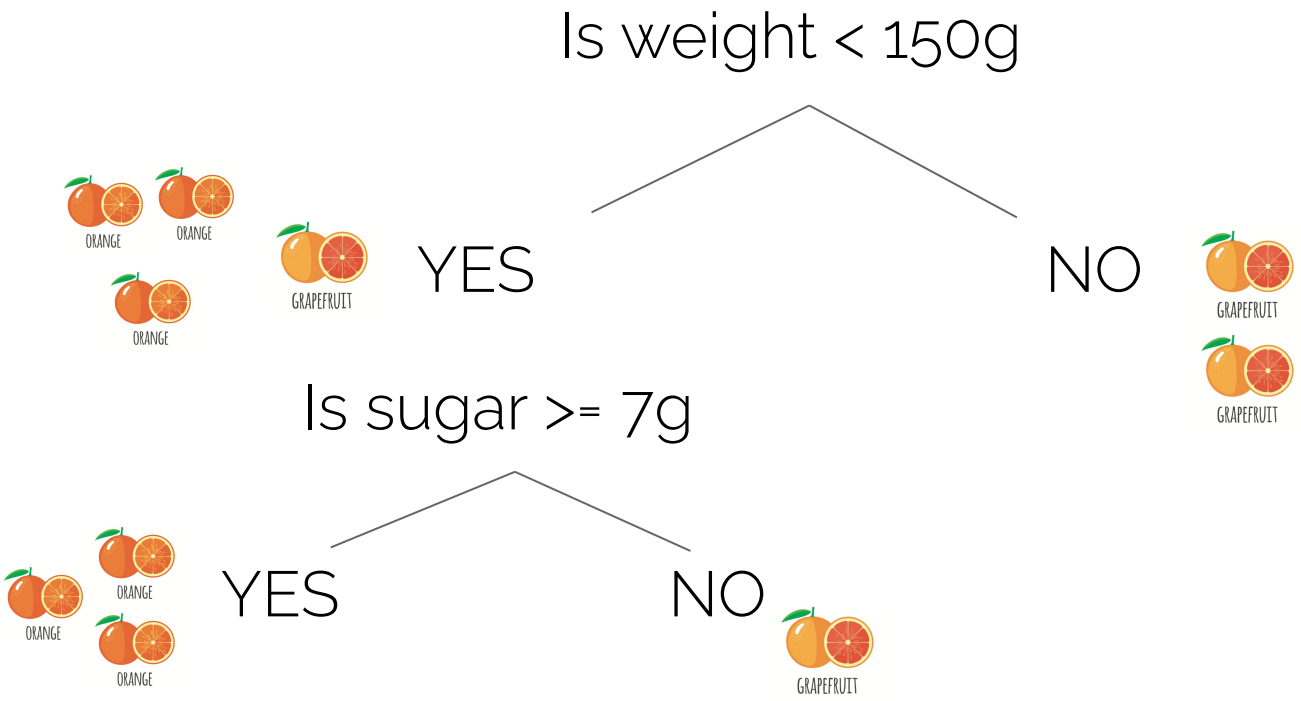
YES



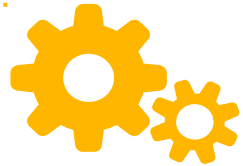
NO

Stack the Questions

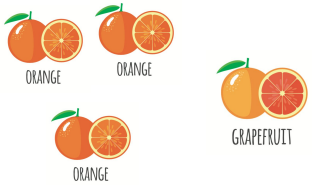




Decision Tree

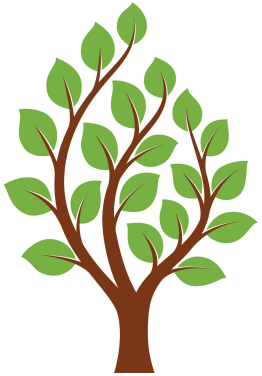


Is weight < 150g

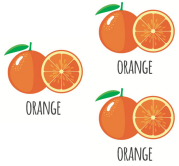


YES

NO



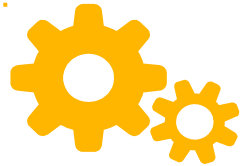
Is sugar >= 7g



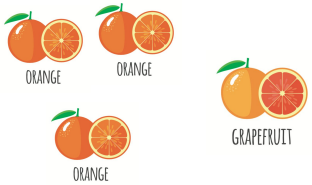
YES

NO

Decision Tree



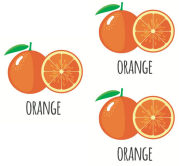
Is weight < 150g



YES

NO

Is sugar >= 7g

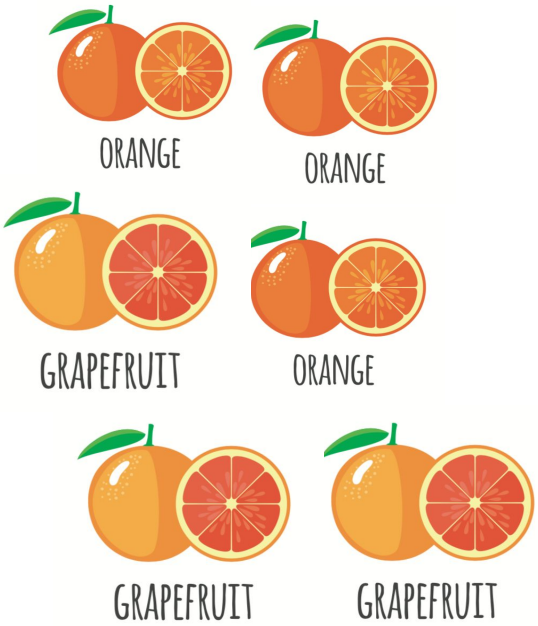


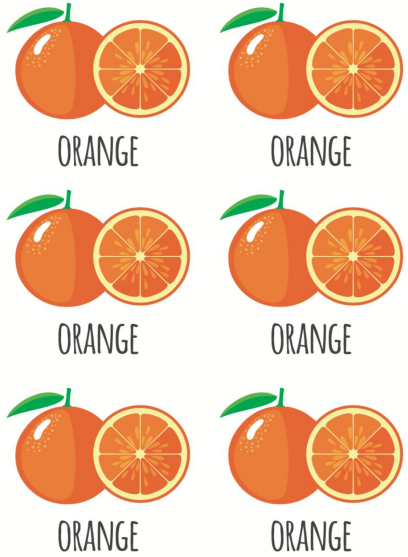
YES

NO

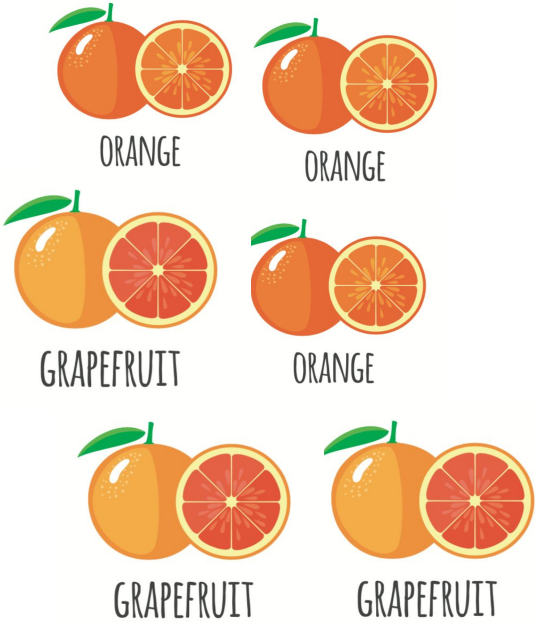
Which questions should you ask first?

Randomness

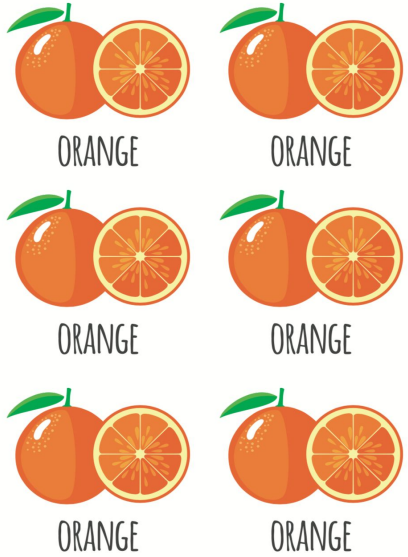

ORANGE
ORANGE
GRAPEFRUIT
ORANGE
GRAPEFRUIT
GRAPEFRUIT


ORANGE
ORANGE
ORANGE
ORANGE
ORANGE
ORANGE

Entropy

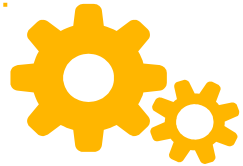


Random! (e=1)



Pure! (e=0)

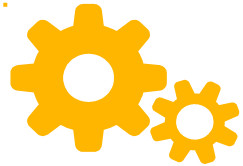
Entropy



$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

A measure of randomness!

Entropy



$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

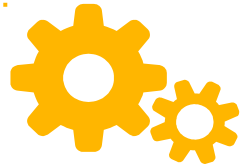
Go through each class and add

Entropy

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Proportion of fruits in each class * log2 proportion of fruits in each class

Entropy



$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

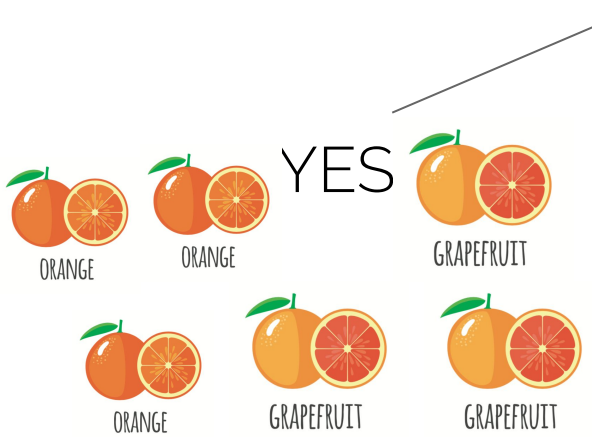
(-prop(oranges)*log2(prop(oranges))

+

(-prop(grapefruits)*log2(prop(grapefruits))

Decision Trees

Is it a citrus fruit?



$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

(3/6)x-log2(3/6) + (3/6)x-log2(3/6)

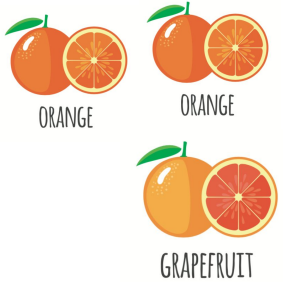
=1

Decision Trees

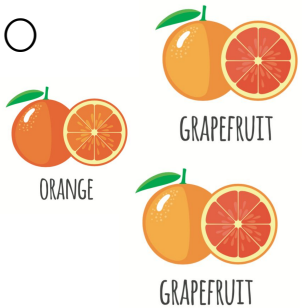
Is sugar >= 10g

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

YES



NO



(⅔)x-log2(⅔) + (⅓)x-log2(⅓)

=0.92

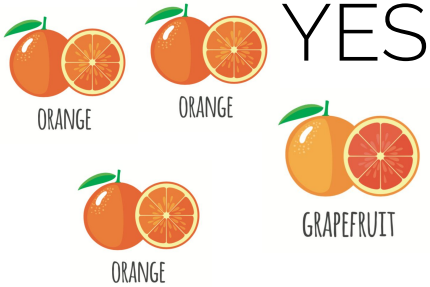
(⅔)x-log2(⅔) + (⅓)x-log2(⅓)

= 0.92

Decision Trees

Is weight < 150g

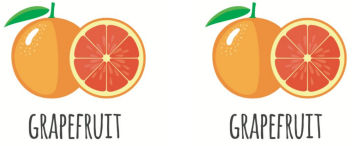
YES



(¾)x-log2(¾) + (¼)x-log2(¼)

=0.81

NO

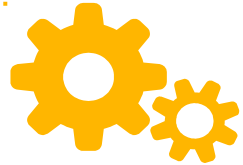


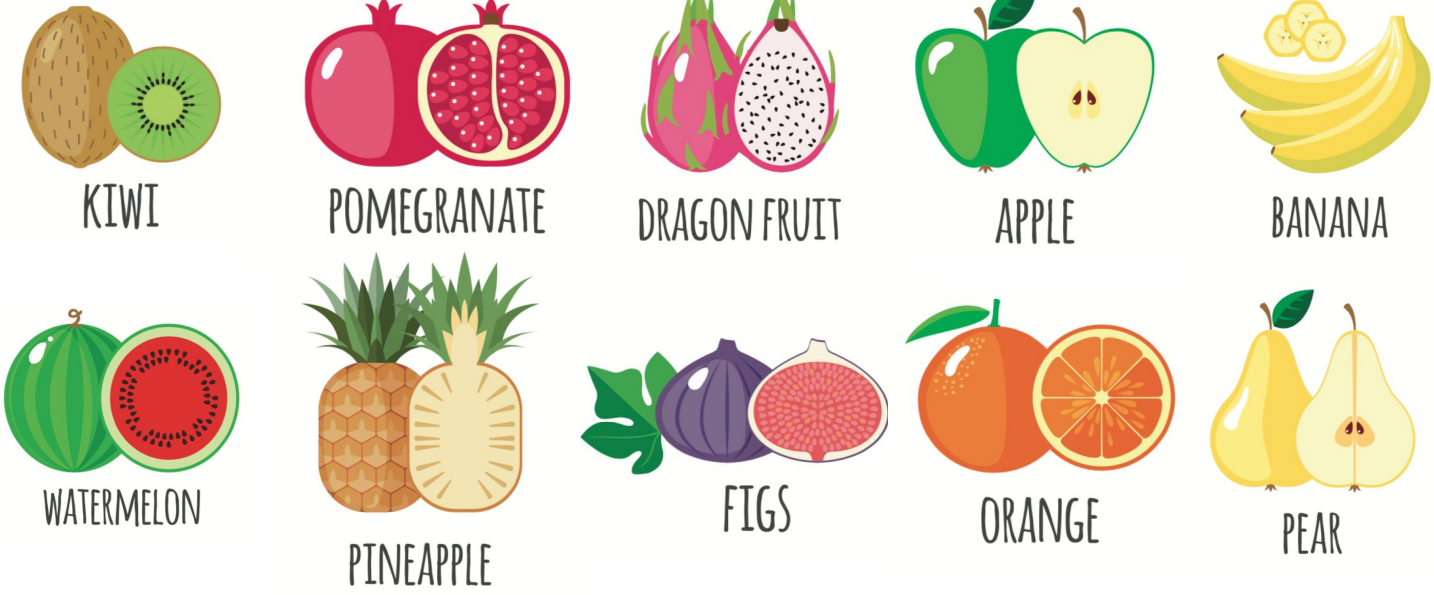
(2/2)x-log2(2/2)

=0

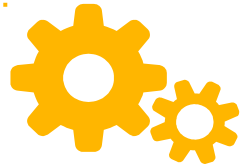
Weight Sugar Citrus

Larger dataset?





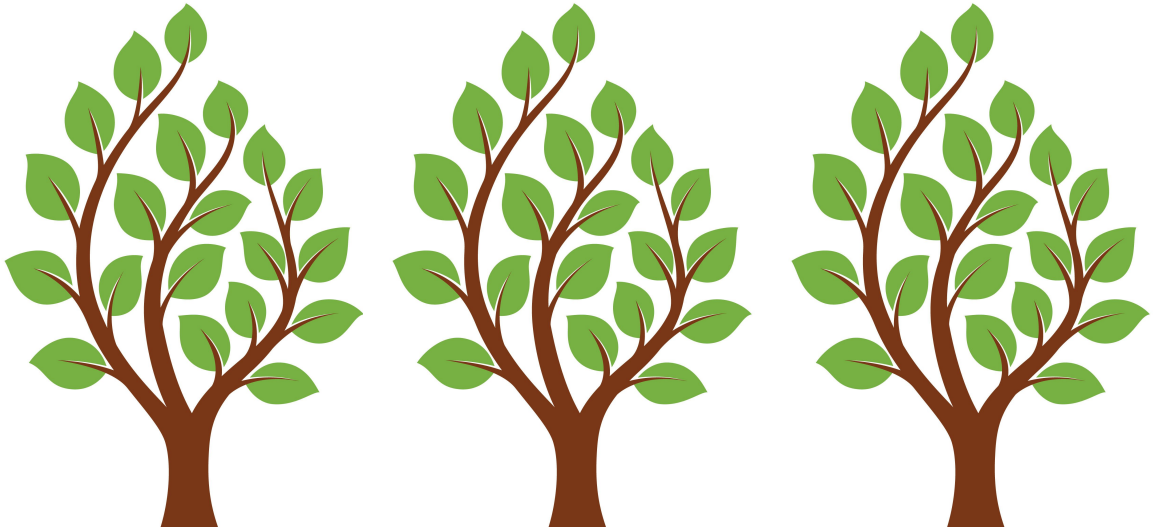
Many more questions!



fruit	citrus	sugar	weight	Col in	Col out	pit	seeds	round	soft	orange
1	yes	10	130	yellow	yellow	yes	no	yes	yes	0
2	yes	11	115	orange	orange	no	yes	no	no	1
3	yes	7	120	red	green	no	yes	no	yes	0
4	yes	10	200	yellow	yellow	no	no	no	no	0
5	yes	6	190	white	yellow	no	yes	yes	no	0
6	yes	5	123	green	green	no	no	no	no	0

Many more questions!

Multiple trees!



Random Forest

Many more columns!

Tree #1

weight
130
115
120
200
190
123

pit	seeds
yes	no
no	yes
no	yes
no	no
no	yes
no	no

orange
0
1
0
0
0
0

Many more columns!

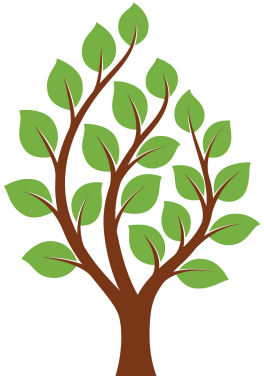
citrus	sugar
yes	10
yes	11
yes	7
yes	10
yes	6
yes	5

Col out
yellow
orange
green
yellow
yellow
green

Tree #2

orange
0
1
0
0
0
0

Many more columns!

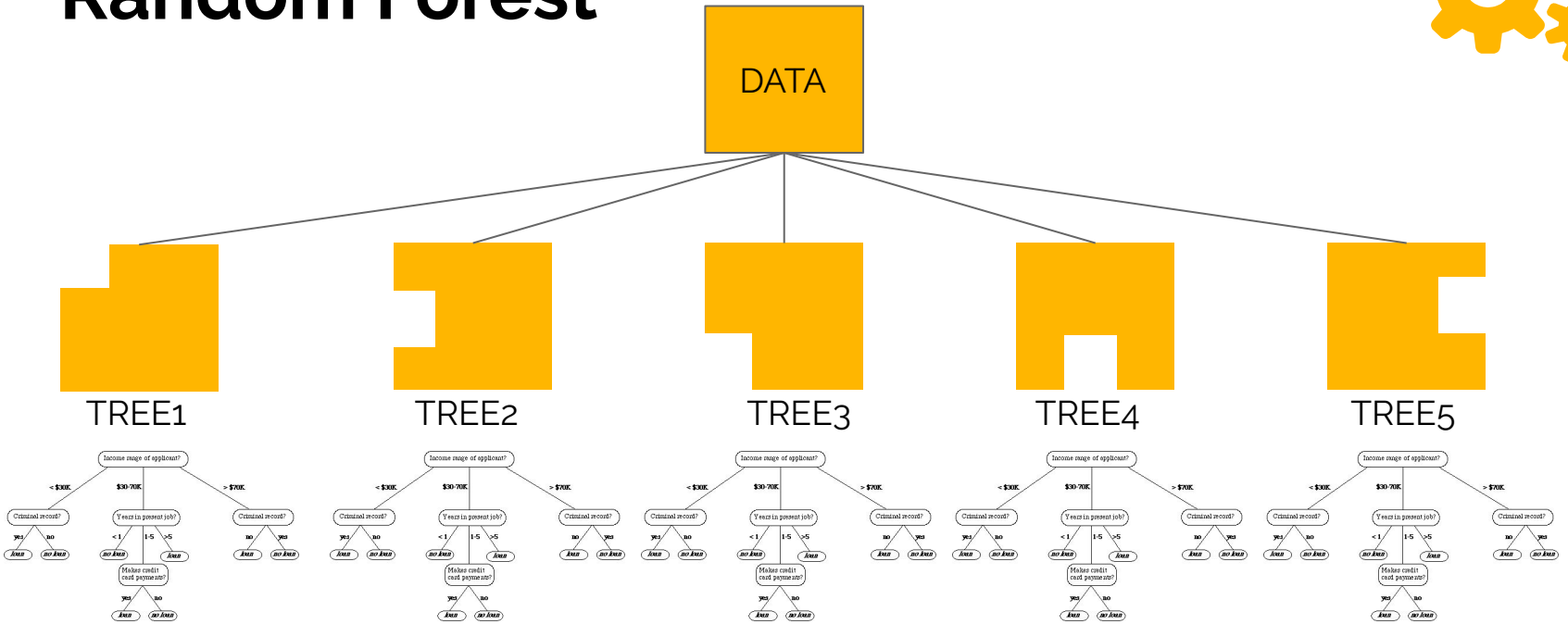


Tree #3

Col in
yellow
orange
red
yellow
white
green

round	soft	orange
yes	yes	0
no	no	1
no	yes	0
no	no	0
yes	no	0
no	no	0

Random Forest



Wisdom of the crowds!

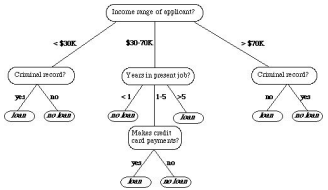
Random Forest

New Fruit

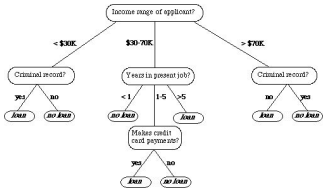
Random Forest

New Fruit

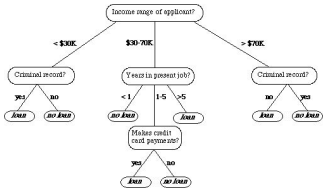
TREE1



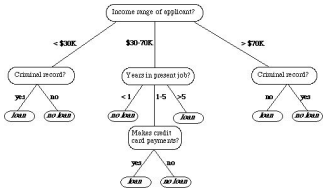
TREE2



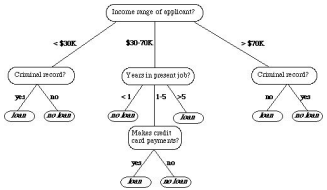
TREE3



TREE4



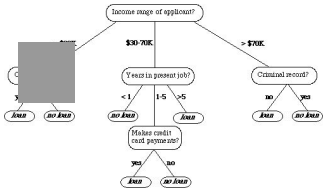
TREE5



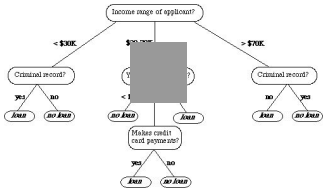
Random Forest

New Fruit

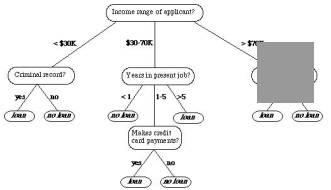
TREE1



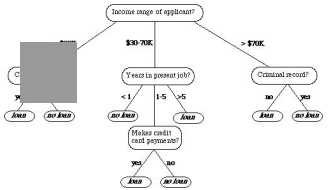
TREE2



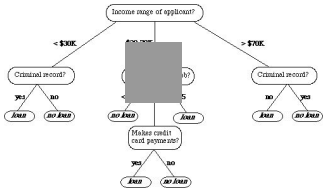
TREE3



TREE4



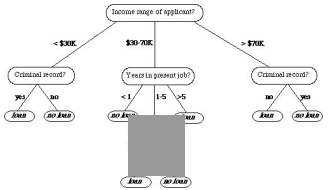
TREE5



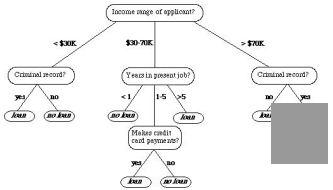
Random Forest

New Fruit

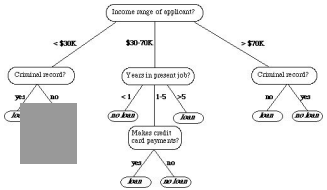
TREE1



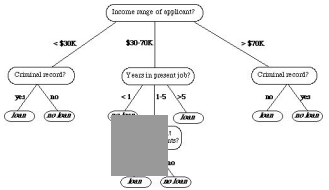
TREE2



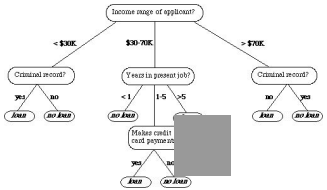
TREE3



TREE4



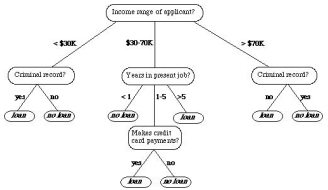
TREE5



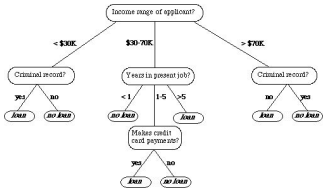
Random Forest

New Fruit

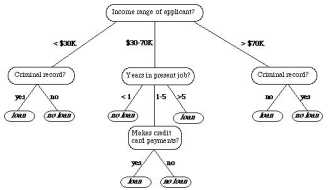
TREE1



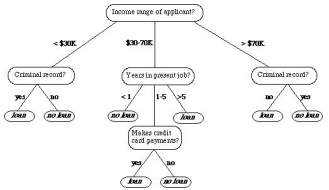
TREE2



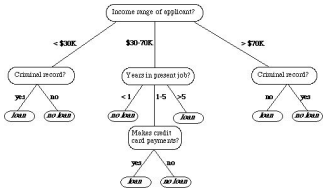
TREE3



TREE4



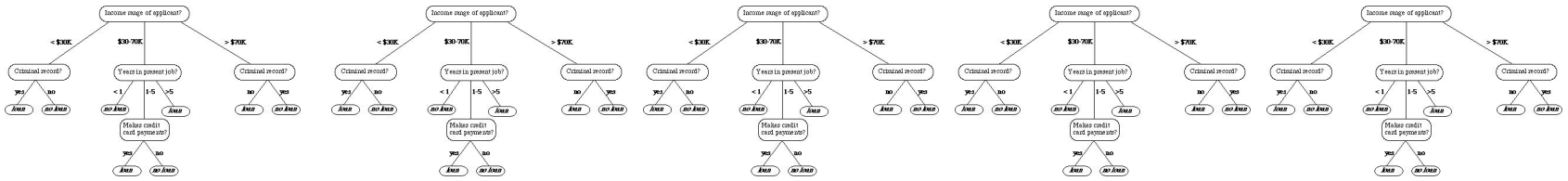
TREE5

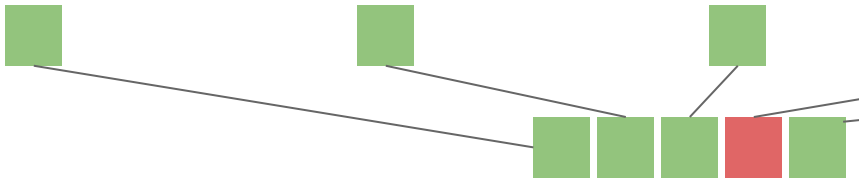


Random Forest

New Client

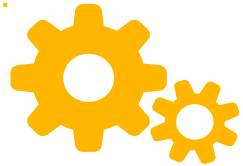
TREE1 TREE2 TREE3 TREE4 TREE5





80%

Random Forest Parameters



max_depth	n_estimators
Maximum number of questions asked for each branch	Number of trees to grow.

Random Forest Implementation

Python Exercise

Model Training

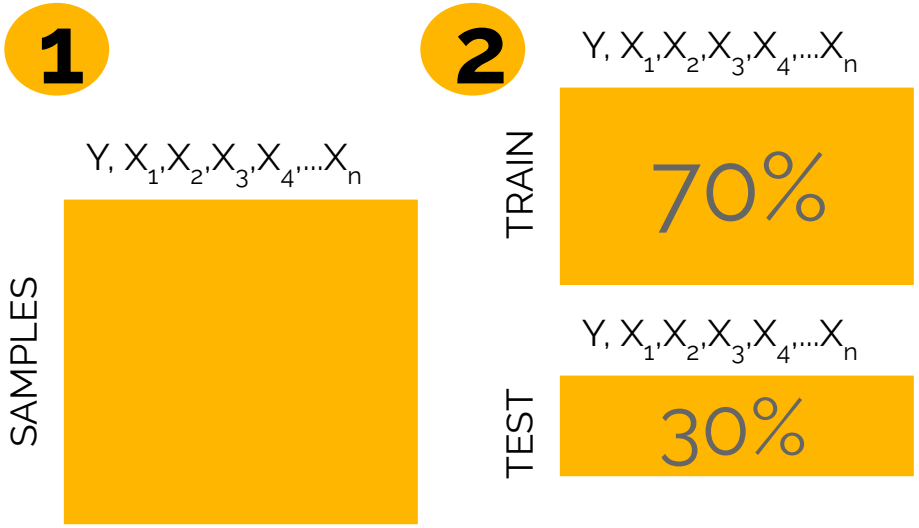
Model Performance

1

$Y, X_1, X_2, X_3, X_4, \ldots X_n$

SAMPLES

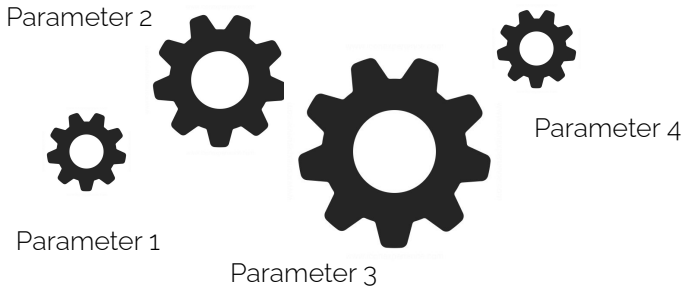
Model Performance

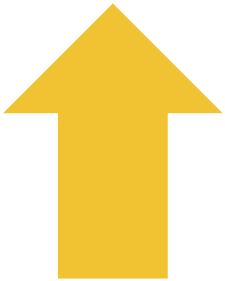


Model Performance

Parameter Tuning

Model

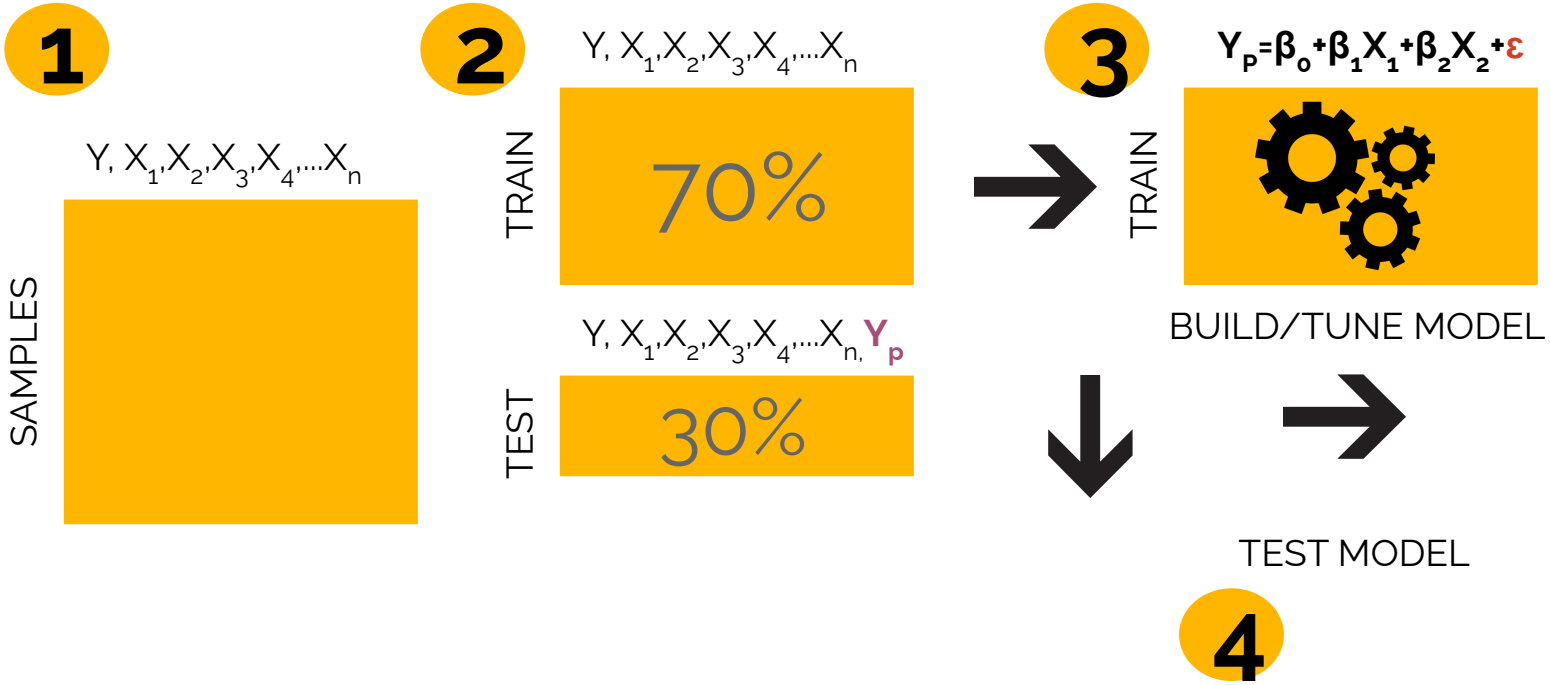




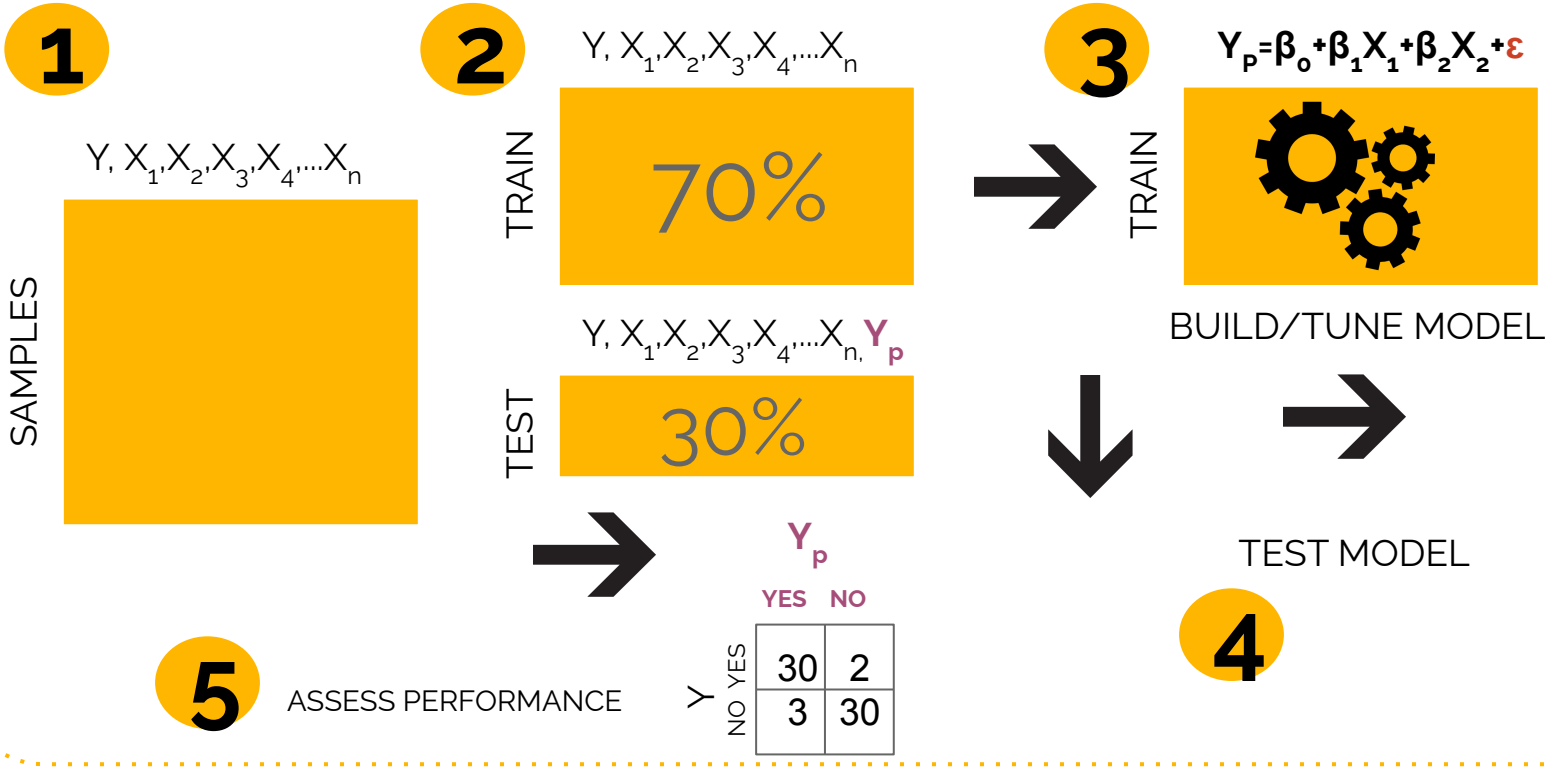
Maximize Performance

Classification Assessment

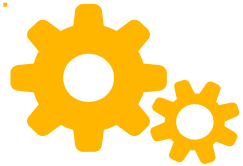
Model Performance



Model Performance



Confusion Matrix



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

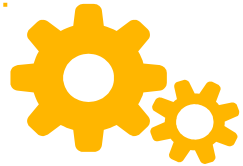
TN	True Negative
FP	False Positive
FN	False Negative
TP	True Positive

Model Performance

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

Source: http://scaryscientist.blogspot.ca/2016/03/confusion-matrix.html

Random Forest



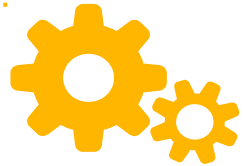
3- Split to Train and Test

```
#split the data to 70% train and 30% test
x_train,x_test,y_train,y_test = train_test_split(X,Y,test_size=0.3,random_state=42)

print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(398, 30)
(171, 30)
(398,)
(171,)
```

Random Forest



4- Train your model: Random Forest

```
rf_model = RandomForestClassifier(max_depth=3,n_estimators=15)          #define the model
rf_model.fit(x_train, y_train)                                          #fit the model (train)
rf_model.score(x_train,y_train)                                         #predict on new observations

#what is the accuracy of this model?
```

```
0.9849246231155779
```