
Introduction to Data Analytics

CSRBOX – IBM Innovation Camp



What should I choose : R or Python ?



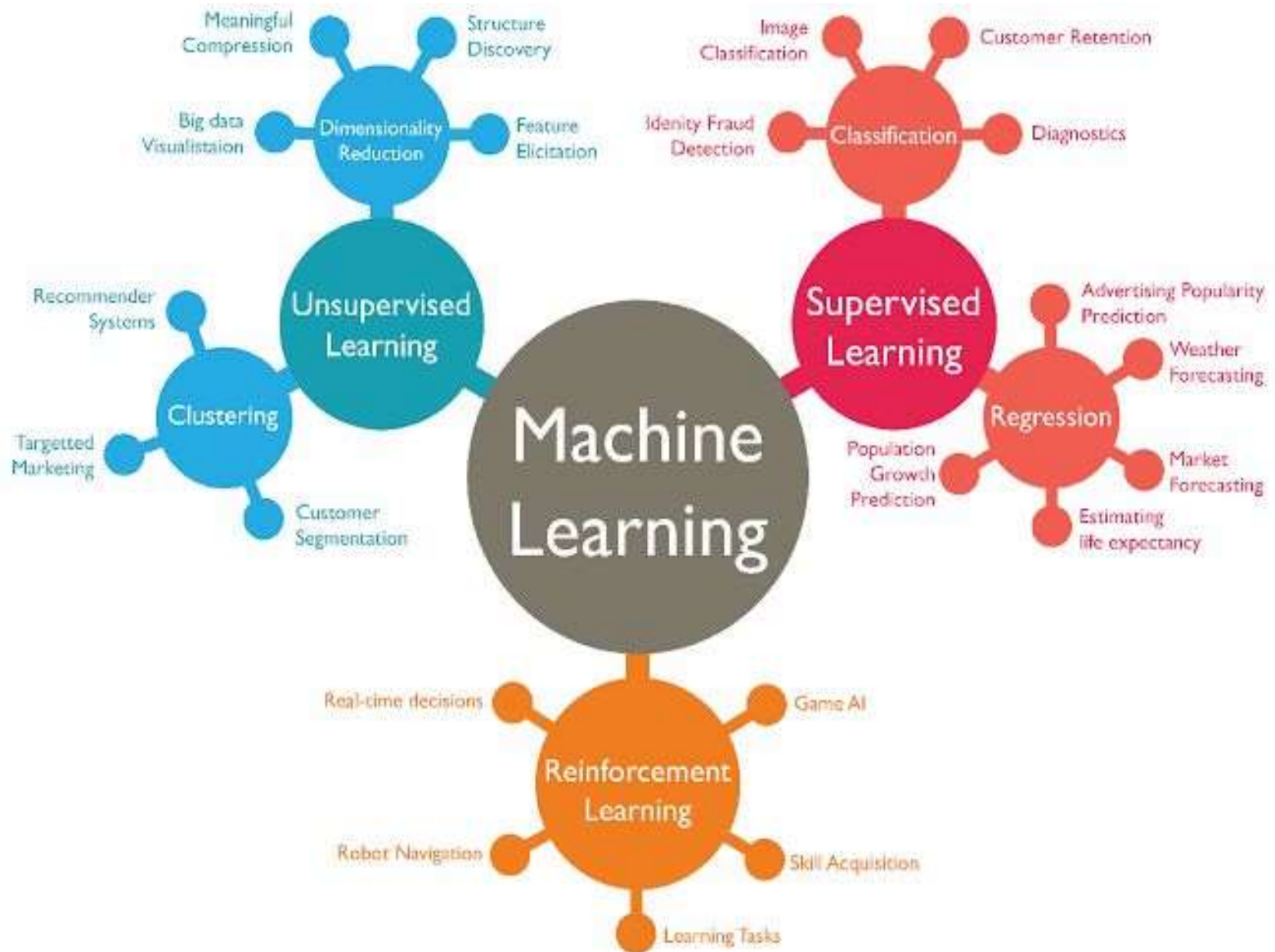
What should I choose R or Python ?



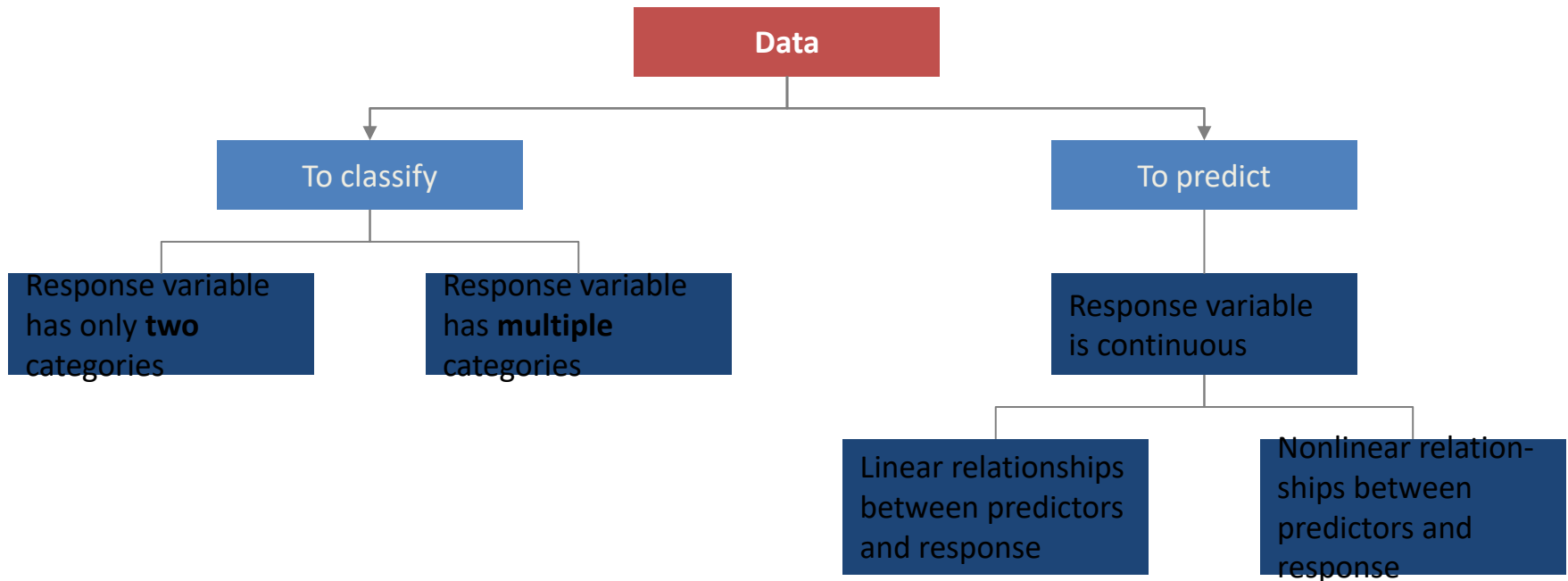
What should I choose R or Python ?



1. What's your end objective ?
2. Syntax Vs Problem Solving



There are 2 types of supervised learning models: Classification vs. Regression



Classification model (Discrete)

Classification models are used to separate the dataset into classes belonging to the response variable

Usually the output variable has two classes: Yes or No, True or false (1 or 0)

Thus classification trees are used when the response or target variable is categorical in nature

Source: <http://www.simafore.com>

Regression model (Continuous)

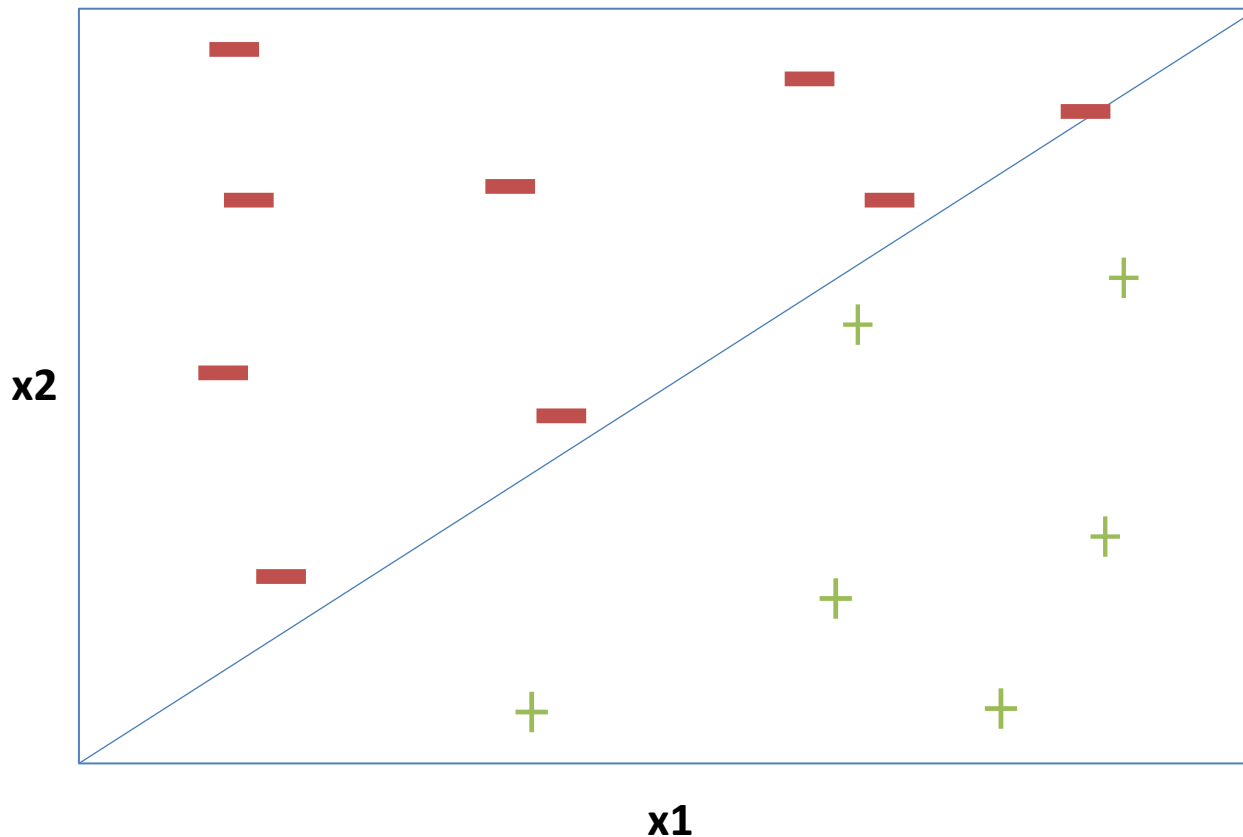
Regression trees are needed when the response variable is numeric or continuous, example- temperature, pressure, strength etc.

Thus regression trees are applicable for prediction type of problems as opposed to classification

Classification model creates decision boundary between 2 or more classes

+ Class 1 ■ Class 0 — Decision boundary of model

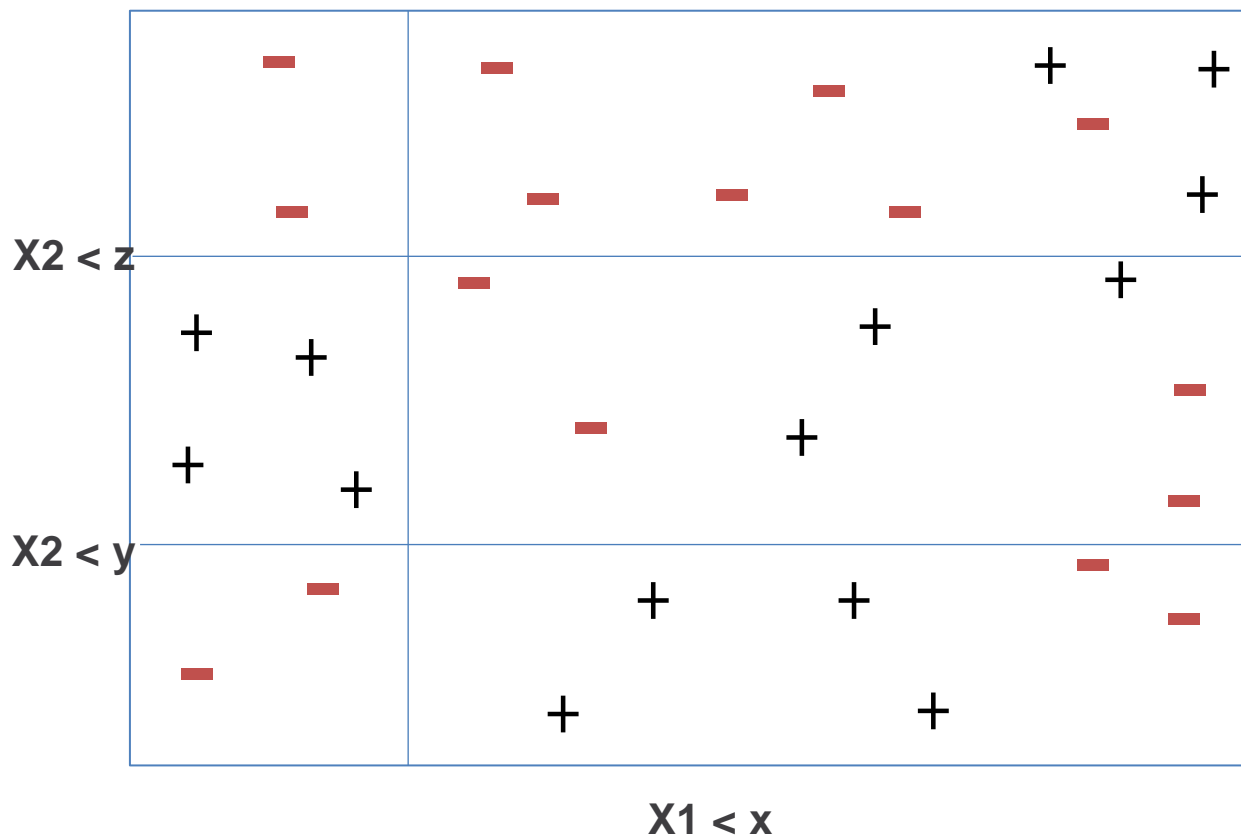
Can you explain this data using a linear model?



Classification model creates decision boundary between 2 or more classes

+ Class 1 - Class 0 — Decision boundary of model

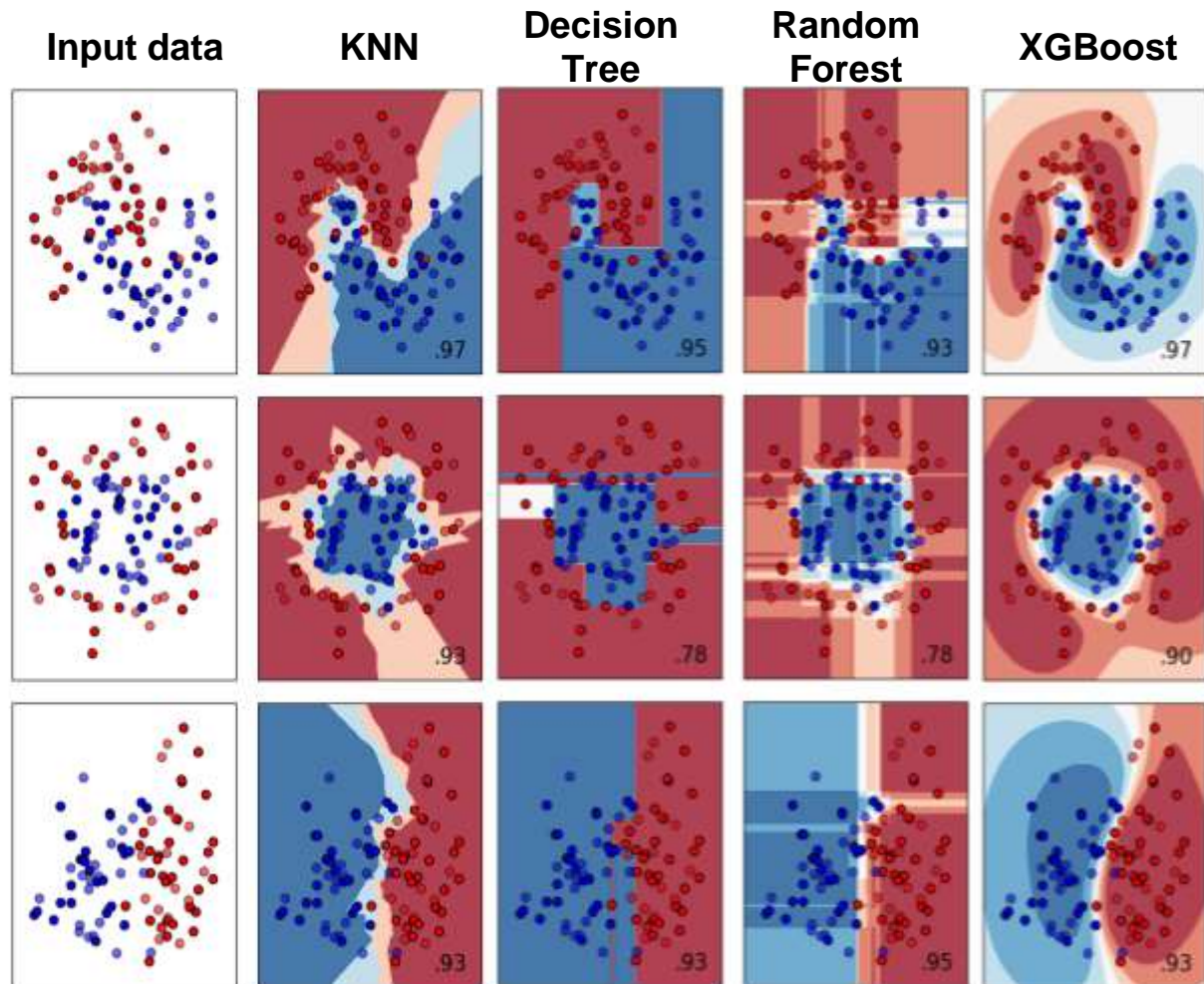
Can you explain this data using a linear model?



Classification – Identifying the problem

Foundational module for sifting incoming data from unified or multiple sources

- **Classification** is a pattern recognition technique used for identifying which category a new observation belongs to based on learning from a training data where category membership is known
- Classification can be binary (2 categories) or multi class (more than 2 categories)
- Different evaluation metrics (AUC, accuracy, precision, etc.) are used understand how effective the model is in classifying data points
- Some applications of classification are:
 - Propensity modeling
 - Text and image classification



Classification Algorithms

- Decision Tree
- Random Forest

3: Decision tree



What is it?

A systematic approach to **segment the space of your predictors into a number of simple regions:**

- We use if-then statements to define patterns in data

We understand **average behavior/response of each region** (using mean or mode)

Then we classify the region with a **single dependent variable label**

We will then have a **tree for**



Why use it?

Output is highly visual and easy to interpret

Neither linearity of relation nor independence of parameters is required



When to use it?

When you have a clear hypothesis about what to solve for

Prediction of segment-specific customer behavior (e.g., buying decisions, churn rate, consumption rate)

Chi-Square automatic interaction detector

Working Principle: The entire tree is grown, and then branches where data is deemed to be an overfit are truncated by comparing the decision tree through the withheld subset.

Splitting Nodes: Binary splits (each node is split into two daughter nodes)

Pruning: CART on the other hand grows a large tree and then **post-prunes** the tree back to a smaller version

Sample size: Can work in smaller samples of data

Use: Can be used for both regression and classification

CART

Classification and regression trees

Working Principle: Uses a statistical rule to stop tree growth called the Chi-Square test. The Chi-Square test qualifies the values observed, to those in theory. Values which are far off from theory, independent, are cut off and the tree stops at that branch

Splitting Nodes: CHAID uses **multiway splits** by default (multiway splits means that the current node is splitted into more than two nodes)

Pruning: CHAID uses a **pre-pruning idea**. A node is only split if a significance criterion is fulfilled.

Sample size: Requires considerably larger data chunks

Use: Intended to work with **categorical/discretized** targets

Random Forest



What is it?

Ensemble of decision trees that produces more stable predictions and is less prone to over fitting than a single decision tree

Each tree uses a random subset from the dataset and a random subset of the variables at each node

The final prediction is the average of the predictions from each tree



Why use it?

Minimize the problem of overfitting as compared to an individual decision tree which might over fit

Very efficient at handling datasets with a large number of input variables and more specifically in identifying the most important variables out of those.



When to use it?

When you have a clear hypothesis about what to solve for

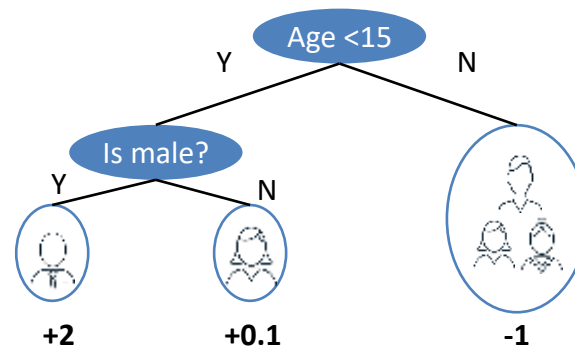
Prediction of segment-specific customer behavior (e.g., buying decisions, churn rate, consumption rate)

How are ensemble models made?

Think of a simple decision tree where you have to predict who among the family members like computer game. We can draw a simple tree:

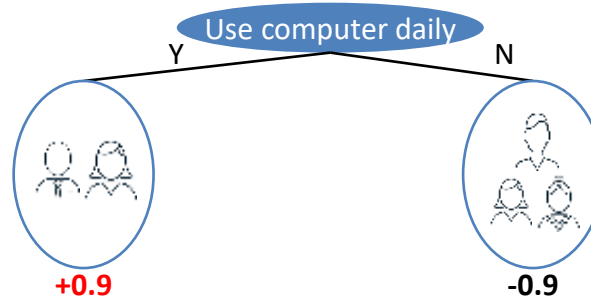
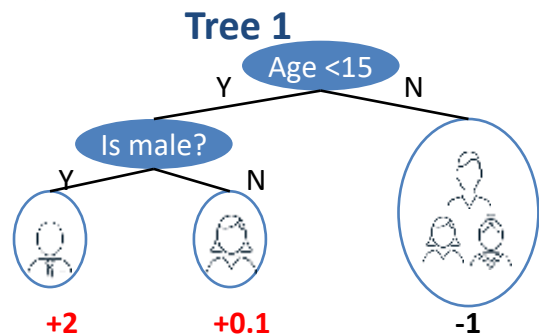
Input: age, gender, occupation, ...

Does the person like computer games



Prediction score in each leaf

Now same prediction can be made by making two separate trees and combining marks of individual scores:



$$F(\text{man}) = 2 + 0.9 = 2.9$$

$$F(\text{woman}) = -1 - 0.9 = -1.9$$

Ensemble model

Prediction of is sum of scores predicted by each of the tree

Performance Metrics

There are 4 main ways to think about measuring “performance”



Accuracy **measures**

How accurate are model outputs compared to actuals



Value

What monetary value does the model generate for the business



Explainability

How easy it is to explain the model, its underlying logic and outputs to business stakeholders



Implementability

How easy is to implement the model and integrate into existing systems and processes

There are
many
ways of
thinking
about
accuracy

Classification

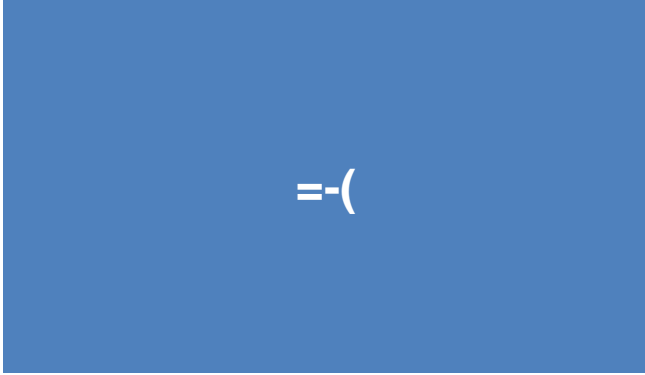
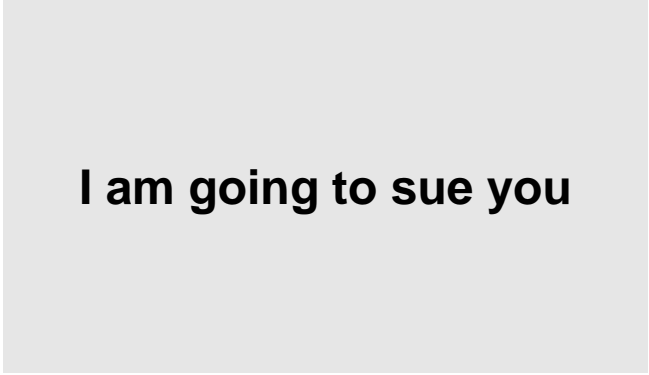
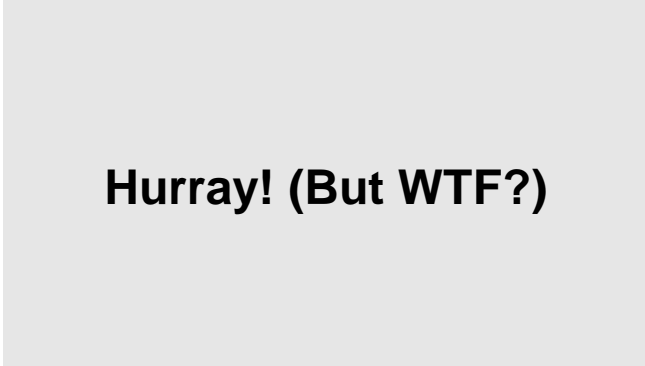

- Basic accuracy
- Confusion matrix metrics
- ROC curves and AUC
- Gini

Details follow

My model has 67% accuracy...
how good is that?

How about 98% accuracy?
Good enough?

Accuracy is not a good measure when errors are costly

		Prediction	
		Cancer	All clear
Actual	Cancer		
	All Clear		

Confusion matrices are a way of capturing different types of errors


		Prediction	
		Positive (e.g. cancer)	Negative (e.g. no cancer)
Actual	Positive	True Positive 30 people	False Negative (I will sue you) 13 people
	Negative	False Positive (Yay but WTF?) 56 people	True Negative 270 people

Different metrics can be derived from the confusion matrix

		Prediction	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)


Accuracy

How often is the model correct?

$$\text{Accuracy} = \frac{TP + TN}{\text{# Observation}}$$


Misclassification rate


How often is the model incorrect?

$$\text{Misclassification rate} = \frac{FP + FN}{\text{# Observation}}$$


True positive rate

What share of **positives** has been **correctly** classified?

Also called: Recall, Specificity

$$\text{True positive rate} = \frac{TP}{\text{Actual positives}}$$



False positive rate

What share of **negatives** has been **incorrectly** classified as positive?

$$\text{False positive rate} = \frac{FP}{\text{Actual negatives}}$$

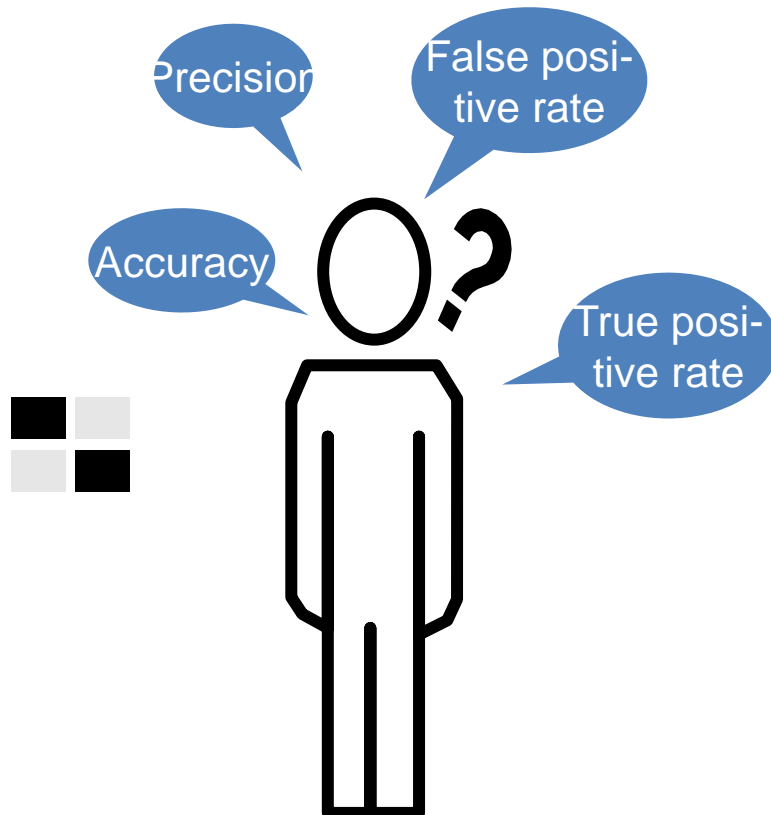

Precision

What share of **positive predictions** are actually positive?

$$\text{Precision} = \frac{TP}{\text{Predicted pos.}}$$


But which metric should we use?

Misclassification rate



It depends...

Which
error
would you
minimise
if you are
predicting
...

...breast cancer occurrence?

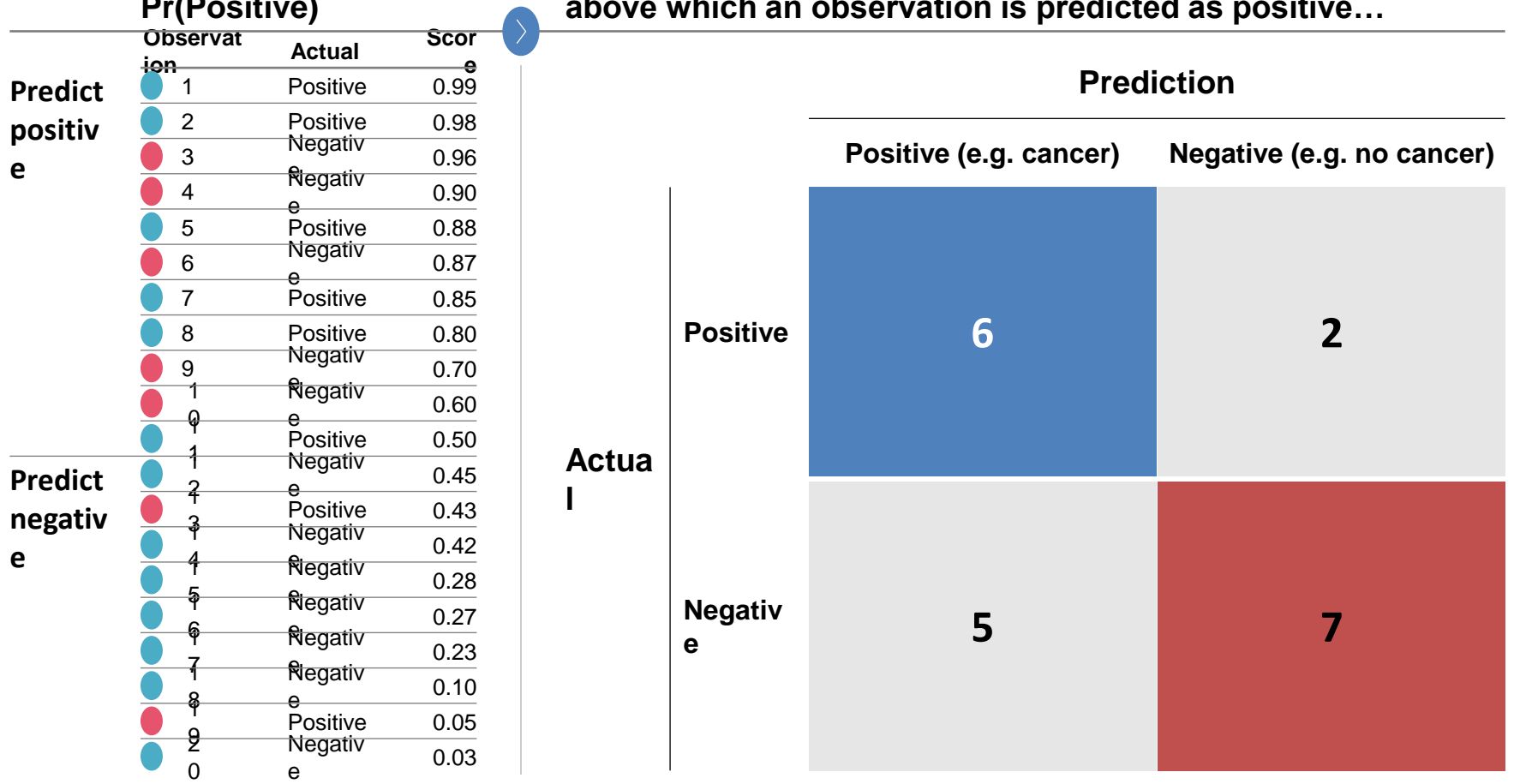
...which bags carry bombs?

...responses to marketing efforts?

Most models predict probability scores rather than a yes/no decision – to create a confusion matrix a probability threshold needs to be chosen

Many models predict a probability score $\text{Pr}(\text{Positive})$

The confusion matrix is then built by picking a threshold above which an observation is predicted as positive...



Thus, the same model can yield very different confusion matrices, depending on the chosen threshold

Many models predict a probability score $\text{Pr}(\text{Positive})$

Observation	Actual	Score
1	Positive	0.99
2	Positive	0.98
3	Negative	0.96
4	Negative	0.90
5	Positive	0.88
6	Negative	0.87
7	Positive	0.85
8	Positive	0.80
9	Negative	0.70
10	Negative	0.60
11	Positive	0.50
12	Negative	0.45
13	Positive	0.43
14	Negative	0.42
15	Negative	0.28
16	Negative	0.27
17	Negative	0.23
18	Negative	0.10
19	Positive	0.05
20	Negative	0.03

Threshold: $\text{Pr}(\text{Positive}) \geq 0.8$

5	3	False positive rate	25%
3	9	True positive rate	63%
		Accuracy	70%

Threshold: $\text{Pr}(\text{Positive}) \geq 0.5$

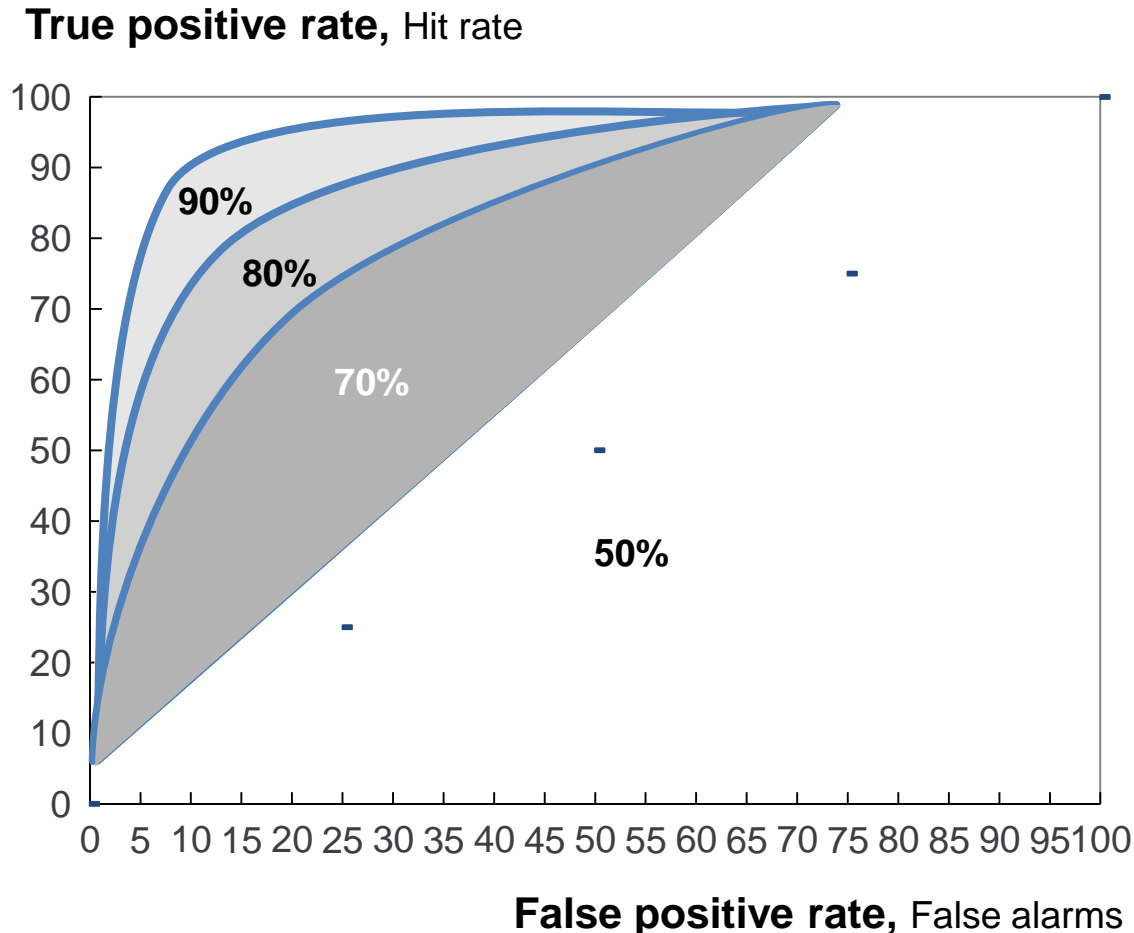
6	2	False positive rate	42%
5	7	True positive rate	75%
		Accuracy	65%

Threshold: $\text{Pr}(\text{Positive}) \geq 0.2$

7	1	False positive rate	83%
10	2	True positive rate	88%
		Accuracy	45%

ROC curves and their related metrics

AUC/Gini are a common method for data scientists to compare across models



ROC curves

- 45 degree line represents random guessing
- Top left corner represents perfect information

Area under the curve (AUC)

- Measure of the explanatory power of the model (generally, higher is better)
- If AUC = 50%, random guessing would have fared as well as the model

Gini coefficient

- $GINI = 2 * AUC - 1$
- If GINI = 0, random guessing would have fared as well as the model

Value is usually a more important measure of model performance

		Prediction			
		Positive	Negative	Value of each outcome	
Actual	Positive	Number of true positives	Number of false negatives	\$ of true positive	\$ false negative
	Negative	Number of false positives	Number of true negatives	\$ false positive	\$ true negative

×

= (Expected) value of the model

