University of Exeter

Predictive Modelling

Week 05-BEM2031

Term2: 2024/25

Today:

- What is random?
- What is predictive modelling?
- Supervised learning
  - Regression
  - Classification
- A decision tree step-by-step

Types of Analytics:

- Descriptive Analytics: WHAT happened (or is happening)?
- Diagnostic Analytics: WHY did it happen?
- Predictive Analytics: WHAT is likely to happen in the future?
- Prescriptive Analytics: WHAT can we do about it?

What do we mean by random?

University of Exeter

List of solar eclipses in the 21st century - Wikipedia

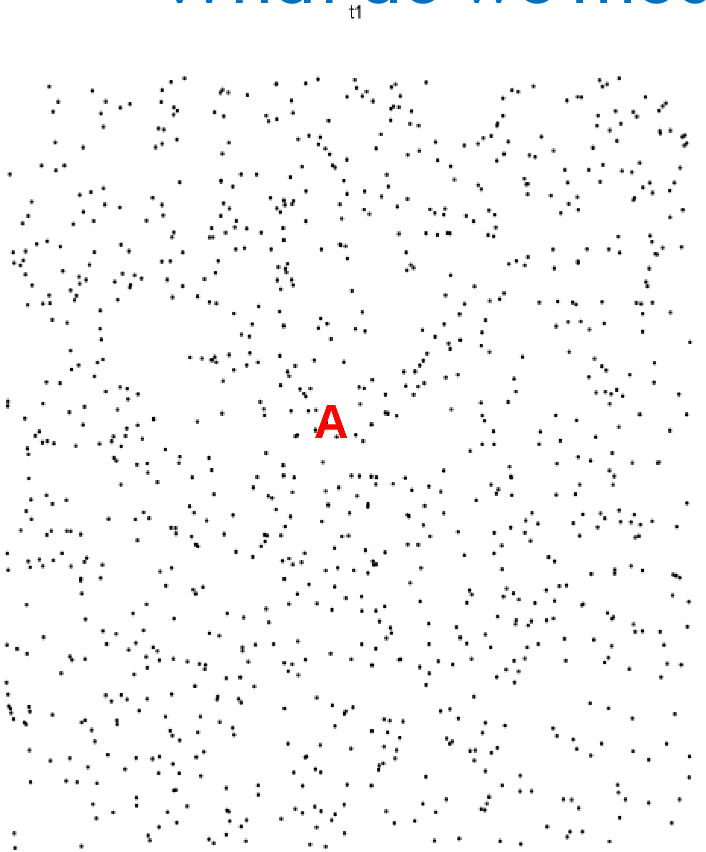
Date	Time	Saros	Type	Magnitude		Duration	Location	Path width		Geography
September 4, 2100	08:49:20	146	Total	−0.3384	1.0402	3:32	10.5°S 39.0°E	142	88	Congo, Rwanda, Uganda, Burundi, Tanzania, Mozambique, Madagascar **Partial:** Africa,

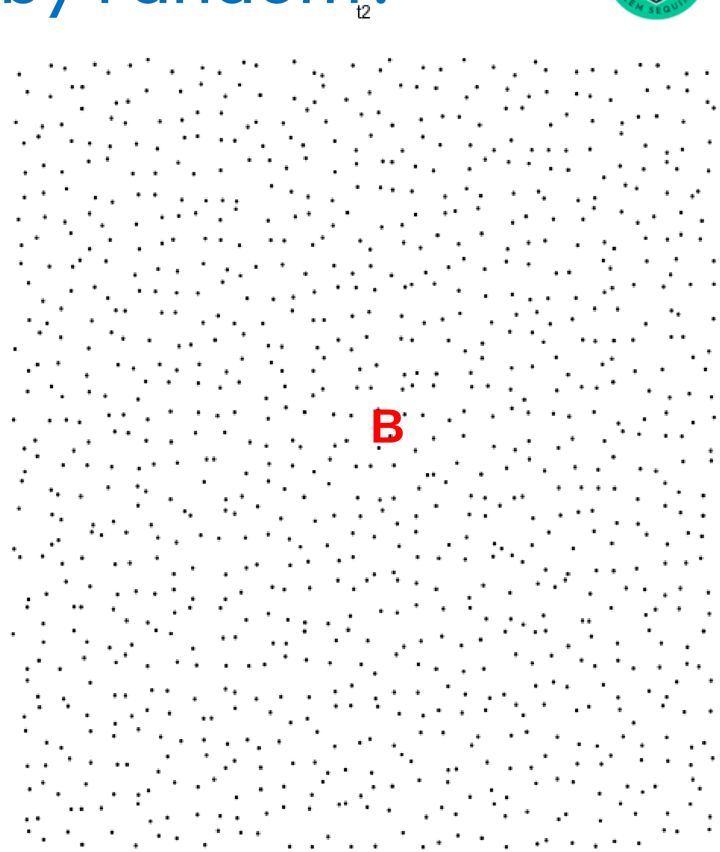
What do we mean by random?

Random in the context of prediction models means that some aspect of the model is determined by chance rather than by design.

Randomness is also a source of uncertainty and error. Random phenomena are difficult to measure, predict, and control.

What do we mean by random?





Which of these could be modelled and which is completely random?

What do we mean by random?

- Create these random fields of points
- Random coin throw, random dice

MODULE MATERIALS

Lecture slides >

Workshop output files >

Workshop interactive code ⌄

Week 1 practice

Week 2 practice

Week 3 practice

Week 4 practice

Week 5 practice

Random and regular fields of

The random field of points from Lecture 5. d1

distribution:

R Code Start Over Run Code

```r
N <- 1024
d1 <- tibble(x = runif(N), y = runif(N), type = 't1')
```

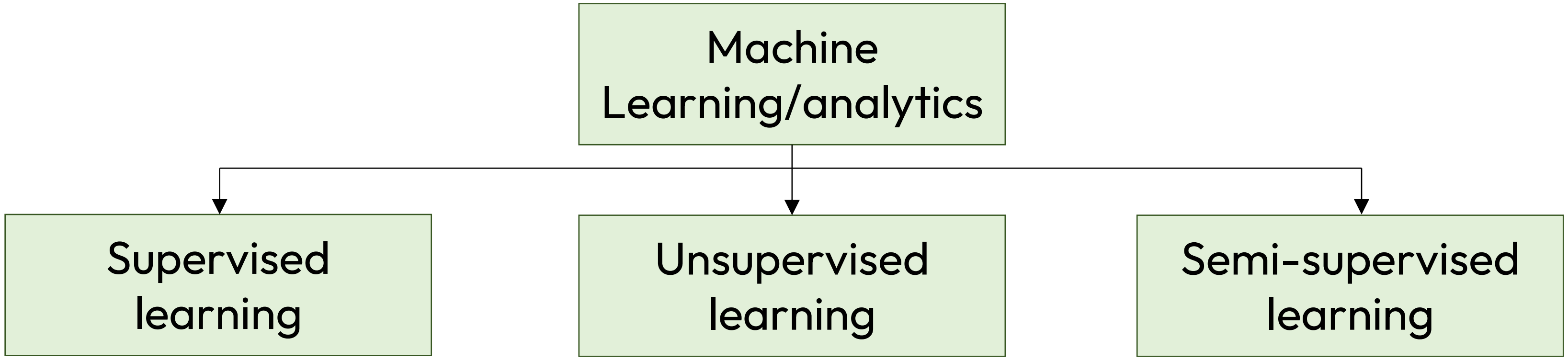
To create the evenly distributed bit (d2), you have to start with an even grid of points:

R Code Start Over Run Code

```r
p <- seq(0,1, length.out = N / 32)

d2 <- expand.grid(p, p) %>%
  data.frame %>%
  rename(x = Var1, y = Var2) %>%
  mutate(type = 't2')
```

Supervised vs Unsupervised methods



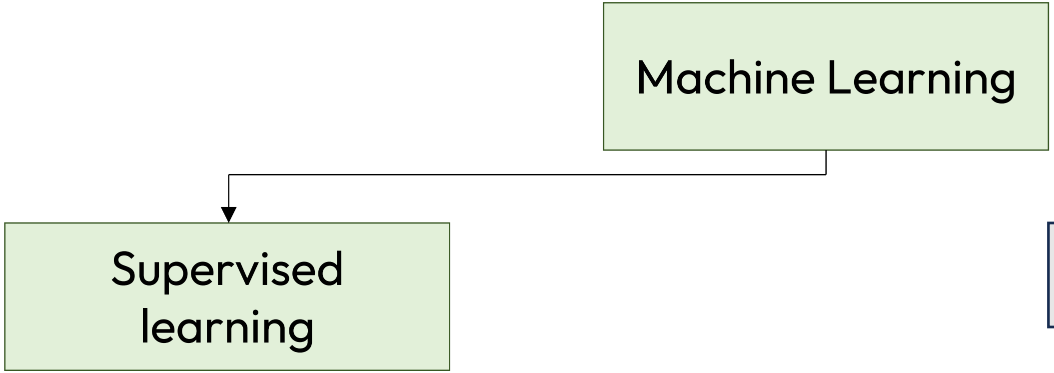
Supervised methods have a target, an objective.

"Can we find groups of customers who have particularly high likelihoods of cancelling their service soon after their contracts expire?"

Unsupervised methods have no specific target.

"Do our customers naturally fall into different groups?"

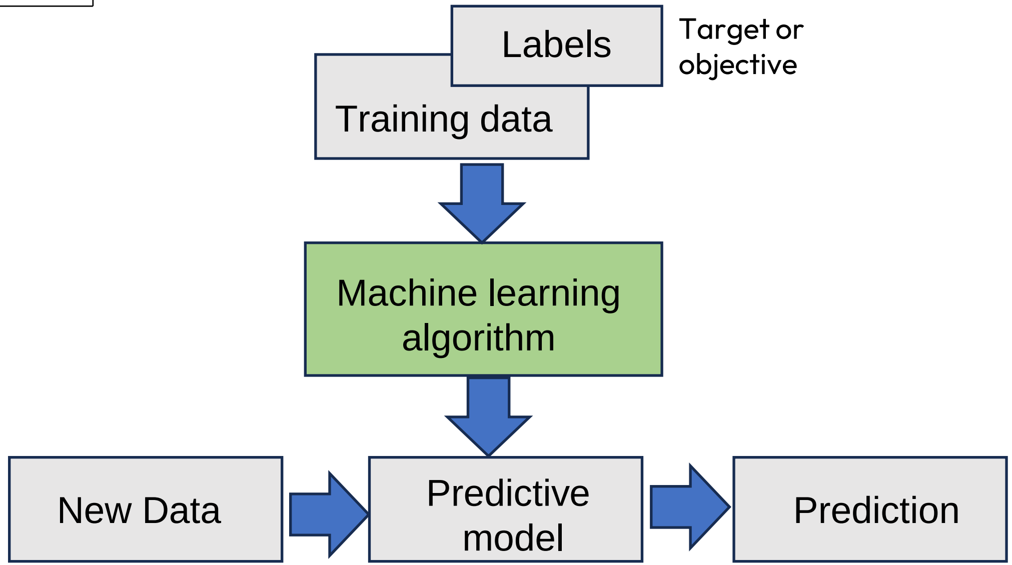
Supervised vs Unsupervised methods



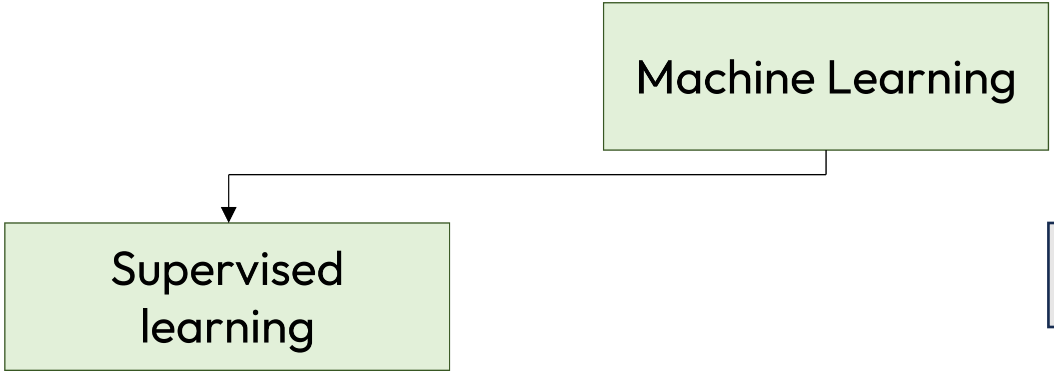
Supervised methods have a target, an objective.

“Can we find groups of **customers** who have particularly high likelihoods of **cancelling their service** soon after their contracts expire?

“Can we predict which **patients** are likely to be **readmitted to hospital** following a previous admission?



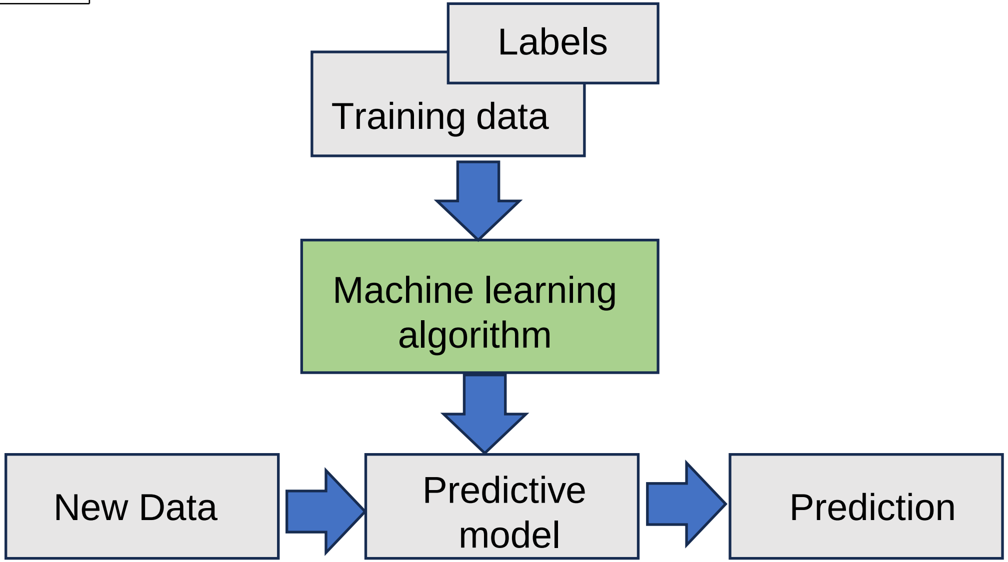
Supervised vs Unsupervised methods



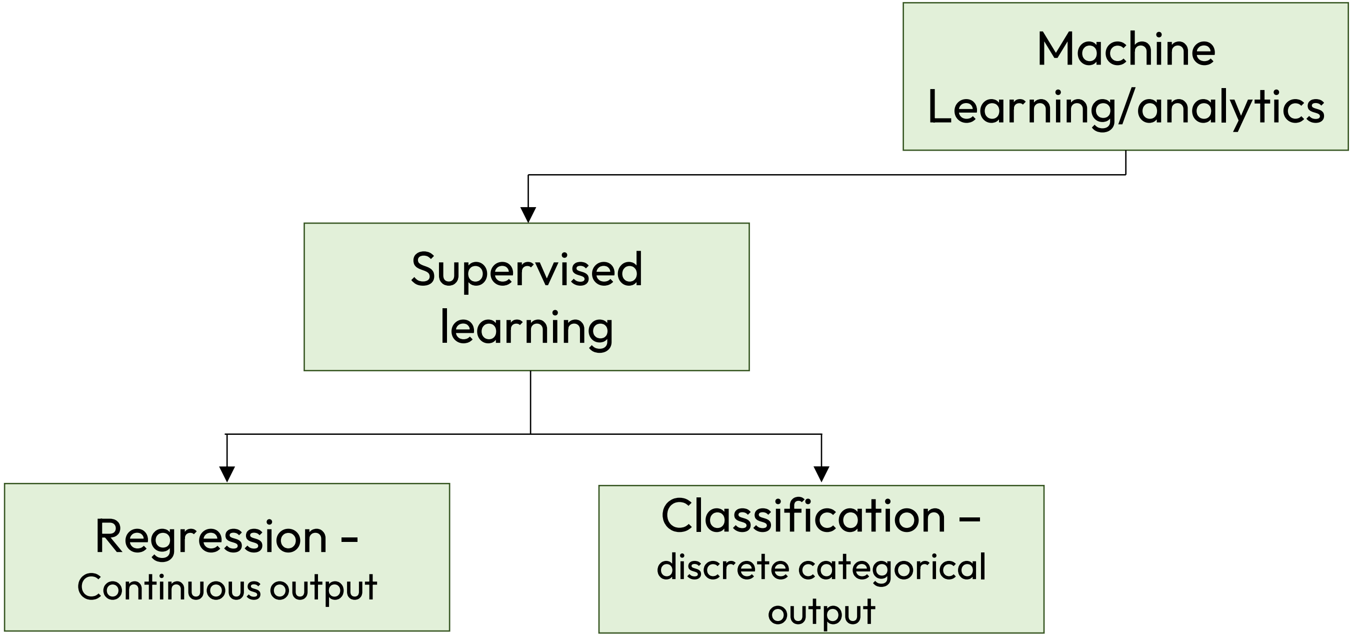
Label: Variable we are predicting (the dependent variable) (y)

Features: the attributes or variables which describe the data (independent variables) $\{x_1, x_2, \ldots, x_N\}$

Unseen (new) data: test the performance of the model $y=f(x)$ on unlabelled data

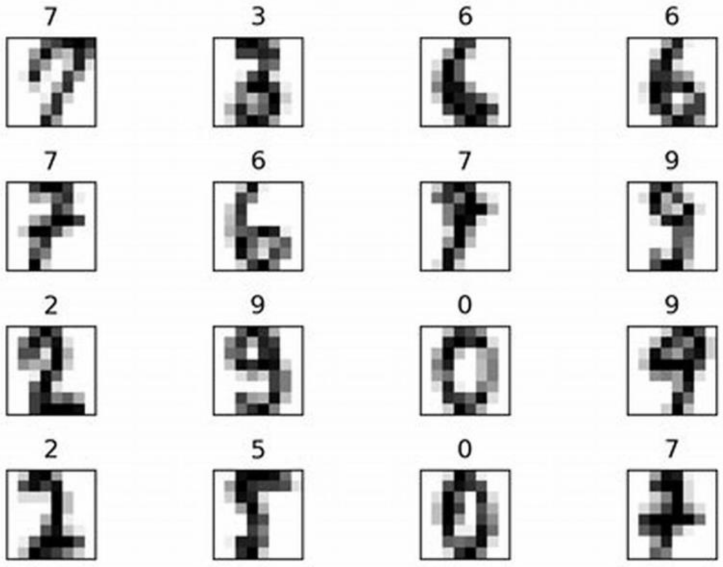


Types of supervised learning

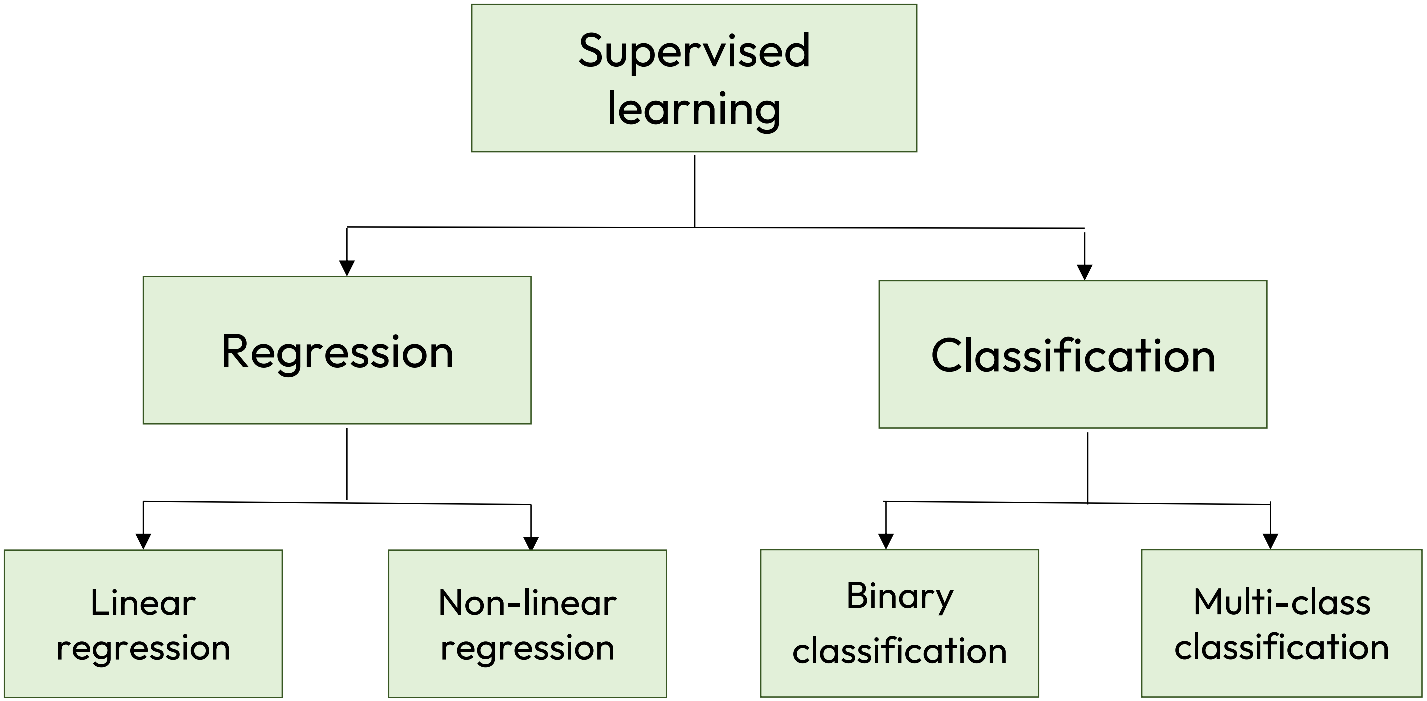


Regression or classification?

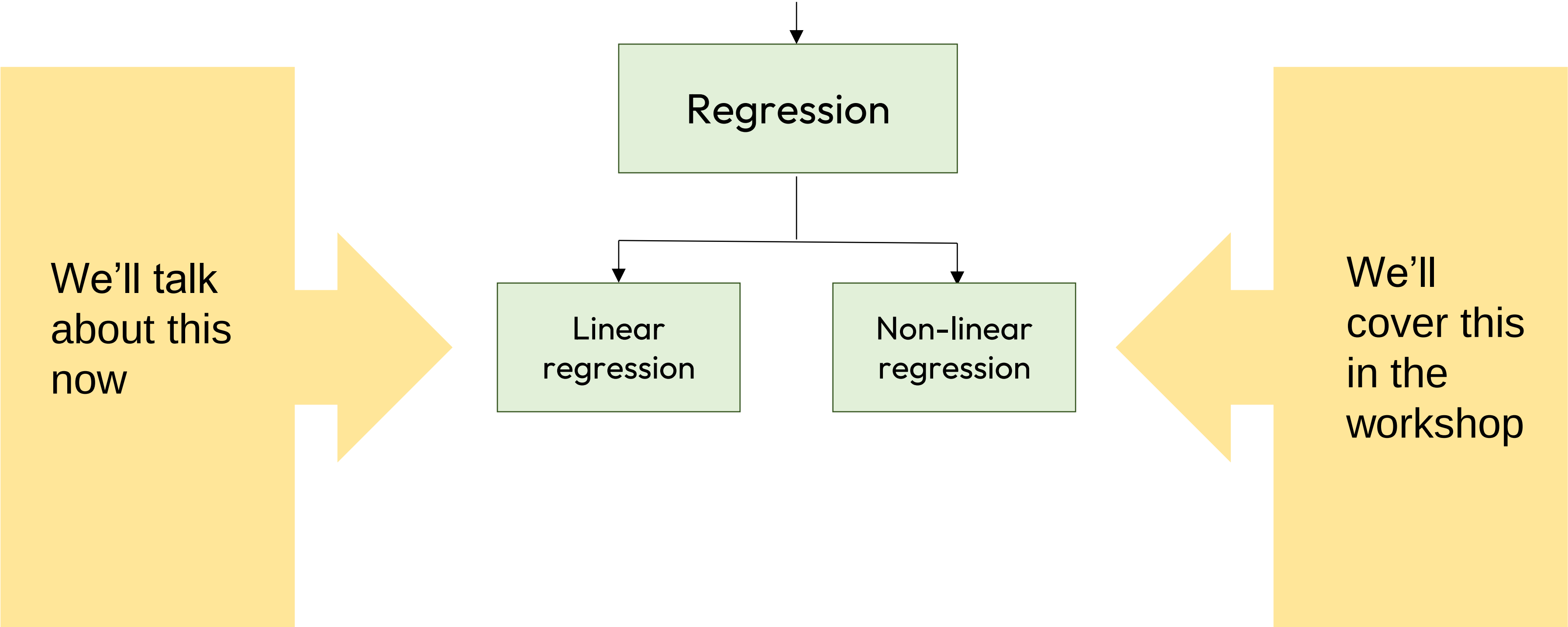
- Weather prediction
- Identification of cancer cells
- Identification of handwritten digits
- Oil price prediction
- Identification of fraudulent credit card transactions
- Monthly income prediction



Machine Learning



Regression modelling:



Linear Regression: House Price Prediction

Given a set of input features (which may influence the price of a house), the goal of the algorithm is to predict the price of a new house going to market

House Price Prediction

Square footage	Num of Rooms	Garden?	Parking Facility?	Num of floors	Price ($) in 1000's
..	3	Yes	..	2	460
1700	..	No	No	1	320
..	5	..	..	..	..
..	..	..	..	..	..
..	..	..	..	..	..
..	..	..	..	..	..

Labeled Exampl

Attributes/Features/Independent Variables ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$)

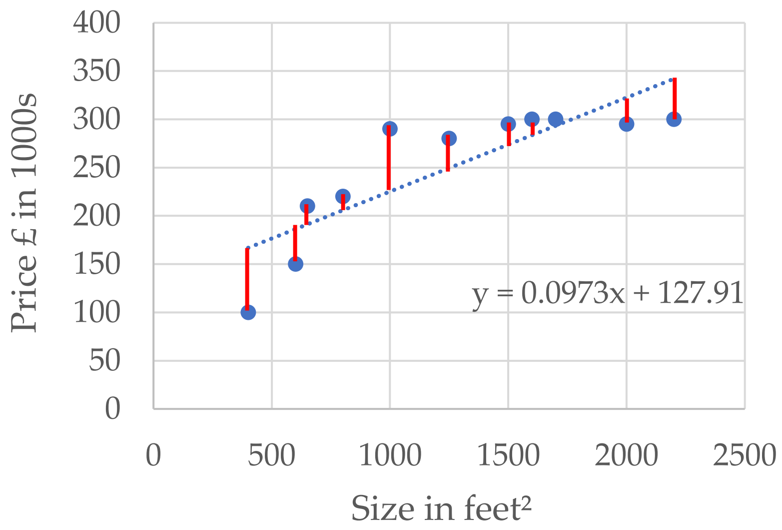
Target/Dependent Variable (y)

Simple Linear Regression

Consider the problem of **predicting house prices (y)**

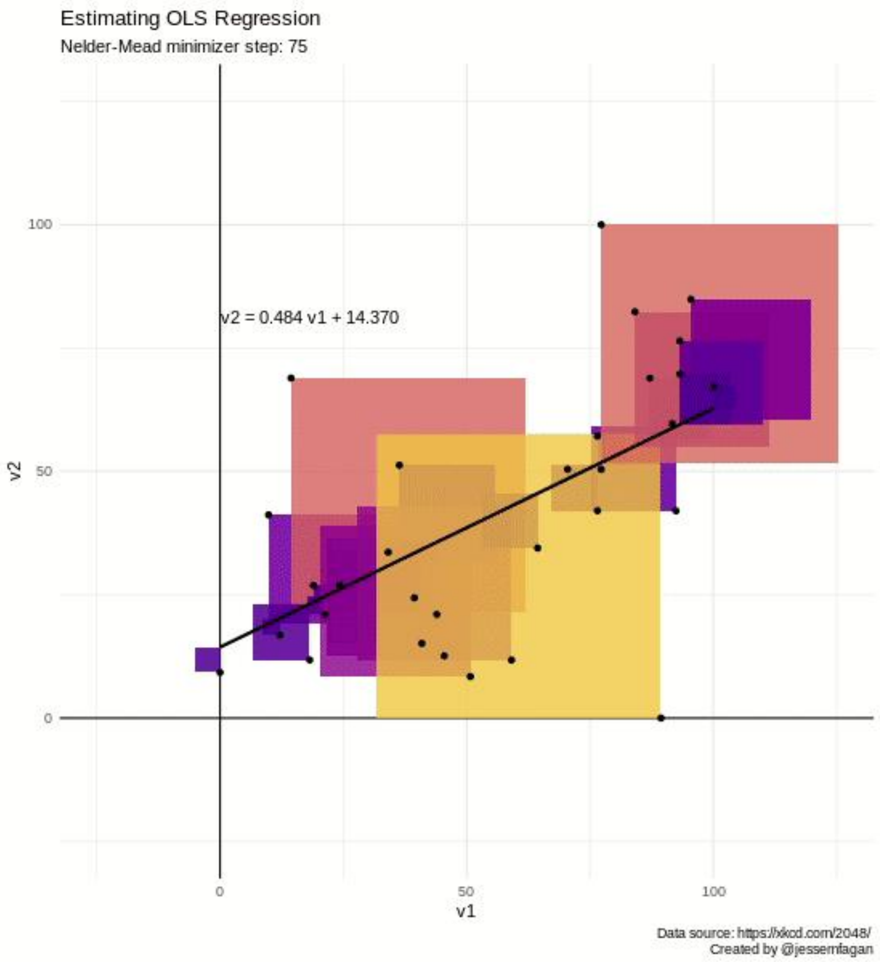
- **Feature Selection** – input variables that can used to predict house prices
  – let's consider one input variable (size in sq.ft) → univariate/simple regression

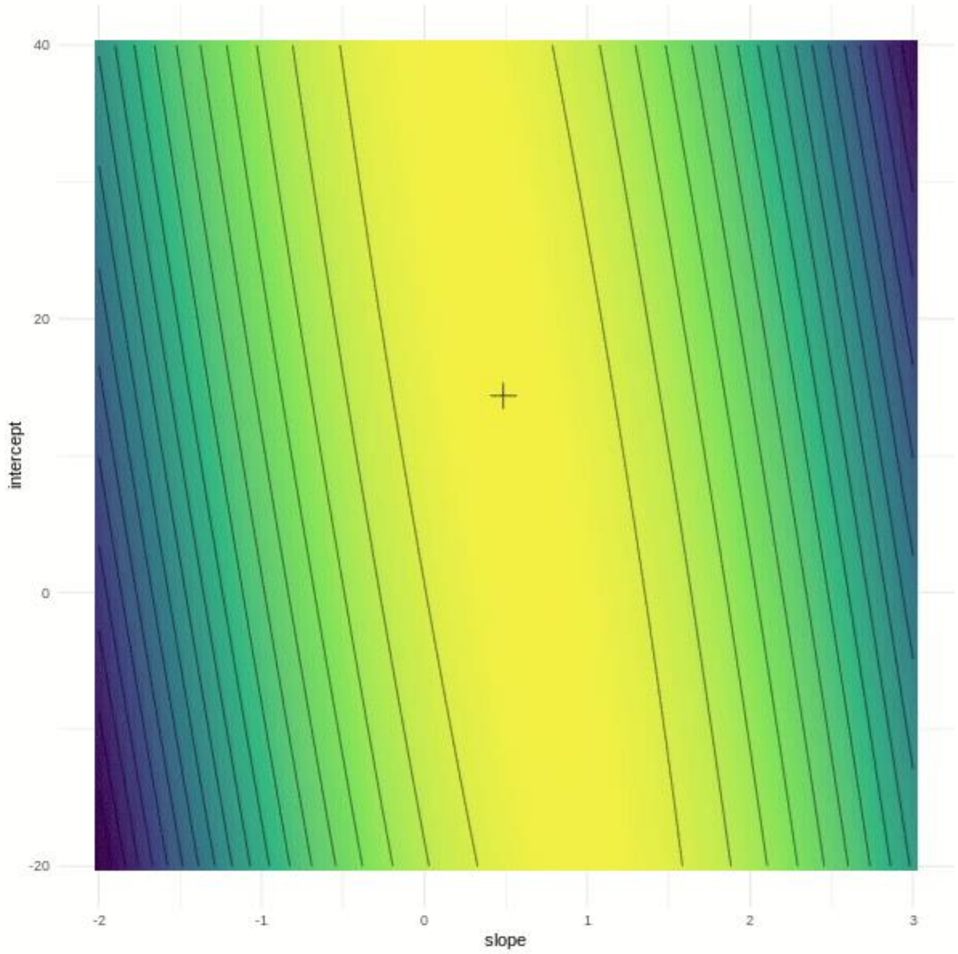
- Simple linear regression finds a linear function (straight line) that predicts the target variable (y) as a function of the features or independent variables (x)
- $y = mx + c$ where m is the slope and c is the intercept



Features/independent variables (x)	Target/dependent variable (y)
Size in feet²	Price £ in 1000s
400	100
600	150
650	210
800	220
1000	290
1250	280
1500	295
1600	300
1700	300
2000	295
2200	300

[Supervised Learning in Machine Learning: Regression and Classification (DeepLearning.AI) | Coursera]


Estimating OLS Regression
Nelder-Mead minimizer step: 75
v2 = 0.484 v1 + 14.370
v2
v1
100
50
0
0
50
100
Data source: https://xkcd.com/2048/
Created by @jessemfagan

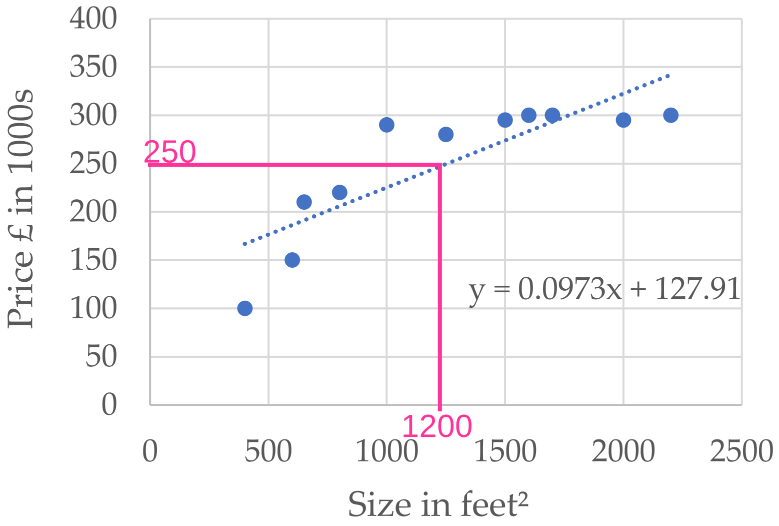

intercept
slope
40
20
0
-20
-2
-1
0
1
2
3

Simple Linear Regression

Consider the problem of **predicting house prices (y)**

- **Feature Selection** – input variables that can used to predict house prices
  – let's consider one input variable (size in sq.ft) → univariate/simple regression

- Simple linear regression finds a linear function (straight line) that predicts the target variable (*y*) as a function of the features or independent variables (*x*)
- $y = mx + c$ where m is the slope and c is the intercept



Features/independent variables (x) Size in feet²	Target/dependent variable (y) Price £ in 1000s
400	100
600	150
650	210
800	220
1000	290
1250	280
1500	295
1600	300
1700	300
2000	295
2200	300

Supervised Learning in Machine Learning: Regression and Classification (DeepLearning.AI) | Coursera

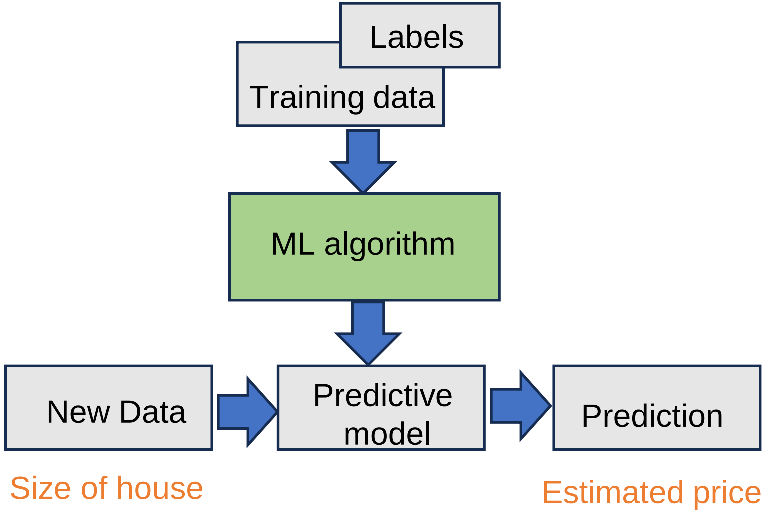
Simple Linear Regression

University of Exeter

Consider the problem of **predicting house prices (y)**

- **Feature Selection** – input variables that can used to predict house prices
  - let's consider one input variable (size in sq.ft) → univariate/simple regression

- Simple linear regression finds a linear function (straight line) that predicts the target variable (*y*) as a function of the features or independent variables (*x*)
- $y = mx + c$ where m is the slope and c is the intercept



Features/independent variables (y) Size in feet²	Target/dependent variable (x) Price £ in 1000s
400	100
600	150
650	210
800	220
1000	290
1250	280
1500	295
1600	300
1700	300
2000	295
2200	300

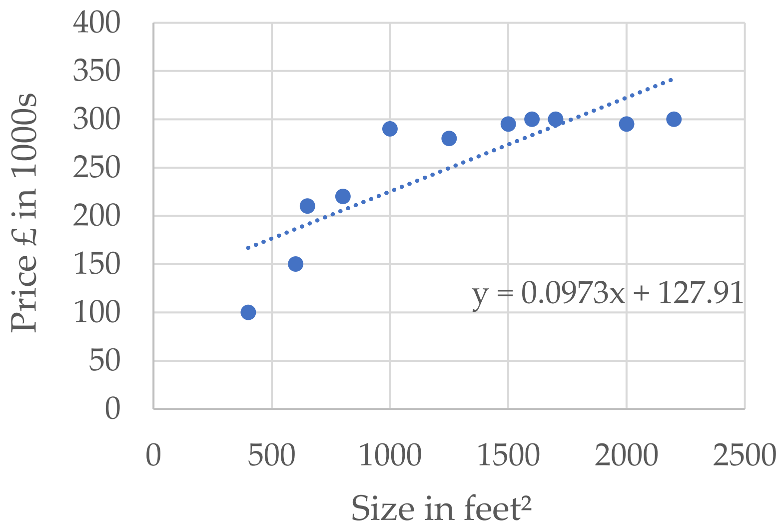
Supervised Learning in Machine Learning: Regression and Classification (DeepLearning.AI) | Coursera

Simple Linear Regression

Consider the problem of **predicting house prices (y)**

- **Feature Selection** – input variables that can used to predict house prices
  – let's consider one input variable (size in sq.ft) → univariate/simple regression

- Simple linear regression finds a linear function (straight line) that predicts the target variable (y) as a function of the features or independent variables (x)
- $y = mx + c$ where m is the slope and c is the intercept



Features/independent variables (x) Size in feet²	Target/dependent variable (y) Price £ in 1000s
400	100
600	150
650	210
800	220
1000	290
1250	280
1500	295
1600	300
1700	300
2000	295
2200	300

Supervised Learning in Machine Learning: Regression and Classification (DeepLearning.AI) | Coursera

Multiple Linear Regression

Consider the problem of **predicting house prices (y)**

- **Feature Selection** – input variables that can used to predict house prices
  - let's consider multiple input variable → multiple linear regression

- Multiple linear regression models a linear function that predicts a target variable as a function of the independent variables: $y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + ...$

Square footage	Num of Rooms	Garden?	Parking Facility?	Num of floors	Price ($) in 1000's
..	3	Yes	..	2	460
1700	..	No	No	1	320
..	5	..	..	..	..
..	..	..	..	..	..
..	..	..	..	..	..
..	..	..	..	..	..

m training examples

Attributes/Features/Independent Variables ($\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}$)

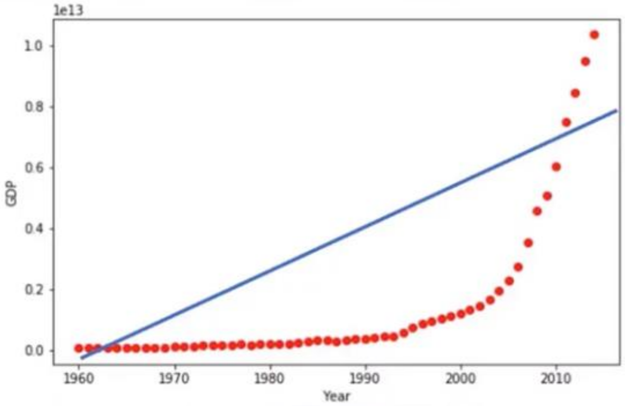
Target/Dependent Variable (y)

[Supervised Learning in Machine Learning: Regression and Classification (DeepLearning.AI) | Coursera]

Problems faced: Underfitting

Should we use linear regression?

	Year	Value
0	1960	5.918412e+10
1	1961	4.955705e+10
2	1962	4.668518e+10
3	1963	5.009730e+10
4	1964	5.906225e+10
5	1965	6.970915e+10
6	1966	7.587943e+10
7	1967	7.205703e+10
8	1968	6.999350e+10
9	1969	7.871882e+10
...	.....	....



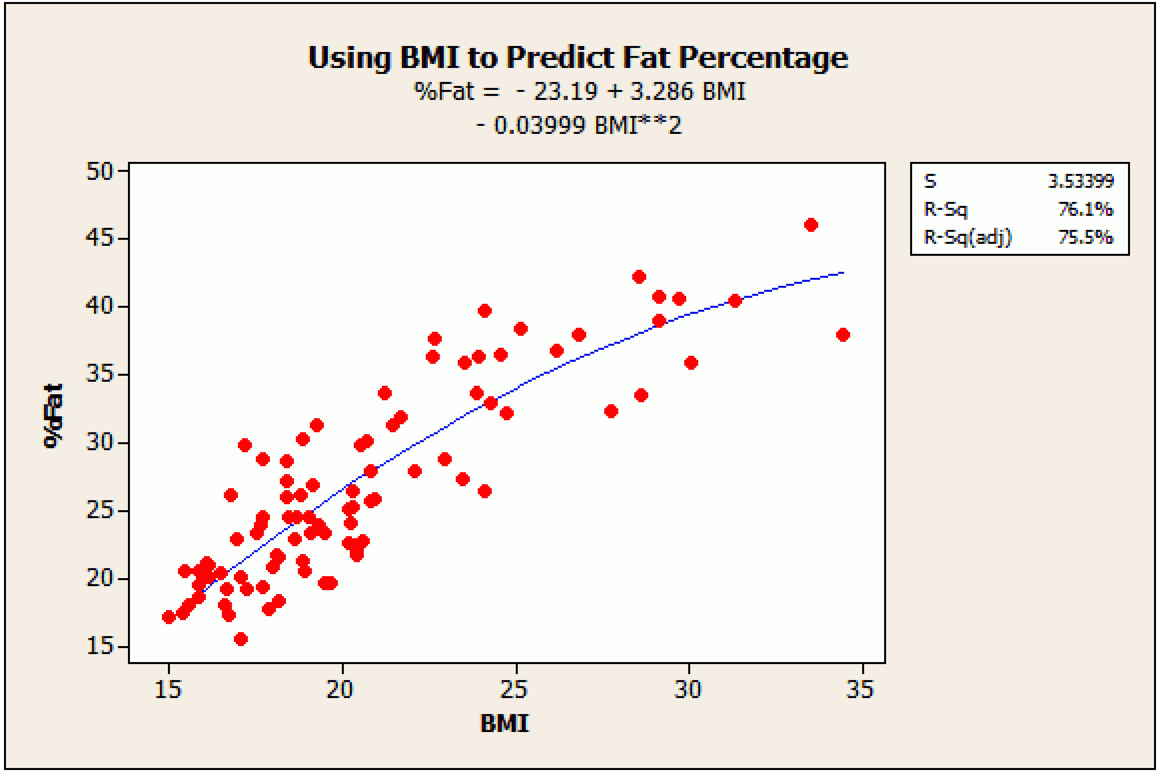
These data points correspond to China's gross domestic product (GDP) from 1960–2014.

Model is not complex enough to capture the underlying patterns in the data.

Leads to **bias**:

The amount of error introduced by approximating real-world phenomena in a simplified model

Non-linear regression



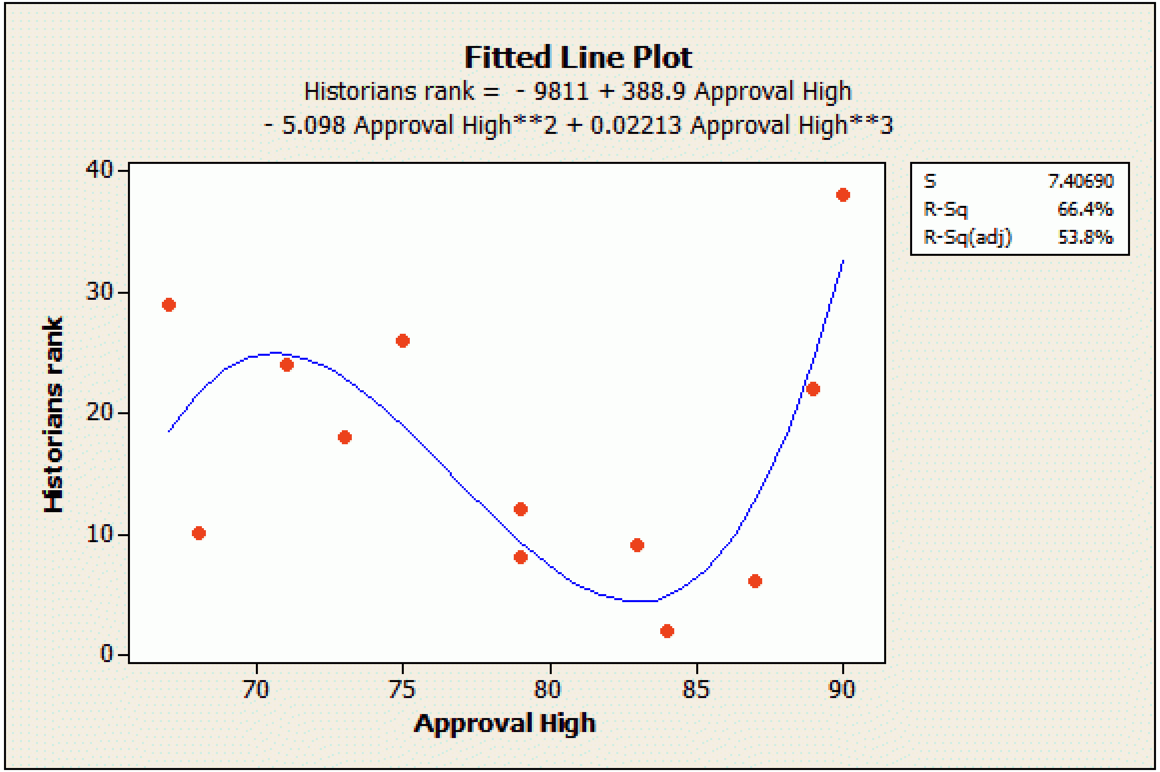
If a regression equation doesn't follow the rules for a linear model, then it must be a nonlinear model.

The regression example models the relationship between body mass index (BMI) and body fat percent.

It is a linear model that uses a quadratic (squared) term to model the curved relationship.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2$$

Problems faced: Overfitting



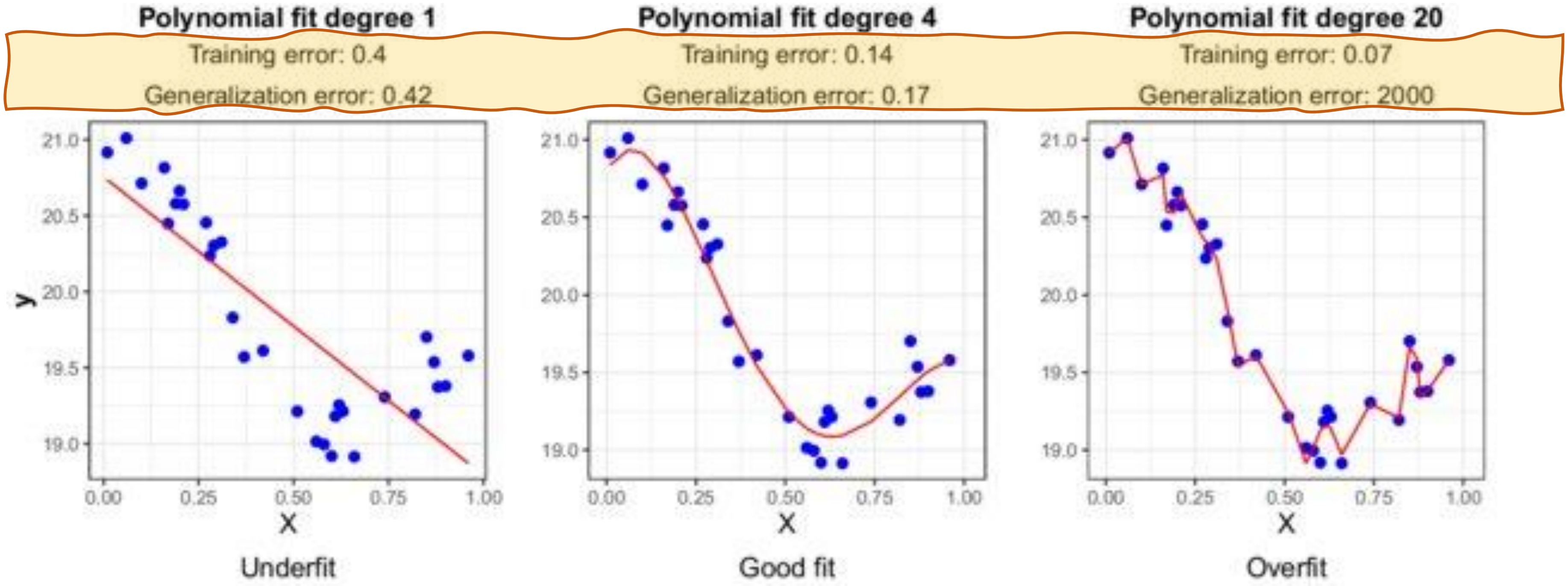
If you try to estimate too many parameters, you will overfit!

The size of your dataset restricts the number of terms you can safely add to your model

If your study calls for a complex model, you must collect a relatively large sample size.

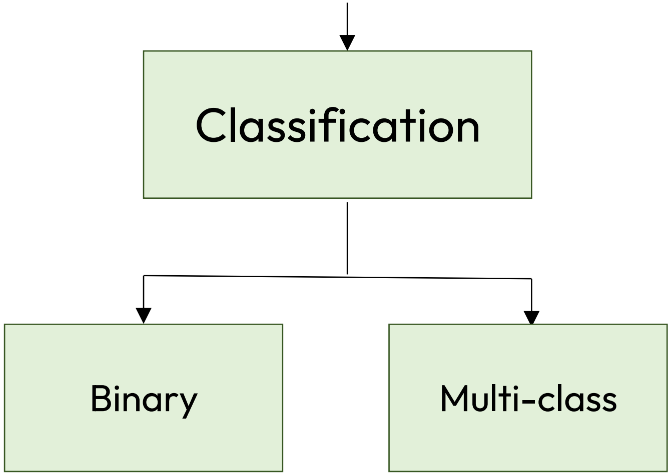
[The Difference between Linear and Nonlinear Regression Models - Statistics By Jim]

Problems faced: Overfitting



Badillo, S., Banfai, B., Birzele, F., Davydov, I.I., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B. and Zhang, J.D. (2020), An Introduction to Machine Learning. Clin. Pharmacol. Ther., 107: 871- 885 https://doi.org/10.1002/cpt.1796

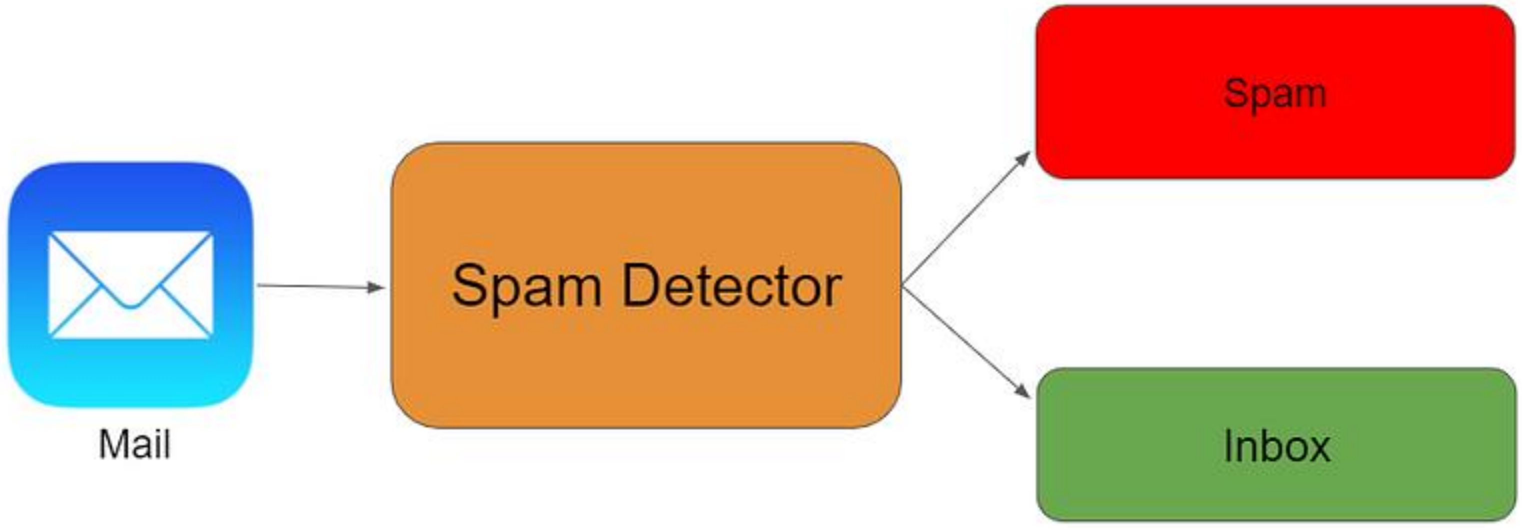
Classification modelling:



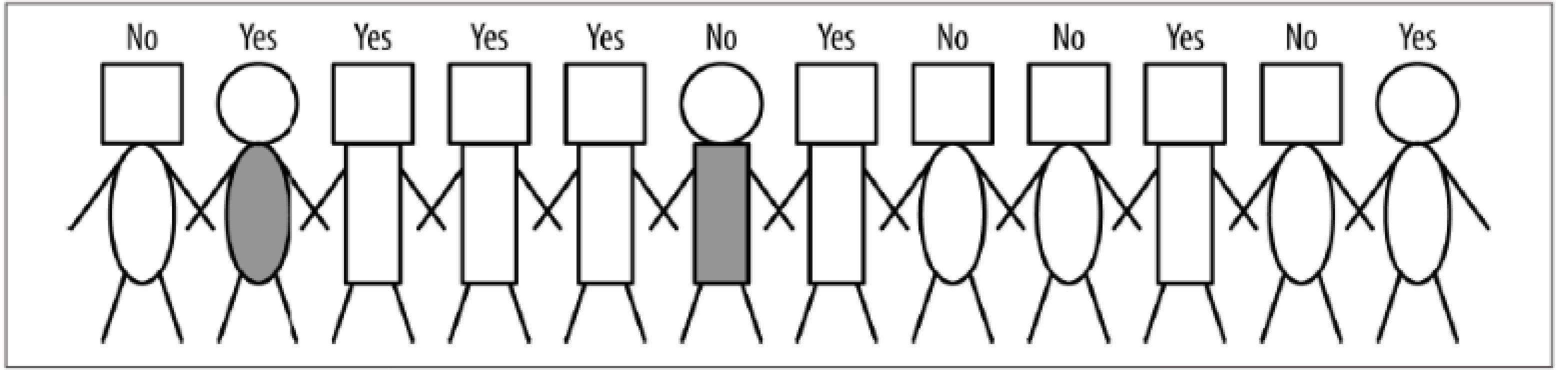
University of Exeter

Binary Classification: Email spam prediction

Trained on a large number of spam and non-spam emails, the algorithm’s goal is to predict whether or not an email is spam



Classification



Attributes:
Head shape: square, circle
Body shape: rectangle, oval
Body colour: grey, white

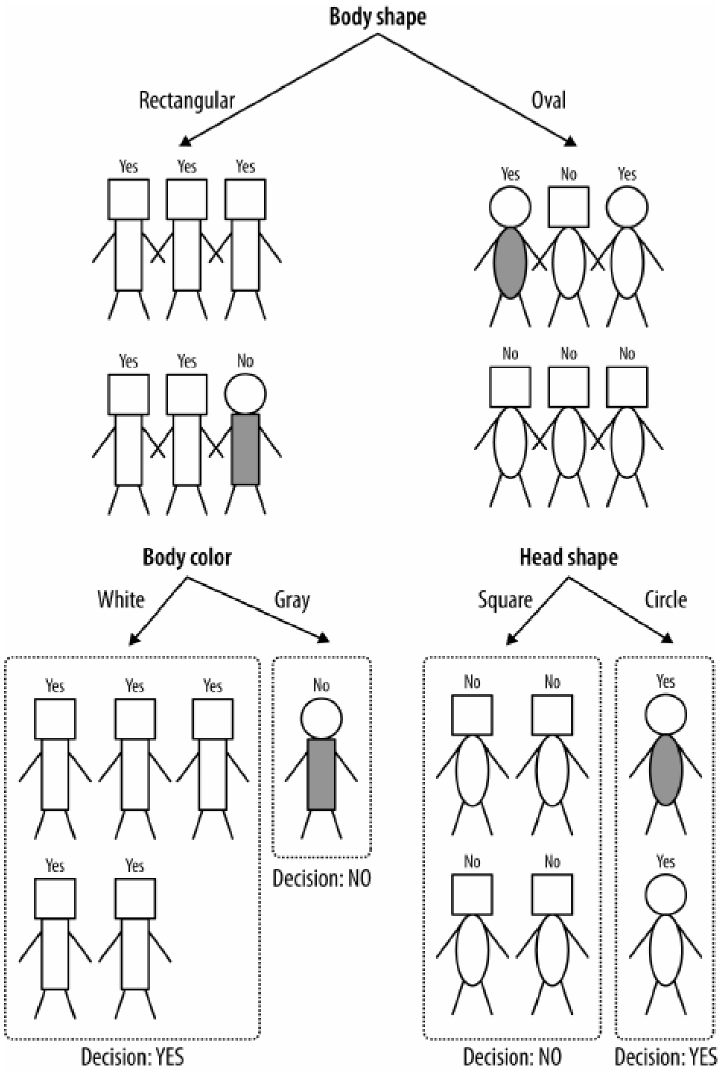
Target:
Write-off: yes, no

Attributes rarely split a group perfectly

Not all attributes are binary

How do we segment for numeric values?

Classification: Decision trees

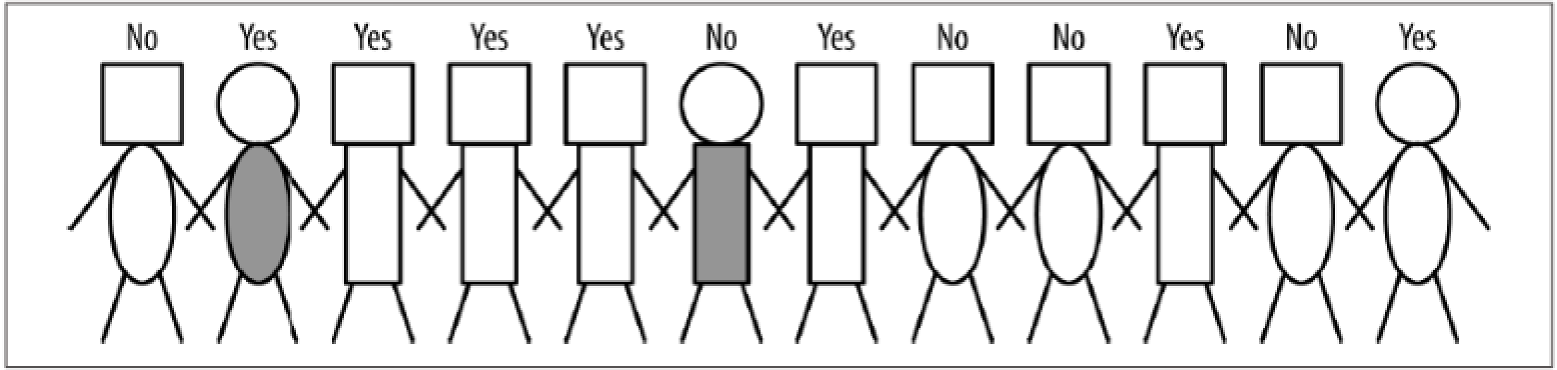


A decision tree is a model with a number of branching options that lead to a decision at the end.

Each point on the tree is called a **node**. The **depth** of the of the tree is maximum number of steps to reach a decision.

A **leaf** of the tree is where the decision is made (when there's no more splitting).

Classification – decision trees



Entropy:
The amount of uncertainty or randomness in a system

→ How *impure* a node (how mixed the training data assigned to that node is)

Information gain:
The reduction in entropy or uncertainty after a dataset is split based on a feature.

→ Helps the algorithm decide which feature to split on at each step. Features with the highest information gain are selected because they reduce uncertainty the most.

Features that result in a higher information gain are considered more important – as they provide more information

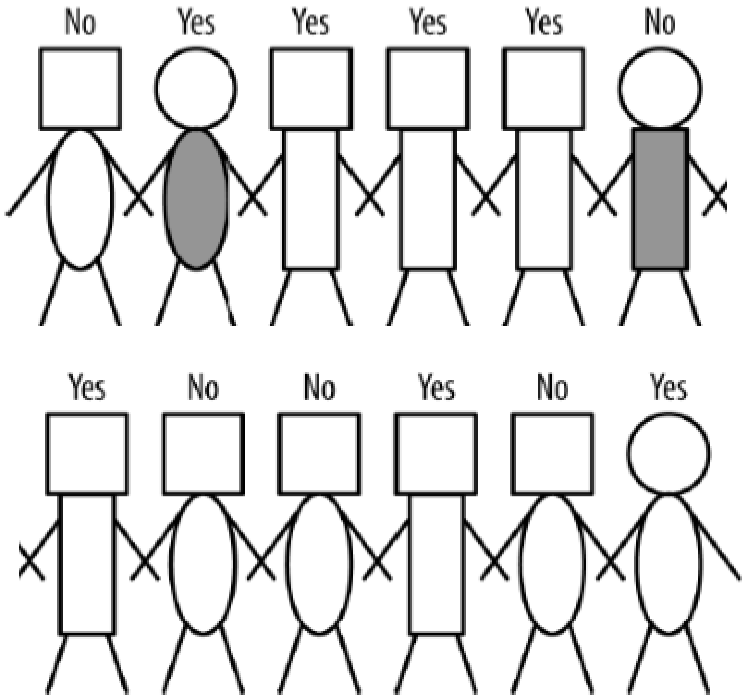
Classification - entropy

An information gain is how much an attribute improves (or decreases) **entropy** (uncertainty) of the model prediction.

$$entropy = -p_1 \log(p_1) - p_2 \log(p_2) - \ \ldots$$

p_yes = 7/12
p_no = 5/12

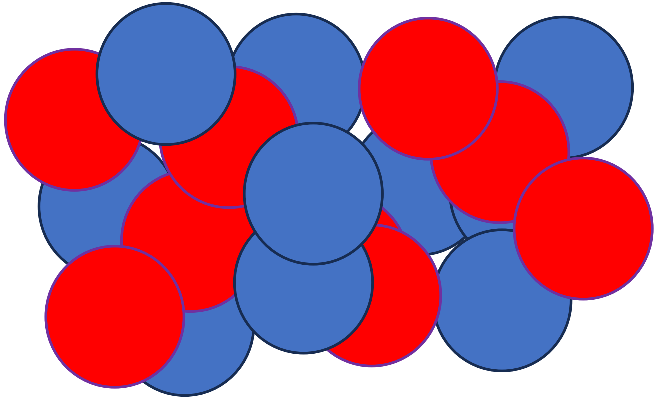
$$entropy(S) = -[(\frac{7}{12}) \times log_2\left(\frac{7}{12}\right) + \left(\frac{5}{12}\right) \times log_2(\frac{5}{12})]$$
$$= 0.98$$



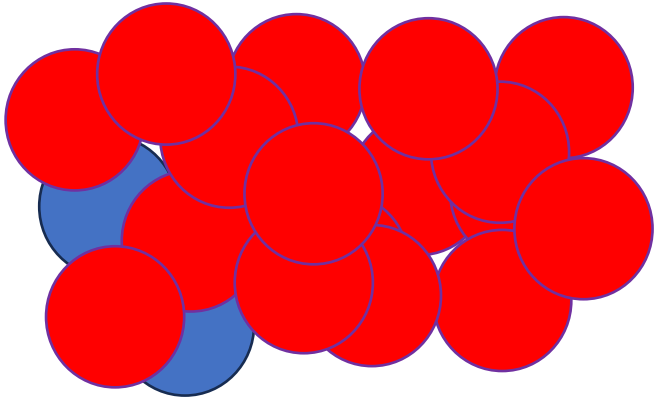
Classification - entropy

Pick a ball at random and guess the colour. The chances of being right or wrong depends on the mix of colours.

If your bag has an equal number of red and blue balls, your uncertainty is highest – this is a state of high entropy.



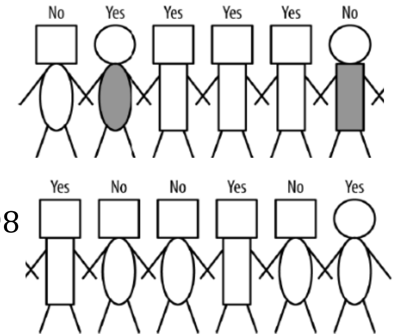
If the bag has mostly red balls, and not many blue balls, you'd probably guess red, and you would be right most of the time. This is a state of low entropy.



Entropy quantifies the uncertainty.

$p\_yes = 7/12$
$p\_no = 5/12$

$entropy = 0.98$

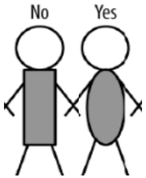


Body colour

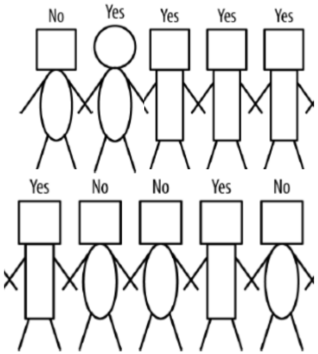
Grey

$p\_yes = 1/2$
$p\_no = 1/2$

$entropy = 1.0$



White



$p\_yes = 6/10$
$p\_no = 4/10$

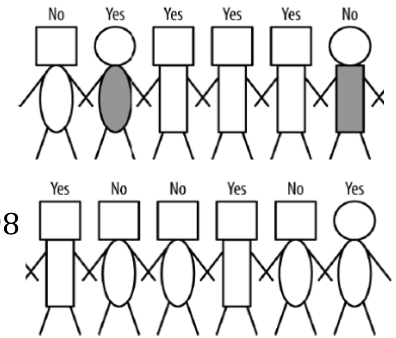
$entropy = 0.97$

$$IG(parent, children) = entropy(parent) - p(c_1) \times entropy(c_1) + p(c_2) \times entropy(c_2) + \ldots$$

$$IG = 0.98 - (0.17 \times 1.0 + 0.83 \times 0.97)$$
$$= 0.005$$

$p\_yes = 7/12$
$p\_no = 5/12$

$entropy = 0.98$



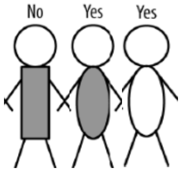
Head Shape

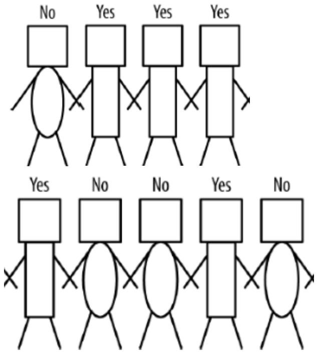
Circle

Square

$p\_yes = 2/3$
$p\_no = 1/3$

$entropy = 0.92$





$p\_yes = 5/9$
$p\_no = 4/9$

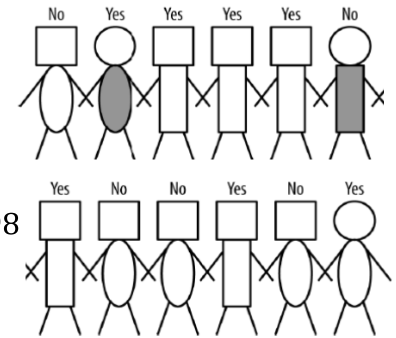
$entropy = 0.99$

$$IG = entropy(base) - p(c_1) \times entropy(c_1) + p(c_2) \times entropy(c_2) + \ \ldots$$

$$\mathrm{IG} = 0.98 - (0.25 \times 0.92 + 0.75 \times 0.99)$$
$$= 0.0075$$

$p\_yes = 7/12$
$p\_no = 5/12$

$entropy = 0.98$



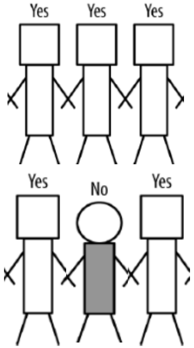
Body Shape

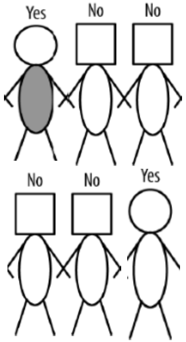
Rectangle

Oval

$p\_yes = 5/6$
$p\_no = 1/6$

$entropy = 0.650$





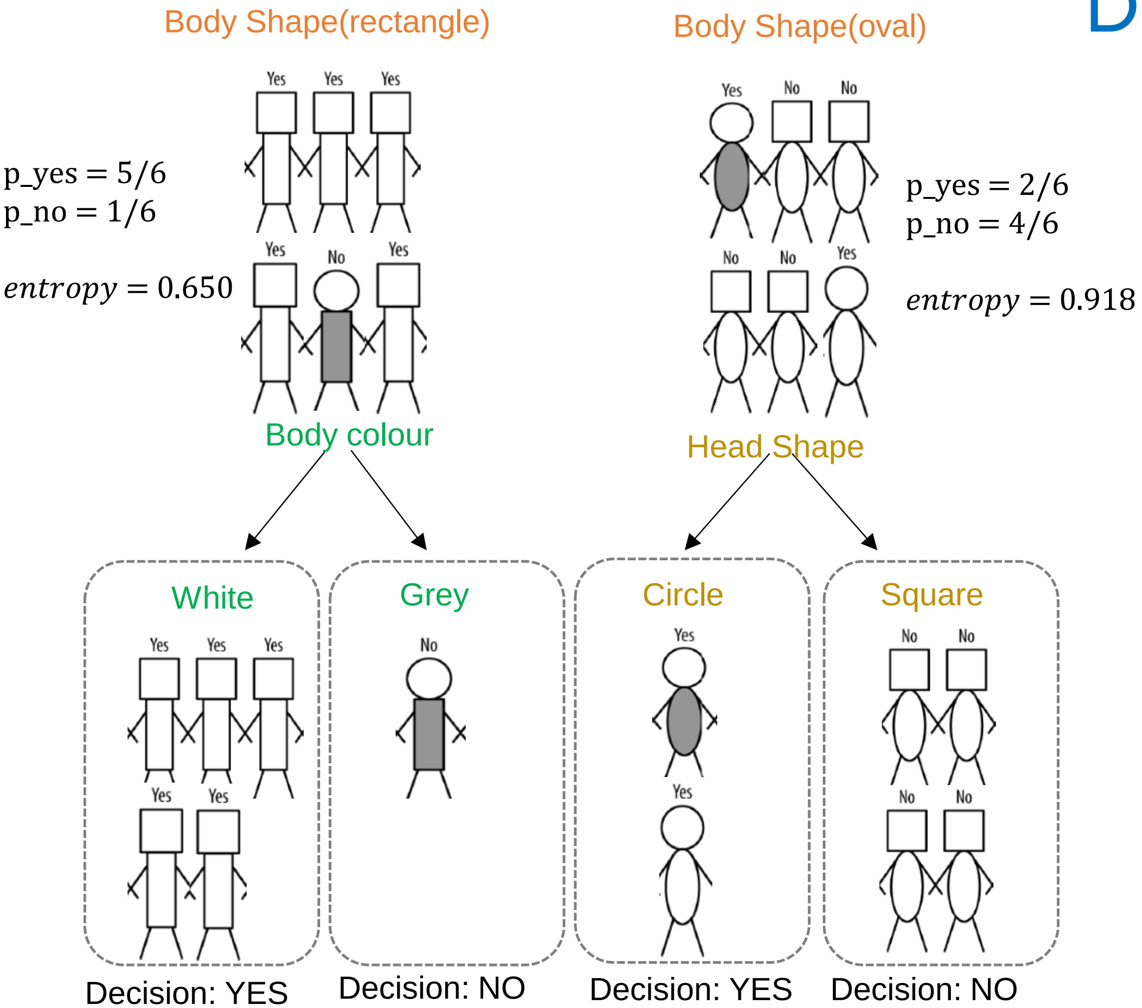
$p\_yes = 2/6$
$p\_no = 4/6$

$entropy = 0.918$

$$IG = entropy(base) - p(c_1) \times entropy(c_1) + p(c_2) \times entropy(c_2) + \ \ldots$$

$$\begin{aligned} \mathrm{IG} &= 0.98 - (0.5 \times 0.650 + 0.5 \times 0.918) \\ &= 0.196 \end{aligned}$$

Decision trees



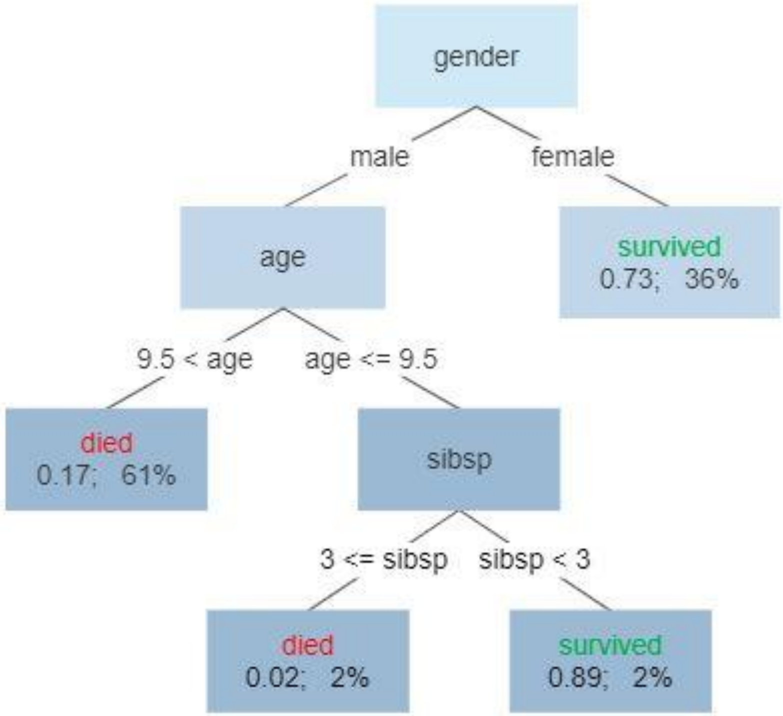
$$IG = entropy(base) - p(c_1) \times entropy(c_1) + p(c_2) \times entropy(c_2) + \ldots$$

$$\mathrm{IG} = 0.650 - (0.17 \times 0 + 0.83 \times 0) = 0.650$$

$$\mathrm{IG} = 0.919 - (0.33 \times 0 + 0.67 \times 0) = 0.918$$

Decision trees

Survival of passengers on the Titanic



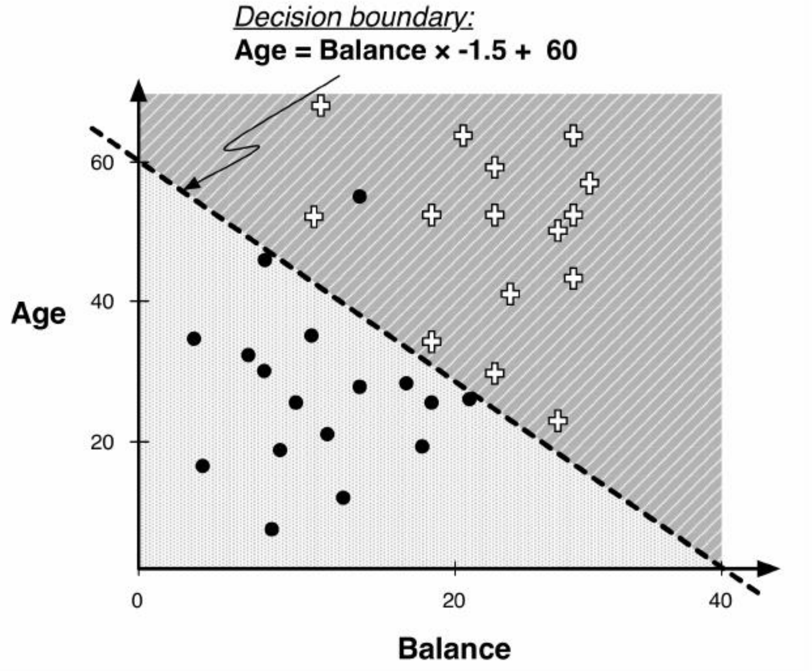
- Widely used for regression and classification problems
- Root at top, leaves at bottom

- Titanic survival model predicts survival for:
  - females
  - males younger than 9.5 years with less than 2.5 siblings
- The figures under the leaves show the probability of survival and the percentage of observations in the leaf.

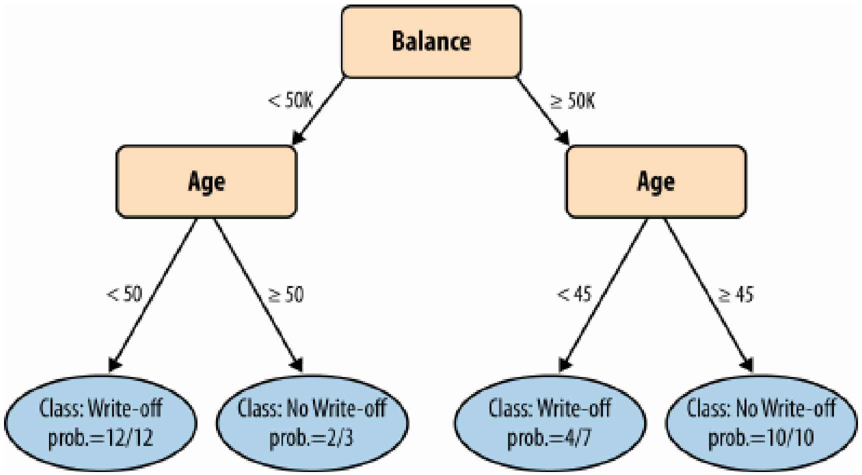
Fitting classification models

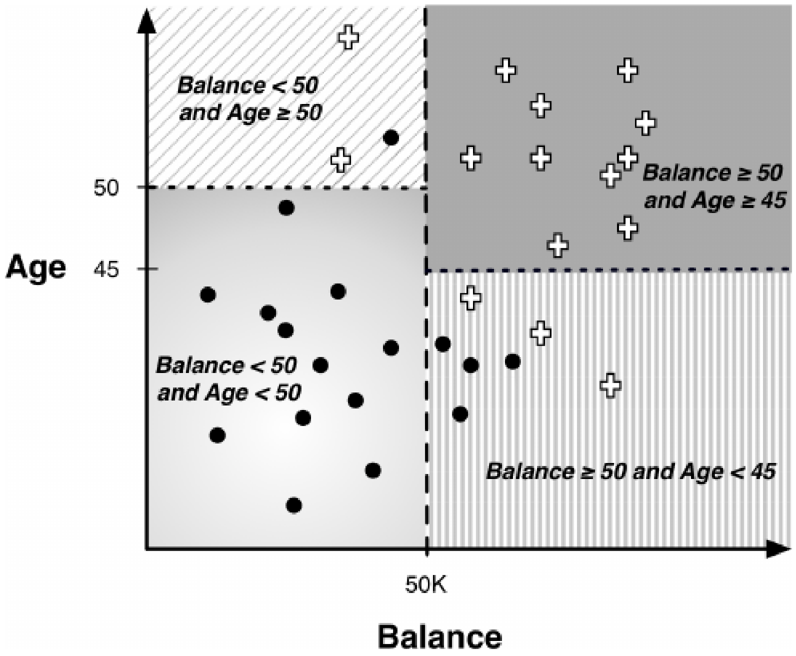
A poorly chosen decision boundary can lead to underfitting (oversimplifying) or overfitting (too closely fitting the data)

- **How do we know how best to draw the boundary?**



Decision trees: hyperplanes

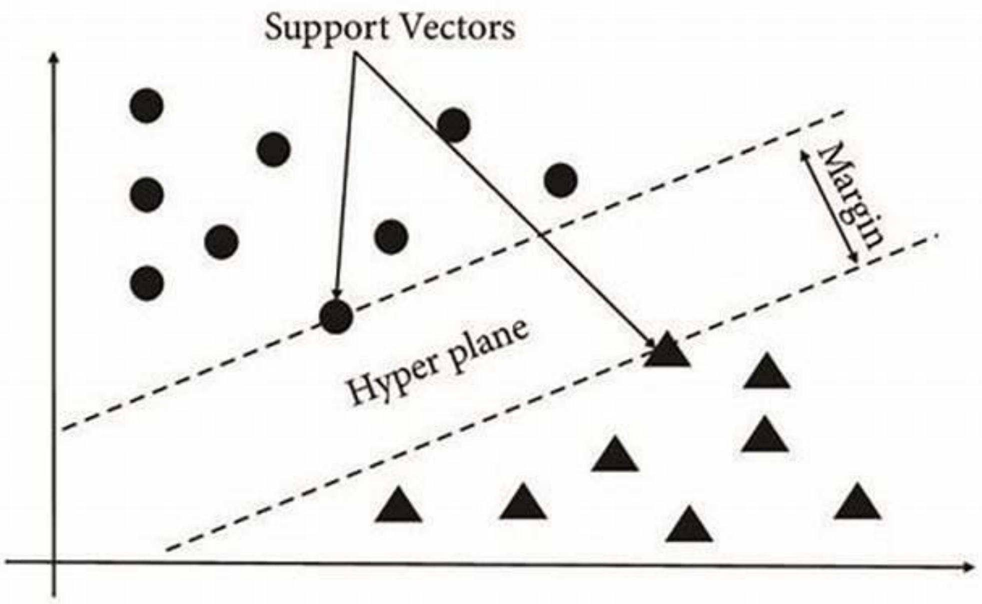




A decision tree can be plotted with each decision segmenting a space into boxes.

The decision boundary is called a **hyperplane**.

Support Vector Machines

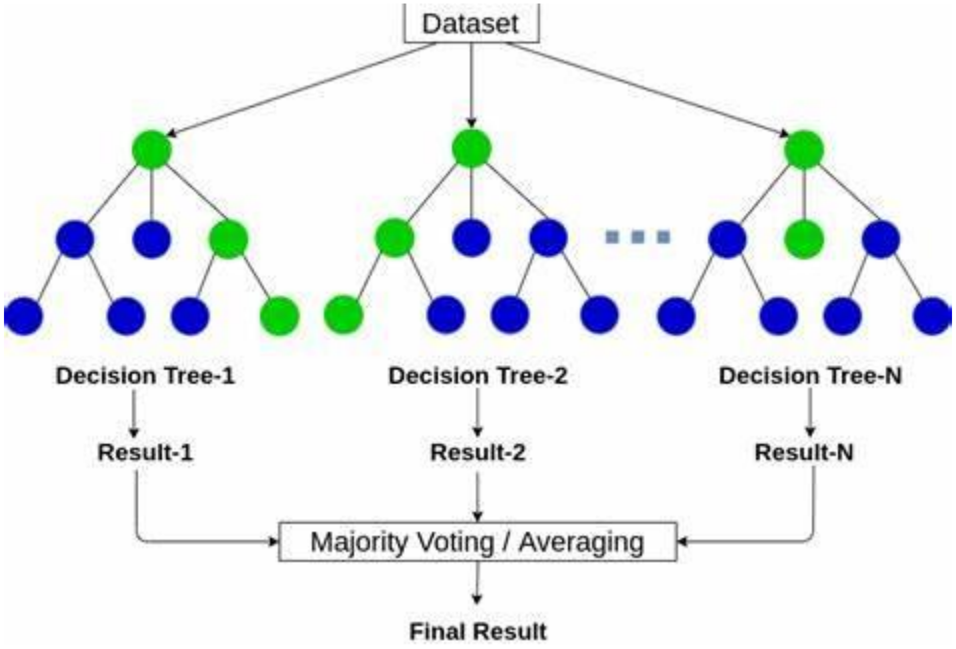


For SVM, part of the objective function (the goal) uses not only the accuracy of the prediction, but also maximises the width of the margin between categories.

- **Improves Generalisation:**
- A larger margin reduces the risk of overfitting and improves the model's ability to generalize to new data.
- **Robustness:**
- By focusing on the support vectors, SVM ignores other points, making it less sensitive to noise or outliers.

Random Forests

University of Exeter



Widely used for regression and classification

Consists of a 'forest' of decision trees:

All fit on *random bootstrap samples* of the data (each tree is trained on a slightly different dataset).

At each split in a tree, Random Forest considers only a *random subset of features* rather than all features.

Each decision tree is *trained independently* on its respective bootstrapped dataset and feature subset.

- The results are averaged for regression
- Majority vote for classification

Next Week: Reading Week – For week 7:

- Read Data Science for Business, chapters 5, 7, 8
- Read AUC-ROC: a really good article
- Watch StatQuest: ROC and AUC, Clearly Explained! – YouTube
- Watch StatQuest: Bias and Variance
- Watch StatQuest: Cross validation
- Watch StatQuest: Sensitivity and Specificity

University of Exeter

Any questions?

?