INTRODUCTION TO MACHINE LEARNING

FUNDAMENTAL CONCEPTS

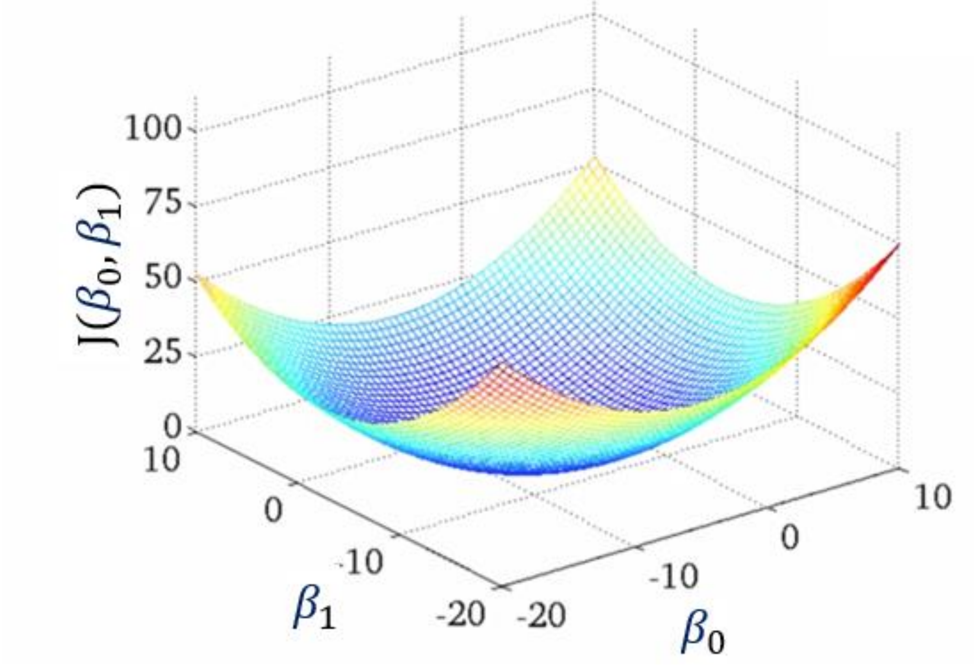
Andreu Arderiu

NORTHIN SUMMER SCHOOL

Barcelona, July 2022

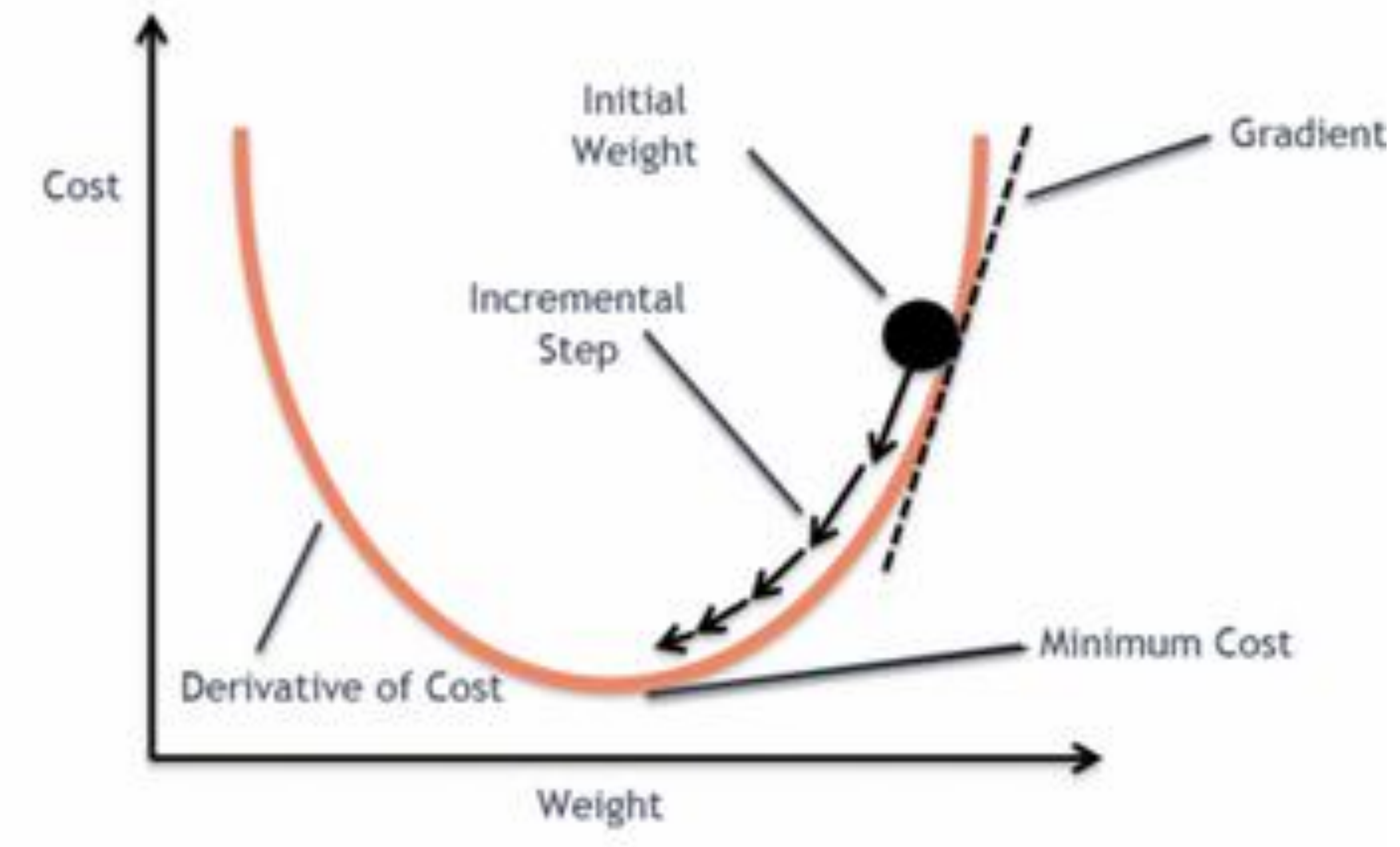
Linear regression: cost function

- Minimize cost function: $J(\beta_0, \beta_1) = \frac{1}{n}\sum_{i=1}^{n}\left(h_\beta\left(x^{(i)}\right) - y^{(i)}\right)^2$
- Gradient descent algorithm: Iterate $\beta_j := \beta_j - \alpha \frac{\partial}{\partial \beta_j} J(\beta_0, \beta_1)$



Source: Coursera

Gradient descent: find the weights minimizing cost



Multivariate linear regression: overview

- Hypothesis: $h_\beta(x) = \beta_0 + \beta_1 x_1 + \ ... + x_m\beta_m = \beta^T x$

Multiple features (variables).

Size (feet^2)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

Notation:

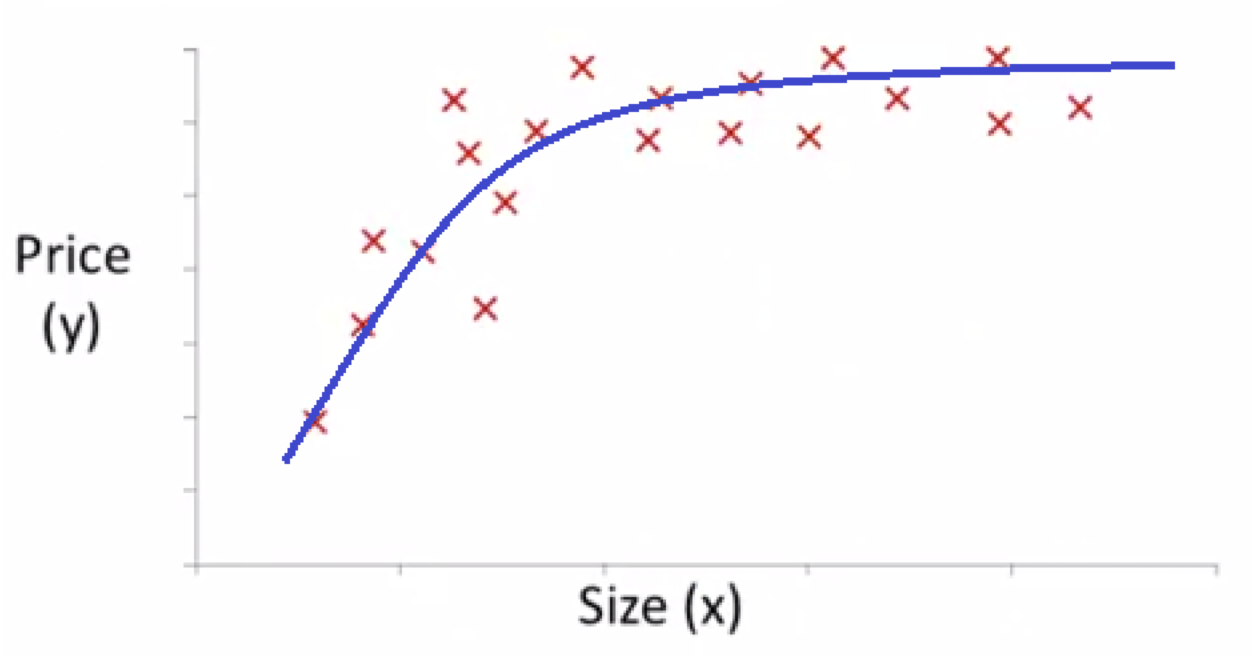
$x^{(i)}$ = input (features) of i^{th} training example.

$x_j^{(i)}$ = value of feature j in i^{th} training example.

Source: Coursera

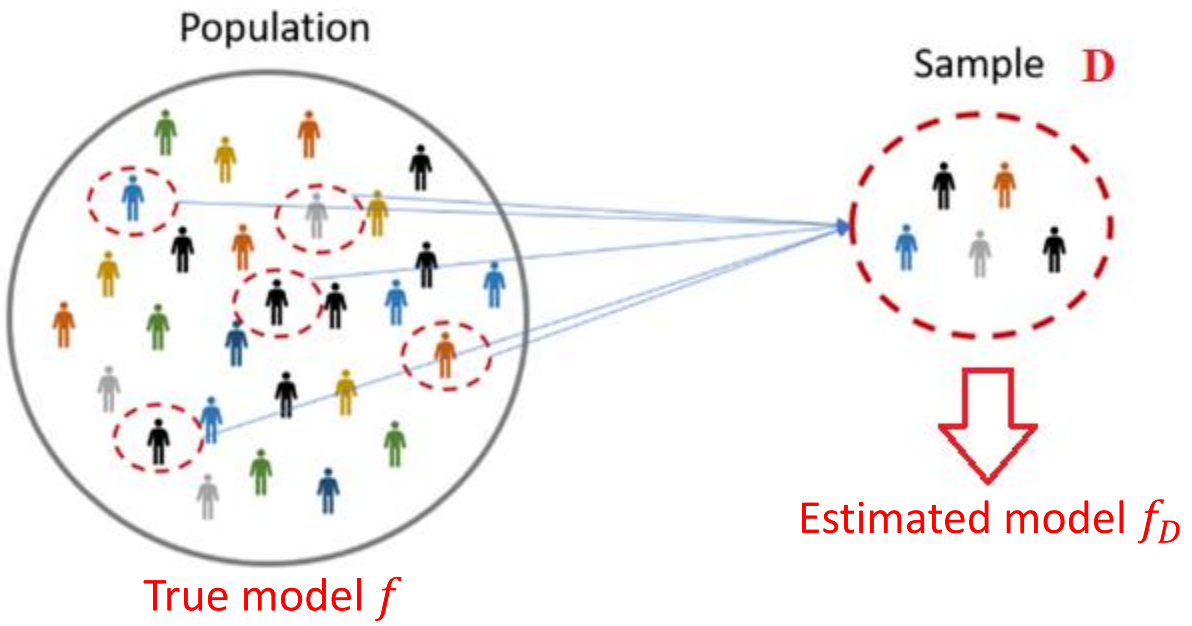
Polynomial regression

- Non-linear combinations of x
- Hypothesis: $h_\beta(x) = \beta_0 + \beta_1 x + x^2\beta_2 + \sqrt{x}\beta_3 + \cdots$



Model performance: finite samples

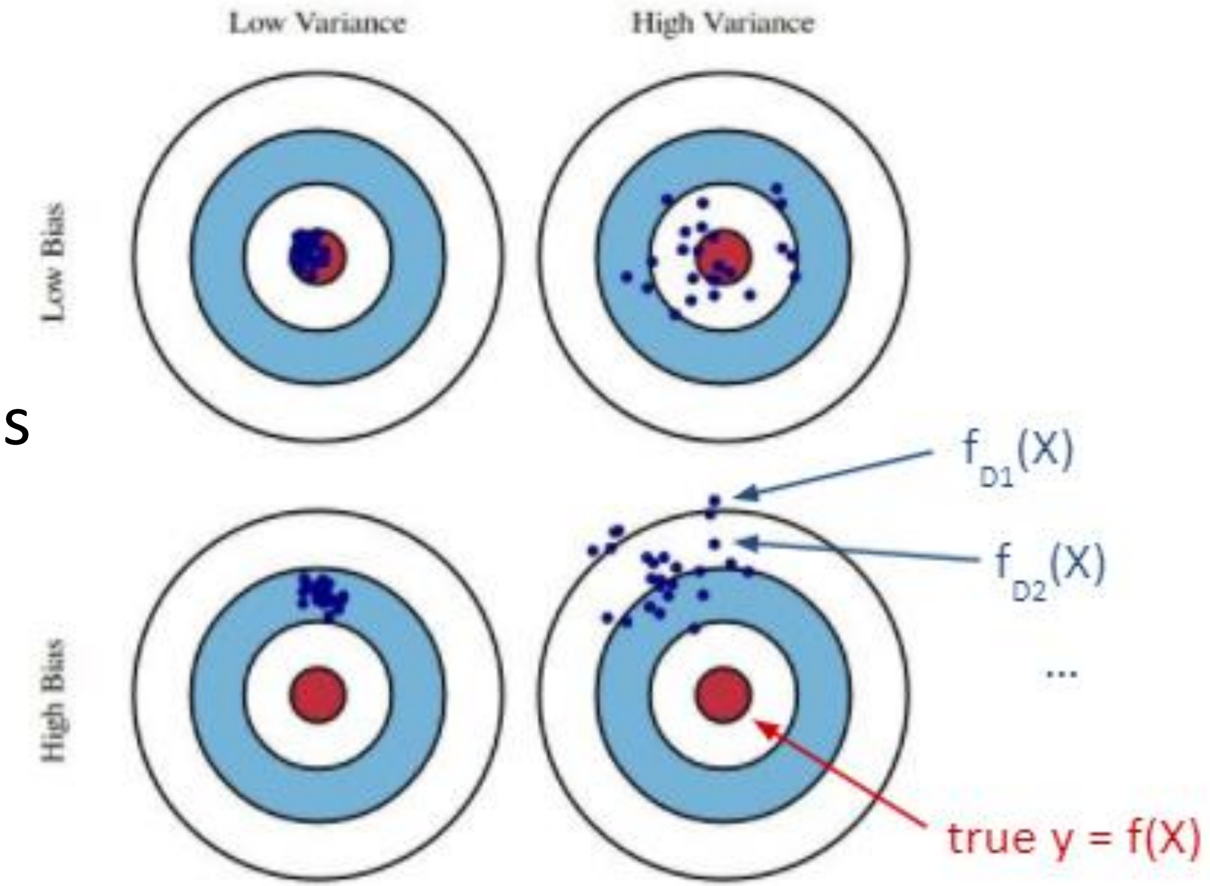
- Most datasets are **samples** taken from an **infinite** population.
- We are interested in modelling the **whole population**, we just have access to a **finite sample**.



Source: Hotcubator

Model performance: Bias and variance

- Consider a fixed X
- **Bias:** Expected difference between predictions and true value
- **Variance**: "Variability" of predictions



Source: Robert West, EPFL

Quizz: Bias and variance

- **Complex** models (**many** parameters) have low/high bias?
  low/high variance?

- **Simple** models (**few** parameters) have low/high bias?
  low/high variance?

Quizz: Bias and variance

- **Complex** models (**many** parameters) have **low**/high bias?
  low/**high** variance?

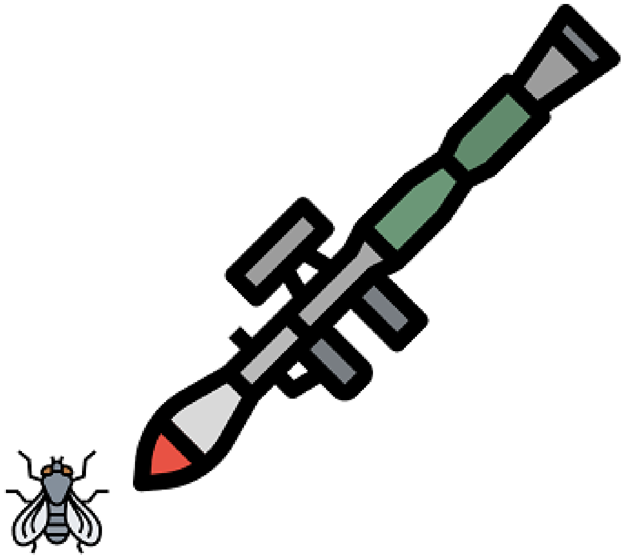
- **Simple** models (**few** parameters) have low/**high** bias?
  low/high variance?

Overfitting: Bias-variance tradeoff

- If we use too many features, the learned model may fit the training data very well (**overfit**), but fail to **generalize** to new examples

Image Credits: https://livebook.manning.com/

Underfitting

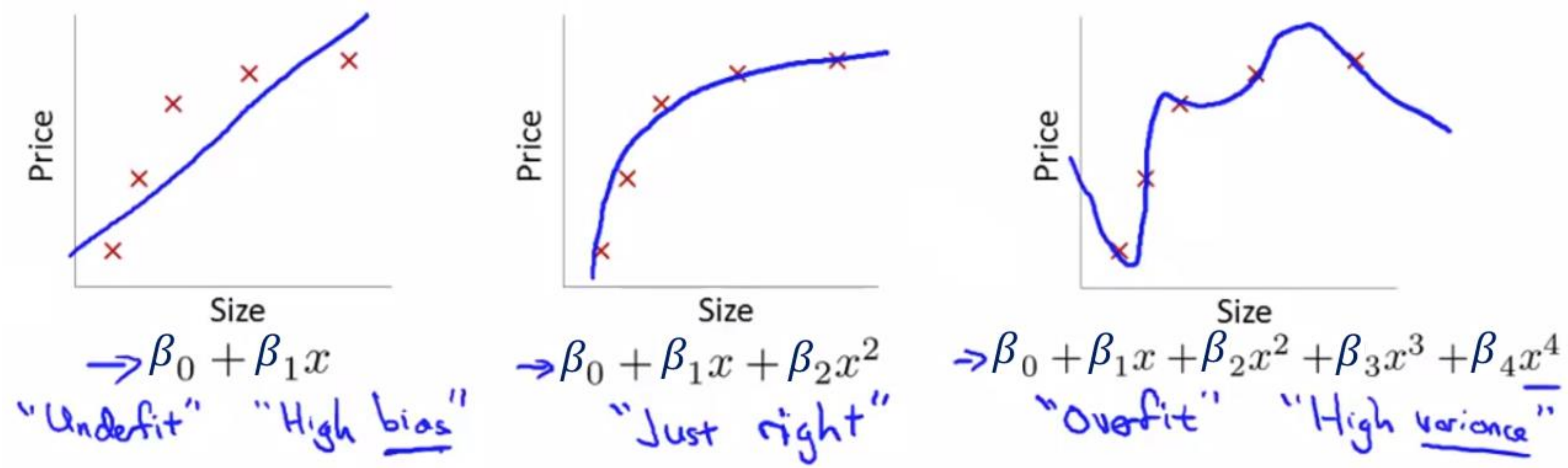


Overfitting

Overfitting: Bias-variance tradeoff

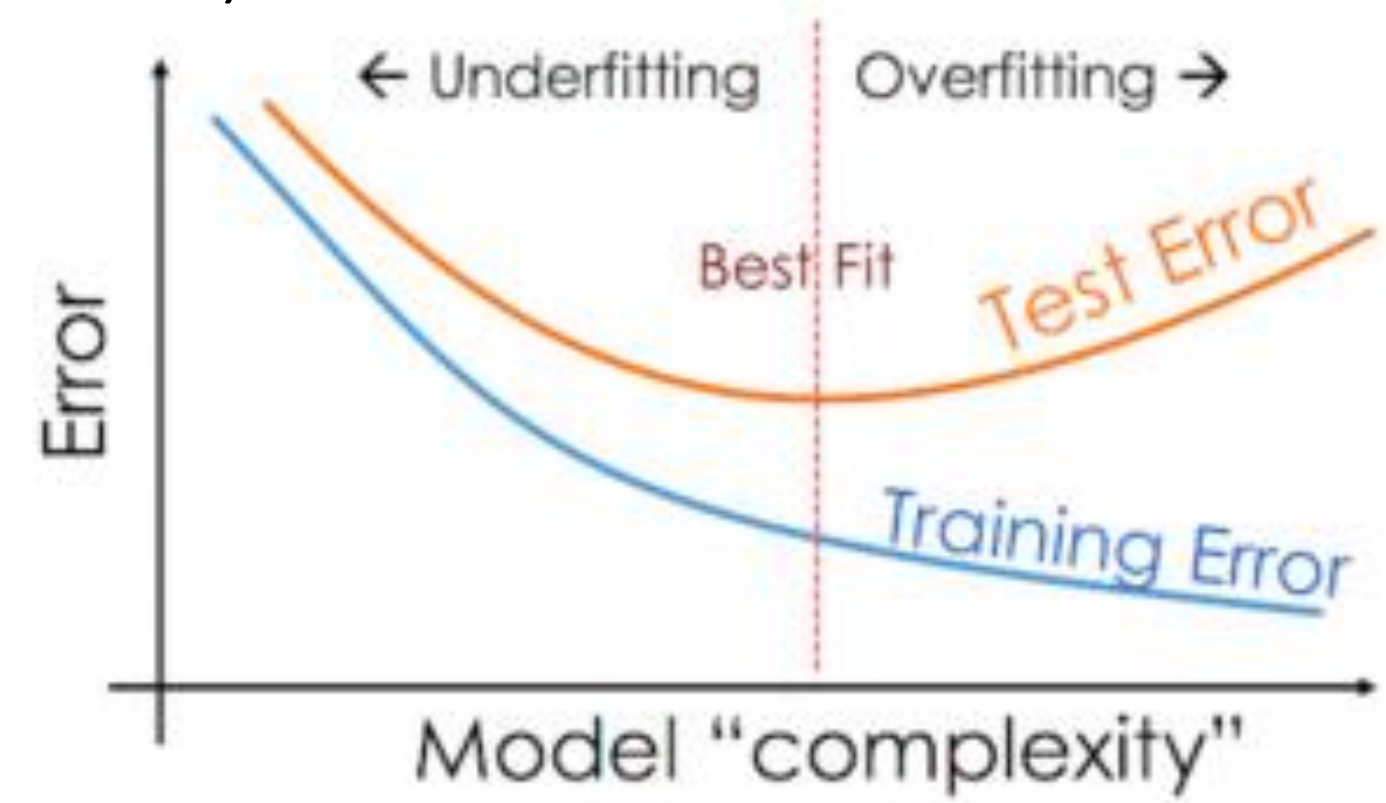
- If we use too many features, the learned model may fit the training data very well (**overfit**), but fail to **generalize** to new examples

Source: Coursera



Overfitting: Bias-variance tradeoff

- Ideally we want **low bias** (small **training error**) and **low variance** (small **test error**)



Source: Ajitesh Kumar

Overfitting: solutions

How to reduce overfitting?

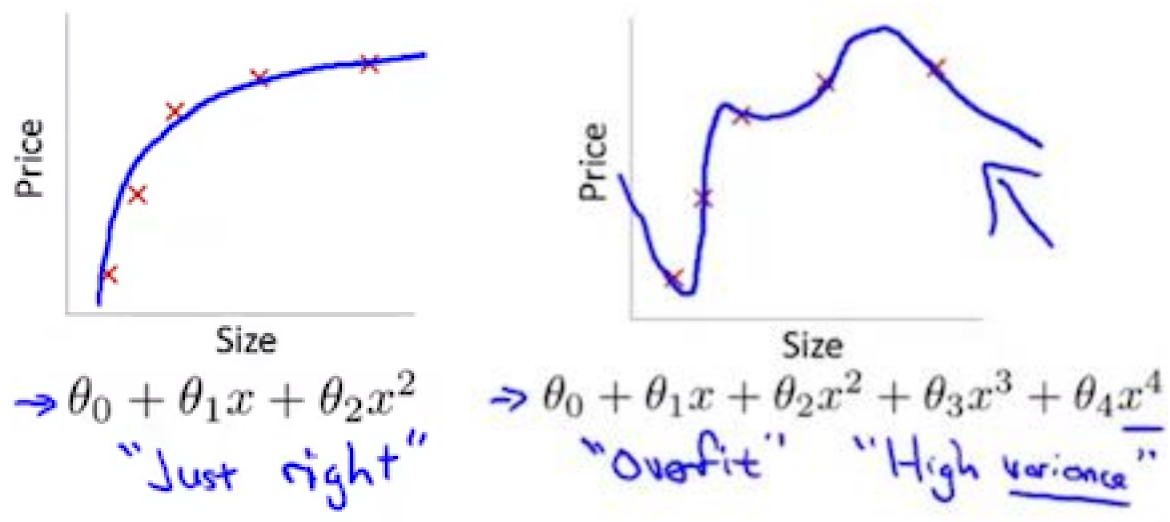
Overfitting: solutions

How to reduce overfitting?

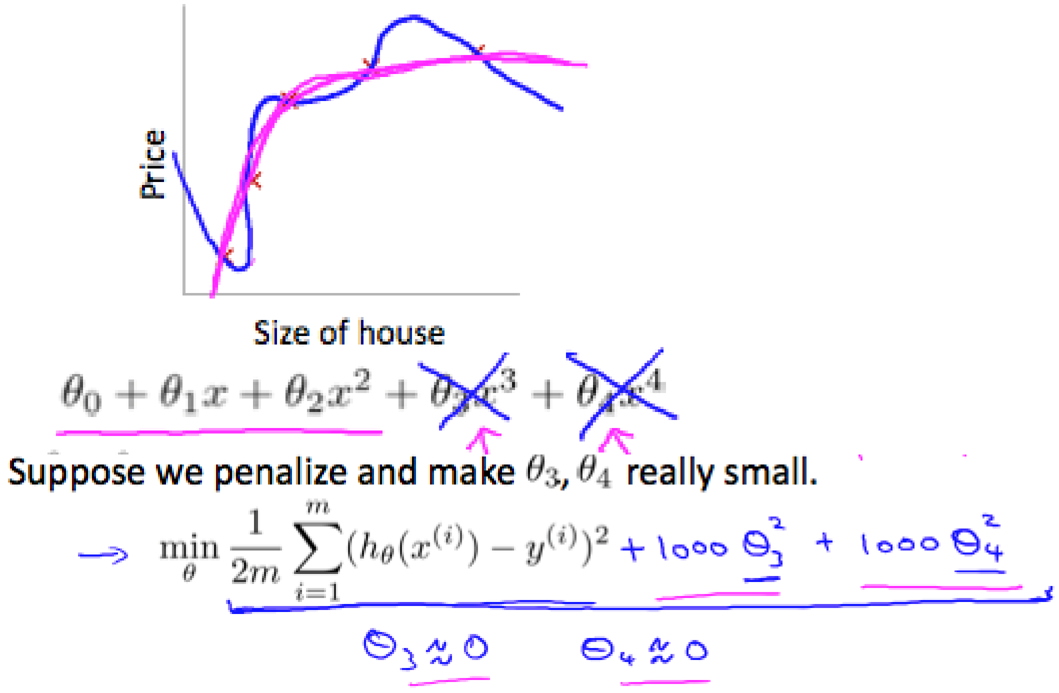
1. **Reduce** the number of **features**: Manually select/reduce the number of features to use
2. **Regularization**: Modify cost function to **penalise** number of **weights** or weights high values

Regularization

- Regularization helps us to maintain all variables or features in the model by reducing the magnitude of the variables. Hence, it maintains accuracy as well as the generalization power of the model.



Source: Coursera

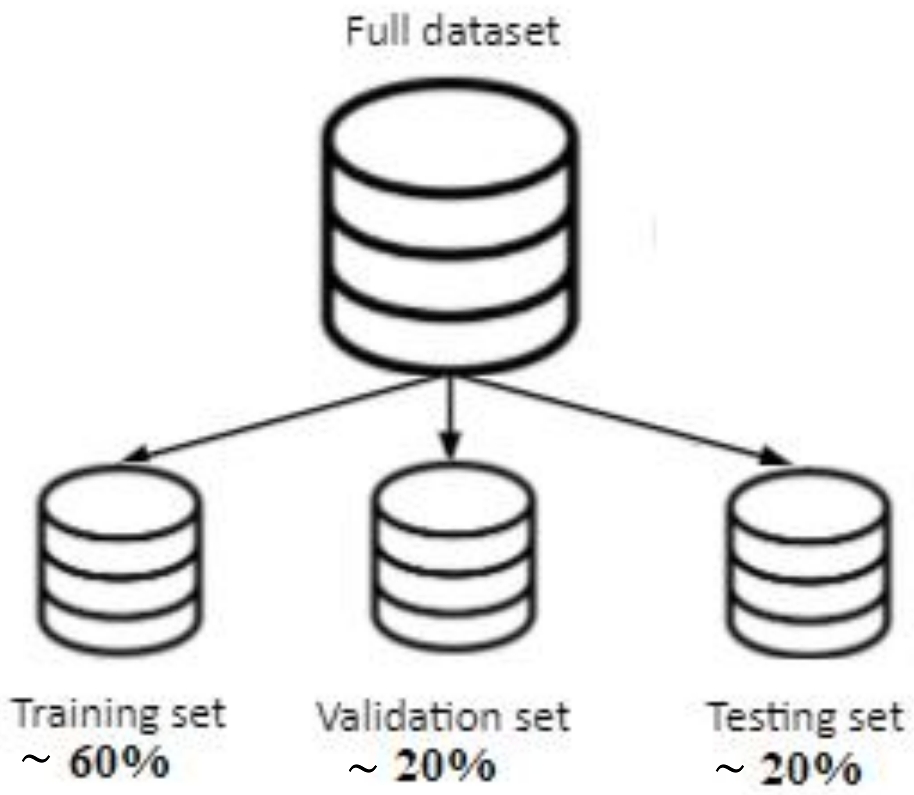


Model selection: overview

- Usually a classifier has some **"hyperparameters"** to be tuned
  - Linear regression: Number of features
  - Polynomial regression: Degree of the polynomial
  - Lasso/ridge regression: Regularization parameter λ
  - …
- Hyperparameters are set before training can begin

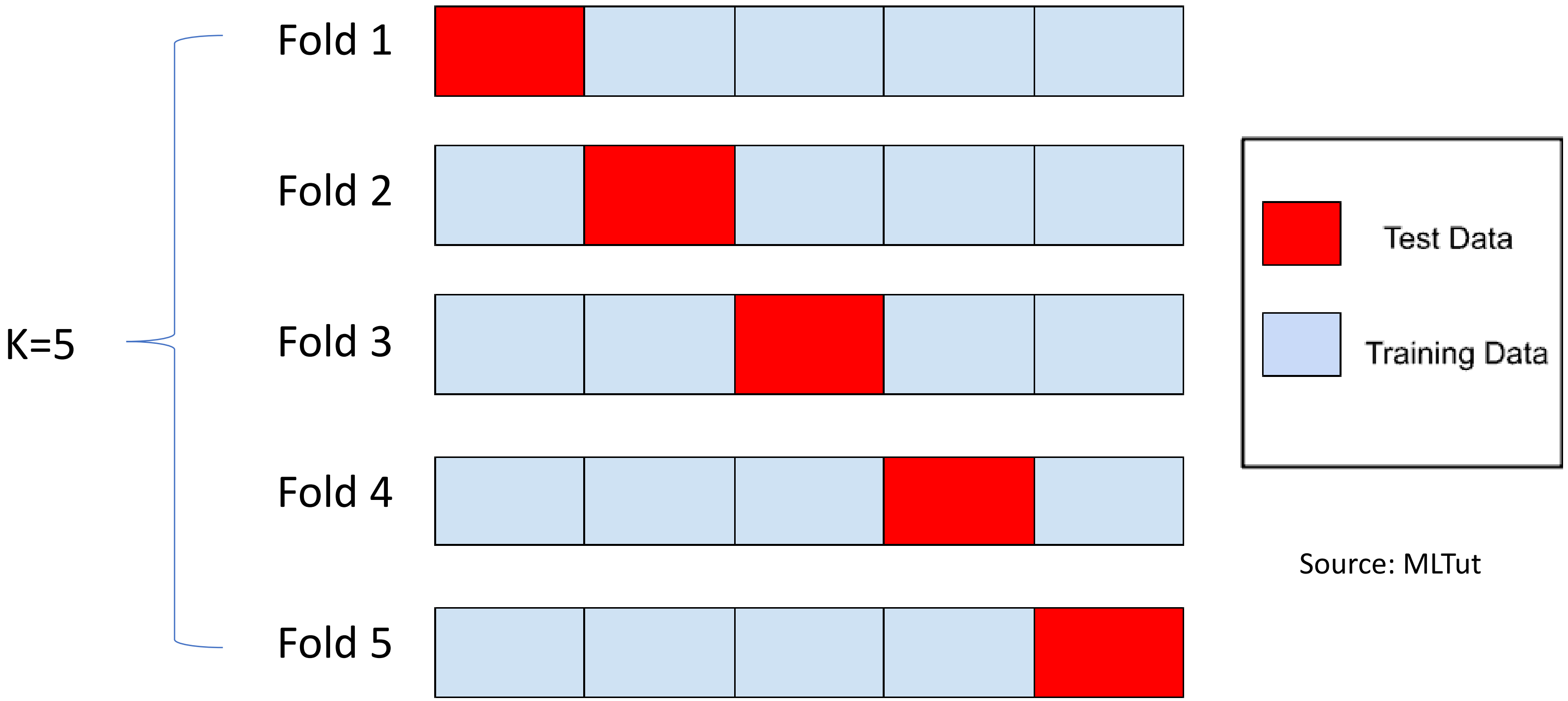
Model selection: Train/Validation/Test sets

1. **Fit** model **parameters** on **training** set
2. **Choose model**/hyperparameter configuration with **lower validation error**
3. **Evaluate** model performance with **testing** set



Model evaluation: k-fold cross-validation

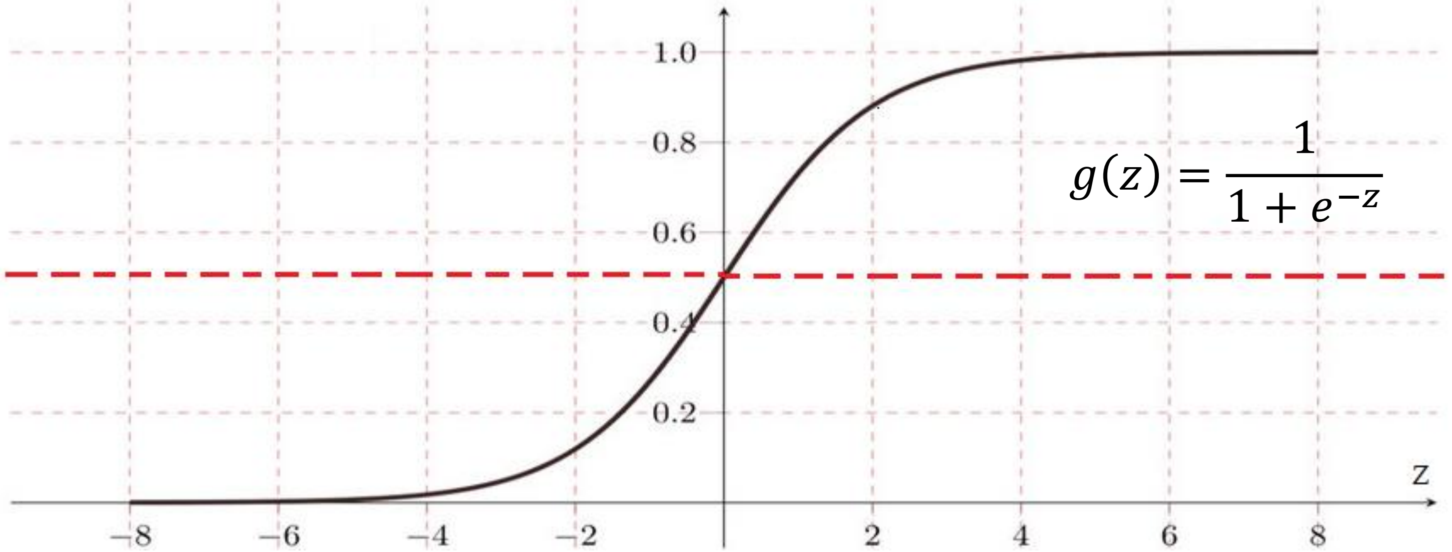
- More **efficient** way to compute validation error if we have **little data**
- Average error over the k red portions → **Validation error**



Source: MLTut

Logistic regression: overview

- Supervised learning algorithm for **classification** tasks
- Hypothesis: $h_\beta(x) = \mathrm{P}(\mathrm{y} = 1|\mathrm{x}; \beta) = \mathrm{g}(\beta^T x) = \frac{1}{1+e^{-\beta^T x}}$
- $g(\cdot)$ **sigmoid** or **logistic** function, values between 0 and 1



$g(z) \geq 0.5\ if\ z \geq 0$

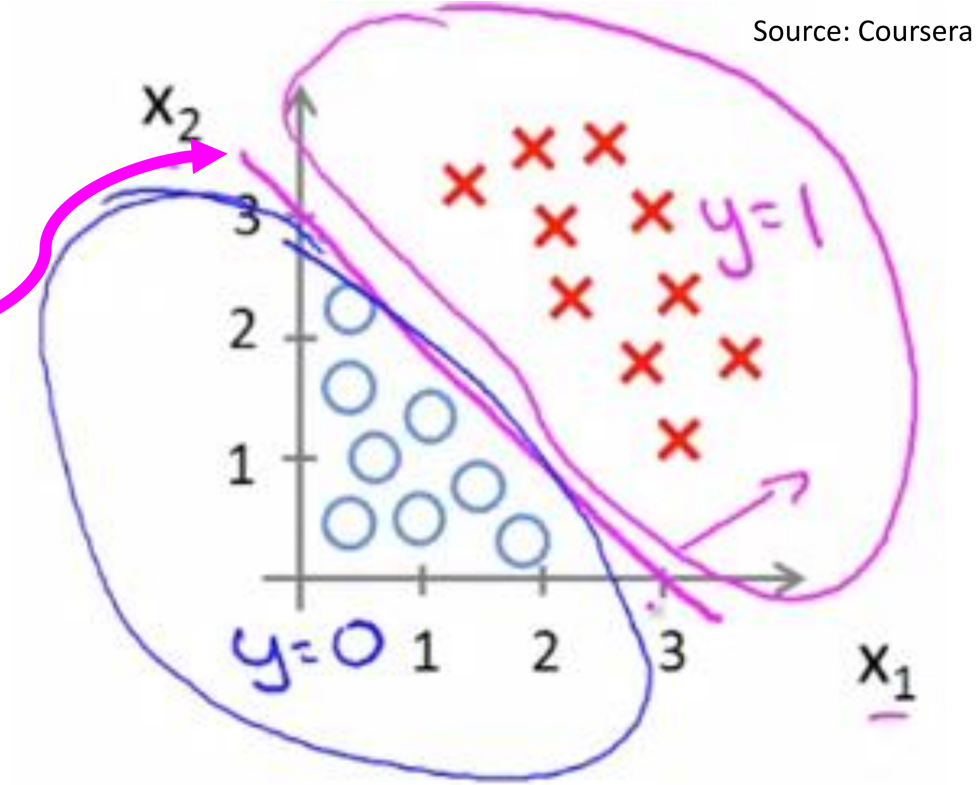
Logistic regression: decision boundary

- Assume we **predict $\boldsymbol{y} = \mathbf{1}$** if $h_\beta(x) = \mathrm{P}(\mathrm{y} = 1|\mathrm{x}; \beta) \geq 0.5$
- Example 1:

$$h_\beta(x) = g(-3 + x_1 + x_2)$$

$$y = 1 \text{ if } \quad -3 + x_1 + x_2 \geq 0;$$

- **Decision boundary** $x_1 + x_2 = 3$



Performance metrics for classification

- In binary classification (Yes/no, 0/1), we use the **confusion matrix** which has 4 values:
  - **True positives:** positive examples classified as positive
  - **True negatives:** negative examples classified as negative
  - **False positives:** negative examples classified as positive
  - **False negatives:** positive examples classified as negative

		Class	
		A	B
Classified	A	TP	FP
	B	FN	TN

Credit: Robert West, EPFL

Accuracy: overview

- Represents the % of correctly predicted cases

$$A = \frac{TP+TN}{TP+TN+FP+FN} = \frac{TP+TN}{N}$$

- Good metric when
  - ➢Classes are not skewed
  - ➢Errors (FP, FN) have the same importance

Source: Avinash Pandey

Accuracy: skewed example

$$A = \frac{85}{100} = 0.85$$

Classifier 1		Class: Fraud	Class: ¬Fraud
Classified	Fraud	5	10
	¬Fraud	5	80

Credit: Robert West, EPFL

$$A = \frac{90}{100} = 0.90$$

Always ¬Fraud		Class: Fraud	Class: ¬Fraud
Classified	Fraud	0	0
	¬Fraud	10	90

Credit: Robert West, EPFL

Accuracy: errors with different importance

$$A = \frac{75}{100} = 0.75$$

Classifier 1

		Class: Cancer	Class: ¬Cancer
Classified	Cancer	45	20
	¬Cancer	5	30

Credit: RobertWest, EPFL

$$A = \frac{80}{100} = 0.80$$

Classifier 2

		Class: Cancer	Class: ¬Cancer
Classified	Cancer	40	10
	¬Cancer	10	40

Credit: RobertWest, EPFL

Precision, recall, F1-score

- **Precision**: What fraction of positive predictions are actually positive?

$$P = \frac{TP}{TP+FP}$$

- **Recall:** What fraction of positive examples did I recognize as such?

$$R = \frac{TP}{TP+FN}$$

- **F1-score:** Harmonic mean of precision and recall

$$\mathrm{F1} = 2\frac{P \cdot R}{P+R}$$