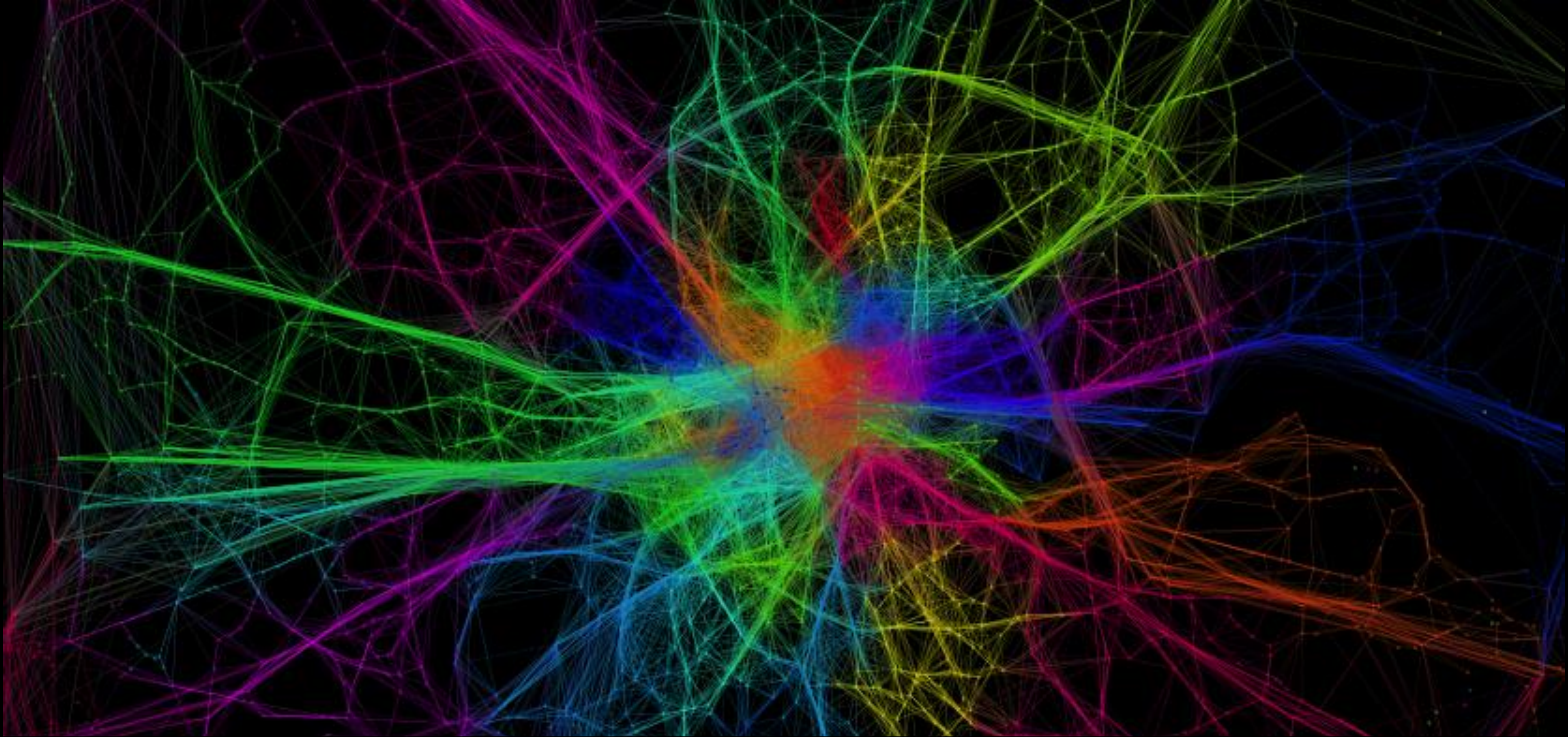


Analysis Workflow

CASA0006: Data Science for Spatial Systems

Huanfa Chen

CASA0006

1. Introduction to Module
2. Supervised Machine Learning
3. Tree-based Methods
4. Artificial Neural Networks
5. Analysis Workflow
6. Panel Regression
7. Difference in Difference
8. Regression Discontinuity
9. Dimensionality Reduction
10. Spatial Clustering

Objectives

- Be able to understand and conduct basic error analysis (*‘is my model overfitting or underfitting on the training data’*)
- Be able to improve machine learning models using strategies (*‘how to tackle overfitting or underfitting’*)
- Be able to detect and prevent data leakage

Outline

1. Basic error analysis
2. Machine learning strategy
3. Data leakage

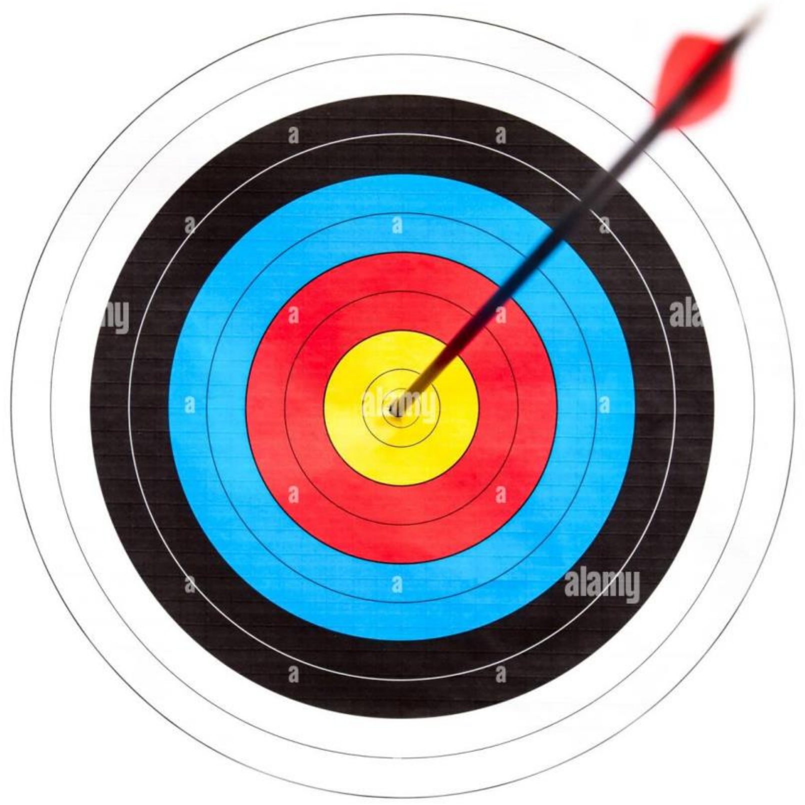
Basic error analysis

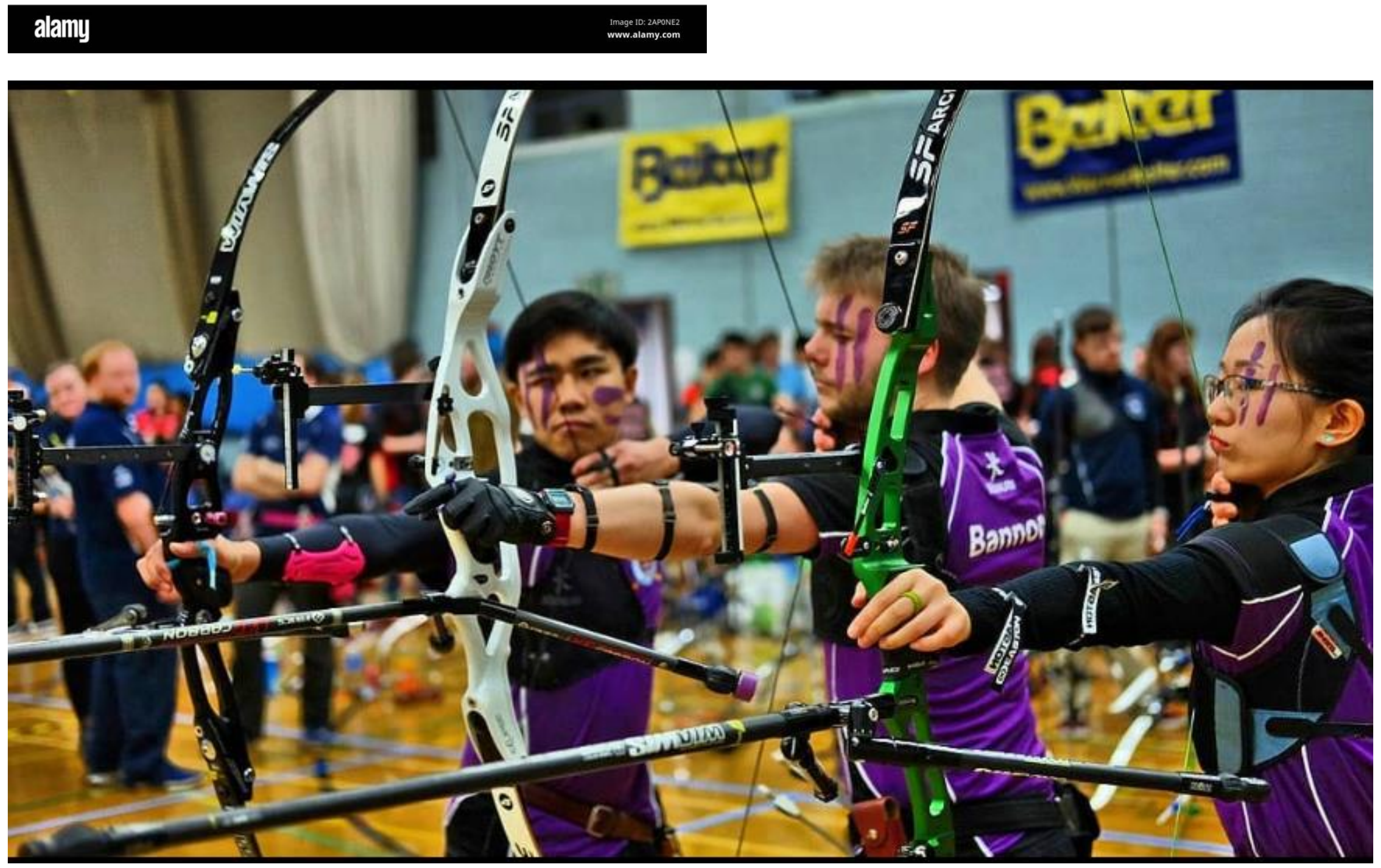
Basic error analysis

- When we discuss supervised learning models, prediction errors can be decomposed into two main subcomponents
  - Error due to bias
  - Error due to variance
- There is a tradeoff between a model's ability to minimise bias and variance (called bias-variance tradeoff)
- Why important? These two errors are linked to over- and under-fitting. Understanding these two types of error can help us diagnose model results and mitigate over- or under-fitting.
- Three ways to understand bias & variance: graphically, conceptually, mathematically

Bias-variance in archery

- Archery is as cool as machine learning.
- Archery: supervised learning
- Archer: prediction algorithm (like, neural network)
- Task: hit the ‘Gold’ or ‘bull’s eye’ (Centre of the target, often in yellow).
- To know your performance: you shoot many times and then summarise the **mean** and **variance** of scores.
- **Bias** = full score - mean score
- **Variance** = variance of the scores



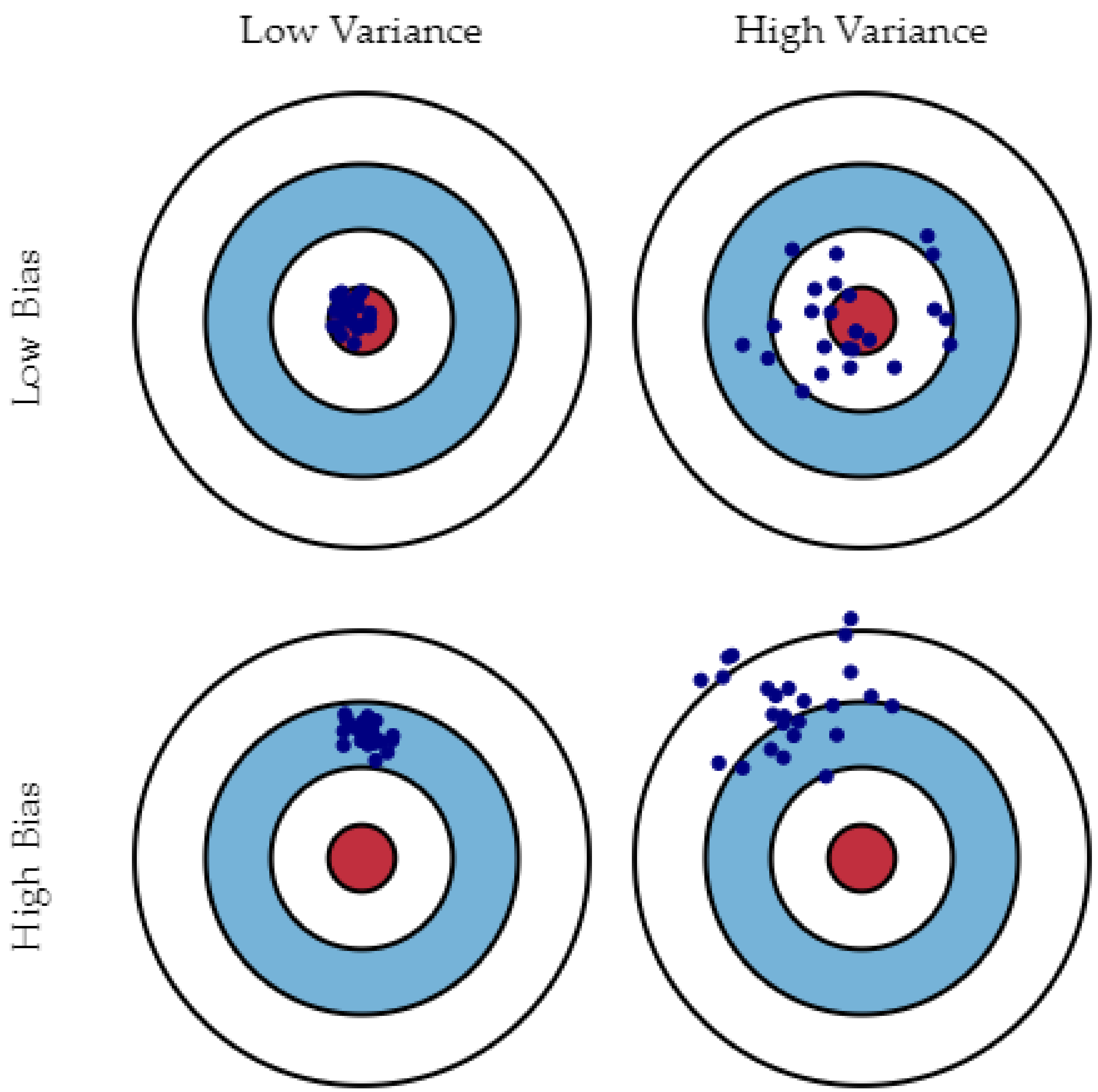


https://clubs-societies.london.ac.uk/organisation/archery/

Bias-variance in archery

- Four combinations of bias and variance (two levels each)

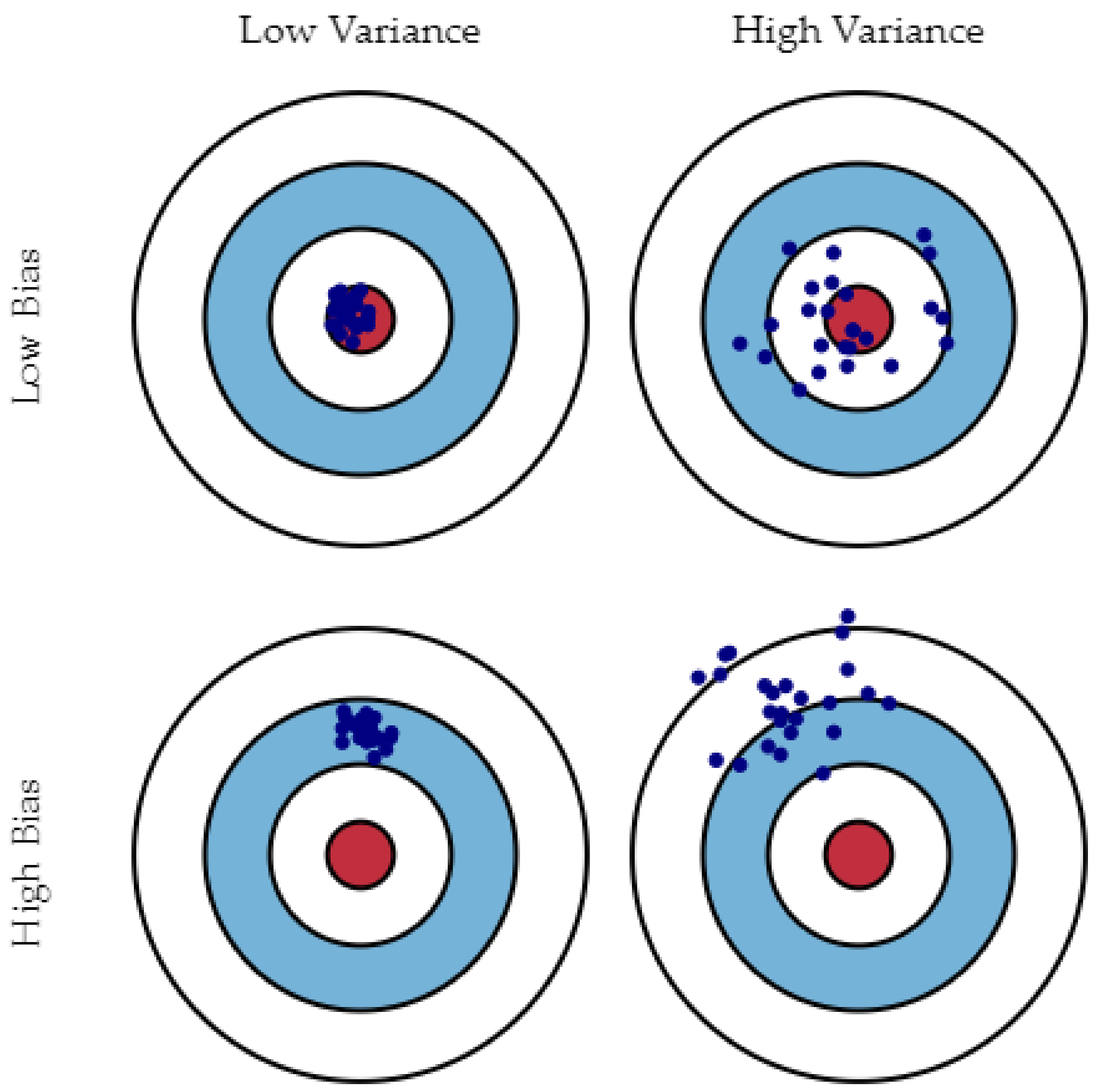
Bias \ Variance	Low	High
Low	Great!	High-variance, not consistent
High	Low accuracy, need more training	*Improvement needed*



Conceptual definition

- **Error due to Bias**: the difference between the expected (or average) prediction of our model and the correct value which we are trying to predict
- **Error due to Variance**: the variability of a model prediction for a given data point

- Note: to measure bias or variance, multiple runs of models are necessary, which is similar to estimate the mean or variance of a variable.



Mathematical definition

- If we denote the target variable as Y and covariates as X, we assume that there is a relationship: Y=f(X)+ϵ
- We build a model to estimate f(X), which leads to $\hat{f}(x)$.
- The expected square error at a data point x is: $Err(x) = \mathrm{E}[\left(\mathrm{Y} - \hat{f}(x)\right)^2]$
- This error can be decomposed into three components:
- $Err(x) = Bias^2 + Variance + Irreducible\ Error$

- Irreducible Error: the noise term, which can't be reduced by any model

Bias-variance in machine learning

In the context of ML models

- **Bias**: how close the model can get to the true relationship between the predictors and the outcome

- Normally, a linear regression model has a higher bias than decision trees or neural networks, as it has <u>strong assumptions</u> regarding linear relationship between y and x and would fail to fit when the actual relationship is non-linear.

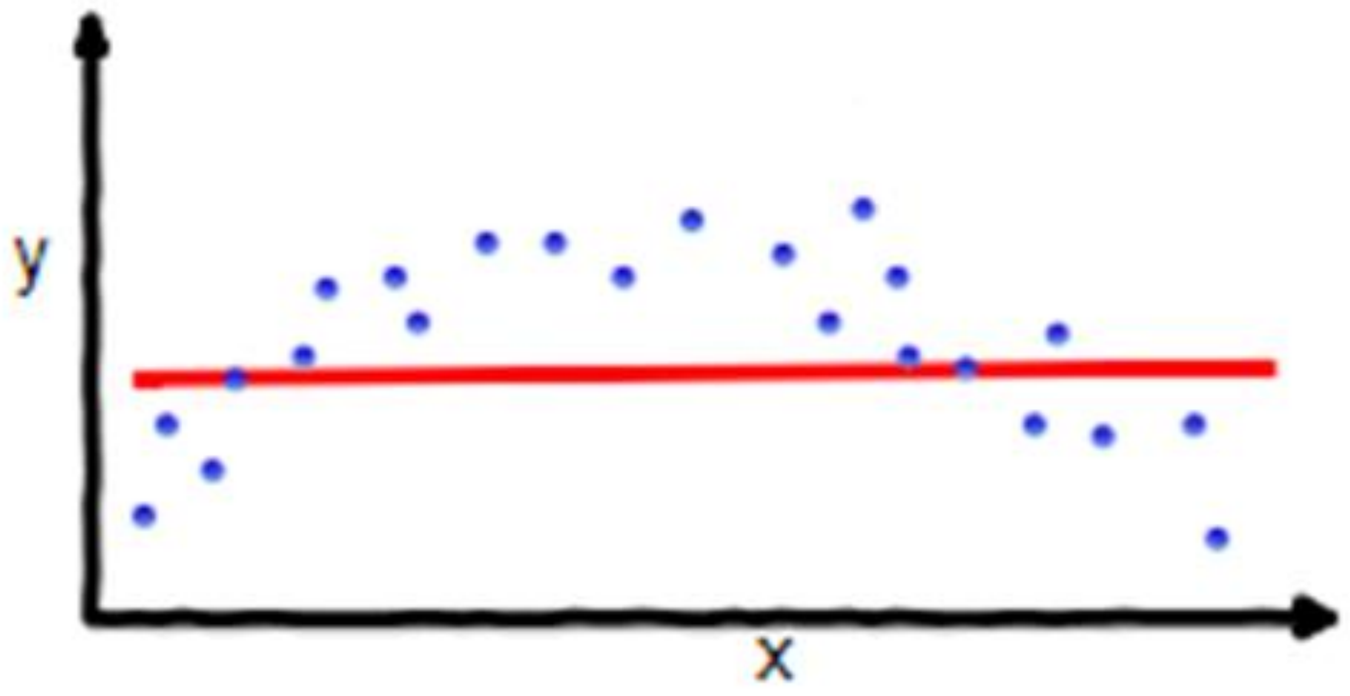
Bias-variance in machine learning

- **Variance**: the amount by which the model would change if we estimated it using a different training data set
- A high-variance model will change a lot with small changes to training dataset
- Given an algorithm, the higher complexity, the higher variance. (Link with model hyperparameters)
- Decision tree (DT): variance ↑ when max_depth ↑ or min_samples_leaf ↓
- Neural networks (NN): variance ↑ when n_layer ↑ or n_neurons ↑
- Note: we can’t directly compare variance of a CART and NN

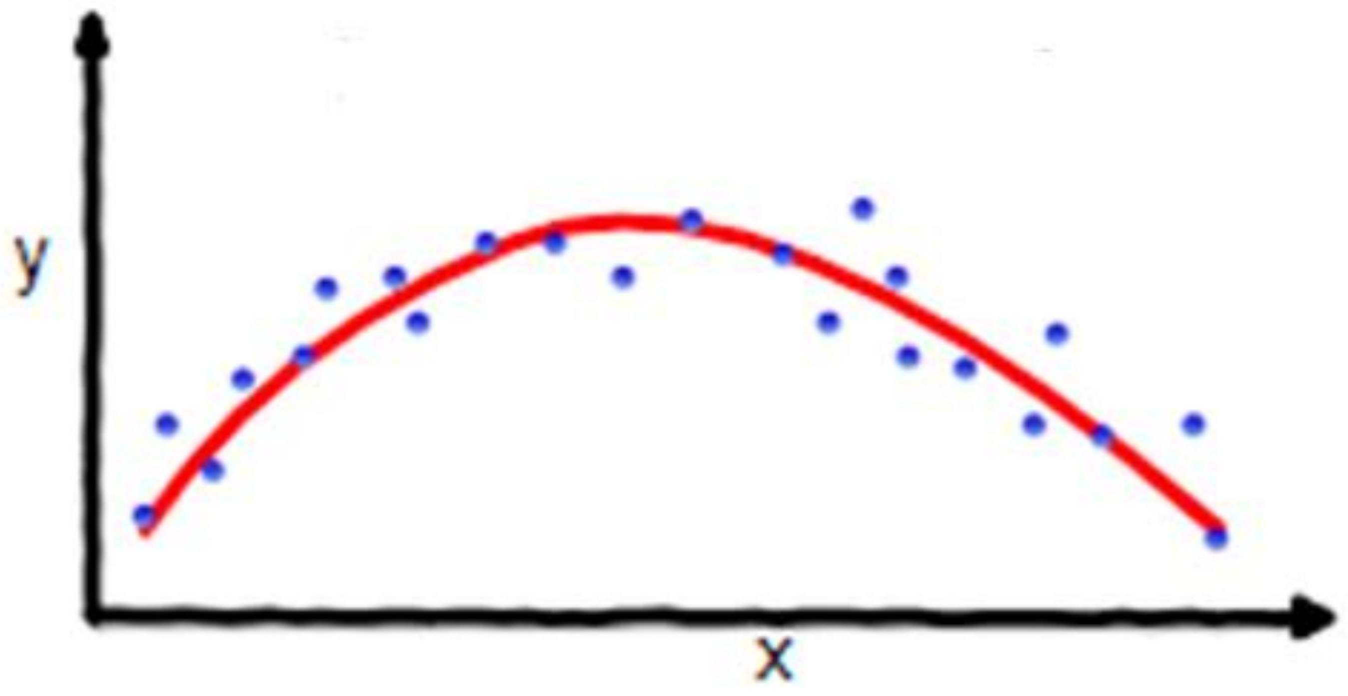
Bias-variance in machine learning

- Example of curve fitting (1-dimensional x)

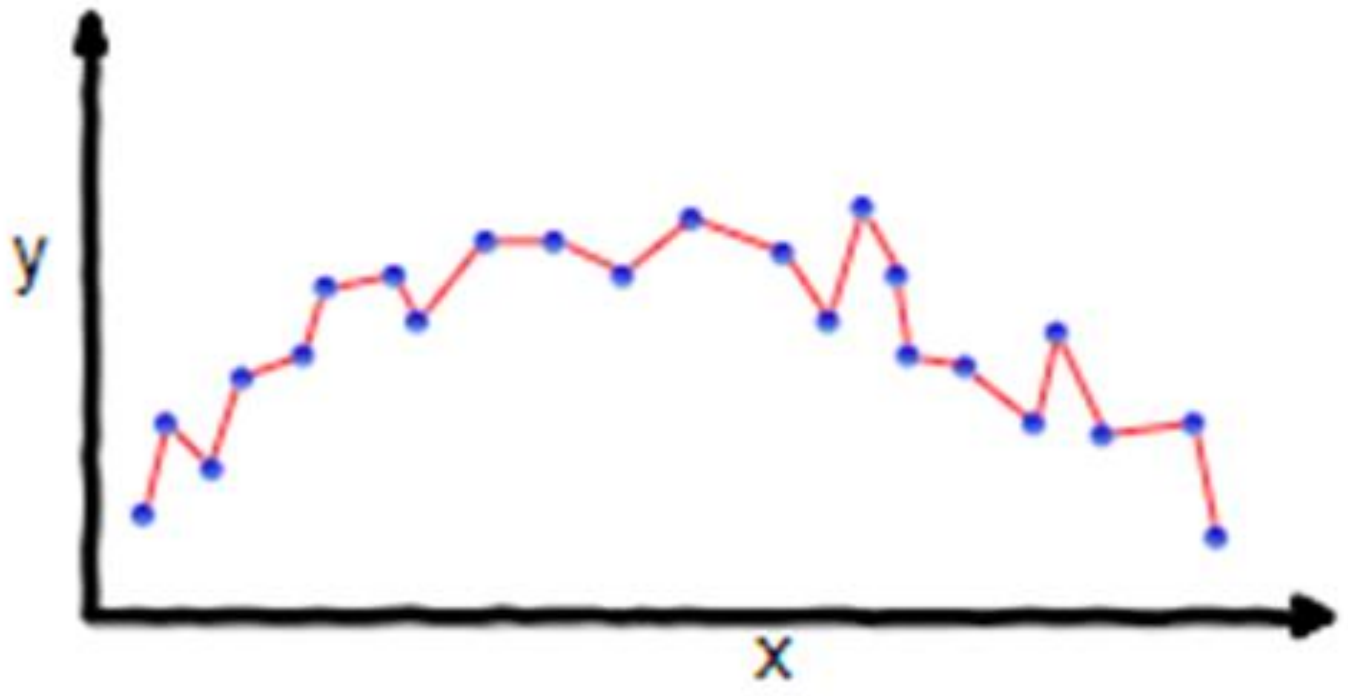
Y = ax+b



Y = ax^2+bx+c



Y = $\sum_{i=0}^{N} a_i x_i$



Model complexity & variance: low -> high

Bias-variance trade-off

- Ideally, we want a model with low bias and low variance, but this is very challenging in practice. In fact, this could be described as the goal of applied machine learning for a given predictive problem.
- The bias-variance trade-off
  - Reducing the bias can easily be achieved by increasing the variance.
  - Reducing the variance can easily be achieved by increasing the bias.
- This is a good conceptual framework for thinking how to choose models and model configuration (or hyperparameters).

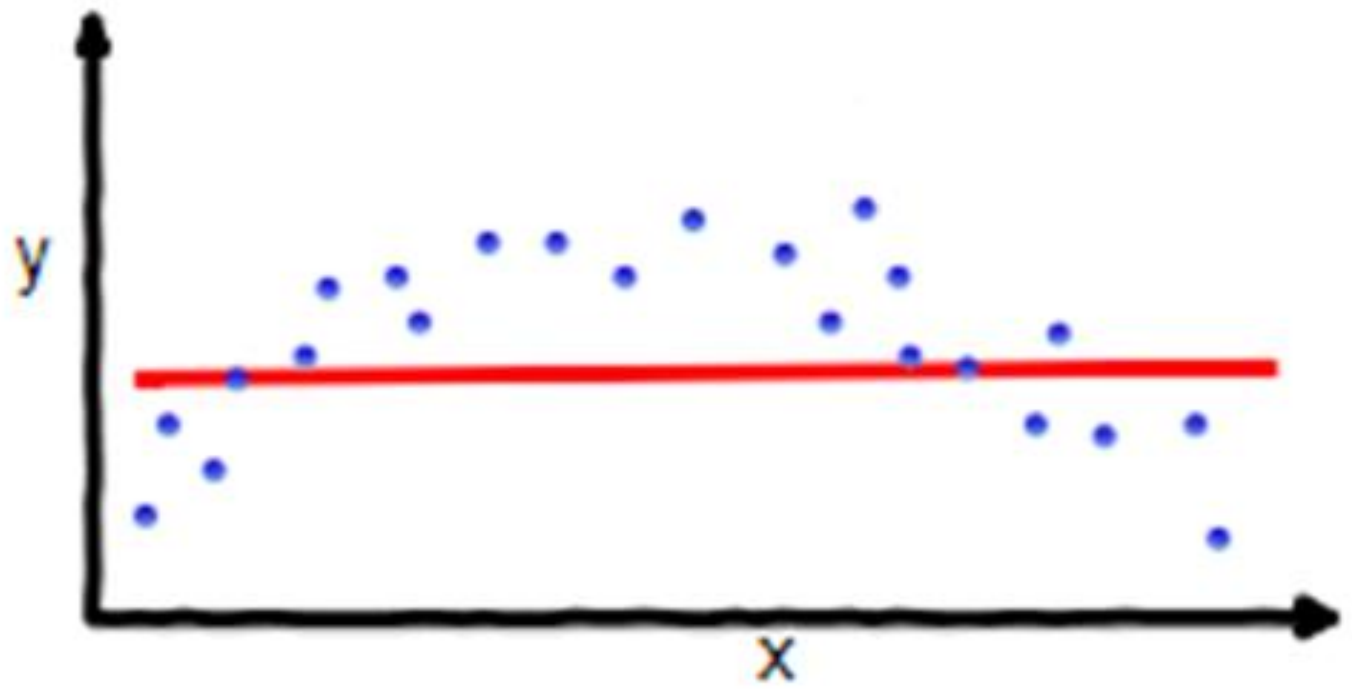
Bias-variance trade-off

- Ideally, we want a model with low bias and low variance, but this is very challenging in practice. In fact, this could be described as the goal of applied machine learning for a given predictive problem.
- The bias-variance trade-off
  - Reducing the bias can easily be achieved by increasing the variance.
  - Reducing the variance can easily be achieved by increasing the bias.
- This is a good conceptual framework for thinking how to choose models and model configuration (or hyperparameters).

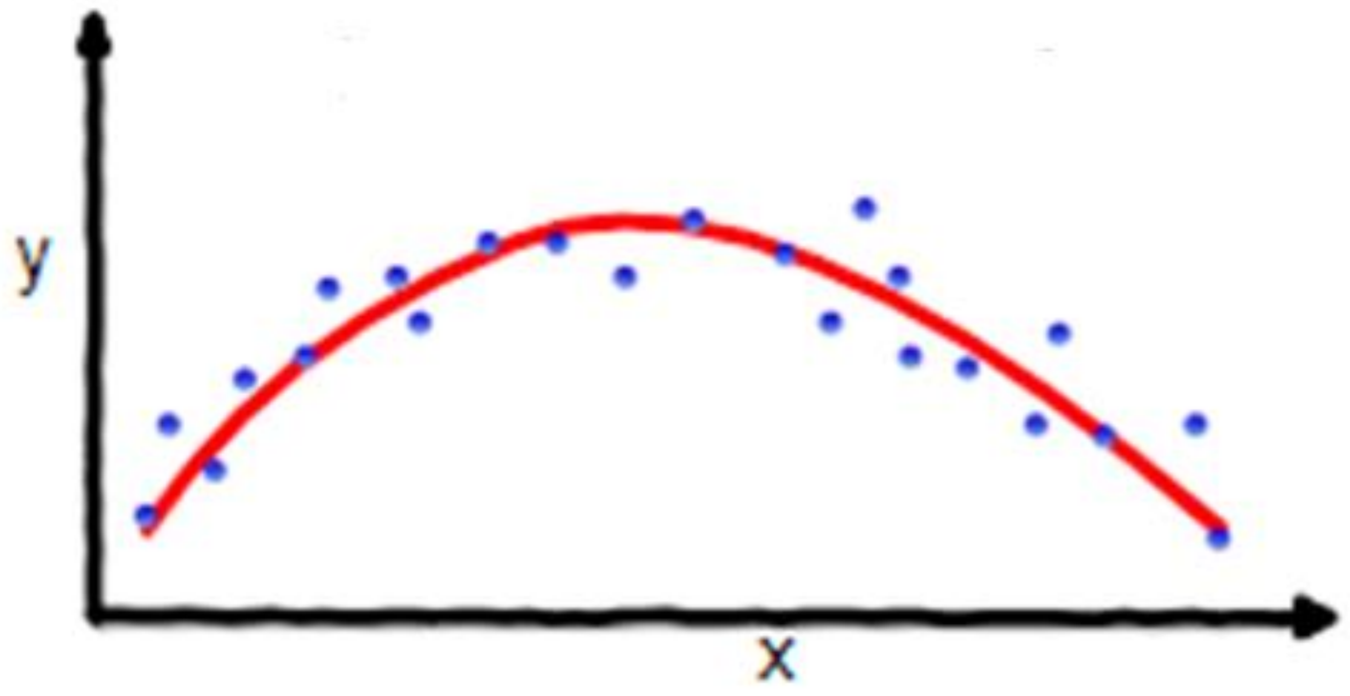
Bias-variance trade-off

- How to choose the N hyperparameter for curve fitting?

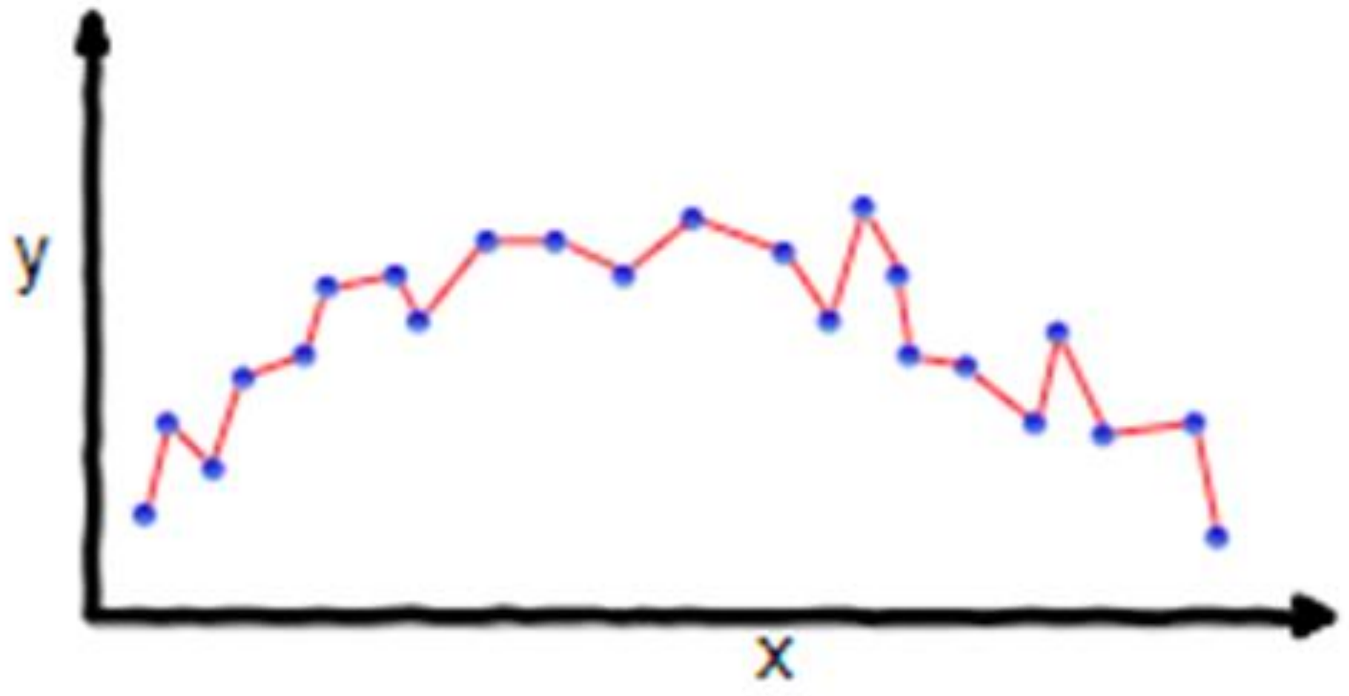
$Y = ax+b$



$Y = ax^2+bx+c$



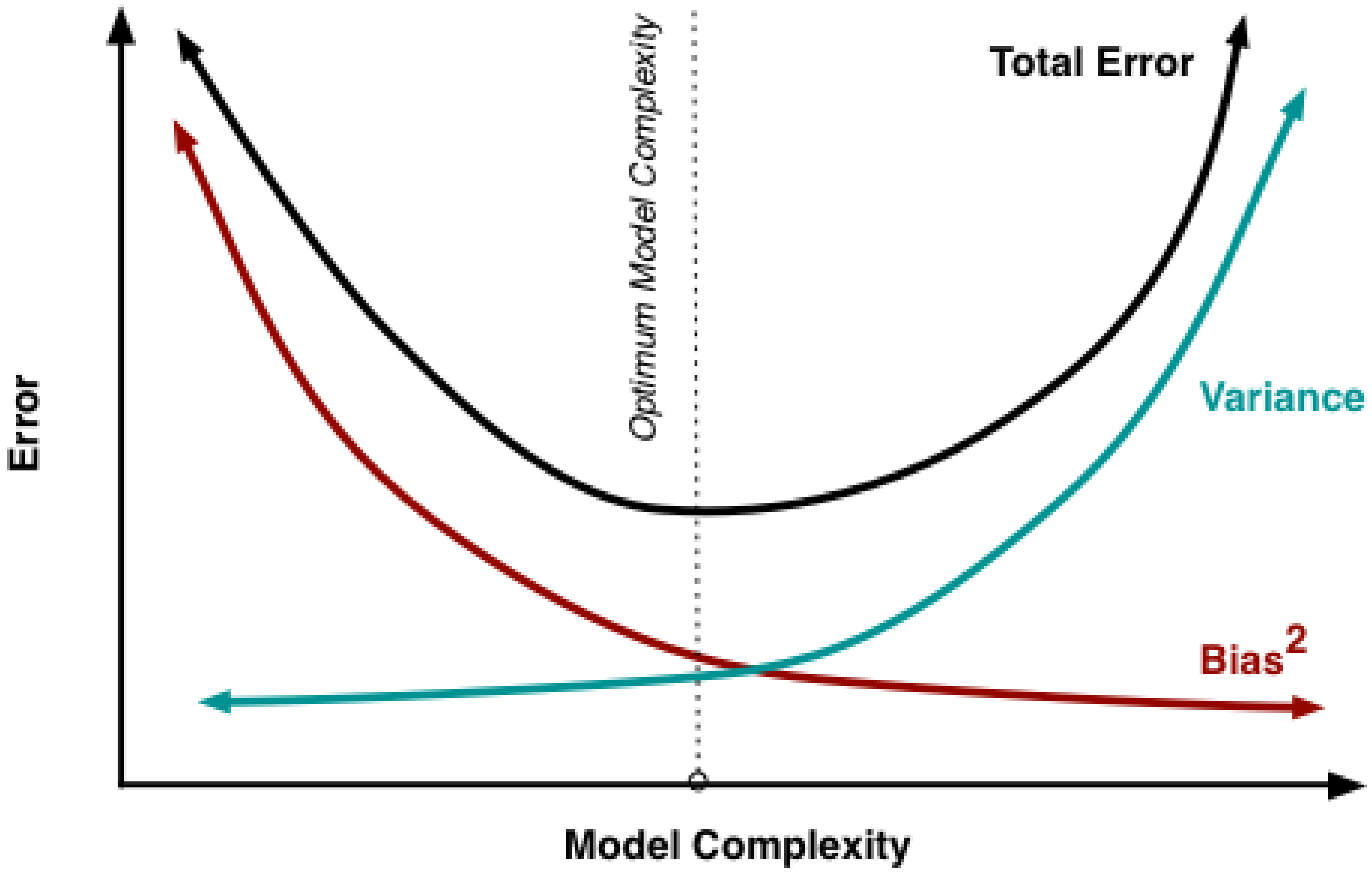
$Y = \sum_{i=0}^{N} a_i x_i$



Variance low → high

Bias high ← low

Bias-variance trade-off



When the model becomes more complex, the model bias decreases, and variance increases, and the total error firstly decreases and then increases.

The challenge is to find the optimum model complexity.

Estimating bias & variance

We can’t really compute the bias and variance of a model, as the actual y=f(x) is unknown.

But, we can estimate the bias and variance using training and testing error

Estimate of bias and variance

- Bias: training error
- Variance: testing error – training error

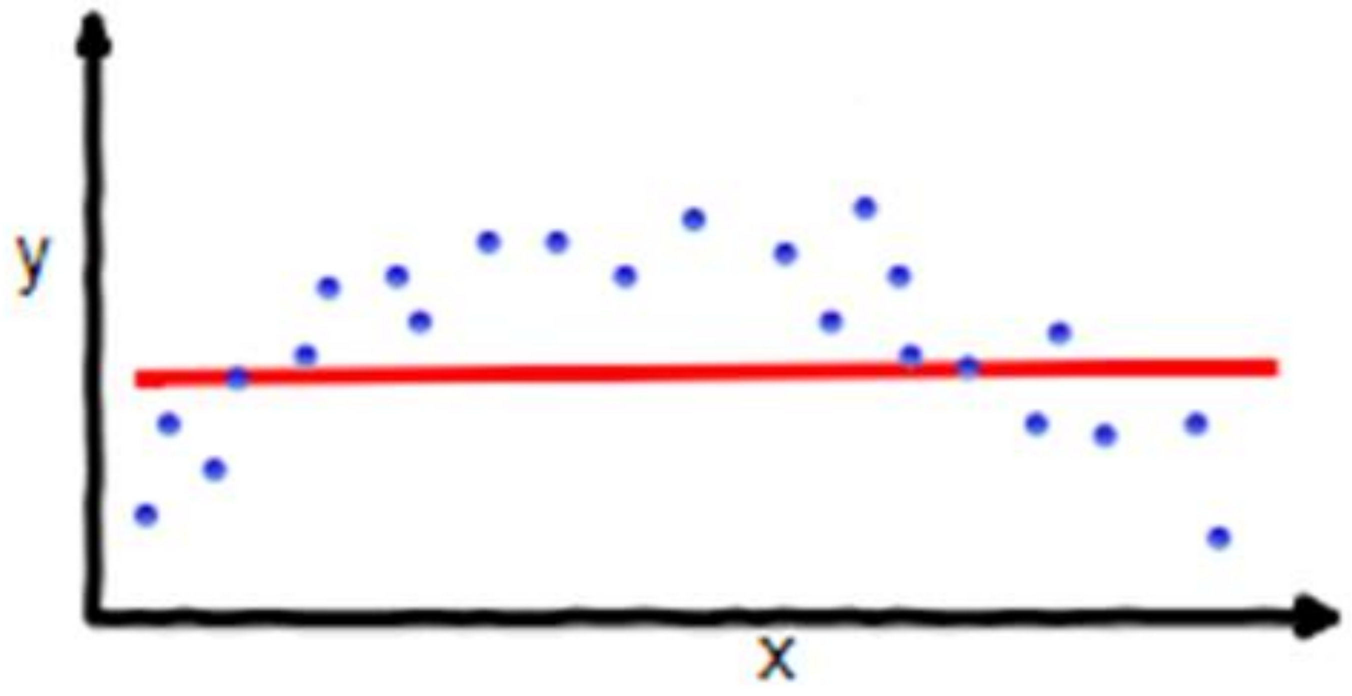
Using bias-variance to diagnose models

Bias \ Variance	Low	High
Low	Good balance	Overfitting
High	Underfitting	*Improvement needed*

Model diagnosis

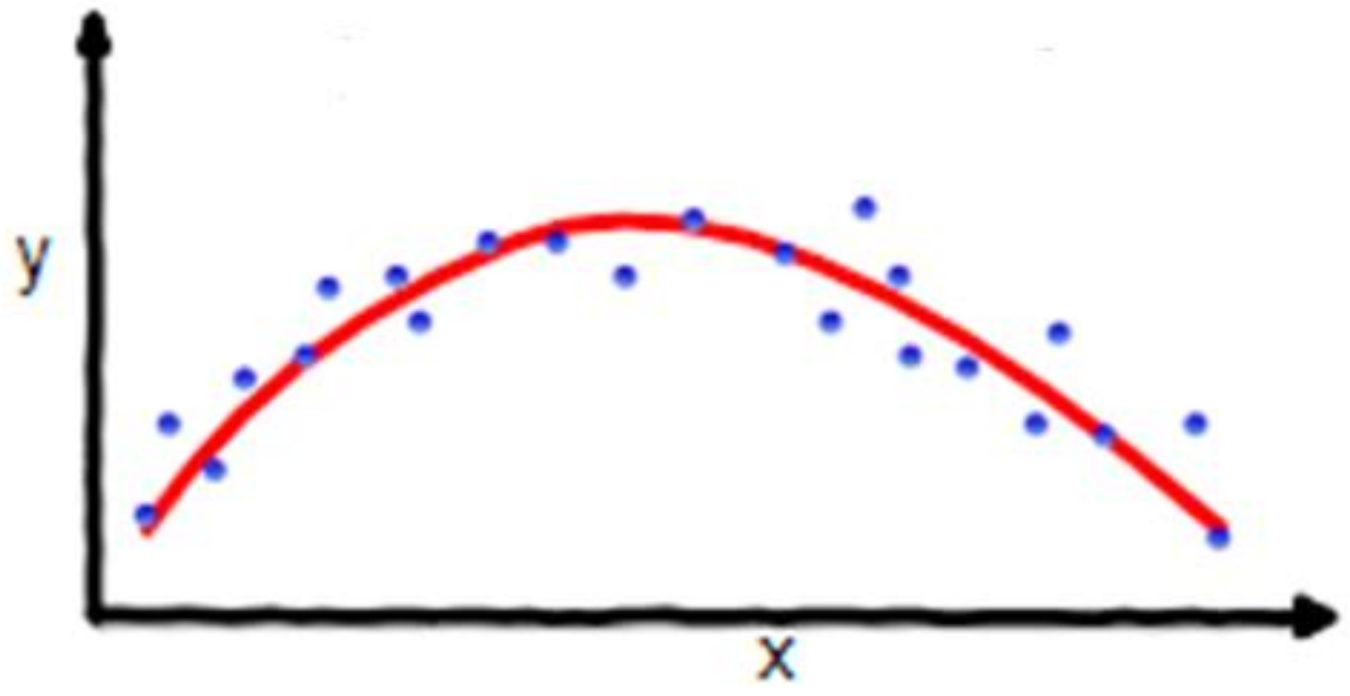
A

Y = ax+b



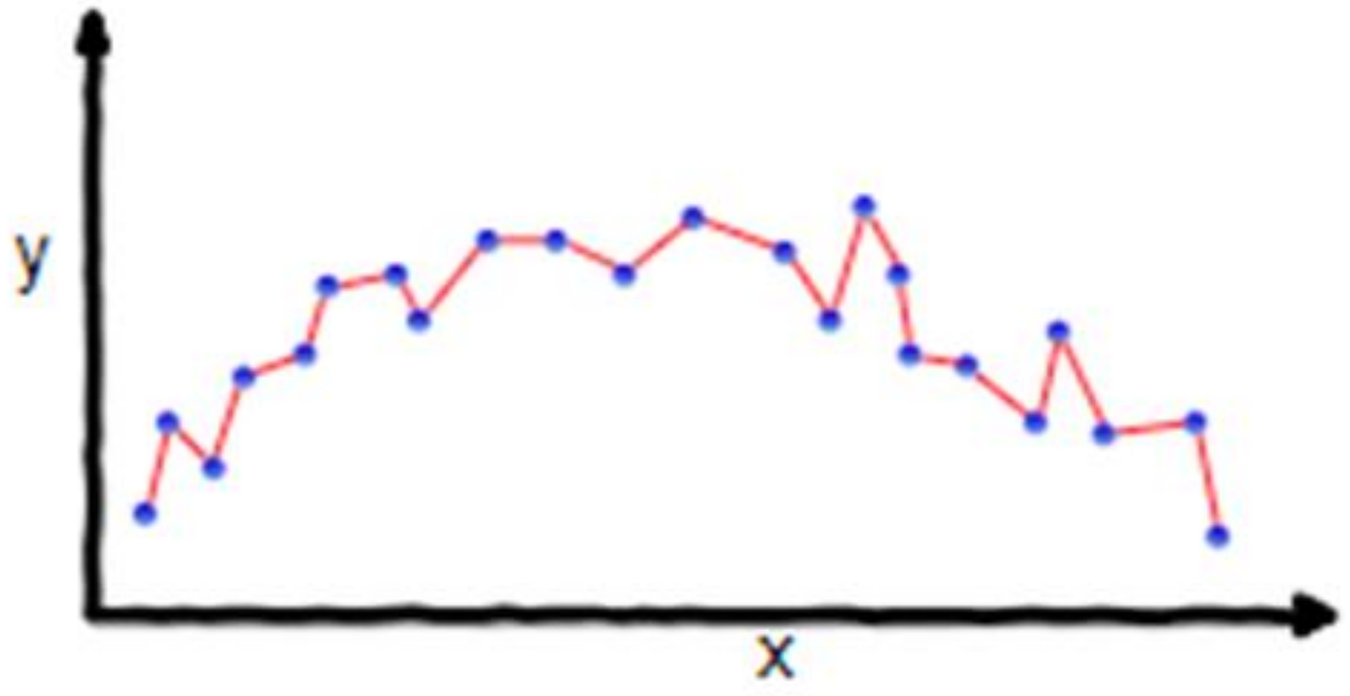
B

Y = ax^2+bx+c



C

Y =$\sum_{i=0}^{N} a_i x_i$



	A	B	C
Bias	High	Intermediate	Low
Variance	Low	Intermediate	High
Model diagnosis	Underfitting	Good balance	Overfitting

Model diagnosis – cat classification tasks

Example: Building an app to detect cats from photos

				You're done!
Training error	1%	15%	15%	0.5%
Test error	11%	16%	30%	1%
	High Variance	High Bias	High Bias High Variance	

Machine learning strategy

Machine Learning Workflow

Fitting the first model

Training error high

Yes

Bigger model

Train longer

High Bias

No

Dev error high

Yes

More train data

Smaller model

High Variance

No

You're Done!

Using the bias-variance conceptual framework as tools

Techniques for reducing bias

Increase model complexity

- Replace a simple linear regression model with a more flexible model, such as random forest or deep learning.
- Add more neurons or layers in a deep learning model.

Modify input features

- Inspect your training data to understand which examples your model is not doing well on.
- See if you can modify data features to eliminate these errors.

Add more training data

- This technique helps with variance problems, but it usually has no significant effect on bias.

Techniques for reducing variance

Add more training data

- This is the simplest and the most reliable way to address variance, so long as you have access to significantly more data.

Reduce model size/complexity

- Replace a large neural network with a random forest.
- Add regularization to your neural network.
- Decrease neural network size.

Feature selection to decrease number/type of input features

- This technique might help with variance problems, but it might also increase bias.
- When using deep learning, there has been a shift away from feature selection, and we are now more likely to give all the data to the algorithm and let the algorithm sort out which ones to use.

Benchmarking with baseline model

- Let’s consider predicting the house price in London.
- You want to build a regression model that predicts house price given the number of bedrooms and the location.
- Which regression model do you start with?

(a) Linear regression

(b) Random forest

- Note: we always start from a simple model

Data leakage

Data leakage in machine learning

- The goal of machine learning is to develop a model that makes accurate predictions on new data
- Principle: We need to estimate the model performance on unseen data, which can’t be used in any stage of model training
- When this principle is broken, data leakage occurs: it leads to invalid predictive models, or overly optimistic models.
- Definition: data leakage is when the data you are using to train a machine learning algorithm happens to have the information you are trying to predict. This invalidate the estimated performance of the model

Examples of data leakage

- Data leakage occurs if we use the training data as the testing data – no train-test split
- In this case: training accuracy = 99.9%; testing accuracy = 99.9%

- Tip: when you achieve performance that seems too good to be true, this is a potential sign of data leakage. But the reverse is NOT true – “when there is a data leakage, the accuracy is always very high”.

More examples of data leakage

1. We randomly split the original data (OD) into two subsets: A (70%) and B (30%).
2. We use the A dataset to train a model (M).
3. We again randomly split OD into two subset: C (70%) and D(30%)
4. Then we apply M on D, getting an accuracy of 80% and report 80% is the finalised model performance.

Is this a case of data leakage?

More examples of data leakage

1. This is a task of predicting house price in London, using data from two companies, Rightmove and Zoopla.
2. We use the data from Rightmove to train a neural network model (M), getting an accuracy of 90%.
3. Then, we use model M to the housing data from Zoopla, getting an accuracy of 60%.

Is this a case of data leakage?

More examples of data leakage

1. Again, you are working to predict house price in London, using data from two companies, Rightmove and Zoopla.
2. This time, you combine all data from the two companies and get dataset OD. You randomly split OD into two subset, A (70%) and B (30%).
3. You train the model using A and evaluate the model performance using B.

Is this a case of data leakage?

More examples of data leakage

1. You are working to predict house price in London, using data from Zoopla.
2. You have removed all duplicate items from the data.
3. You normalise the house price using the average value of the whole dataset. Then, you split the data into A (70%) and B (30%).
4. You train the model using A and evaluate the model performance using B.

Is this a case of data leakage?

Time series datasets

1. You are working to predict future house price in London, using three-year data (2019/20/21) from Zoopla.
2. This time, there are two ways to do train-test split: first, shuffle the data and split it into A (66%) and B(33%); second, use 2019/2020 data as training data and the 2021 data as testing data.

Which split is more reasonable? Why?

Techniques to minimise data leakage

1. **Hold back a testing dataset** for final sanity check of your developed models
2. **Perform data preparation without using testing dataset**. If you are using cross validation, perform data preparation within your cross validation folds
3. **Add Noise**. Add random noise to input data to try and smooth out the effects of possibly leaking variables.
4. **Use Pipelines**. Heavily use pipeline architectures that allow a sequence of data preparation steps to be performed within cross validation folds, such as the caret package in R and Pipelines in scikit-learn.

Summary

- Basic error analysis
- Machine learning strategy
- Data leakage

Workshop

- Weekly quiz on Moodle: please finish them before the workshop and we will discuss the quiz in the workshop
- Python notebooks for workshop: will be ready by 5pm Thursday.
- See you in the workshop on Friday 1-3pm