Cross Validation Techniques

Jayin Khanna
Roll no: 2210110324
MAT485: Statistical simulation and Computing

1. Need for Cross Validation
2. Bias vs Variance Trade-off
3. Examples
4. LOOCV Technique
5. K-fold CV Technique

6. Choosing optimal K value

7. Stratified Cross-Validation

8. Code Illustration

Need for Cross-Validation

The fundamental problem of prediction:
we never observe how our model performs on unseen data

The usual Algorithm:

- Dataset → Fit model → Report training error
- But the model was *optimised* for that particular sample dataset (selection bias). Not the entire true population.
- So we need a model that can learn the general pattern from this sample dataset alone
- Normally we dive the data into test and training set only;
- Cross-validation simulates unseen data to estimate true risk.

- **Algorithm:** Randomly split the dataset into two parts.
  - **Training Set (~70-80%):** Used to build/train the model.
  - **Test Set (~20-30%):** Used *only once* to evaluate the final model's performance.
- **Advantage:** Simple, fast, gives an unbiased estimate of performance on unseen data.
- **The Fatal Flaw (High Variance):**
  - The estimate is highly dependent on *which* data points end up in the training and test sets.
  - If our test set does not have too many points, our test set might just be lucky or unlucky

Bias vs Variance trade-off

Effect of the training data

- Goal: We will attempt to learn a function f that is as close to the True f as possible
- Mean squared error (MSE) is the average squared difference of a prediction f_hat(x) from its true value y.

$$\mathrm{MSE} = \mathbb{E}[(y - \hat{f}(x))^2]$$

- The function f_hat that is learned depends upon the training data given
  a. f_hat can be different for different training data, and f_hat(x) can change even though x is fixed
  b. f_hat is a random variable affected by the randomness in which we obtain training data

Bias and Variance

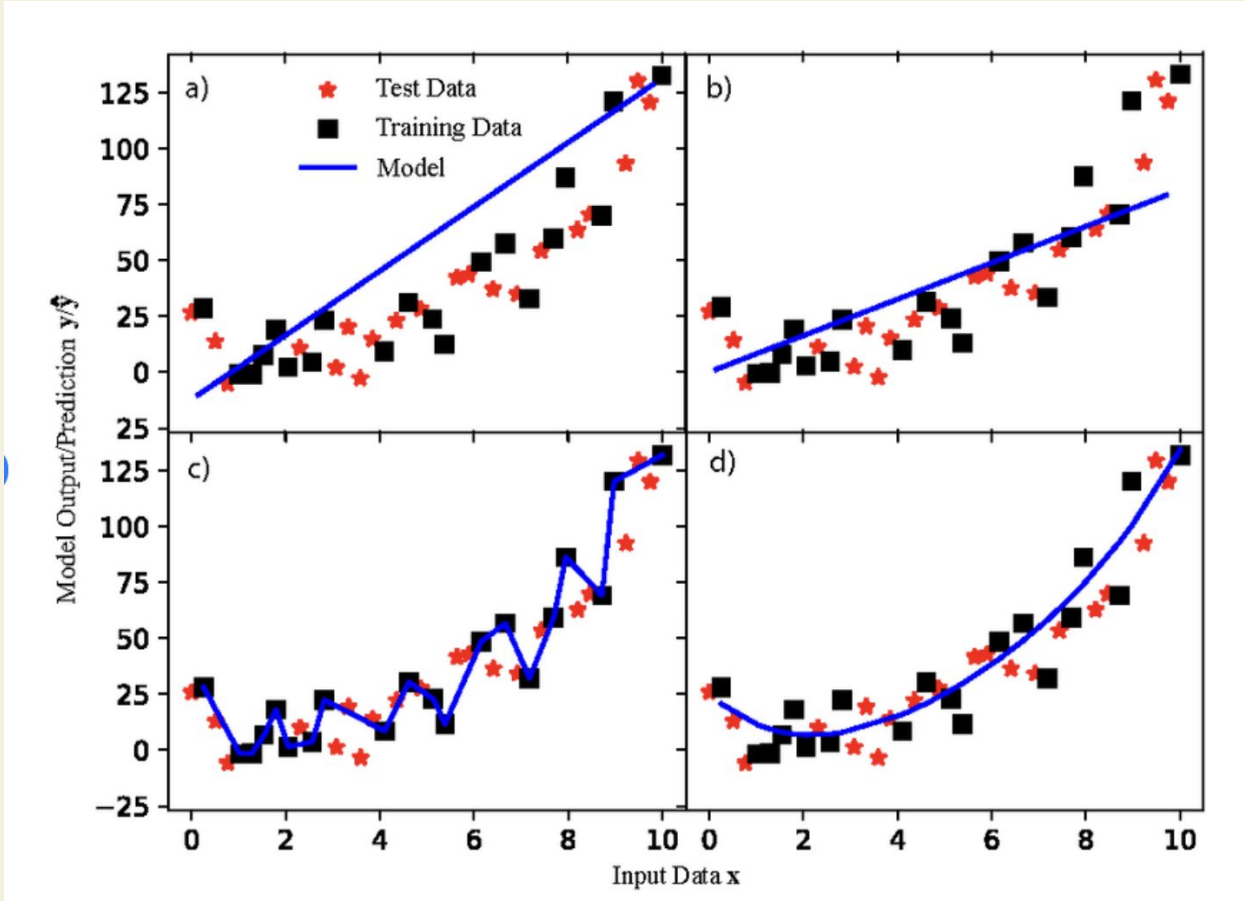
- Since f_hat(x) is a random variable, we can talk of its expectation (over different realizations of training data)
- Bias: difference of the average prediction (over different realizations of training data) to the true function f(x), for a given unseen (test) point x

$$\mathrm{bias}[\hat{f}(x)] = \mathbb{E}[\hat{f}(x)] - f(x)$$

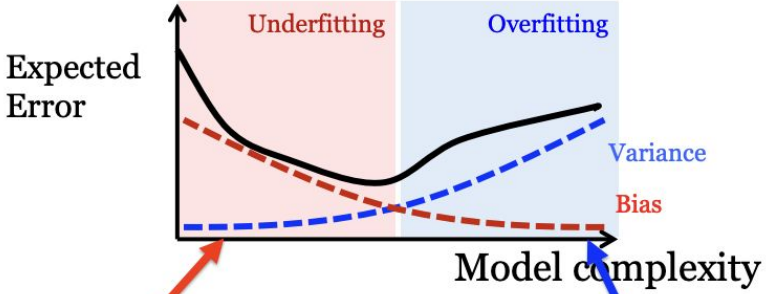
- Variance: mean squared deviation of f(x) from its expected value E[f(x)] over different realizations of training data

$$\mathrm{var}(\hat{f}(x)) = \mathbb{E}[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2]$$

Illustration for Bias Variance trade off

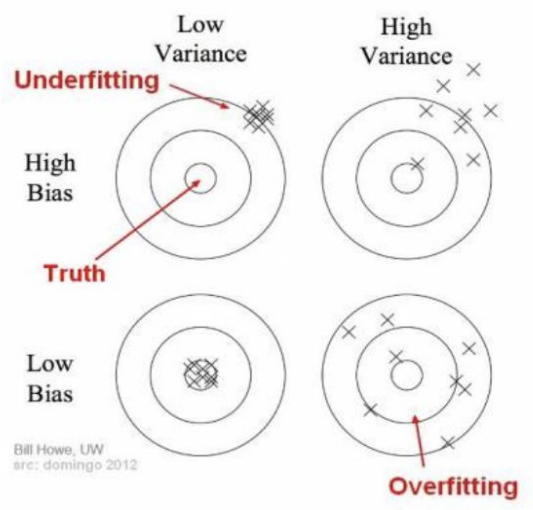


Underfitting and Overfitting



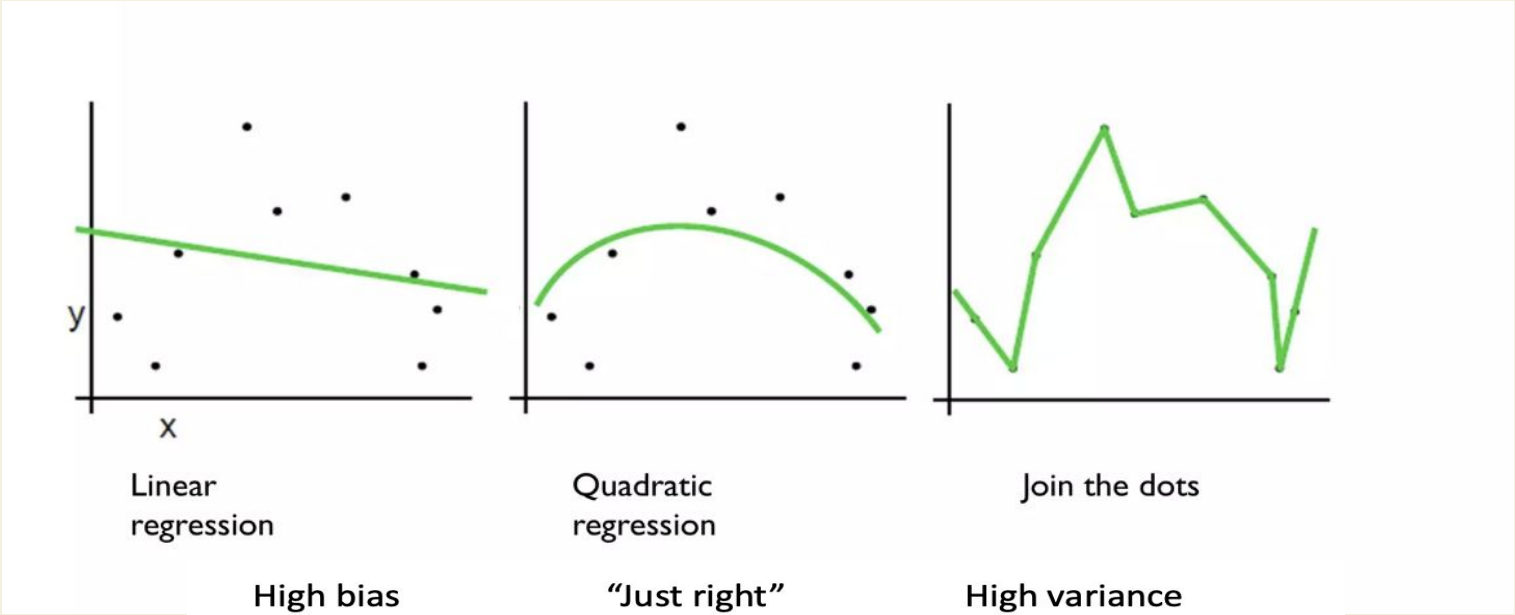
Simple models:
High bias and low variance

Complex models:
High variance and low bias



Crosses represent the function learned over different realizations of the training data

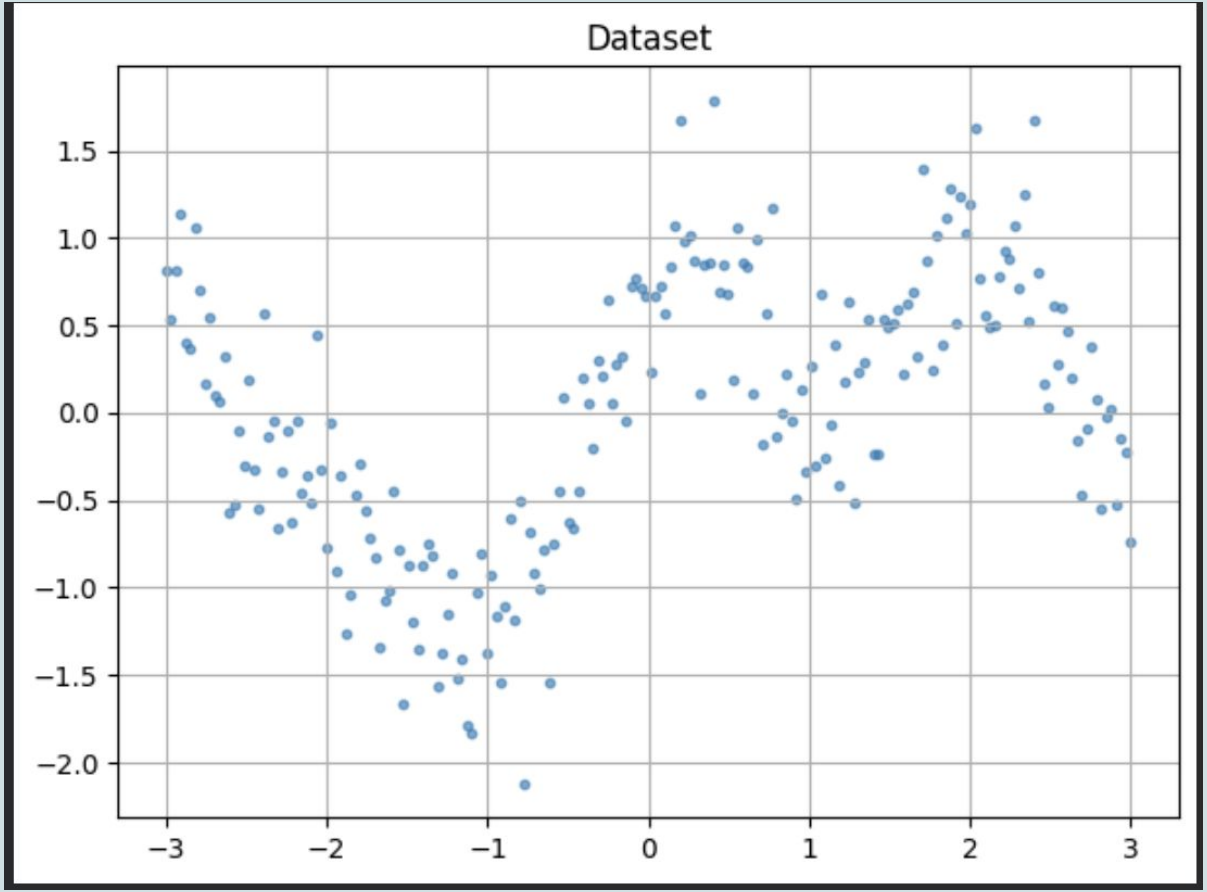
Which is best?



Two important questions:

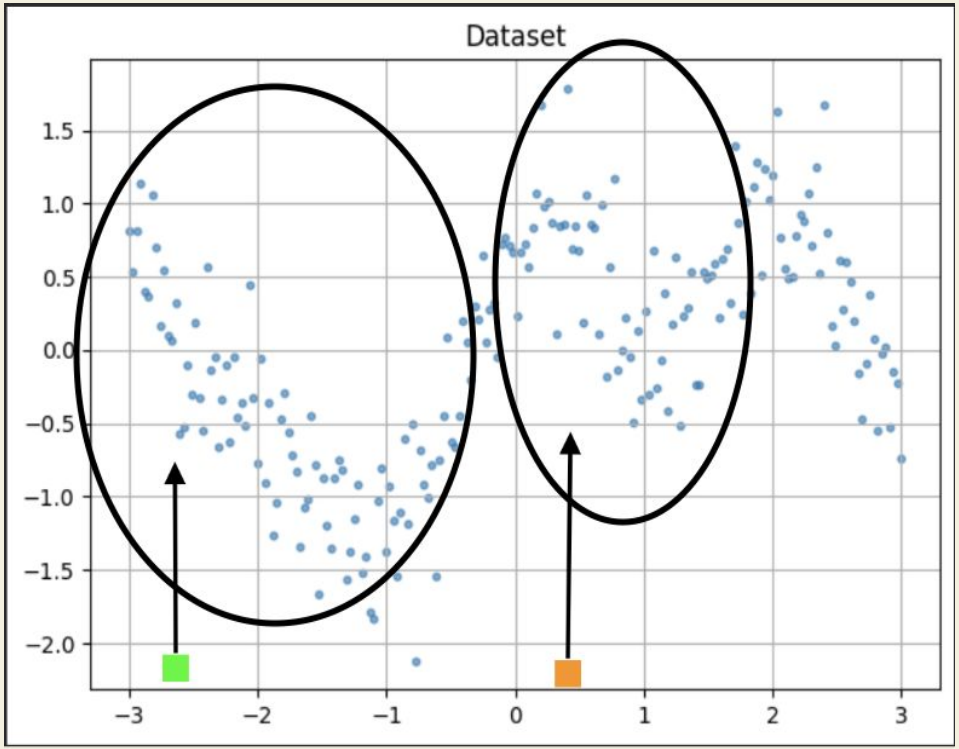
1. Can we use the 70-30 data split methodology here?
2. How can we come up with a better method that helps us decide which model is best based on which performs best under multiple rounds of surprise?

Example from generated dataset

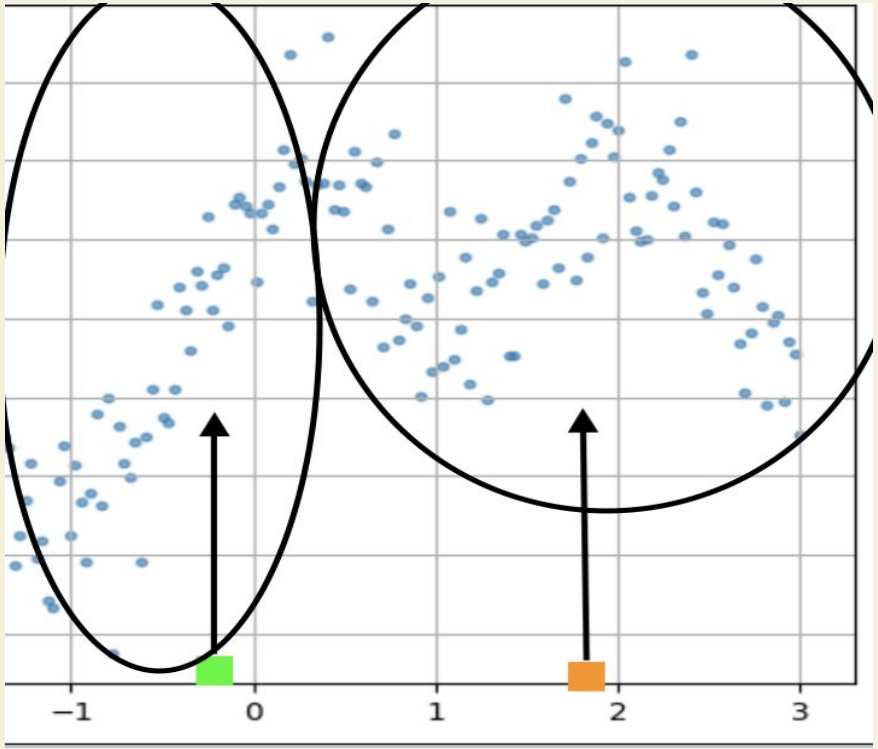


Motivating example: In which split would Linear regression fail?

CASE I



CASE II



The Algorithm

Procedure to estimate prediction error by LOOCV (leave-one-out) cross validation:

Step 1

A single observation is used for the validation set; the remaining (n-1) observations make up the training set.

Step 2

Model is fit on (n-1) training observations.

Step 3

The error on the heldout observation is an unbiased estimate for the error on new data.

Compute the prediction error:

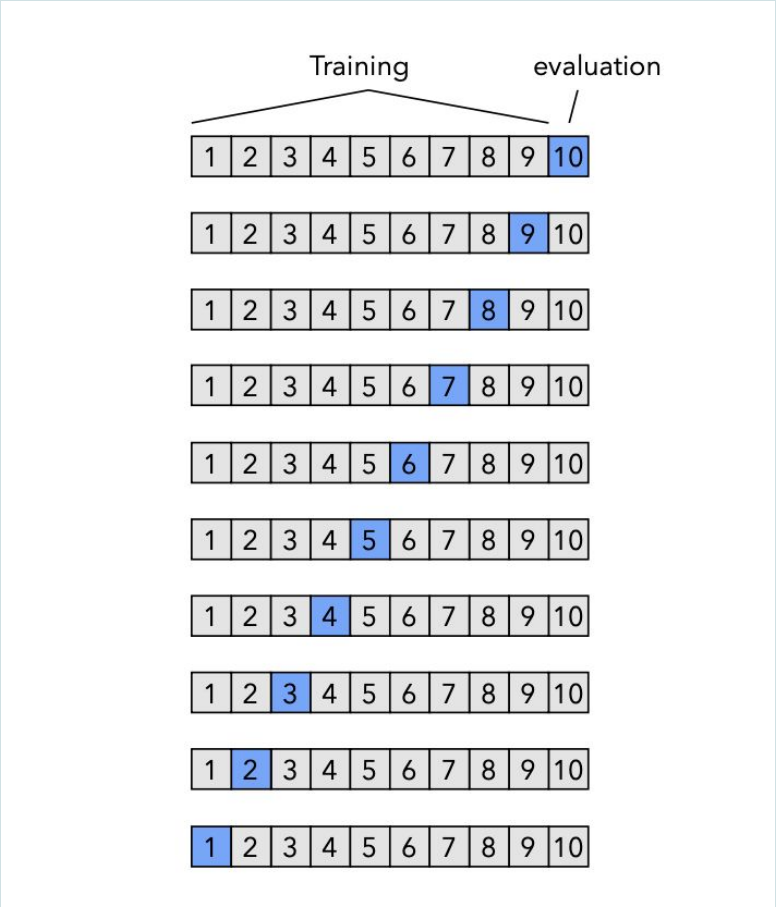
e_k = y_k − y_k_pred

Step 4

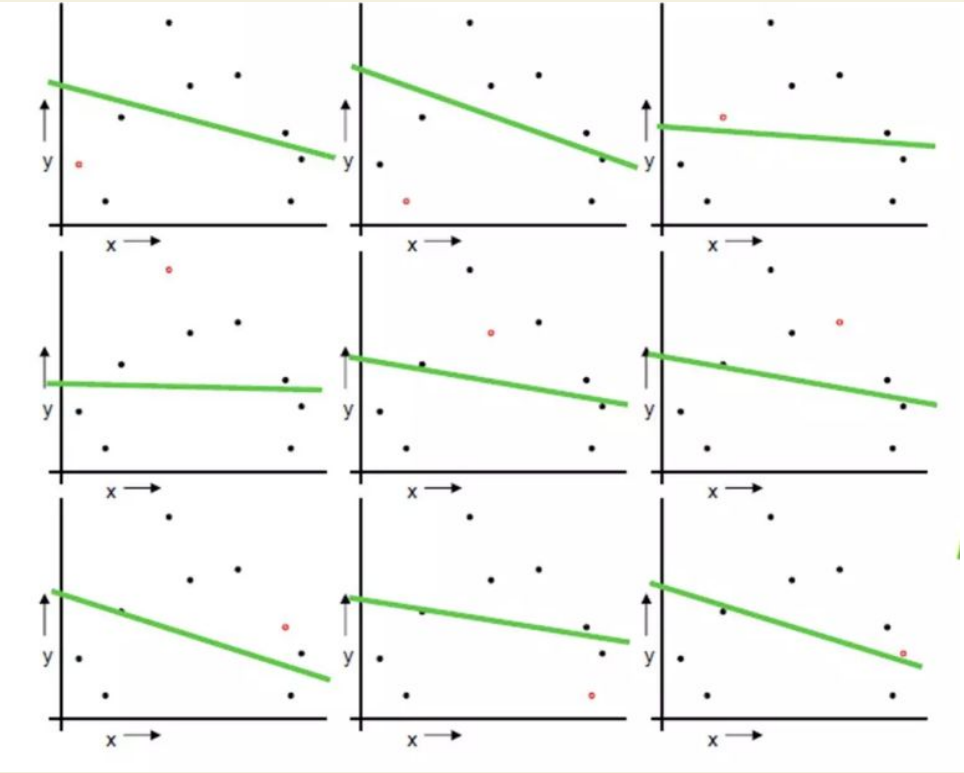
Estimate the mean of the squared prediction errors

$$\hat{\sigma}_\varepsilon^2 = \frac{1}{n}\sum_{k=1}^{n} e_k^2.$$

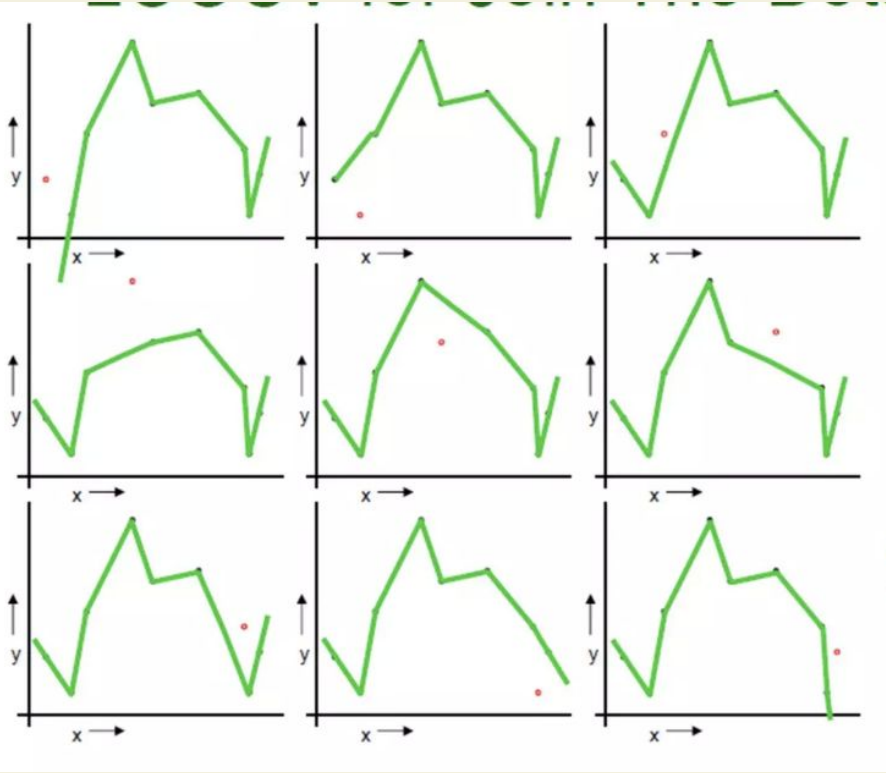
Illustration of Leave-one-out Cross Validation



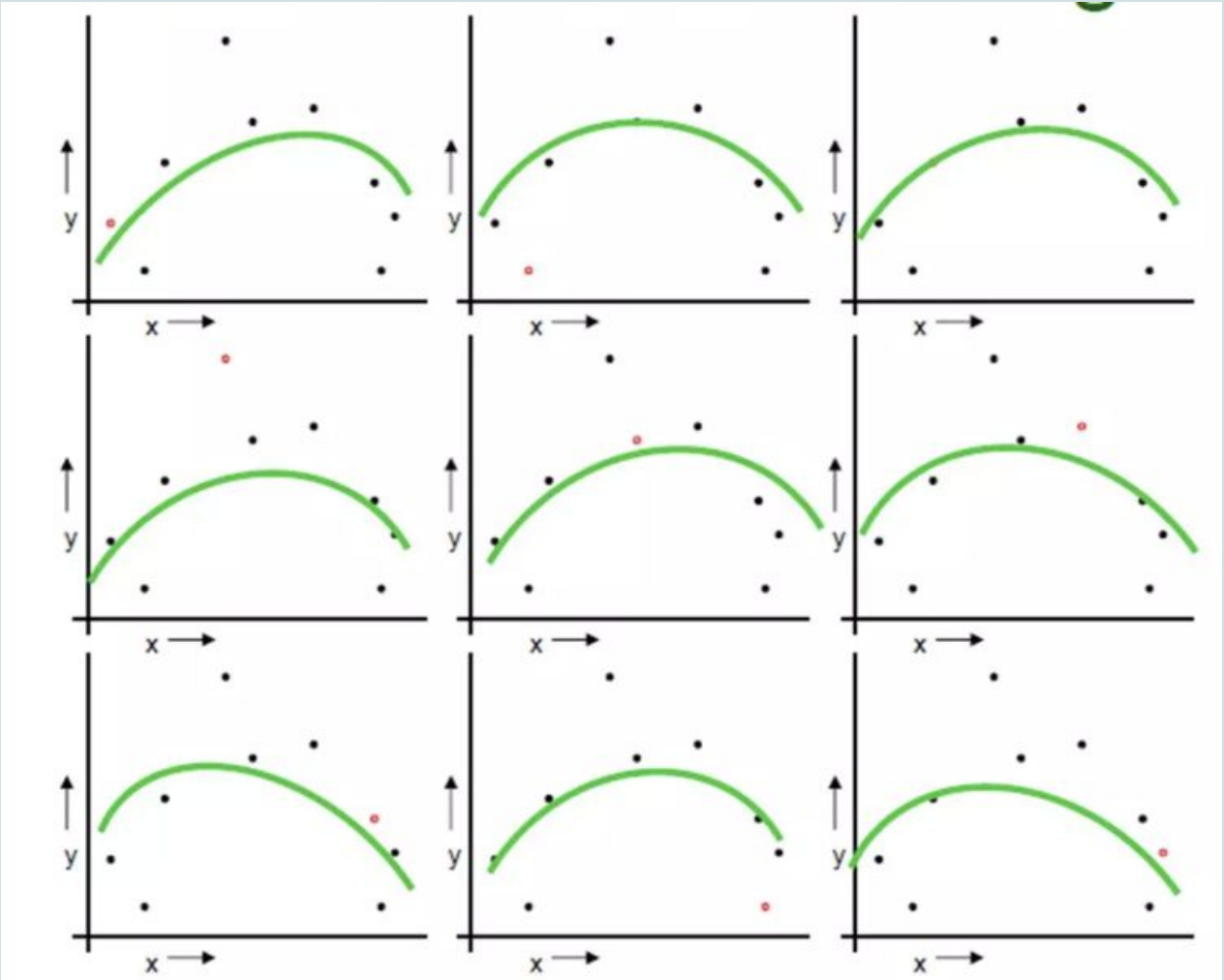
Model selection with MSE_LOOCV



MSE_LOOCV=2.12



MSE_LOOCV=3.33



MSE_LOOCV= 0.962

Beyond LOOCV: The Flexibility of k-Fold Cross-Validation

The Limitations of LOOCV:

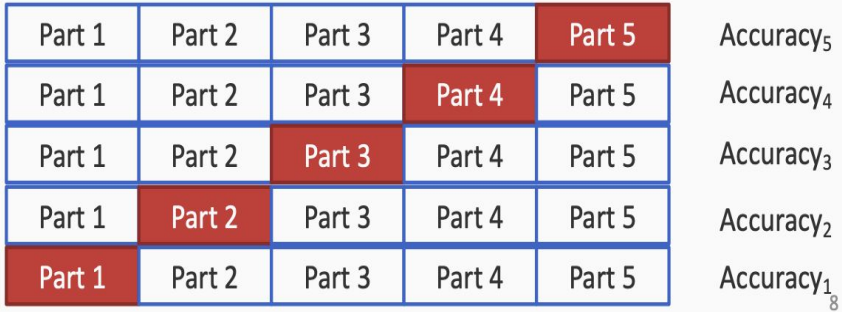
1. High computational cost:
   For a dataset of 10,000 points, LOOCV requires 10,000 model fits. This is very infeasible for complex models like SVMs or Neural Networks

2. High Variance in estimation
   Each training set in LOOCV is almost identical to the others (differing by just one point). This means the performance scores S_i are very correlated, leading to a high-variance estimate of the test error.

K-Fold Cross-Validation

if k = 5, then you have 5 parts T1, . . . , T5
you would run 5 training runs

1. Train on T1,T2,T3,T4 evaluate on T5.
2. Train on T1,T2,T3,T5 evaluate on T4.
3. Train on T1,T2,T4,T5 evaluate on T3.
4. Train on T1,T3,T4,T5 evaluate on T2.
5. Train on T2,T3,T4,T5 evaluate on T1.



Variance Reduction (Stability):

- A single validation set gives us a **single point estimate** of the test error, which is highly sensitive (as seen in the previous slides).
- By averaging K different error estimates, we reduce the variance of our final estimate, producing a more reliable metric of model performance.

Bias Reduction (Data Efficiency):

- **The Problem with Simple Splits:** If we split data 50/50, we train on only half the data. Statistical models generally perform worse with fewer samples, leading to an **overestimation of the test error** (optimistic bias).
- **The K-Fold Solution:** With K=10, we train on 90% of the data in every iteration. This allows the model to approximate the performance of the full model (trained on all n) much more closely.

$$\mathrm{Var}(\bar{X}) = \rho\sigma^2 + \frac{1-\rho}{k}\sigma^2$$

Choosing the optimal value of K

High Values of K (Approaching LOOCV)

- **Pros (Low Bias):** Since we train on N-1 points, the model is nearly identical to the one trained on the full dataset. The error estimate is very accurate (low bias).
- **Cons (High Variance):**
  - Because the training sets are almost identical (they share N-2 points), the error estimates are **highly correlated**. Averaging highly correlated quantities does not reduce variance as effectively as averaging independent ones.
  - **Computation:** Very expensive (N separate training runs).

Low Values of K (e.g., K=2 or 3)

- **Pros (Computational Efficiency):** Very fast to compute.
- **Cons (High Bias):** We only train on a fraction (e.g., 50%) of the data. Models generally perform worse with less data. Therefore, the CV score will **overestimate** the true error rate.

4. Optimal value: Why K=5 or K=10?

- Empirically, K=5 or K=10 has been shown to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance.

Stratified Cross-Validation

Motivation

The Problem: What if there is Imbalanced Data?

Example- Imagine we have to build a model to detect a rare disease. The dataset has 100 patients:

- 95 are Healthy (Class 0)
- 5 have the Disease (Class 1)

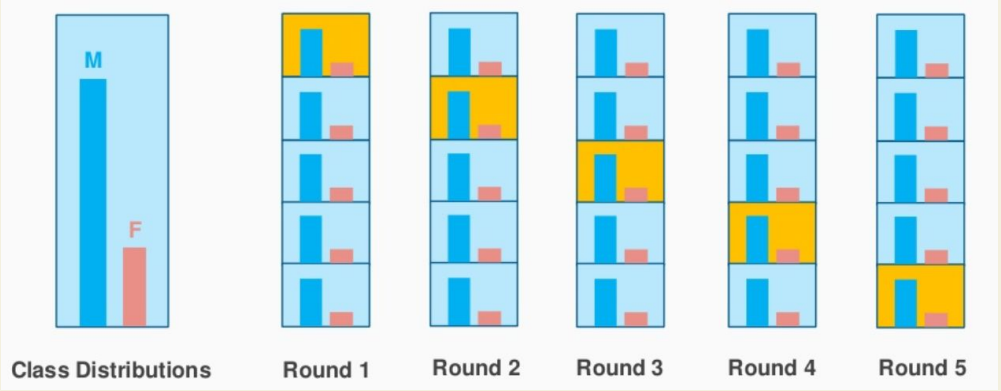
If we do a standard **Random K-Fold** (let's say K=5), randomly grabbing 20 patients for each fold.

- **The Risk:** By pure random chance, one of the folds might contain **all 5 sick patients**.
- **The Result:** When that fold is used for *testing*, the *training* set (the other 4 folds) will have **zero** sick patients. **The model will learn nothing about the disease.** Conversely, if that fold is used for *training*, **the *test* set has no sick patients to evaluate the model on.**

The Solution: Stratification

Stratified Cross-Validation forces the ratio of classes in each fold to be the same as the ratio in the full dataset.

- It looks at the target variable Y
- It ensures that every fold of 20 patients contains exactly **19 Healthy** and **1 Sick** patient.
- It preserves the **class distribution**.



Thank You

All Code Illustrations can be found at:
https://colab.research.google.com/drive/1z9Kcv4BdTk4GHQ8riTtTfN_TnbU7HGgx?usp=sharing