Cross-Validation

In previous chapters we discuss the difference between the **test error rate** and **training error rate** and that the **test error** is always higher than the **training error** and made these following points :

- The **training error rate** can be easily calculated by comparing the predicted values to the true labels
- While the **test error rate** is complicated to calculate
- If the training error rate is low and test error rate is high it might indicate that the model is Overfitting

Cost Function Vs Test and Training error rate

First the **Cost function** is a measure of the difference between the predicted values and actual values

- Used to **evaluate** the accuracy of a model's predictions
- Guide the process of **adjusting** the model's parameters during training
- **Minimize** the difference between the predicted and actual values

the **loss function** measures how well the model preforms on a single training example while the **cost function** is on the entire training data set

Second the **test** and **training** error rate they are just the rate of the wrong predictions or Miss-classification divided by the total test or training examples :

$$\text{Training error rate } = \frac{\text{Number of incorrect predictions}}{\text{Total number of training examples}}$$

$$\text{Test error rate} = \frac{\text{Number of incorrect predictions}}{\text{Total number of test examples}}$$

Note : the test examples are new unseen observations not used in the fitting process

Cross-Validation approaches

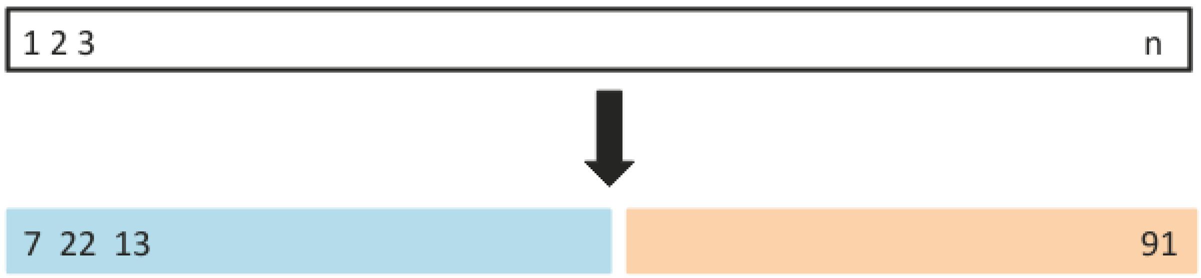
In this section we gonna cover the approaches of cross-validation and the main differences between them , these are the following :

1. The Validation Set Approach
2. Leave One Out Cross-Validation
3. K-Fold Cross-validation

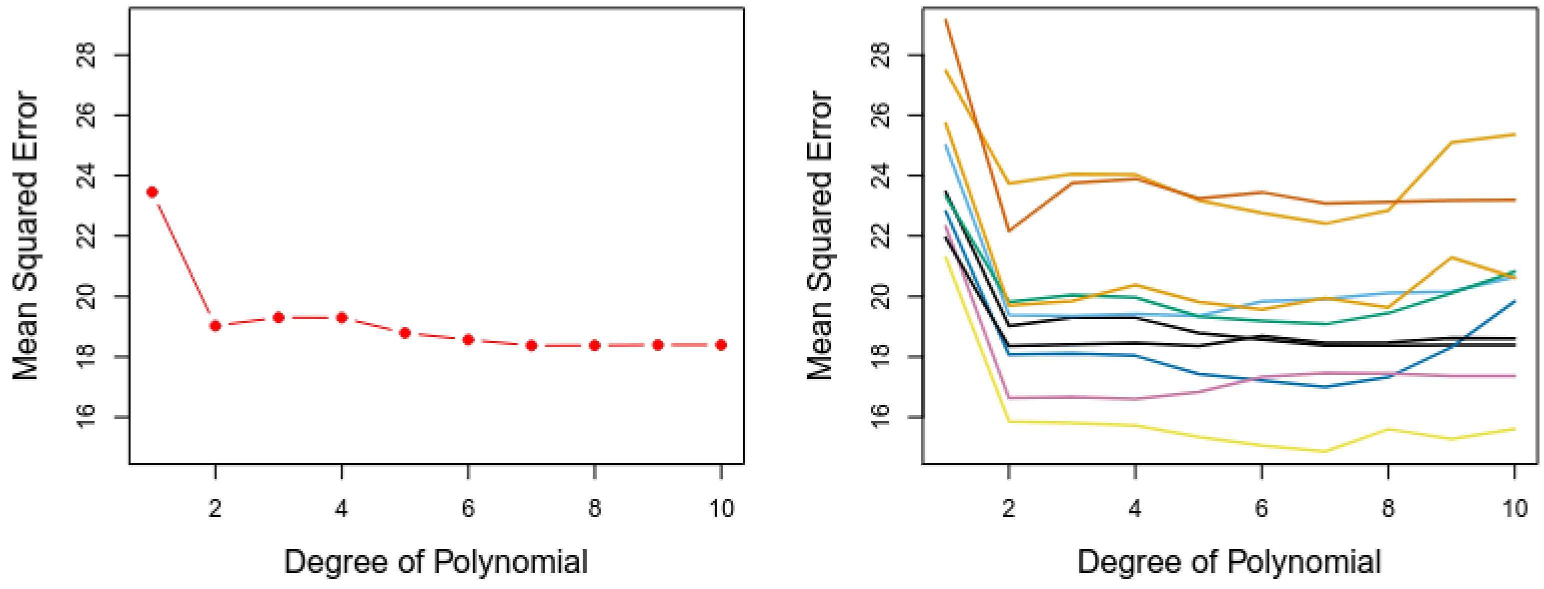
The Validation Set Approach

This approach simply involves randomly dividing the set of observations into two parts :

- **Training set**
- **Validation set** (Hold-out set)



- Splitting the set of Observations randomly into training and validations sets



- In this example we used the Validation set approach for multiple Degrees of Polynomial Regression
- After running it many times we can clearly notice that the quadratic degree is the best one that fits the data increasing it further doesn't results in substantial difference in the MSE

Pros:

- Quick and simple to implement on large and small data sets

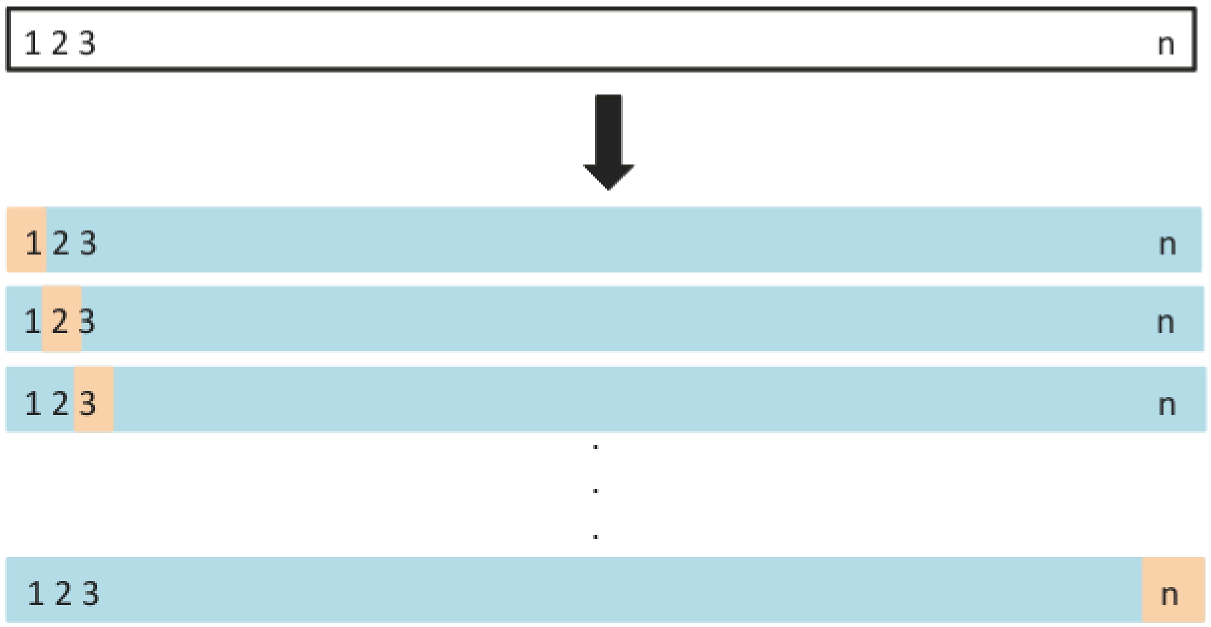
Cons:

- May **overestimate** the test error rate
- Struggles with high variance
- Results depends heavily on the split

Leave One Out Cross-Validation

Leave one out cross-validation also known **LOOCV** resembles the set approach validation , With splitting the data :

- Into two parts
  - **Training set** $\rightarrow n - 1$
  - **Validation set** $\rightarrow 1$
- Repeated for all samples n



Example :
Taking a single observation (x_1, y_1) to validate and the rest $(x_2, y_2) \ldots (x_n, y_n)$ as the training set to fit the model and doing the same for the rest of samples (x_n, y_n)

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$

Pros:

- **Exhaustive validation** the **LOOCV** results in a comprehensive and accurate evaluation since it covers every possible test scenario
- Very low variance and bias it captures all the complex relationships

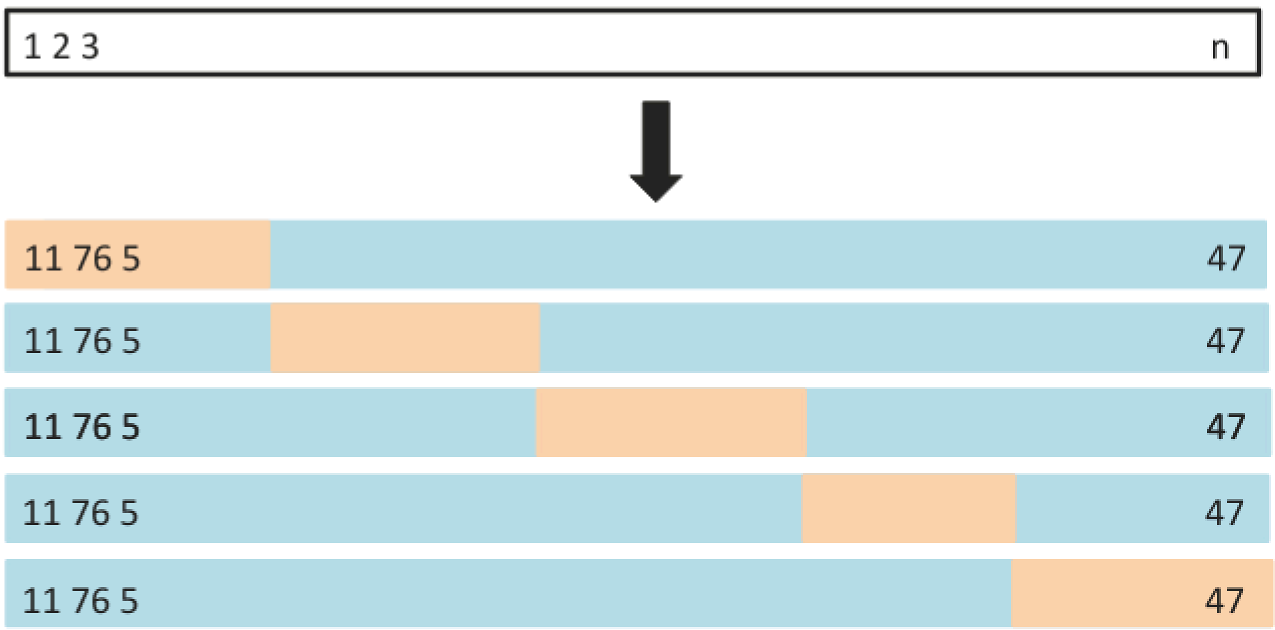
Cons:

- Heavy to compute and not scalable with large data sets
- Sensitivity to outliers which can results in unreliable performance

K-Fold Cross-Validation

An alternative approach to **LOOCV** approach where instead of taking $n-1$ as a training set and single observation for validation , **K-Fold** takes :

- Randomly Split the data into groups (**Folds**) ,K Folds
- Taking $K-1$ as a **training set**
- and the remaining fold as a **validation set**
- Its a special case of **LOOCV** where $K=n$
- Taking the mean and variance of the scores to estimate the model performance model



- Usually $k=5$ or $k=10$ as number of folds

Pros :

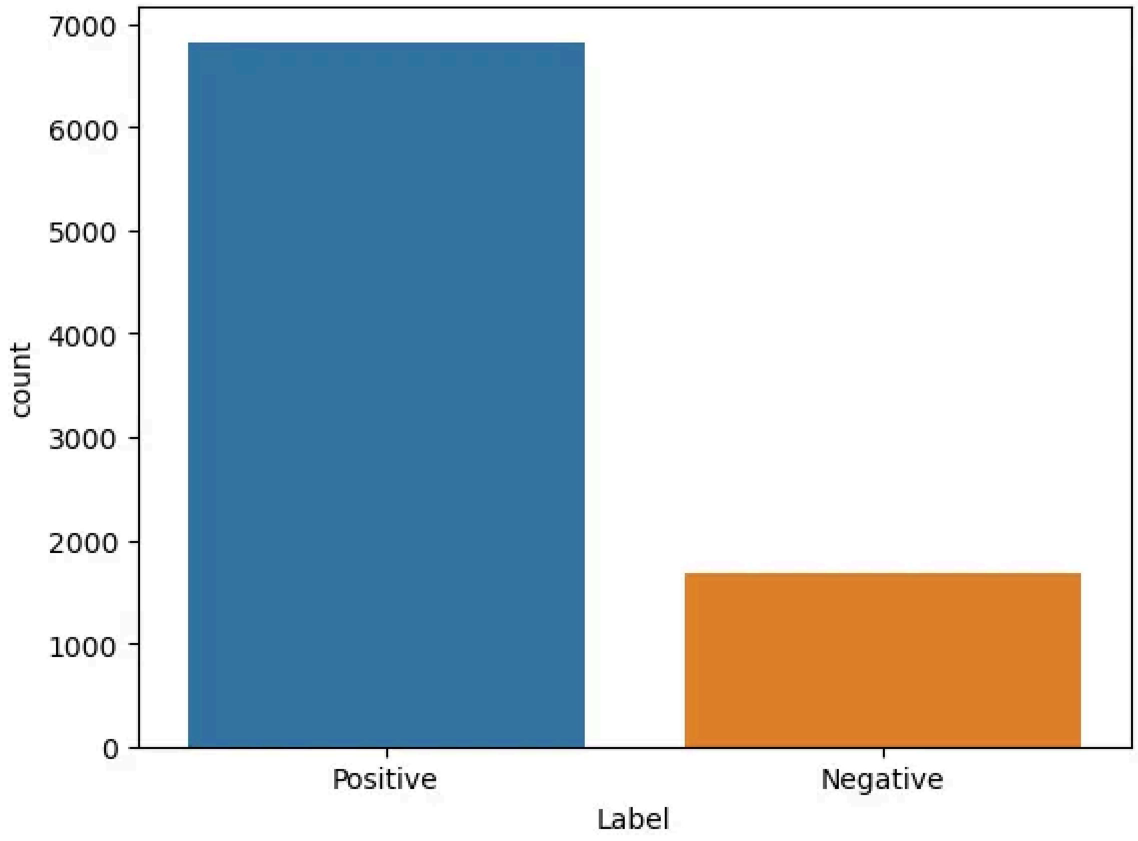
- More stable and less computationally expansive
- Accurate on estimating the test error rate

Cons:

- May overestimate the test error in higher flexibility
- Struggles with data imbalance in the classification problems which can result in a **biased estimate and overfitting** in the favor of the dominating class

Stratified K-Fold Cross-Validation

an Important approach when dealing with classification problems and there is imbalance in the data-set

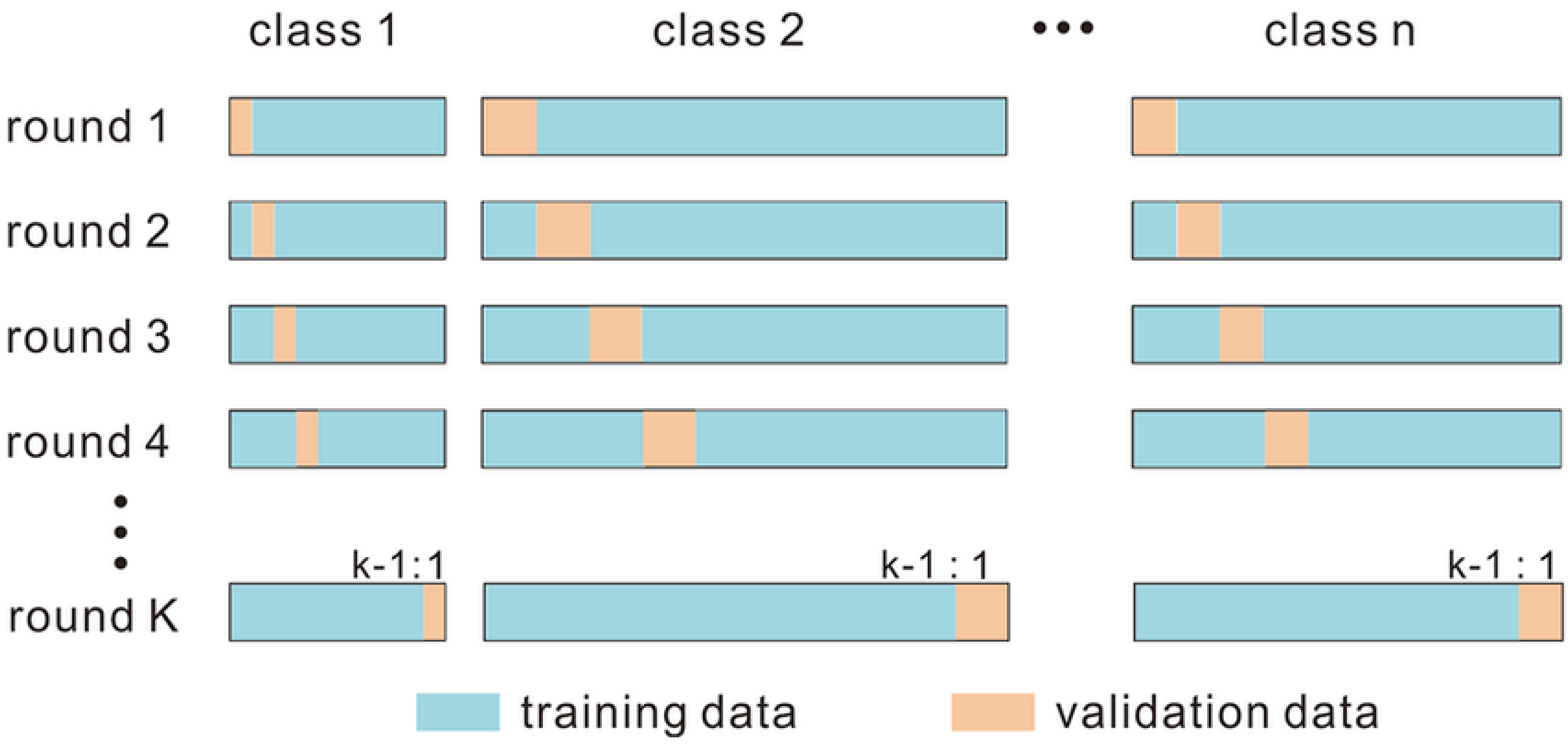


- Here clearly there is a dominance in the **Positive** class , which can results in overfitting in the favor of the majority class

Stratified K-Fold Cross-Validation solves this by ensuring each class gets a fair representation in the training and the validation process which results in **unbiased** estimates.

Process step by step :

1. **Order Samples per Class** : Ordering the samples by class, grouping all the samples from the same class together
2. **Creating Strata** : for each class k the samples are divided into k **non-overlapping** sets or strata
3. **Combining Strata into Folds** : Remember that Strata and the samples from a single class divided into validation and training sets , we create folds from the strata of each class creating folds that represent the true **class distribution** in the data set
4. **Perform K-Fold CV** : Now the Folds are created we can perform the same process we do in the traditional **K-Fold Cross-Validation** with $K-1$ Training and the remaining fold which is the combination of all the validation folds in the Divided Strata


class 1
class 2
•••
class n
round 1
round 2
round 3
round 4
round K
k-1:1
k-1 : 1
k-1 : 1
training data
validation data