Crossvalidation

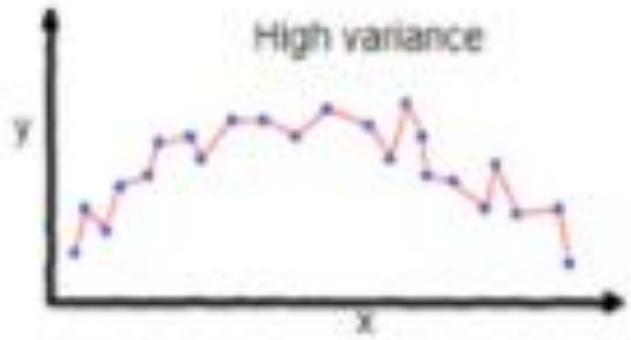
Introduction

- Whenever we build any machine learning model, we feed it with initial data to train the model. And then we feed some unknown data (test data) to understand how well the model performs and generalized over unseen data. If the model performs well on the unseen data, it’s consistent and is able to predict with good accuracy on a wide range of input data; then this model is stable.
- But this is not the case always! Machine learning models are not always stable and we have to evaluate the stability of the machine learning model. That is where Cross Validation comes into the picture.

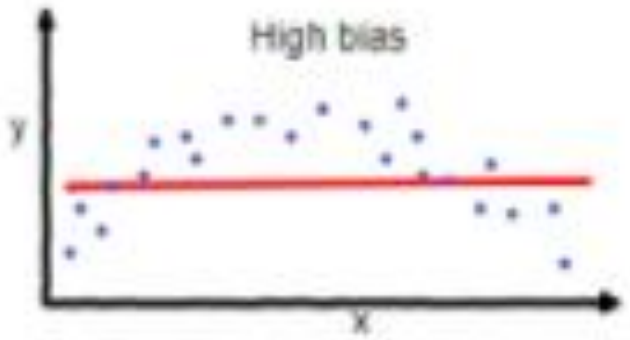
Why do we need Cross-Validation?

- Suppose you build a machine learning model to solve a problem, and you have trained the model on a given dataset. When you check the accuracy of the model on the training data, it is close to 95%. Does this mean that your model has trained very well, and it is the best model because of the high accuracy?
- No, it’s not! Because your model is trained on the given data, it knows the data well, captured even the minute variations(noise), and has generalized very well over the given data. If you expose the model to completely new, unseen data, it might not predict with the same accuracy and it might fail to generalize over the new data. This problem is called over-fitting

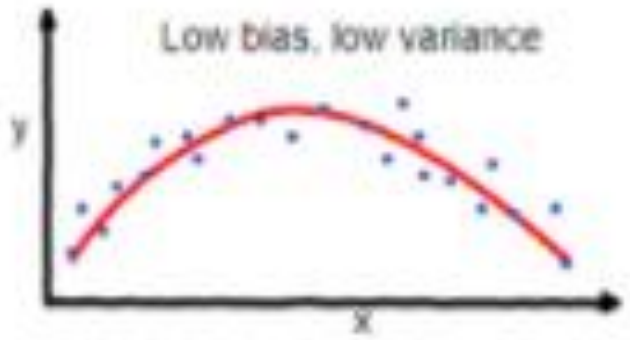
- Sometimes the model doesn’t train well on the training set as it’s not able to find patterns. In this case, it wouldn’t perform well on the test set as well. This problem is called Under-fitting
- To overcome over-fitting problems, we use a technique called Cross-Validation



overfitting



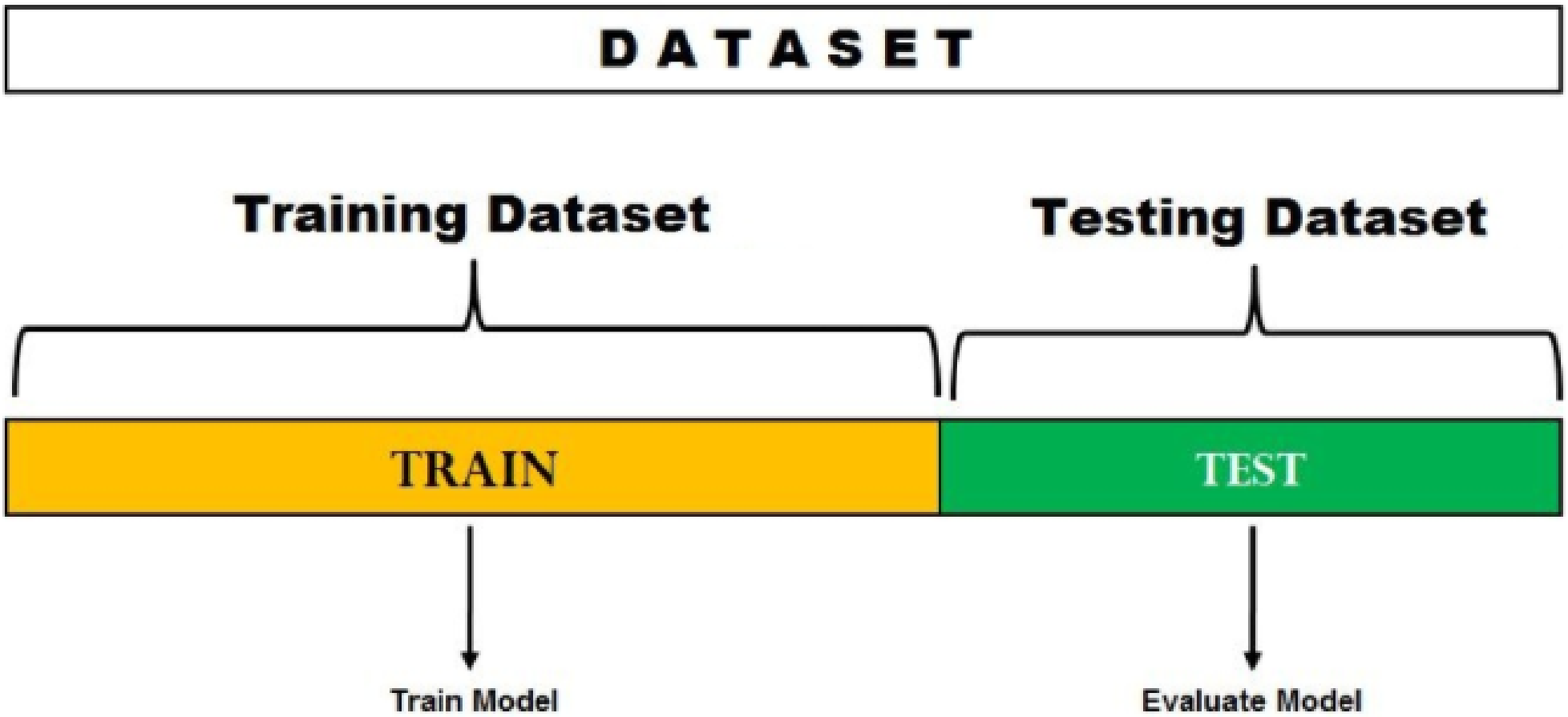
underfitting



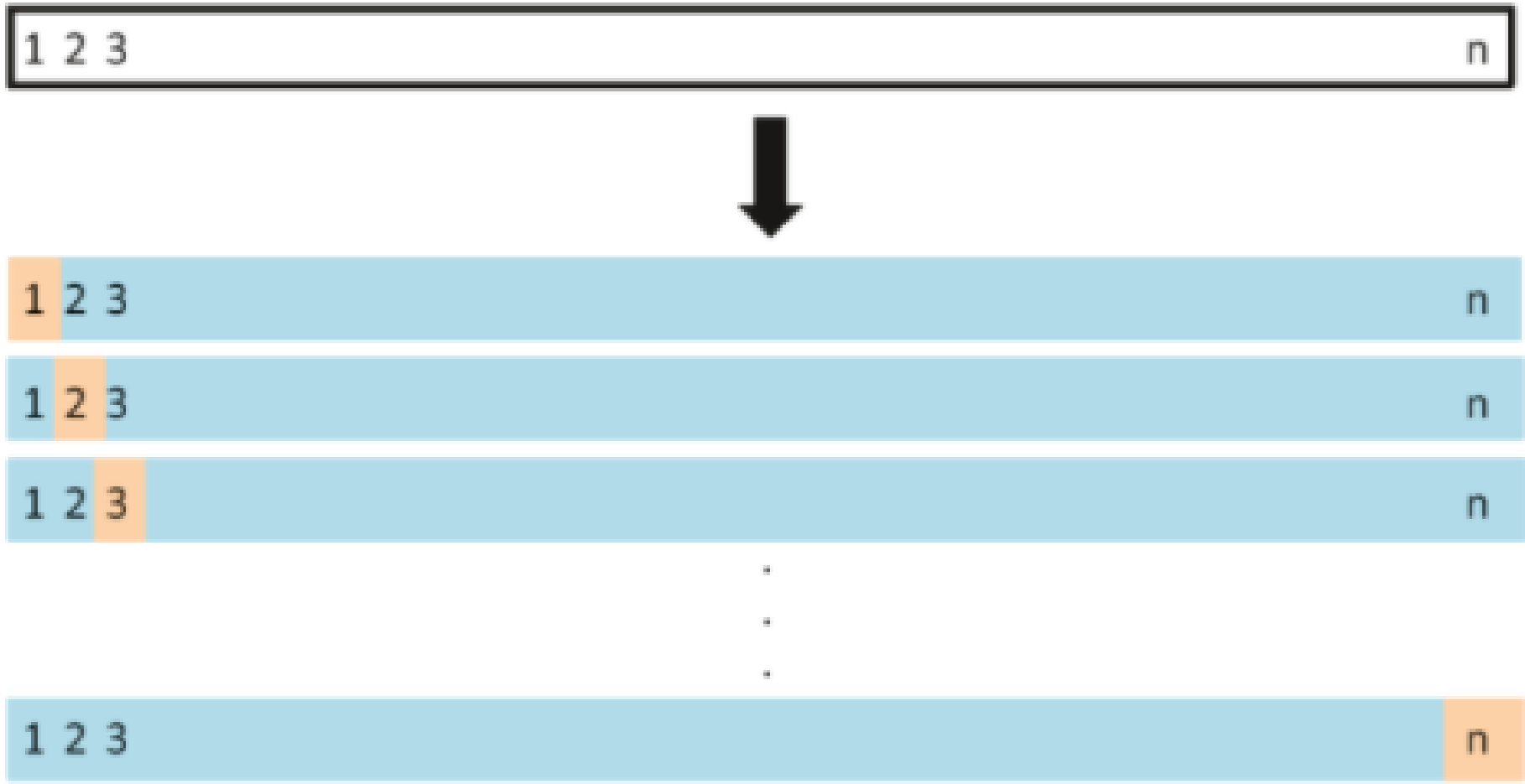
Good balance

- **Cross-Validation** is a resampling technique with the fundamental idea of splitting the dataset into 2 parts- training data and test data. Train data is used to train the model and the unseen test data is used for prediction. If the model performs well over the test data and gives good accuracy, it means the model hasn’t overfitted the training data and can be used for prediction.

1. Hold Out method



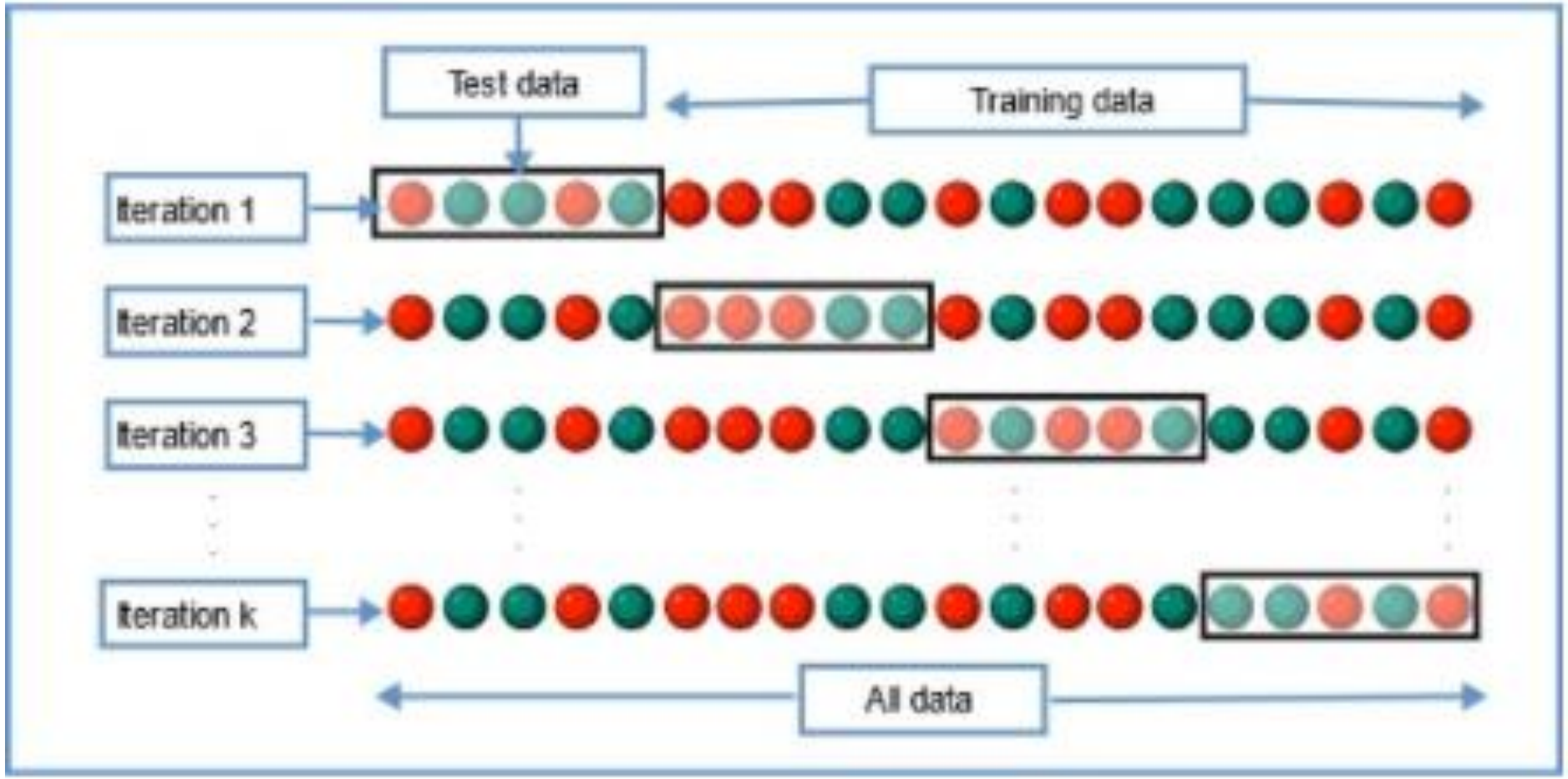
2. Leave One Out Cross-Validation



- In this method, we divide the data into train and test sets – but with a twist. Instead of dividing the data into 2 subsets, we select a single observation as test data, and everything else is labeled as training data and the model is trained. Now the 2nd observation is selected as test data
- This process continues ‘n’ times and the average of all these iterations is calculated and estimated as the test set error and the model is trained
- on the remaining data.

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

3. K-Fold Cross-Validation



- In this resampling technique, the whole data is divided into k sets of almost equal sizes. The first set is selected as the test set and the model is trained on the remaining k-1 sets. The test error rate is then calculated after fitting the model to the test data.
- In the second iteration, the 2nd set is selected as a test set and the remaining k-1 sets are used to train the data and the error is calculated. This process continues for all the k sets.
- The mean of errors from all the iterations is calculated as the CV test error estimate

$$\mathrm{CV}_{(k)} = \frac{1}{k} \sum_{i=1}^{k} \mathrm{MSE}_i .$$

4. Stratified K-Fold Cross-Validation