Sanjivani Rural Education Society's

Sanjivani College of Engineering, Kopargaon-423603

(An Autonomous Institute Affiliated to Savitribai Phule Pune University, Pune)

NAAC 'A' Grade Accredited

Department of Information Technology

NBA Accredited-UG Programme

Machine Learning

Ms. K. D. Patil

Assistant Professor

Contents

- Introduction: Definition, Real life applications, Introduction to Data in Machine Learning Types of Learning: Supervised Learning Unsupervised Learning, Semi-Supervised Learning, Reinforcement Learning, Concept of Feature, Feature Construction, Feature Selection and Transformation, Curse of Dimensionality. Dataset Preparation: Training Vs. Testing Dataset, Dataset Validation Techniques – Hold-out, kfold Cross validation, Leave-One-Out Cross- Validation (LOOCV)

Course Outcome

- **CO1:** To recognize the characteristics of machine learning that makes it useful to real-world problems.

Curse of Dimensionality

- Curse of Dimensionality arises when working with high-dimensional data, leading to increased computational complexity, over fitting, and spurious correlations.
- Techniques like dimensionality reduction, feature selection, and careful model design are essential for mitigating its effects and improving algorithm performance.
- Navigating this challenge is crucial for unlocking the potential of high-dimensional datasets and ensuring robust machine-learning solution.

Curse of Dimensionality

- Consider a machine learning model trying to predict whether a customer will click on an online advertisement.
- In a low-dimensional space, you might have features like age, location, and browsing history.
- As you add more features, like the color of the user's socks, the number of hairs on their head, or the exact time they had lunch last Tuesday, the data becomes increasingly sparse.
- This sparsity makes it harder for the model to learn meaningful patterns.
- Furthermore, the distance between users in this high-dimensional space becomes less meaningful, making algorithms that rely on proximity less effective.

Dataset Validation

- **Generalization:**
  - How well a model trained on the training set predicts the right output for new instances is called **Generalization.**
  - Generalization refers to how well the concepts learned by a machine learning model apply to specific examples not seen by the model when it was learning.
  - The goal of a good machine learning model is to generalize well from the training data to any data from the problem domain.
  - This allows us to make predictions in the future on data the model has never seen.

Dataset Validation

- **Bias:**
  - Assumptions made by a model to make a function easier to learn.
  - It is actually the error rate of the **training data**.
  - When the error rate has a high value, we call it **High Bias** and when the error rate has a low value, we call it **Low Bias**.
  - Bias is simply defined as the inability of the model because of that there is some difference or **error occurring between the model's predicted value and the actual value**.
  - These differences between actual or expected values and the predicted values are known as **error** or **bias error** or **error due to bias**.
  - Bias is a systematic error that occurs due to wrong assumptions in the ML process.

Dataset Validation

- **Bias:**
  - **Low Bias:** Low bias value means fewer assumptions are taken to build the target function. In this case, **the model will closely match the training dataset.**
  - **High Bias:** High bias value means more assumptions are taken to build the target function. In this case, **the model will not match the training dataset closely.**
  - The high-bias model will not be able to capture the dataset trend. It is considered as the **under-fitting model which has a high error rate**. It is due to a very simplified algorithm.
  - For example, a linear regression model may have a high bias if the data has a non-linear relationship.

Dataset Validation

- **Variance:**
  - The difference between the error rate of training data and testing data is called **variance**.
  - Gives proper results on training data but not on testing data.
  - If the difference is high then it's called **high variance** and when the difference of errors is low then it's called **low variance**.
  - Usually, we want to make a low variance for generalized our model.
  - **Variance is the measure of spread in data from its mean position**.
  - In machine learning variance is the amount by which the performance of a predictive model changes when it is trained on different subsets of the training data.
  - It is the variability of the model that how much it is sensitive to another subset of the training dataset. i.e. how much it can adjust on the new subset of the training dataset.

Dataset Validation

- **Variance:**
  - **Low variance:** Low variance means that the model is less sensitive to changes in the training data and can produce consistent estimates of the target function with different subsets of data from the same distribution. However, low variance can also indicate **under-fitting** if the model is too simple and fails to capture the underlying patterns in the data. This is when the model **performs poorly on both the training data and testing data.**
  - **High variance**: High variance means that the model is very sensitive to changes in the training data and can result in significant changes in the estimate of the target function when trained on different subsets of data from the same distribution. This is the case of **over-fitting** when the model performs well on the training data but poorly on new, unseen test data. **It fits the training data too closely that it fails on the new training dataset.**

Dataset Validation

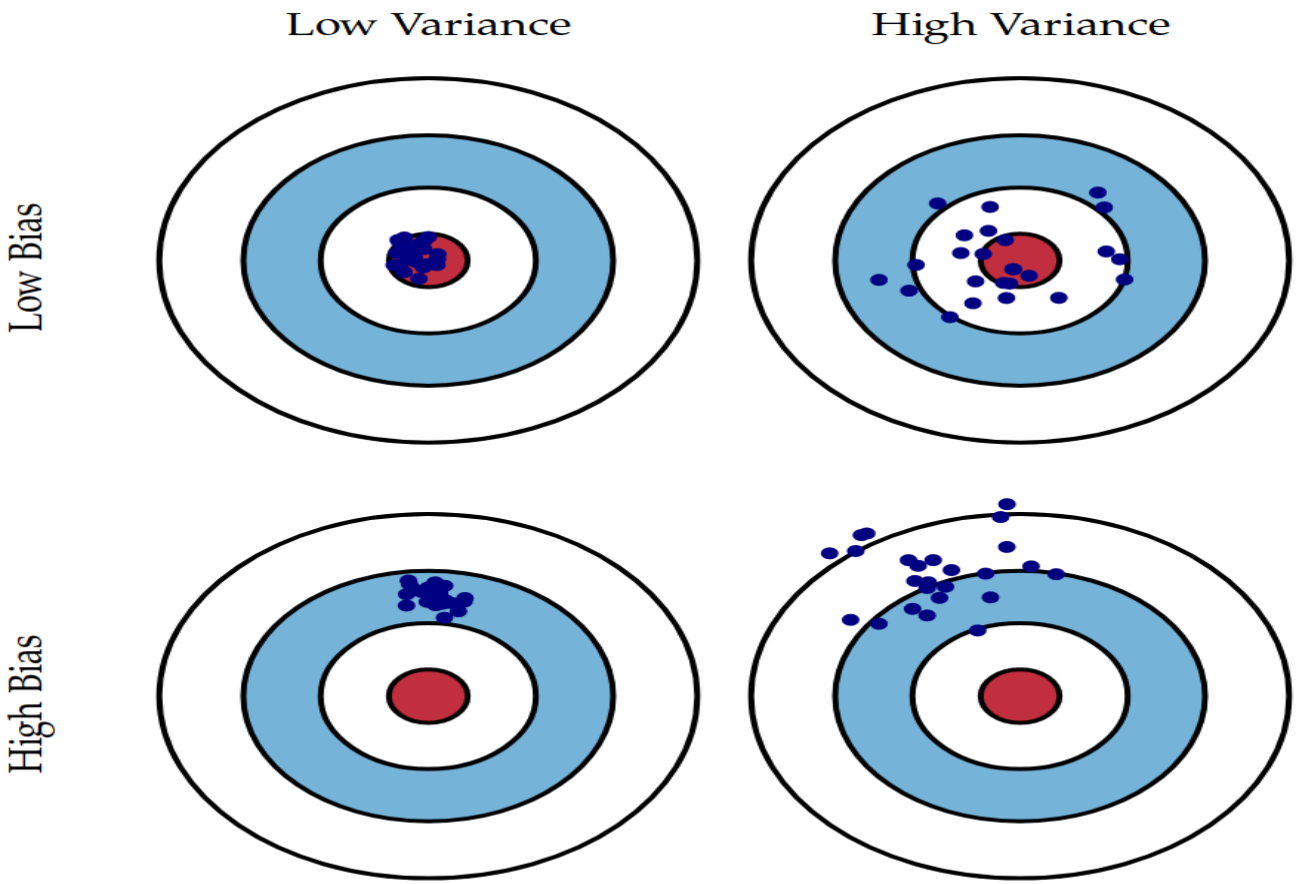


Fig. 1 Graphical illustration of bias and variance.

Low Bias, High Variance
Training Accuracy – 90%
Testing Accuracy – 75%

High Bias, Low Variance
Training Accuracy – 45%
Testing Accuracy – 95%

High Bias, High Variance
Training Accuracy – 60%
Testing Accuracy – 55%

Our Target
Low Bias, Low Variance
Training Accuracy – 90%
Testing Accuracy – 92%

Dataset Validation

- Over fitting and Under fitting are the two biggest causes for poor performance of machine learning algorithms.
- The model should be selected having the best generalization.
- This is said to be the case if these problems are avoided.

- **Under-fitting:**
  - A statistical model or a machine learning algorithm is said to have under-fitting when it cannot capture the underlying trend of the data.
  - It is not complex enough to accurately capture relationships between a dataset features and a target variable.

Dataset Validation

- **Reasons for Under-fitting:**
  - High bias and low variance
  - The size of the training dataset used is not enough.
  - The model is too simple.
  - Training data is not cleaned and also contains noise in it.

- **Techniques to reduce Under-fitting:**
  - Increase model complexity.
  - Increase the number of features, performing feature engineering.
  - Remove noise from the data.
  - Increase the number of epochs or increase the duration of training to get better results.

Dataset Validation

- **Over-fitting:**
  - Over-fitting is the production of an analysis which corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably.
  - A statistical model is said to be over-fitted when the model does not make accurate predictions on testing data.
  - When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set and when testing with test data results in High variance.
  - Then the model does not categorize the data correctly, because of too many details and noise.

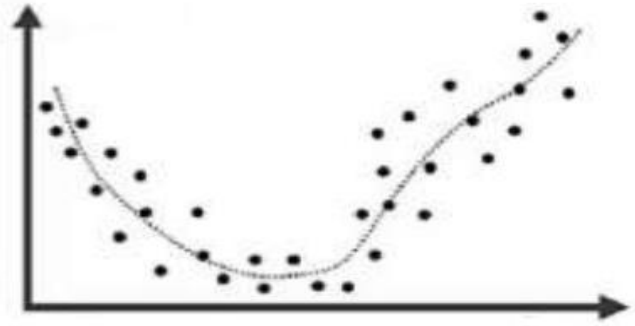
Dataset Validation

- **Over-fitting:**
  - The causes of over-fitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models.
  - Over-fitting is a problem where the evaluation of machine learning algorithms on training data is different from unseen data.

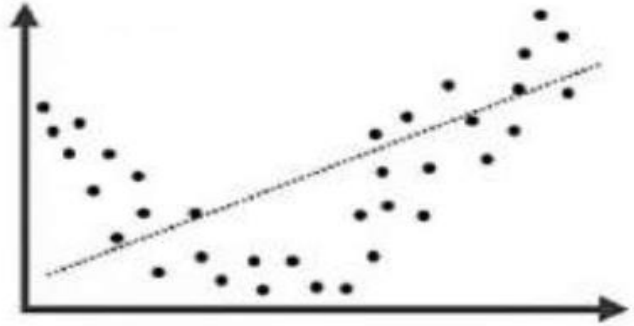
  - **Reasons for Over-fitting are as follows:**
    - High variance and low bias
    - The model is too complex
    - The size of the training data

Dataset Validation

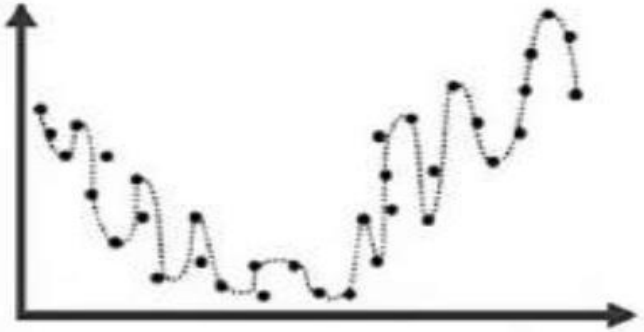
(a) Given dataset



(b) "Just right" model



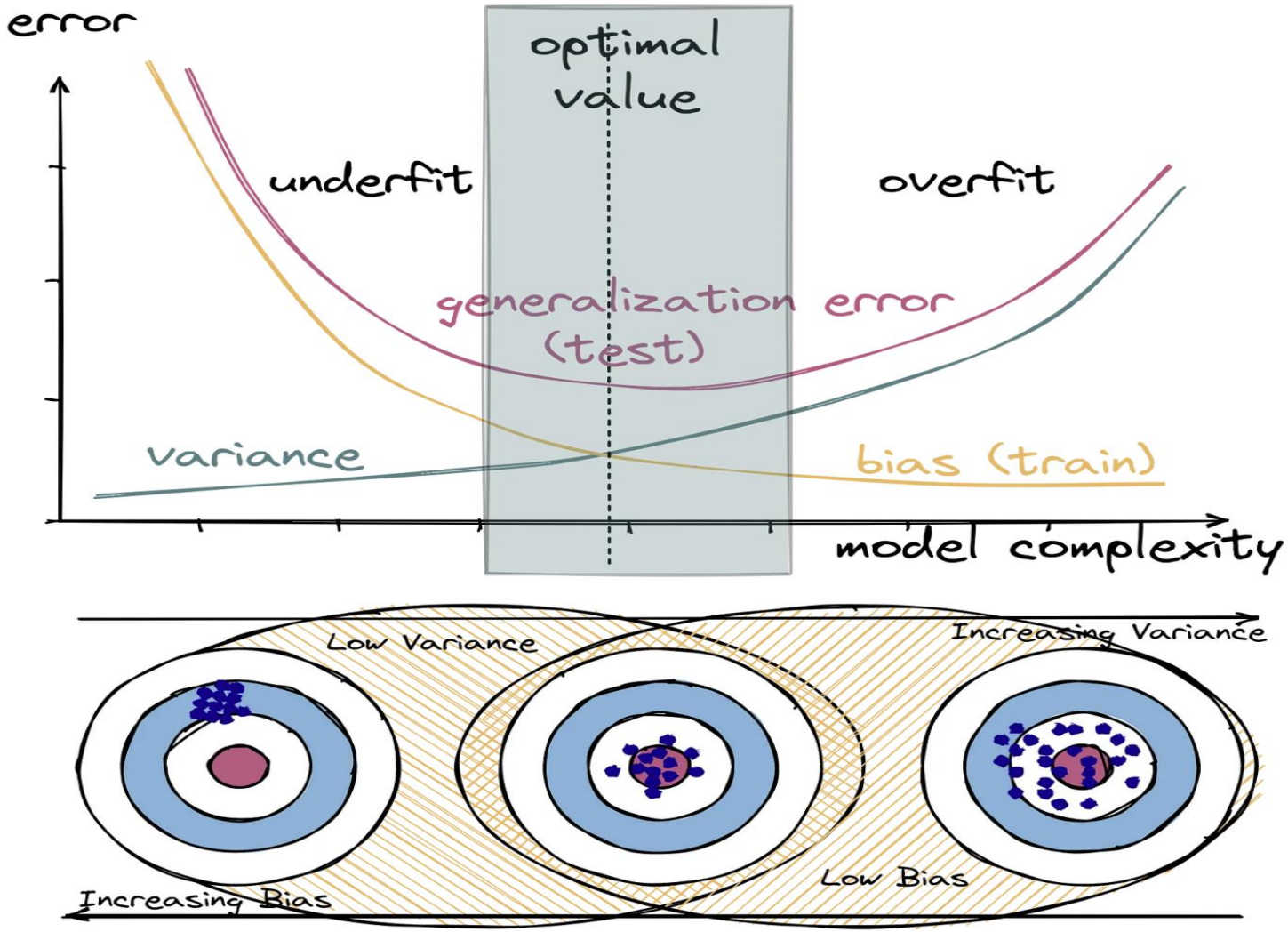
(c) Underfitting model



(d) Overfitting model

Example: https://www.analyticsvidhya.com/blog/2020/02/underfitting-overfitting-best-fitting-machine-learning/

Dataset Validation



Cross Validation

- Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set.

- **The three steps involved in cross-validation are as follows:**
  - Reserve some portion of sample data-set.
  - Using the rest data-set train the model.
  - Test the model using the reserve portion of the data-set.

Validation Techniques

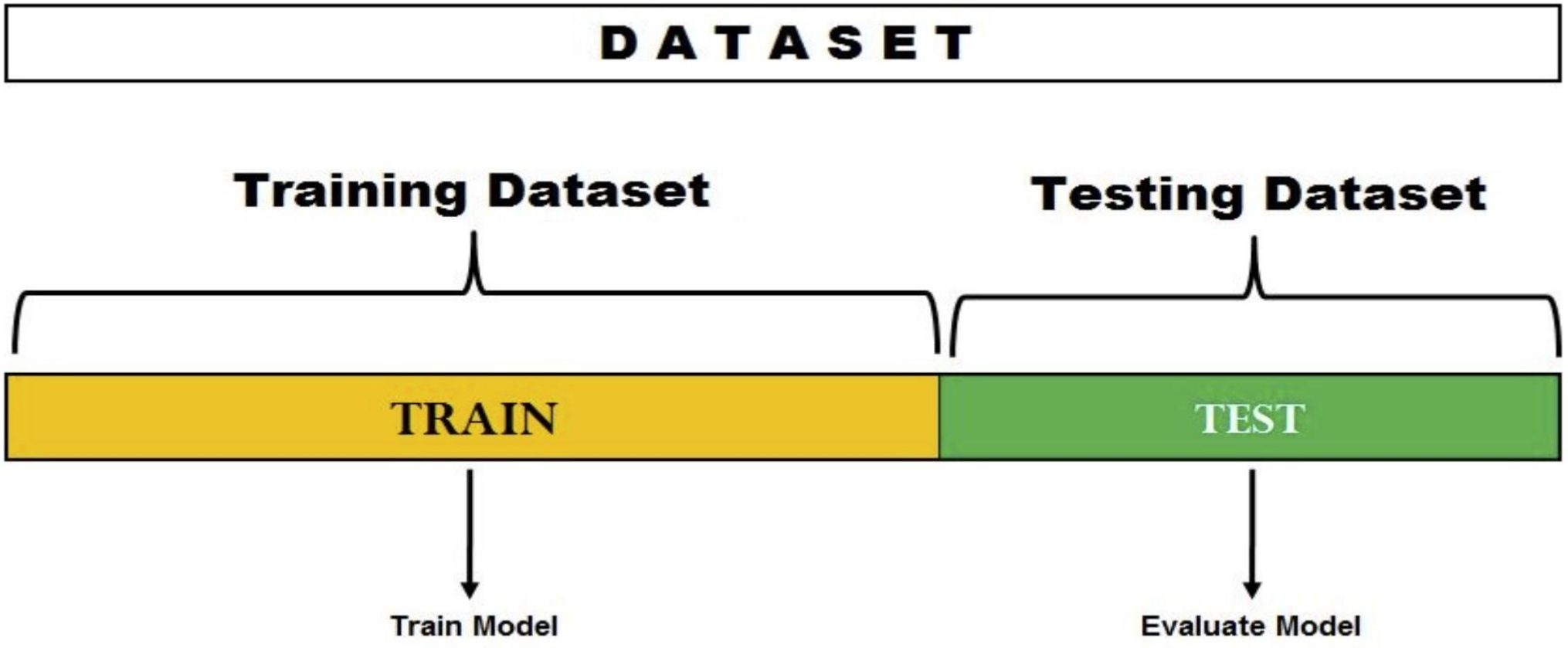
- **Validation:**
  - In this method, we perform training on the 50% of the given data-set and rest 50% is used for the testing purpose.
  - The major drawback of this method is that we perform training on the 50% of the dataset, it may possible that the remaining 50% of the data contains some important information which we are leaving while training our model i.e higher bias.

Validation Techniques

- **Hold-Out method:**
  - This is the simplest evaluation method and is widely used in Machine Learning projects.
  - Here the entire dataset (population) is divided into 2 sets – train set and test set.
  - The data can be divided into 70-30 or 60-40, 75-25 or 80-20, or even 50-50 depending on the use case.
  - As a rule, the proportion of training data has to be larger than the test data.

Validation Techniques

- **Hold-Out method:**



Validation Techniques

- **Hold-Out method:**
  - The data split happens randomly, and we can't be sure which data ends up in the train and test bucket during the split.
  - This can lead to extremely high variance and every time, the split changes, the accuracy will also change.

- **There are some drawbacks to this method:**
  - The test error rates are highly variable (high variance) and it totally depends on which observations end up in the training set and test set.
  - Only a part of the data is used to train the model (high bias) which is not a very good idea when data is not huge and this will lead to overestimation of test error.

Validation Techniques

- **Hold-Out method:**

```
1  from sklearn.model_selection import train_test_split
2  X = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
3  X_train, X_test= train_test_split(X, test_size=0.3, random_state=1)
4  print('Train:', X_train, 'Test:', X_test)
5
```

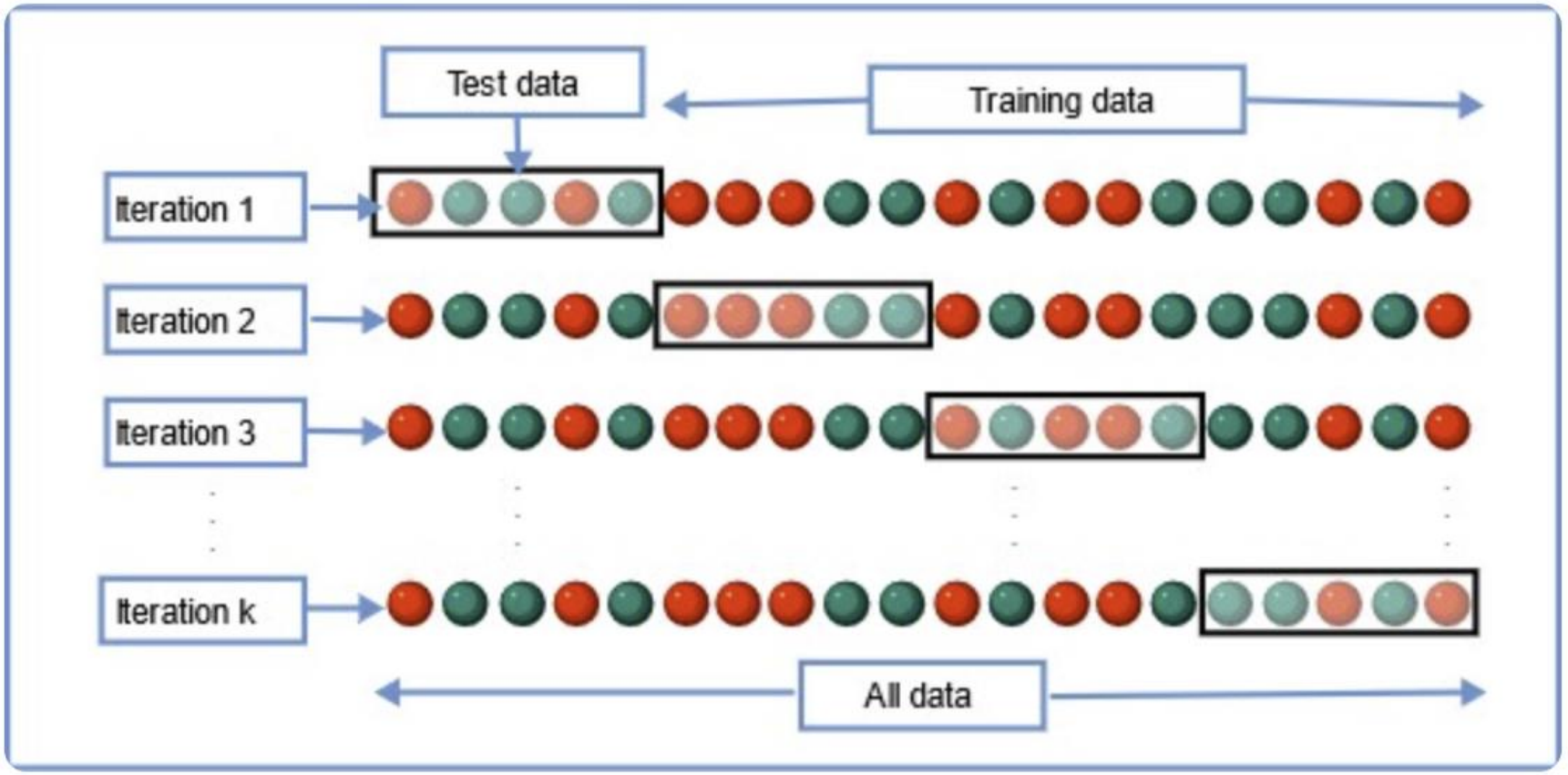
```
Output

Train: [50, 10, 40, 20, 80, 90, 60] Test: [30, 100, 70]
```

Validation Techniques

- **K-Fold Cross-Validation:**
  - In this 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.

Validation Techniques

- **K-Fold Cross-Validation:**



Validation Techniques

- **K-Fold Cross-Validation:**
  - The mean of errors from all the iterations is calculated as the CV test error estimate.

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

  - In K-Fold CV, the no of folds k is less than the number of observations in the data (k<n) and we are averaging the outputs of k fitted models that are somewhat less correlated with each other since the overlap between the training sets in each model is smaller.
  - This leads to low variance.

Validation Techniques

- **K-Fold Cross-Validation:**
  - The best part about this method is each data point gets to be in the test set exactly once and gets to be part of the training set k-1 times.
  - As the number of folds k increases, the **variance also decreases** (low variance).
  - This method leads to **intermediate bias.**
  - Typically, K-fold Cross Validation is performed using k=5 or k=10 as these values have been empirically shown to yield test error estimates that neither have high bias nor high variance.
  - Note: It is always suggested that the value of k should be 10 as the lower value of k is takes towards validation and higher value of k leads to LOOCV method.

Validation Techniques

- **K-Fold Cross-Validation:**
  - The major disadvantage of this method is that the model **has to be run from scratch k-times** and is **computationally expensive than the Hold Out method.**

Simple implementation of K-Fold Cross-Validation in Python

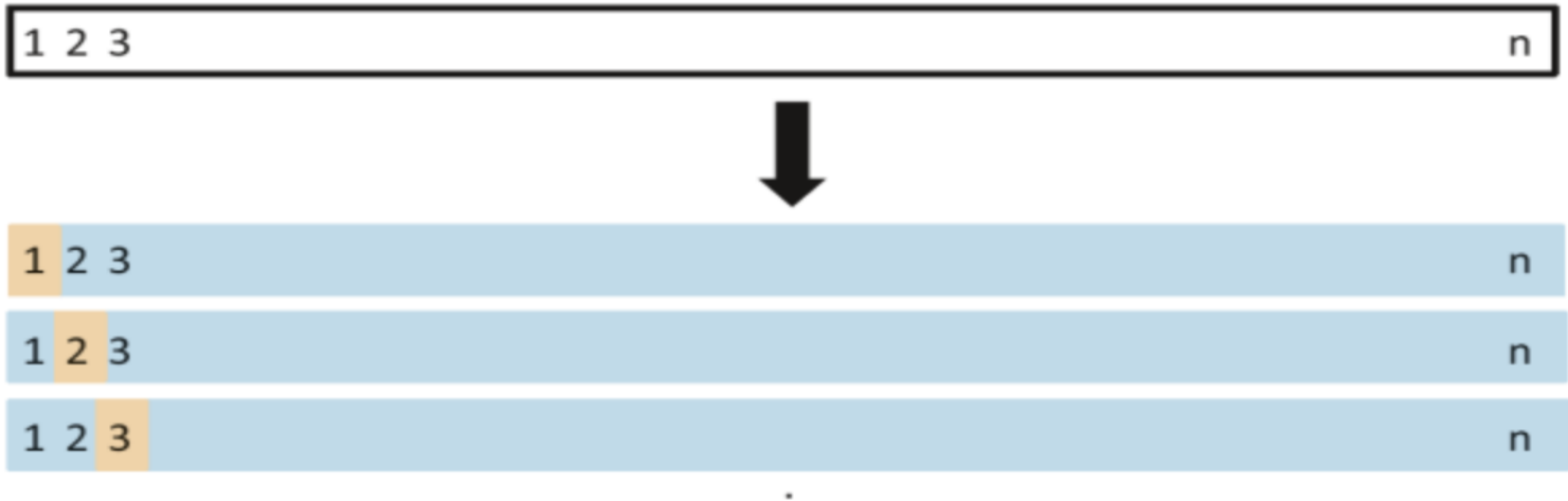
```
from sklearn.model_selection import KFold

X = ["a",'b','c','d','e','f']

kf = KFold(n_splits=3, shuffle=False, random_state=None)

for train, test in kf.split(X):

    print("Train data",train,"Test data",test)
```

Output

```
Train: [2 3 4 5] Test: [0 1]
Train: [0 1 4 5] Test: [2 3]
Train: [0 1 2 3] Test: [4 5]
```

Validation Techniques

- **Leave-one-Out Cross-Validation (LOOCV):**
  - 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 and the model is trained on the remaining data.



Validation Techniques

- **Leave-one-Out Cross-Validation (LOOCV):**
  - This process continues ‘n’ times and the average of all these iterations is calculated and estimated as the test set error.

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

  - When it comes to test-error estimates, LOOCV gives unbiased estimates (**low bias**). But bias is not the only matter of concern in estimation problems. We should also consider variance.

Validation Techniques

- **Leave-one-Out Cross-Validation (LOOCV):**
  - LOOCV has an extremely **high variance** because we are averaging the output of n-models which are fitted on an almost identical set of observations, and their outputs are highly positively correlated with each other.
  - This is **computationally expensive** as the model is run ‘n’ times to test every observation in the data.

Validation Techniques

- **Leave-one-Out Cross-Validation (LOOCV):.**

Quick implementation of Leave One Out Cross-Validation in Python

```
from sklearn.model_selection import LeaveOneOut

X = [10,20,30,40,50,60,70,80,90,100]

l = LeaveOneOut()

for train, test in l.split(X):

    print("%s %s"% (train,test))
```

Output

```
[1 2 3 4 5 6 7 8 9] [0]
[0 2 3 4 5 6 7 8 9] [1]
[0 1 3 4 5 6 7 8 9] [2]
[0 1 2 4 5 6 7 8 9] [3]
[0 1 2 3 5 6 7 8 9] [4]
[0 1 2 3 4 6 7 8 9] [5]
[0 1 2 3 4 5 7 8 9] [6]
[0 1 2 3 4 5 6 8 9] [7]
[0 1 2 3 4 5 6 7 9] [8]
[0 1 2 3 4 5 6 7 8] [9]
```

This output clearly shows how LOOCV keeps one observation aside as test data and all the other observations go to train data.

Reference

- Ethem Alpaydin, Introduction to Machine Learning, PHI 4th Edition-2020, The MIT Press,ISBN:9780262043793.
- Peter Flach, “Machine Learning The Art and Science of Algorithms that Make Sense of Data”, Cambridge University Press India. ISBN 13: 9781107422223

Thank You!!!

Happy Learning!!!