Principal Component Analysis

Eigenvalues and Eigenvectors

- Find the eigenvalues and eigenvectors of the matrix (real symmetric matrix, typical of covariance matrix) $\begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix}$
- $Ax = \lambda x$, x is an eigenvector of A and λ its corresponding eigenvalue. We can re-write this as $Ax = \lambda Ix$, where I is the identity matrix, or as $(\lambda I - A)x = 0$
- For λ to be an eigenvalue, there must be a $\neq 0$ solution of this, which occurs when $\det(\lambda I - A) = 0$

$$\det(\lambda I - A) = \begin{bmatrix} \lambda - 5 & -3 \\ -3 & \lambda - 5 \end{bmatrix} = 0$$

$\lambda^2 - 10\lambda + 16 = 0$; $(\lambda - 2)(\lambda - 8) = 0$; the eigenvalues of A are $\lambda_1 = 2$ and $\lambda_2 = 8$

Principal Component Analysis

Eigenvectors

- Substituting $\lambda_1 = 2$ into $Ax = \lambda x$ we get

$$\begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

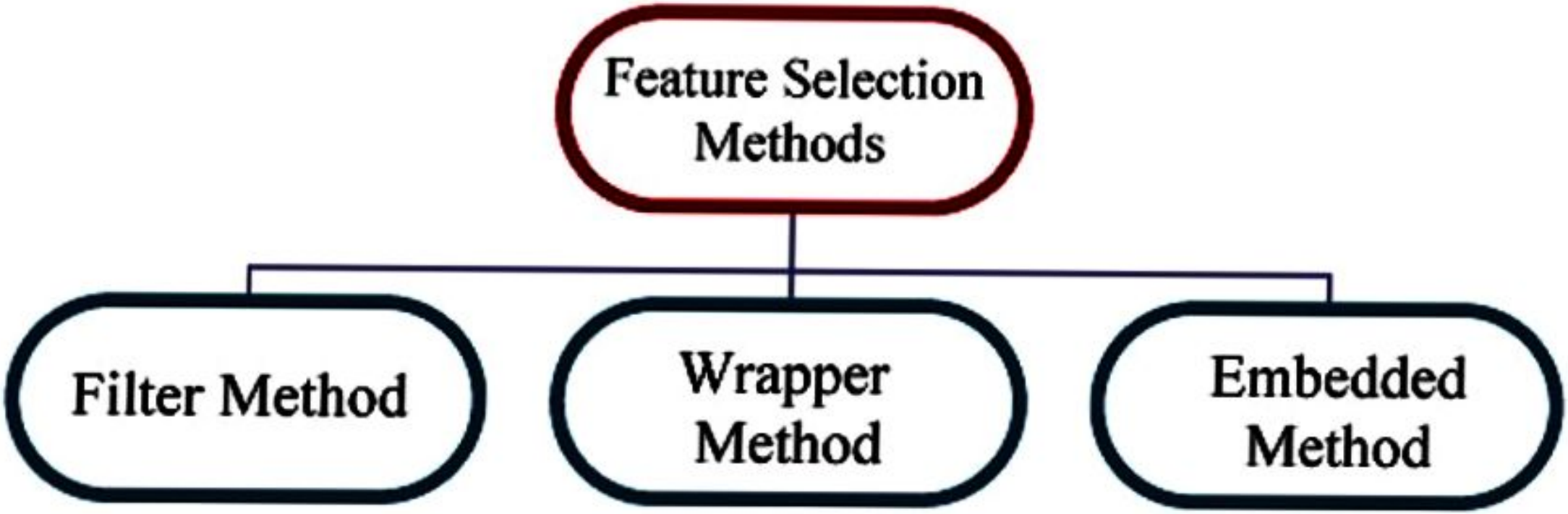
  which gives $3x_1 + 3x_2 = 0$ from which we deduce, $e_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$. In a similar manner, for eigenvalue $\lambda_2 = 8$, we get $-3x_1 + 3x_2 = 0$ from which, $e_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

- The eigenvectors are orthogonal (which only happens for a real symmetric matrix) and in this case are rotated $\pi/4$ rad from the original axes.
- In the coordinate system of these new principal axes the isocontours will be ellipses and the ellipse corresponding to one standard deviation will be $\frac{u^2}{8^2} + \frac{v^2}{2^2} = 1$

Principal Component Analysis

Example

- Find the eigenvalues and eigenvectors of the matrix
$$\begin{bmatrix} 0.8 & 0.3 \\ 0.2 & 0.7 \end{bmatrix}$$
- $$\det(A - \lambda I) = \begin{bmatrix} 0.8 - \lambda & .3 \\ .2 & .7 - \lambda \end{bmatrix} = 0$$
- $\lambda^2 - \frac{3}{2}\lambda + \frac{1}{2} = 0$; $(\lambda - 1)(\lambda - \frac{1}{2}) = 0$; the eigenvalues of A are $\lambda_1 = 1$ and $\lambda_2 = \frac{1}{2}$
- $(A - I)x_1 = 0 \Longrightarrow Ax_1 = x_1$; the first eigenvector $= (0.3, 0.2)$
  $(A - \frac{1}{2}I)x_2 = 0 \Longrightarrow Ax_2 = \frac{1}{2}x_2$; the second eigenvector $=$ $(1, -1)$


Feature Selection Methods
Filter Method
Wrapper Method
Embedded Method

So, we can define feature Selection as, "***It is a process of automatically or manually selecting the subset of most appropriate and relevant features to be used in model building***." Feature selection is performed by either including the important features or excluding the irrelevant features in the dataset without changing them.

So, we can define feature Selection as, "***It is a process of automatically or manually selecting the subset of most appropriate and relevant features to be used in model building***." Feature selection is performed by either including the important features or excluding the irrelevant features in the dataset without changing them.

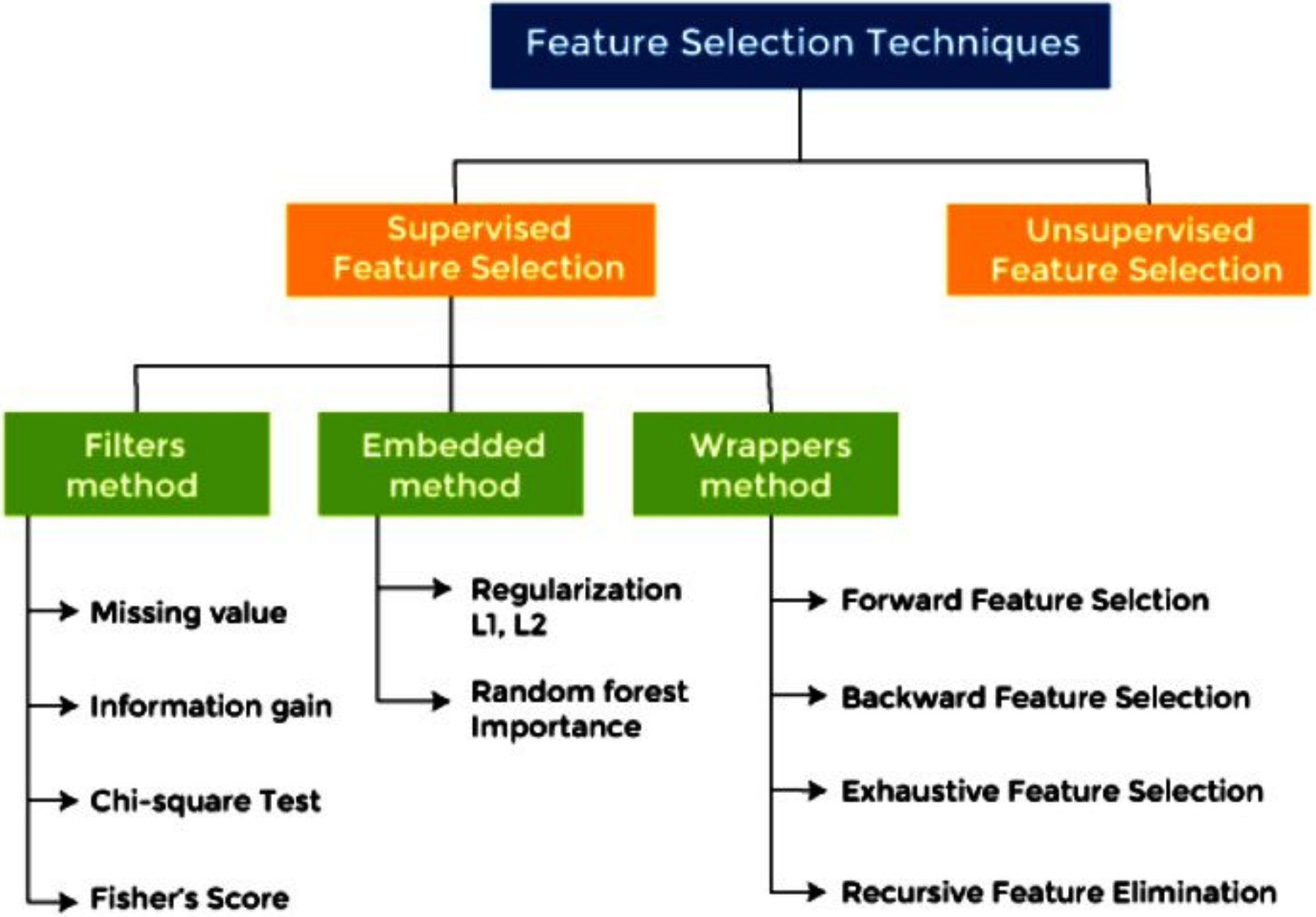
Need for Feature Selection

Before implementing any technique, it is really important to understand, need for the technique and so for the Feature Selection. As we know, in machine learning, it is necessary to provide a pre-processed and good input dataset in order to get better outcomes. We collect a huge amount of data to train our model and help it to learn better. Generally, the dataset consists of noisy data, irrelevant data, and some part of

Feature Selection Techniques

There are mainly two types of Feature Selection techniques, which are:

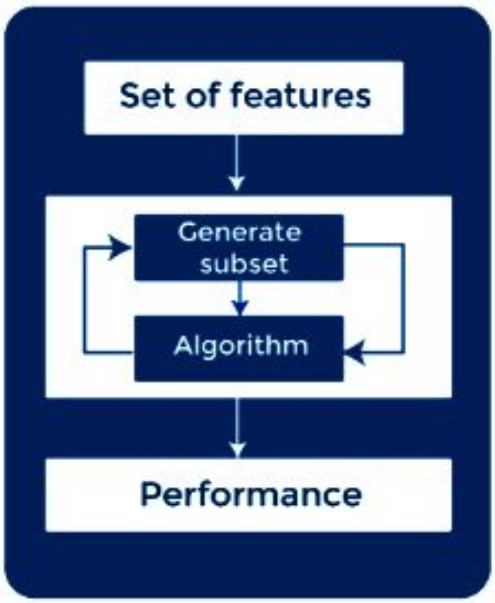
- **Supervised Feature Selection technique**
  Supervised Feature selection techniques consider the target variable and can be used for the labelled dataset.
- **Unsupervised Feature Selection technique**
  Unsupervised Feature selection techniques ignore the target variable and can be used for the unlabelled dataset.


Feature Selection Techniques
Supervised Feature Selection
Unsupervised Feature Selection
Filters method
Embedded method
Wrappers method
Missing value
Information gain
Chi-square Test
Fisher's Score
Regularization L1, L2
Random forest Importance
Forward Feature Selction
Backward Feature Selection
Exhaustive Feature Selection
Recursive Feature Elimination

There are mainly three techniques under supervised feature Selection:

1. Wrapper Methods

In wrapper methodology, selection of features is done by considering it as a search problem, in which different combinations are made, evaluated, and compared with other combinations. It trains the algorithm by using the subset of features iteratively.



Some techniques of wrapper methods are:

- **Forward selection** - Forward selection is an iterative process, which begins with an empty set of features. After each iteration, it keeps adding on a feature and evaluates the performance to check whether it is improving the performance or not. The process continues until the addition of a new variable/feature does not improve the performance of the model.
- **Backward elimination** - Backward elimination is also an iterative approach, but it is the opposite of forward selection. This technique begins the process by considering all the features and removes the least significant feature. This elimination process continues until removing the features does not improve the performance of the model.
- **Exhaustive Feature Selection-** Exhaustive feature selection is one of the best feature selection methods, which evaluates each feature set as brute-force. It means this method tries & make each possible combination of features and return the best performing feature set.
- **Recursive Feature Elimination-**
  Recursive feature elimination is a recursive greedy optimization approach, where features are selected by recursively taking a smaller and smaller subset of features. Now, an estimator is trained with each set of features, and the importance of each feature is determined using *coef_attribute* or through a *feature_importances_attribute.*

2. Filter Methods

In Filter Method, features are selected on the basis of statistics measures. This method does not depend on the learning algorithm and chooses the features as a pre-processing step.

The filter method filters out the irrelevant feature and redundant columns from the model by using different metrics through ranking.

The advantage of using filter methods is that it needs low computational time and does not overfit the data.

- Information Gain
- Chi-square Test
- Fisher's Score
- Missing Value Ratio

Information Gain: Information gain determines the reduction in entropy while transforming the dataset. It can be used as a feature selection technique by calculating the information gain of each variable with respect to the target variable.

Chi-square Test: Chi-square test is a technique to determine the relationship between the categorical variables. The chi-square value is calculated between each feature and the target variable, and the desired number of features with the best chi-square value is selected.

Fisher's Score:

Fisher's score is one of the popular supervised technique of features selection. It returns the rank of the variable on the fisher's criteria in descending order. Then we can select the variables with a large fisher's score.

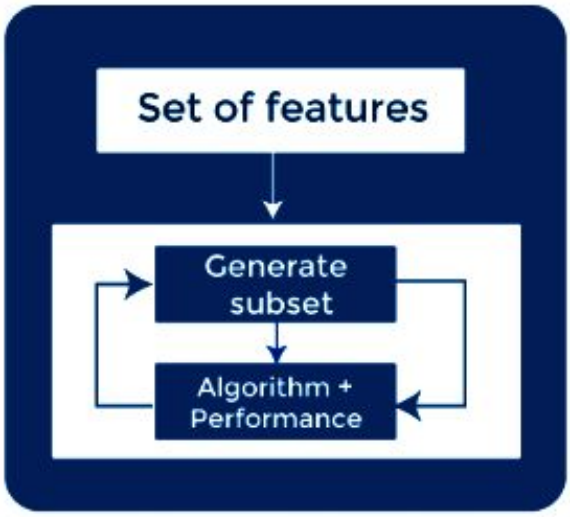
Missing Value Ratio:

The value of the missing value ratio can be used for evaluating the feature set against the threshold value. The formula for obtaining the missing value ratio is the number of missing values in each column divided by the total number of observations. The variable is having more than the threshold value can be dropped.

$$\text{Missing Value Ratio} = \frac{\text{Number of Missing values} * 100}{\text{Total number of observations}}$$

3. Embedded Methods

Embedded methods combined the advantages of both filter and wrapper methods by considering the interaction of features along with low computational cost. These are fast processing methods similar to the filter method but more accurate than the filter method.



These methods are also iterative, which evaluates each iteration, and optimally finds the most important features that contribute the most to training in a particular iteration. Some techniques of embedded methods are:

most to training in a particular iteration. Some techniques of embedded methods are:

- **Regularization**- Regularization adds a penalty term to different parameters of the machine learning model for avoiding overfitting in the model. This penalty term is added to the coefficients; hence it shrinks some coefficients to zero. Those features with zero coefficients can be removed from the dataset. The types of regularization techniques are L1 Regularization (Lasso Regularization) or Elastic Nets (L1 and L2 regularization).
- **Random Forest Importance** - Different tree-based methods of feature selection help us with feature importance to provide a way of selecting features. Here, feature importance specifies which feature has more importance in model building or has a great impact on the target variable. Random Forest is such a tree-based method, which is a type of bagging algorithm that aggregates a different number of decision trees. It automatically ranks the nodes by their performance or decrease in the impurity (Gini impurity) over all the trees. Nodes are arranged as per the impurity values, and thus it allows to pruning of trees below a specific node. The remaining nodes create a subset of the most important features.

Evaluating Machine Learning algorithms and Model Selection.

Overfitting: Good performance on the training data, poor generliazation to other data.

Underfitting: Poor performance on the training data and poor generalization to other data.

Bias & Variance:

What is bias?

Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.

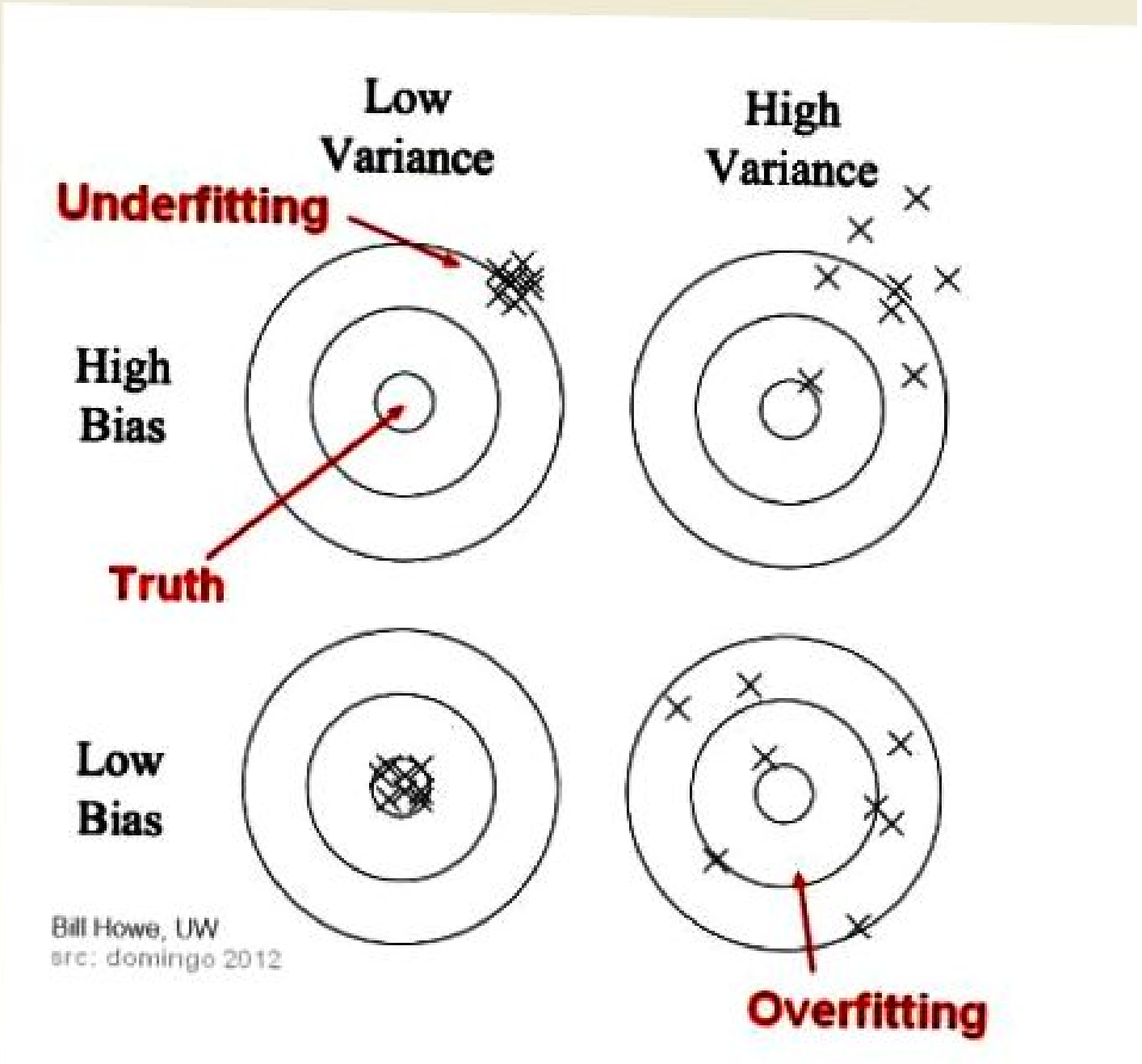
What is variance?

Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models perform very well on training data but has high error rates on test data.

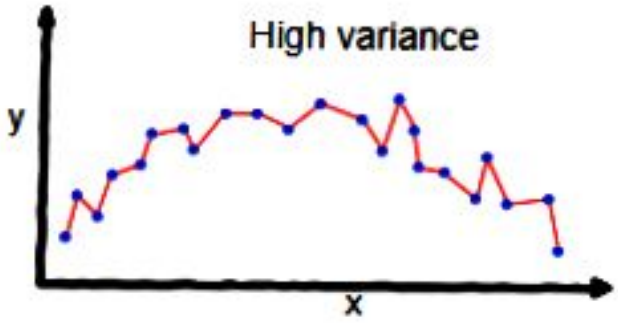
High bias implies our estimate based on the observed data is not close to the true parameter. (aka underfitting).

High variance implies our estimates are sensitive to sampling. They'll vary a lot if we compute them with a different sample of data (aka overfitting).

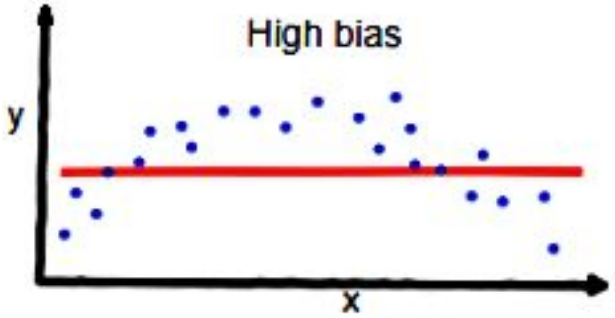
Bias and variance using bulls-eye diagram



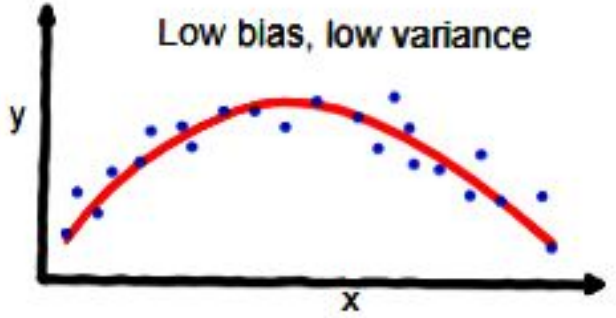
Evaluating Machine Learning algorithms and Model Selection.



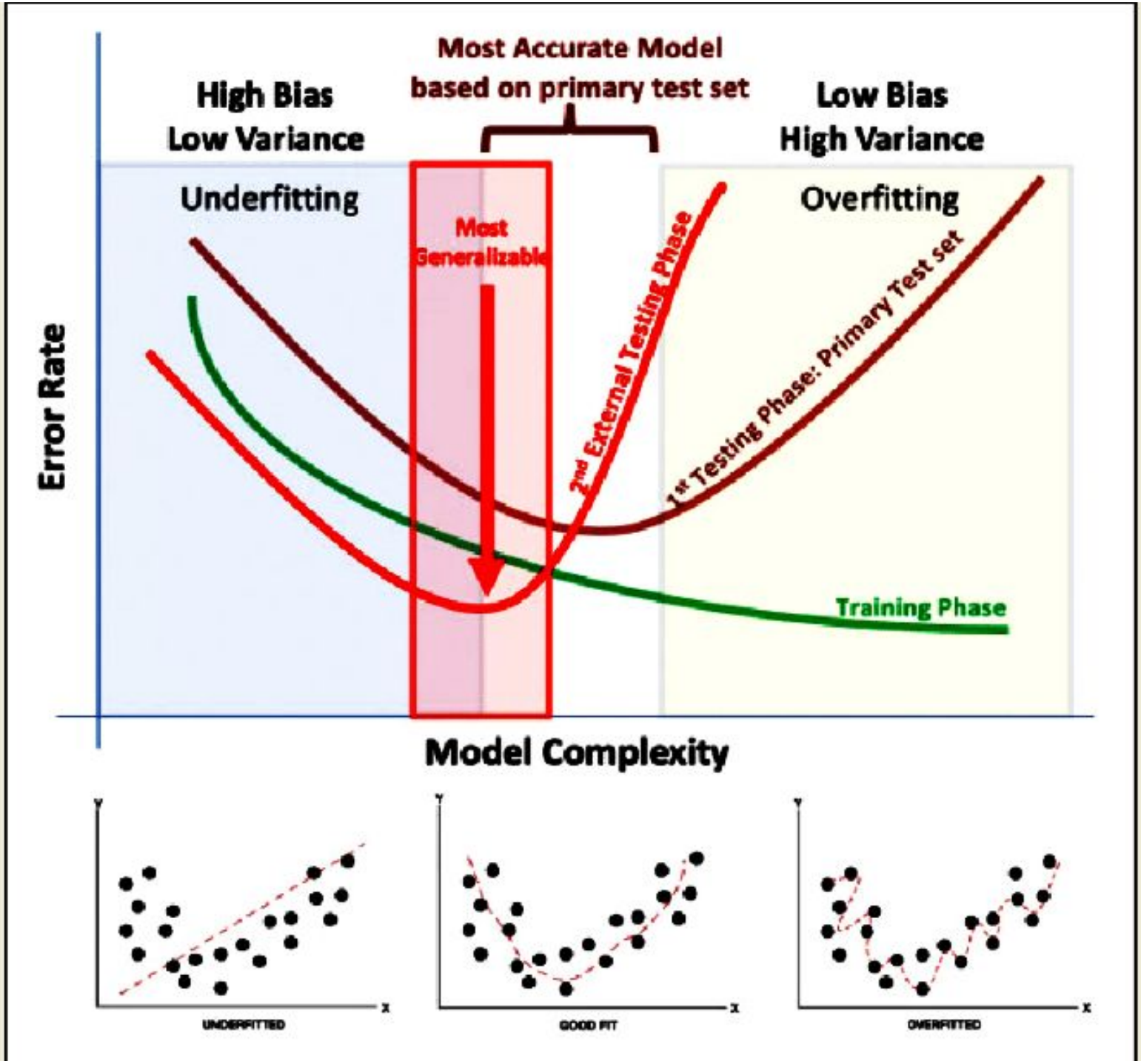
overfitting



underfitting



Good balance


Most Accurate Model
based on primary test set
High Bias
Low Variance
Low Bias
High Variance
Underfitting
Most
Generalizable
Overfitting
2nd External Testing Phase
1st Testing Phase: Primary Test set
Training Phase
Error Rate
Model Complexity
UNDERFITTED
GOOD FIT
OVERFITTED

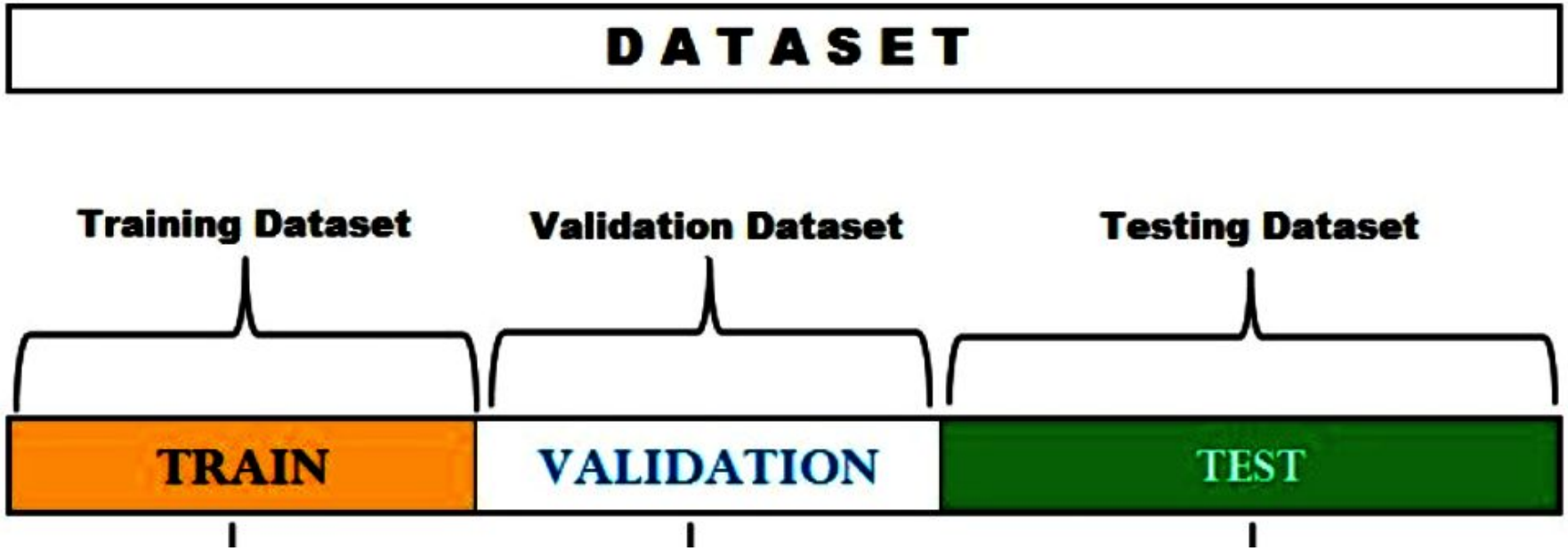
What is the Hold-out method for training ML models?

The hold-out method for training a machine learning model is the process of splitting the data into different splits and using one split for training the model and other splits for validating and testing the models. The hold-out method is used for both **model evaluation** and **model selection.**

When the entire data is used for training the model using different algorithms, the problem of evaluating the models and selecting the most optimal model remains. The primary task is to find out which model out of all models has the lowest generalization error. In other words, which model makes a better prediction on future or unseen datasets than all other models. This is where the need to have some mechanism arises wherein the model is trained on one data set and tested on another dataset. This is where the hold-out method comes into the picture.

Hold-out method for Model Selection

The hold-out method can also be used for model selection or hyperparameters tuning. As a matter of fact, at times, the model selection process is referred to as hyper-parameters tuning. In the hold-out method for model selection, the dataset is split into three different sets – training, validation, and test dataset. When using the hold out method by splitting data into three different sets, it is important to ensure that the training, validation and test datasets are representative of the entire dataset. Otherwise, the model may perform poorly on unseen data.



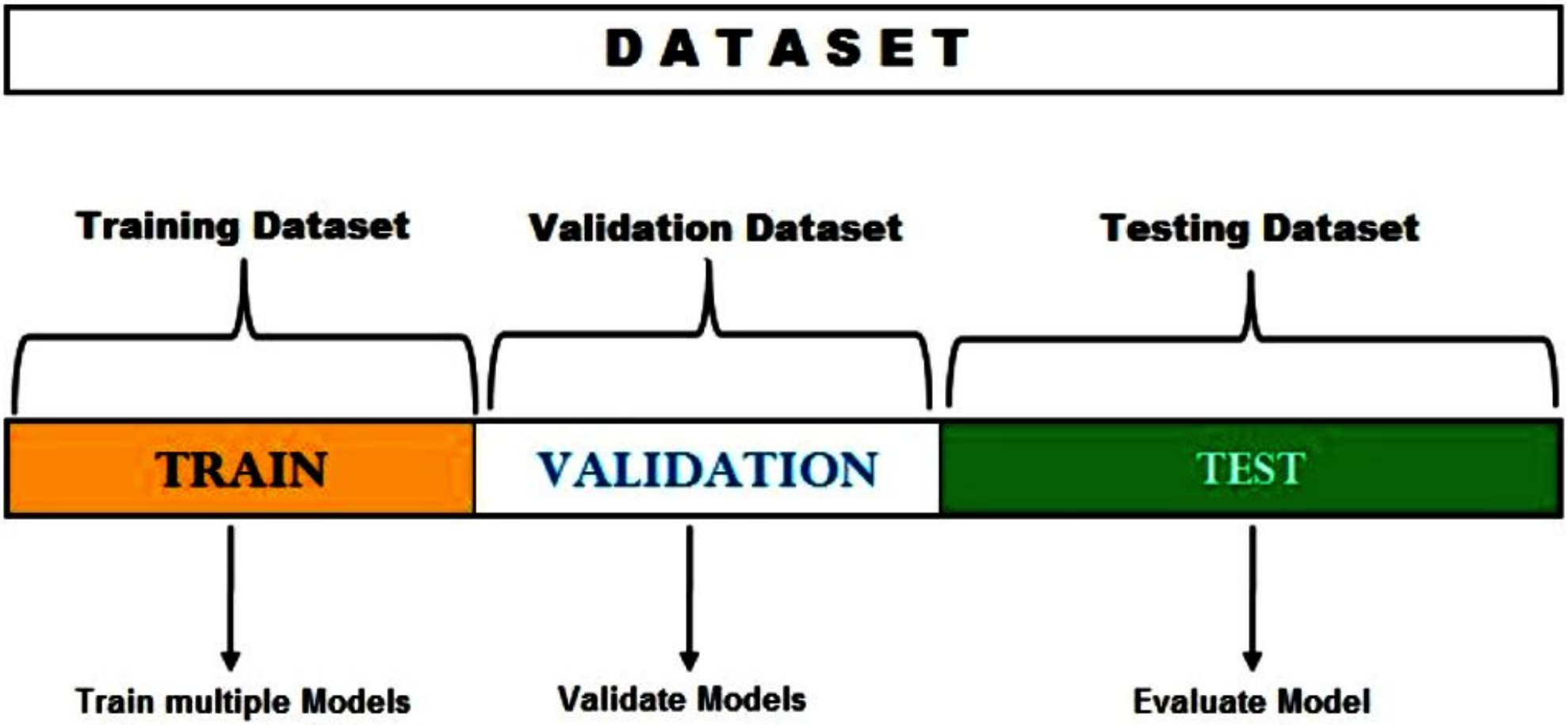


Fig 2. Hold out method – Training – Validation – Test Dataset

The following process represents the hold-out method for model selection:

1. Split the dataset in three parts – Training dataset, validation dataset and test dataset.
2. Train different models using different machine learning algorithms. For example, train the classification model using logistic regression, random forest, XGBoost, etc.
3. For the models trained with different algorithms, tune the hyper-parameters and come up with different models. For each of the algorithms mentioned in step 2, change hyperparameters settings and come with multiple models.
4. Test the performance of each of these models (belonging to each of the algorithms) on the validation dataset.
5. Select the most optimal model out of the models tested on the validation dataset. The most optimal model will have the most optimal hyperparameters settings for a specific algorithm. Going by the above example, let's say the model trained with XGBoost with the most optimal hyperparameters gets selected.
6. Test the performance of the most optimal model on the test dataset.

original dataset. The process of training, tuning, and evaluation is repeated multiple times, and the most optimal model is selected. The final model is evaluated on the test dataset.

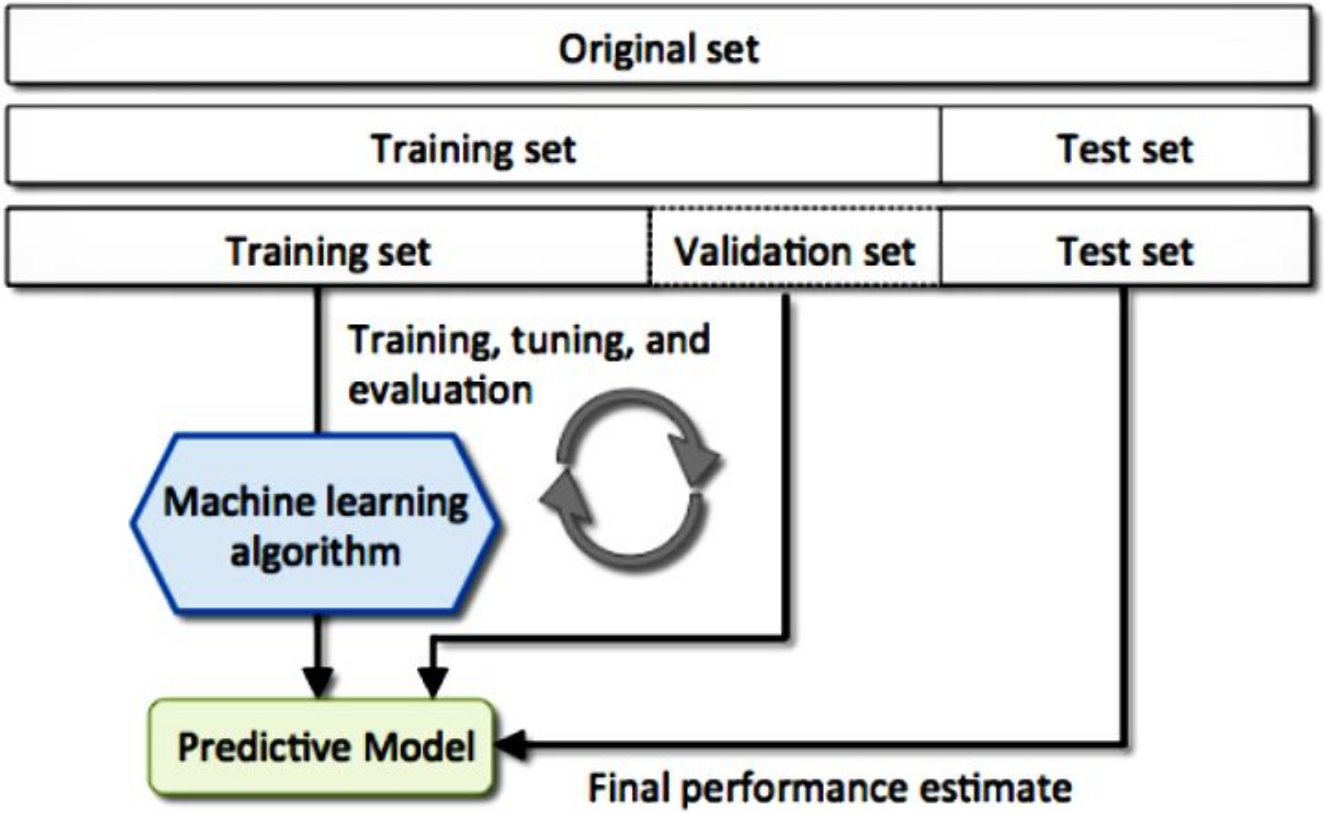


Fig 3. Hold out method for model selection

Cross-Validation in Machine Learning

Cross-validation is a technique for validating the model efficiency by training it on the subset of input data and testing on previously unseen subset of the input data. ***We can also say that it is a technique to check how a statistical model generalizes to an independent dataset***.

In machine learning, there is always the need to test the stability of the model. It means based only on the training dataset; we can't fit our model on the training dataset. For this purpose, we reserve a particular sample of the dataset, which was not part of the training dataset. After that, we test our model on that sample before deployment, and this complete process comes under cross-validation. This is something different from

1. **Validation Set Approach**
2. **Leave-P-out cross-validation**
3. **Leave one out cross-validation**
4. **K-fold cross-validation**
5. **Stratified k-fold cross-validation**

Validation Set Approach

We divide our input dataset into a training set and test or validation set in the validation set approach. Both the subsets are given 50% of the dataset.

But it has one of the big disadvantages that we are just using a 50% dataset to train our model, so the model may miss out to capture important information of the dataset. It also tends to give the underfitted model.

Leave-P-out cross-validation

In this approach, the p datasets are left out of the training data. It means, if there are total n datapoints in the original input dataset, then n-p data points will be used as the training dataset and the p data points as the validation set. This complete process is repeated for all the samples, and the average error is calculated to know the effectiveness of the model.

Leave one out cross-validation

This method is similar to the leave-p-out cross-validation, but instead of p, we need to take 1 dataset out of training. It means, in this approach, for each learning set, only one datapoint is reserved, and the remaining dataset is used to train the model. This process repeats for each datapoint. Hence for n samples, we get n different training set and n test set. It has the following features:

- In this approach, the bias is minimum as all the data points are used.
- The process is executed for n times; hence execution time is high.
- This approach leads to high variation in testing the effectiveness of the model as we iteratively check against one data point.

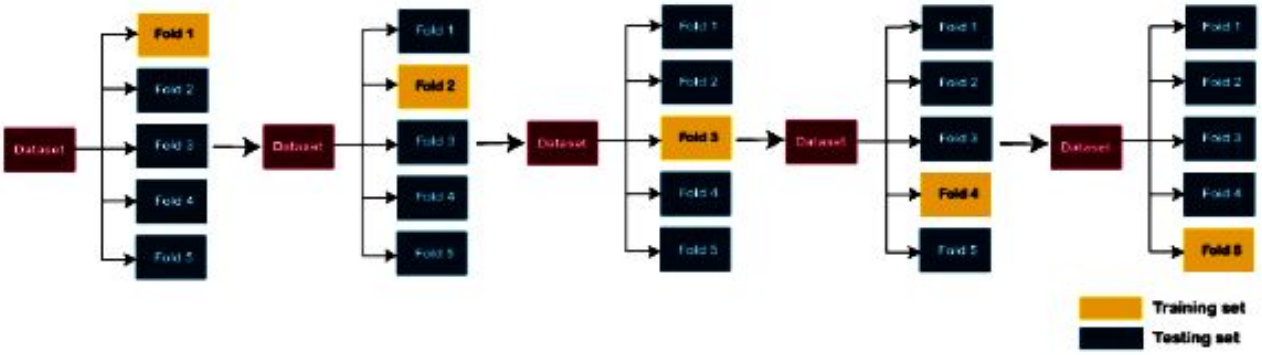
K-Fold Cross-Validation

K-fold cross-validation approach divides the input dataset into K groups of samples of equal sizes. These samples are called **folds**. For each learning set, the prediction function uses k-1 folds, and the rest of the folds are used for the test set. This approach is a very popular CV approach because it is easy to understand, and the output is less biased than other methods.

The steps for k-fold cross-validation are:

- Split the input dataset into K groups
- For each group:
  - Take one group as the reserve or test data set.
  - Use remaining groups as the training dataset
  - Fit the model on the training set and evaluate the performance of the model using the test set.

Consider the below diagram:



Stratified k-fold cross-validation

This technique is similar to k-fold cross-validation with some little changes. This approach works on stratification concept, it is a process of rearranging the data to ensure that each fold or group is a good representative of the complete dataset. To deal with the bias and variance, it is one of the best approaches.

It can be understood with an example of housing prices, such that the price of some houses can be much high than other houses. To tackle such situations, a stratified k-fold cross-validation technique is useful.