Modelling and Evaluation

Modelling

Modeling

It tries to emulate human learning by applying mathematical and statistical formulations.

1. Data Input
2. Abstraction(Learning Process)
3. Generalization

Model

- This structured representation of raw input data to the meaningful pattern is called a model.
- when the problem is related to prediction and the target field is numeric and continuous, the regression model is assigned.
- The process of assigning a model, and fitting a specific model to a data set is called model training.
- Once the model is trained, the raw input data is summarized into an abstracted form.
- If the outcome is systematically incorrect, the learning is said to have a bias.

Modelling

Target Function

A machine learning algorithm creates its cognitive capability by building a mathematical formulation or function, known as target function, based on the features in the input data set.

Hyper-parameters

- Just like a child learning things for the first time needs her parents guidance to decide whether she is right or wrong,
- In machine learning someone has to provide some non-learnable parameters, also called hyper-parameters.
- Without these human inputs, machine learning algorithms cannot be successful.

SELECTING A MODEL

Input variables

predictors, attributes, features, independent variables, or simply variables.

Input variables can be denoted by X,

while individual input variables are represented as X1, X2, X3, ..., Xn

Output variables

response or dependent variable

output variable by symbol Y

The relationship between X and Y is represented in the general form

$Y = f(X) + e$

where 'f ' is the target function and

'e' is a random error term.

Modelling and Evaluation

Cost Function

- Error function
- determines how well a machine learning model performs for a given dataset.
- a measure of how wrong the model is in terms of its ability to estimate the relationship between X and y.
- Helps to measure the extent to which the model is going wrong in estimating the relationship between X and Y.
- It calculates the difference between the expected value and predicted value and represents it as a single real number.

Loss function

It's a method of evaluating how well your algorithm models your dataset.

Determined as the difference between the actual output and the predicted output from the model for the single training example

Example : Loss function

Predicted Sales Price (In lakh)	Actual Sales Price(In lakh)	Deviation (Loss)
Bangalore: 45 Pune: 35 Chennai: 40		0 (All predictions are correct)
Bangalore: 40 Pune: 35 Chennai: 38	Bangalore: 45 Pune: 35 Chennai: 40	5 lakh for Bangalore, 2 lakh for Chennai
Bangalore: 43 Pune: 30 Chennai: 45		2 lakh for Bangalore, 5 lakh for, Pune2 lakh for Chennai,

A loss function is for a single training example,
while a cost function is an average loss over the complete train dataset.

Predictive models

Models for supervised learning or predictive models.

Try to predict certain value using the values in an input data set.

To predict the value of a category or class to which a data instance belongs to.

Examples:

1. Predicting win/loss in a cricket match
2. Predicting whether a transaction is fraud
3. Predicting whether a customer may move to another product

Classification models:

The models which are used for prediction of target features of categorical value

target feature - class label

Categories to which classes are divided into are called levels.

Classification models:

k-Nearest Neighbor (kNN), Naïve Bayes, and Decision Tree.

Predictive models

To predict numerical values of the target feature based on the predictor features.

Examples:

- Prediction of revenue growth in the succeeding year
- Prediction of rainfall amount in the coming monsoon
- Prediction of potential flu patients and demand for flu shots next winte

Regression models.

The models which are used for prediction of the numerical value of the target feature of a data instance

Examples:

- Linear Regression
- Logistic Regression
- Support Vector Machines
- Neural Network

Descriptive models

Models for unsupervised learning(clustering).

No target feature or single feature of interest in case of unsupervised learning.

Which group together similar data instances.

Data instances having a similar value of the different features are called clustering models.

Examples:

- Customer grouping or segmentation based on social, demographic, ethnic,
- etc. factors
- Grouping of music based on different aspects like genre, language, time_x0002_period, etc.
- Grouping of commodities in an inventory

Models:

K-means

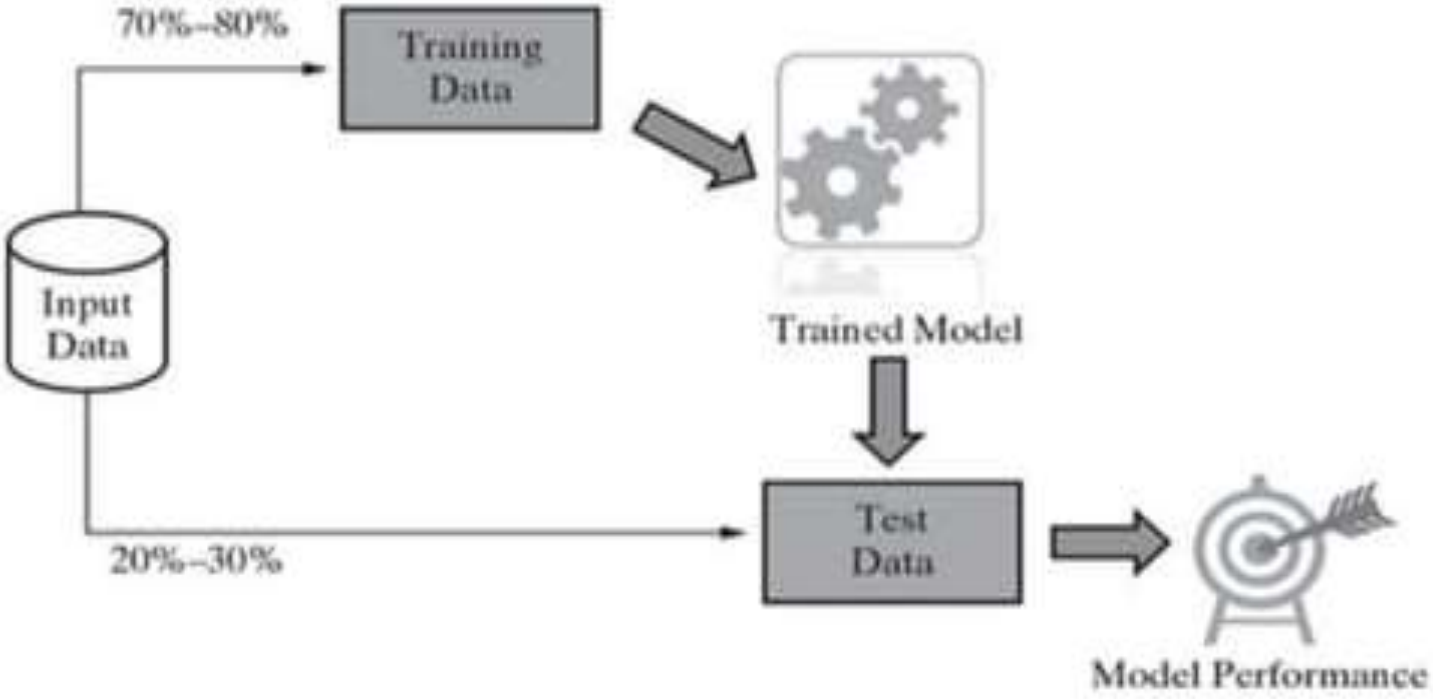
TRAINING A MODEL (FOR SUPERVISED LEARNING)

Holdout method

- This method of partitioning the input data into two parts – training and test data
- which is by holding back a part of the input data for validating the trained model

- Subset of the input data is used as the test data for evaluating the performance of a trained model.
- In general 70%–80% of the input data (which is obviously labelled) is used for model training
- The remaining 20%–30% is used as test data for validation of the performance of the model.
- Once the model is trained using the training data, the labels of the test data are predicted using the model's target function.
- Then the predicted value is compared with the actual value of the label
- The performance of the model is in general measured by the accuracy of prediction of the label value.

Holdout Method



Stratified random sampling, the whole data is broken into several homogenous groups or strata and a random sample is selected from each such stratum.

This ensures that the generated random partitions have equal proportions of each class

K-fold Cross-validation method

Process of repeated holdout is the basis of k-fold cross- validation technique.

In k-fold cross-validation, the data set is divided into k-completely distinct or non-overlapping random partitions called folds.

A special variant of holdout method, called repeated holdout, is sometimes employed to ensure the randomness of the composed data sets.

In repeated holdout, several random holdouts are used to measure the model performance.

In the end, the average of all performances is taken.

The value of 'k' in k-fold cross-validation can be set to any number.

K-Fold is validation technique in which we split the data into k-subsets

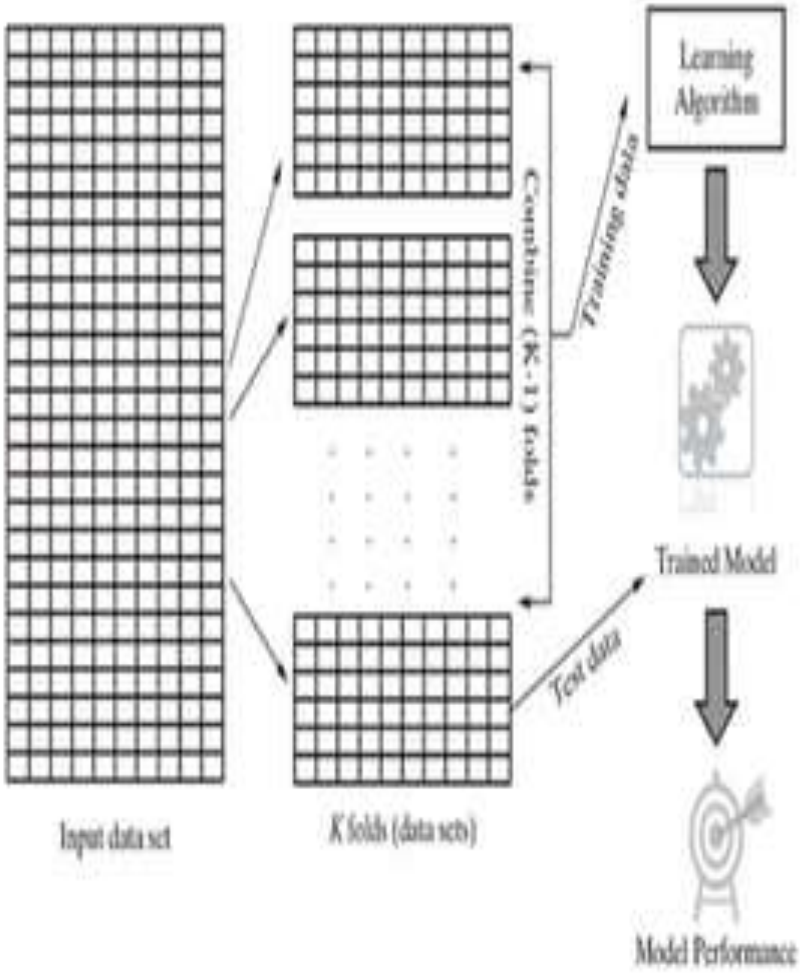
holdout method is repeated k-times where each of the k subsets are used as test set

other k-1 subsets are used for the training purpose

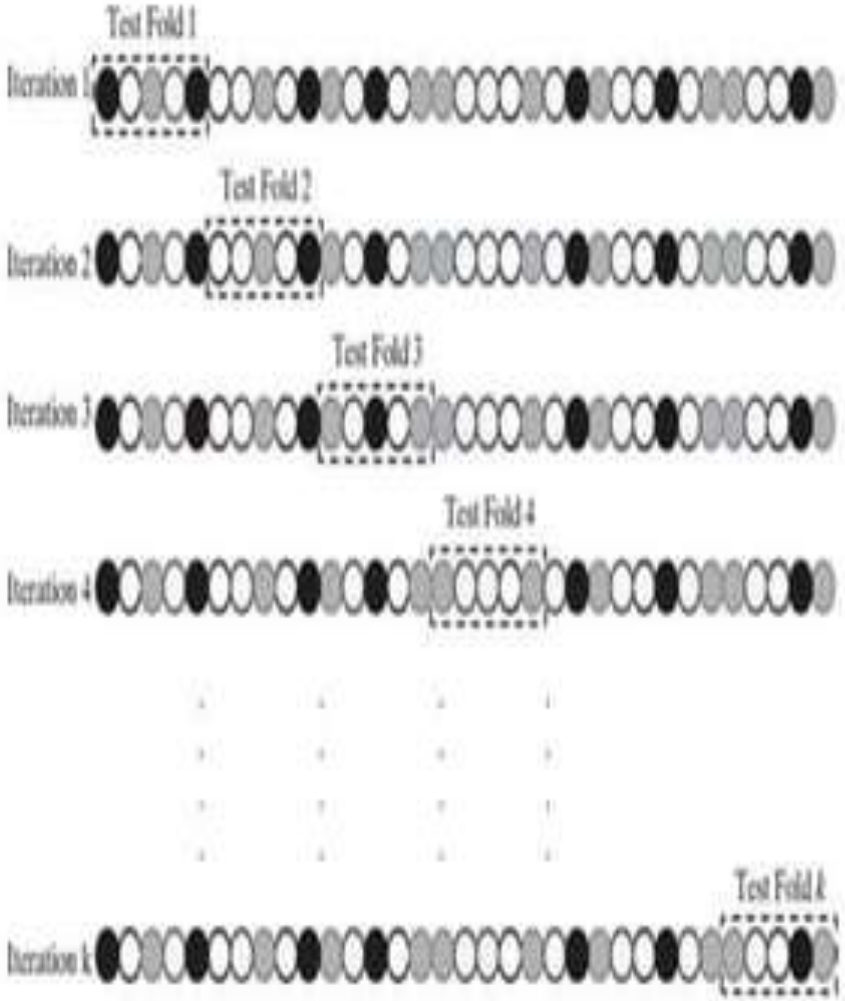
Two approaches

- 10-fold cross-validation (10-fold CV)
- Leave-one-out cross-validation (LOOCV)

Detailed approach for fold selection



Detailed approach for fold selection



Leave-one-out cross-validation (LOOCV)

Using one record or data instance at a time as a test data

The number of iterations for which it has to be run is equal to the total number of data in the input data set.

Imagine if k is equal to n where n is the number of samples in the dataset.

10-fold cross-validation

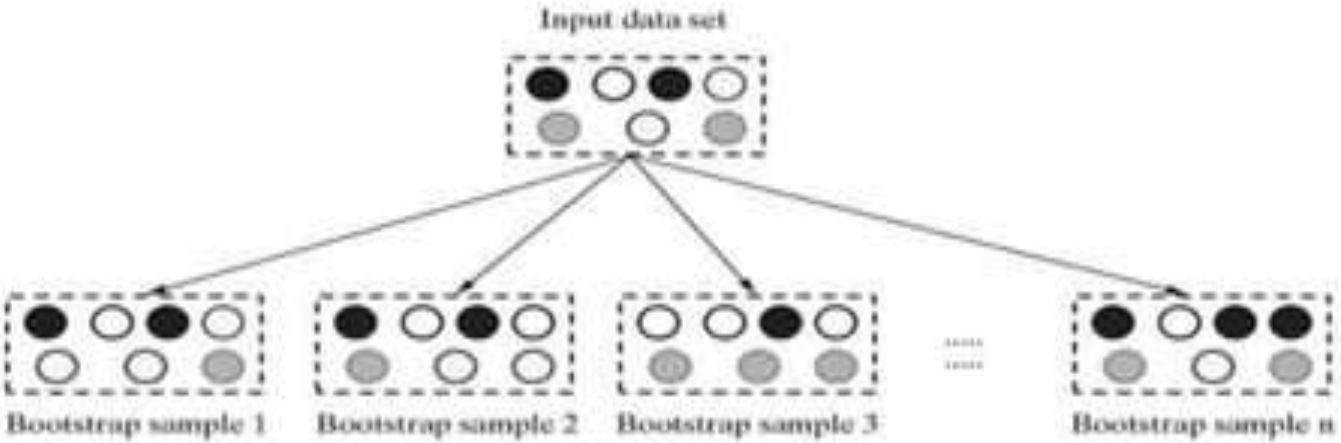
For each of the 10-folds, each comprising of approximately 10% of the data, one of the folds is used as the test data for validating model performance trained based on the remaining 9 folds (or 90% of the data).

This is repeated 10 times, once for each of the 10 folds being used as the test data and

the remaining folds as the training data.

The average performance across all folds is being reported

Bootstrap Sampling vs Cross Validation



CROSS-VALIDATION	BOOTSTRAPPING
It is a special variant of holdout method, called repeated holdout. Hence uses stratified random sampling approach (without replacement). Data set is divided into 'k' random partitions, with each partition containing approximately $\frac{n}{k}$ number of unique data elements, where 'n' is the total number of data elements and 'k' is the total number of folds.	It uses the technique of Simple Random Sampling with Replacement (SRSWR). So the same data instance may be picked up multiple times in a sample.
The number of possible training/test data samples that can be drawn using this technique is finite.	In this technique, since elements can be repeated in the sample, possible number of training/test data samples is unlimited.

Eager learner

Eager Learning learner

When a machine learning algorithm builds a model soon after receiving training data set, it is called eager learning.

It is called eager; because, when it gets the data set, the first thing it does – build the model. Then it forgets the training data.

Later, when an input data comes, it uses this model to evaluate it. Most machine learning algorithms are eager learners.

A learning algorithm that explores an entire training record set during a training phase to build a decision structure that it can exploit during the testing phase

Given a set of training set, constructs a classification model before receiving new (e.g., test) data to classify.

An eager learner abstracts away from the data during training and uses this abstraction to make predictions

- When it receive data set it starts classifying (learning)
- Then it does not wait for test data to learn
- So it takes long time learning and less time classifying data

Example: Decision Tree, Naive Bayes, Artificial Neural Networks,Support Vector Machine

Lazy learner

when a machine learning algorithm does not build a model immediately after receiving the training data, rather waits till it is provided with an input data to evaluate

Completely skips the abstraction and generalization processes

Any machine learning process that defers the majority of computation to consultation time.

It uses training data as-is,

it is also known as rote learning (i.e. memorization technique based on repetition

It is heavy dependency on the given training data instance, it is also known as instance learning.

also called non-parametric learning

- Just store Data set without learning from it
- Start classifying data when it receive Test data
- So it takes less time learning and more time classifying data

Example

K - Nearest Neighbour, Case - Based Reasoning

MODEL REPRESENTATION AND INTERPRETABILITY

The goal of supervised machine learning is to learn or derive a target function which can best determine the target variable from the set of input variables.

A key consideration in learning the target function from the training data is the extent of generalization.

Fitness of a target function approximated by a learning algorithm determines how correctly it is able to classify a set of data it has never seen.

Underfitting

Its occurrence simply means that our model or the algorithm does not fit the data well enough.

It usually happens when we have fewer data to build an accurate model .

the model is not able to learn enough from the training data.

Underfitting results in both poor performance with training data as well as poor generalization to test data

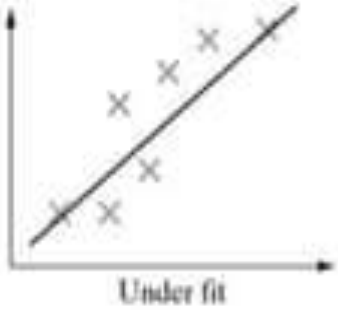
A data model is unable to capture the relationship between the input and output variables accurately,

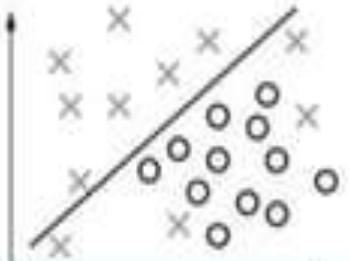
Generating a high error rate on both the training set and unseen data.

Uderfitting occurs due to high bias and low variance.

Underfitting can be avoided by

- using more training data
- reducing features by effective feature selection





Overfitting

When a model performs **very well** for training data but has **poor performance** with test data

The machine learning model learns the details and noise in the training data such that it negatively affects the performance of the model on test data.

Overfitting can happen due to low bias and high variance.

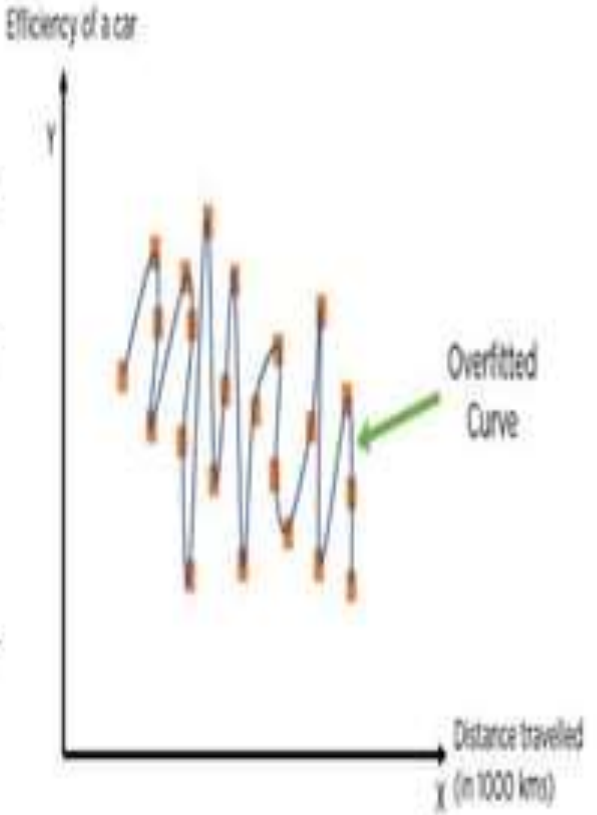
Good performance on the training data, poor generliazation to other data.

This is because the model is memorizing the data it has seen unseen examples.

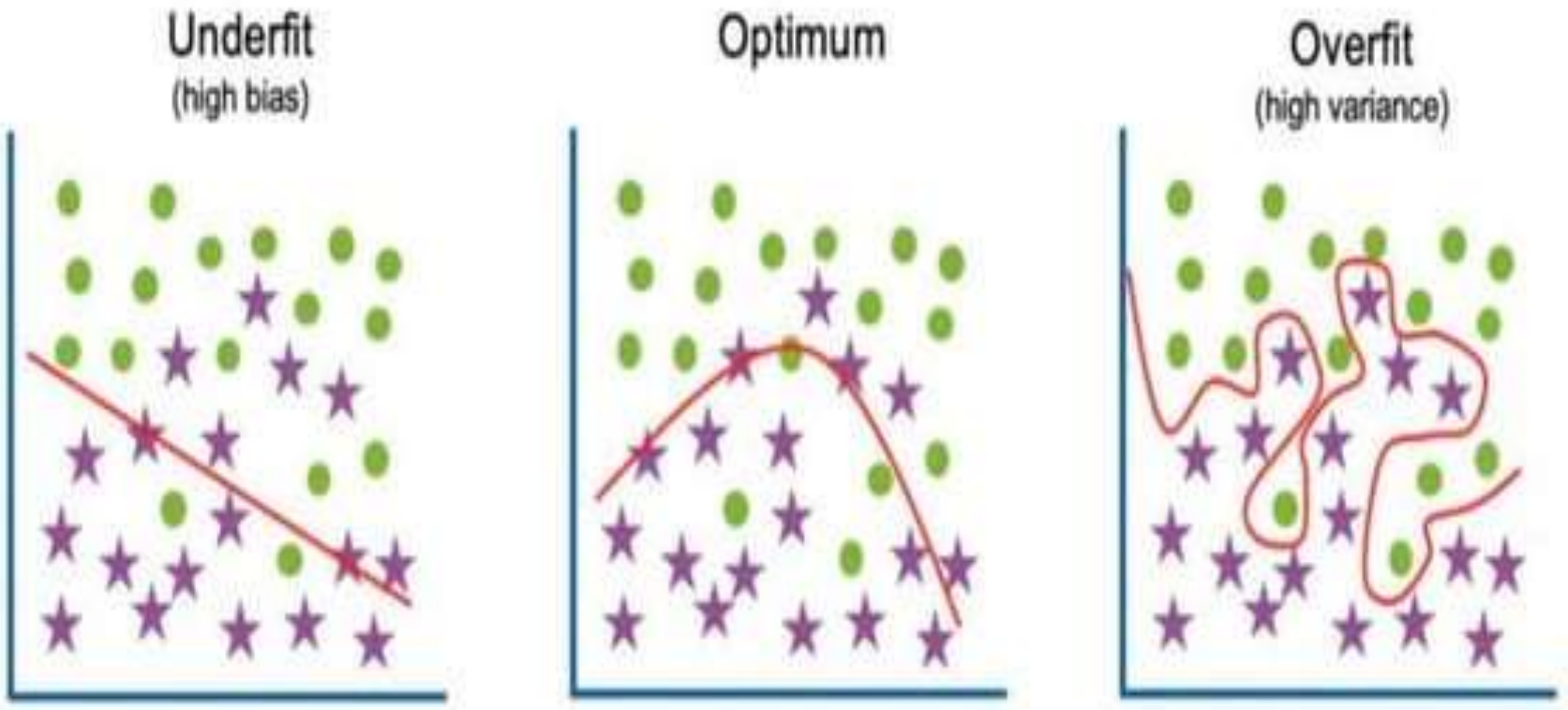
It means the more we train our model, the more chances of

Overfitting can be avoided by

1. using re-sampling techniques like k-fold cross validation
2. hold back of a validation data set
3. remove the nodes which have little or no predictive power machine learning problem.



Underfitting Vs Overfitting



High training error
High test error

Low training error
Low test error

Low training error
High test error

Bias

If the machine learning model is not accurate, it can make predictions errors, and these prediction errors are usually known as **Bias and Variance.**

An error is a measure of how accurately an algorithm can make predictions for the previously unknown dataset.

Difference between the model predictions and actual predictions.

A model has either:

Low Bias

A low bias model will make fewer assumptions about the form of the target function.

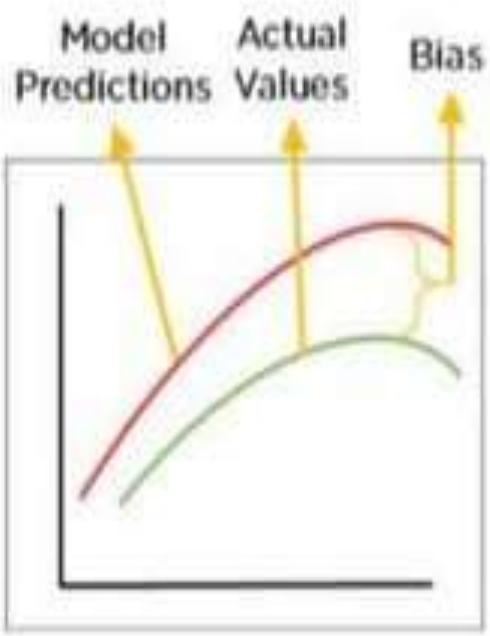
High Bias

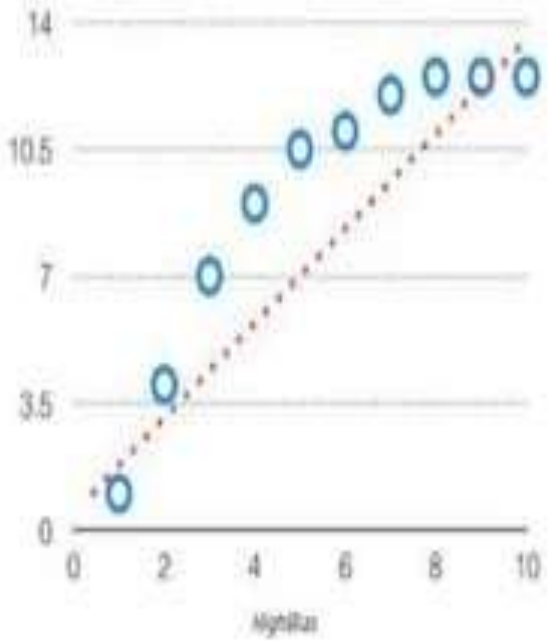
A model with a high bias makes more assumptions, and the model becomes unable to capture the important features of our dataset.

A high bias model also cannot perform well on new data.

High in biasing gives a large error in training as well as testing data.

Bootstrap Sampling





Variance

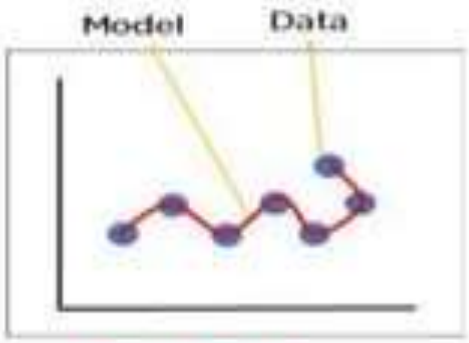
During training, it allows our model to 'see' the data a certain number of times to find patterns in it.

If it does not work on the data for long enough, it will not find patterns and bias occurs.

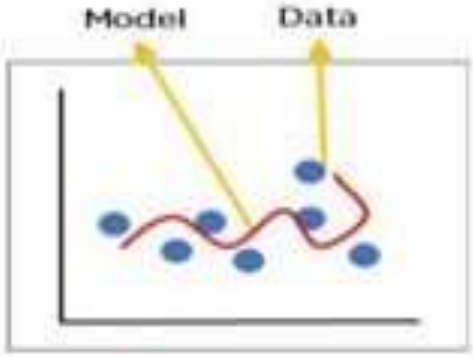
our model will perform really well on traing data and get high accuracy but will fail to perform on new, unseen data.

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.

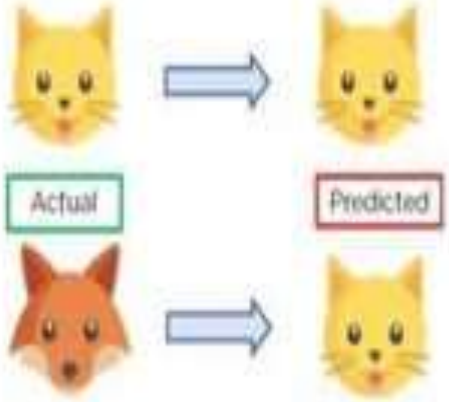
As a result, such models perform very well on training data but has high error rates on test data.



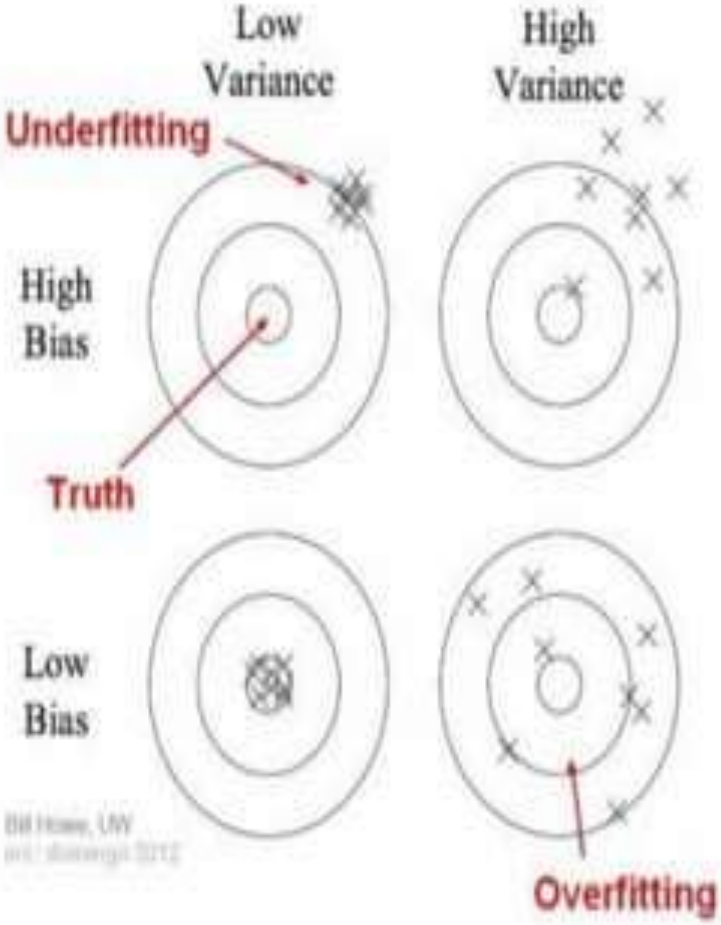
Training Data

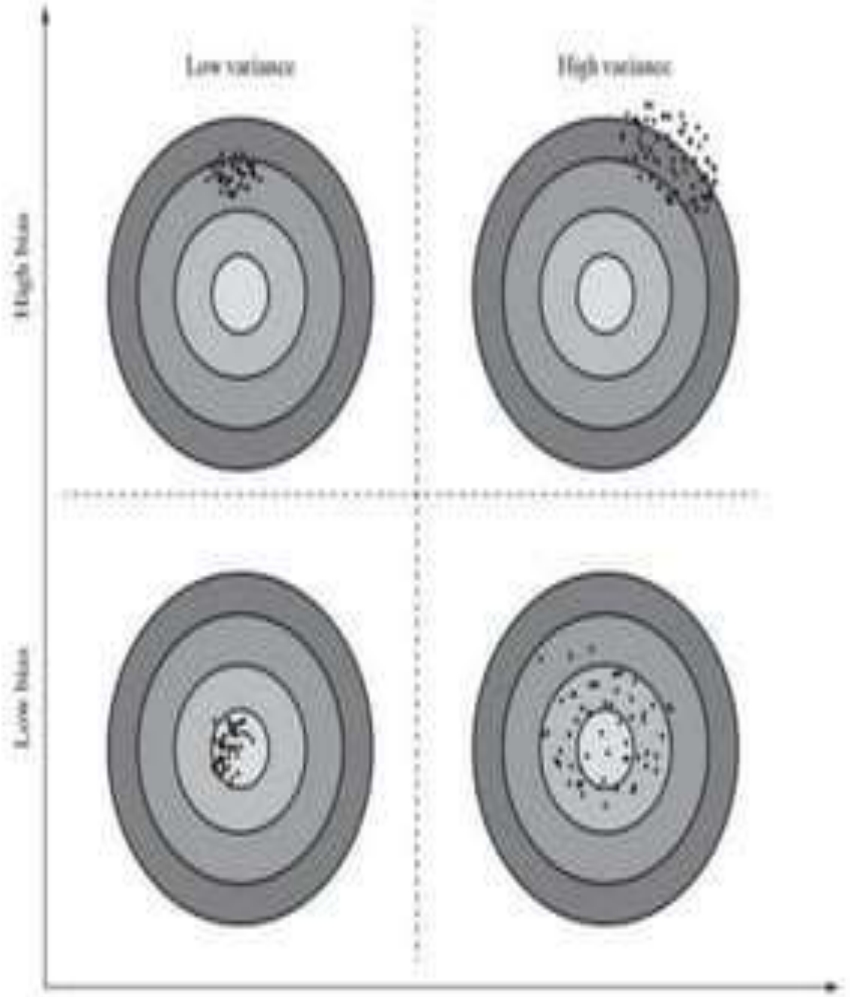


New Data



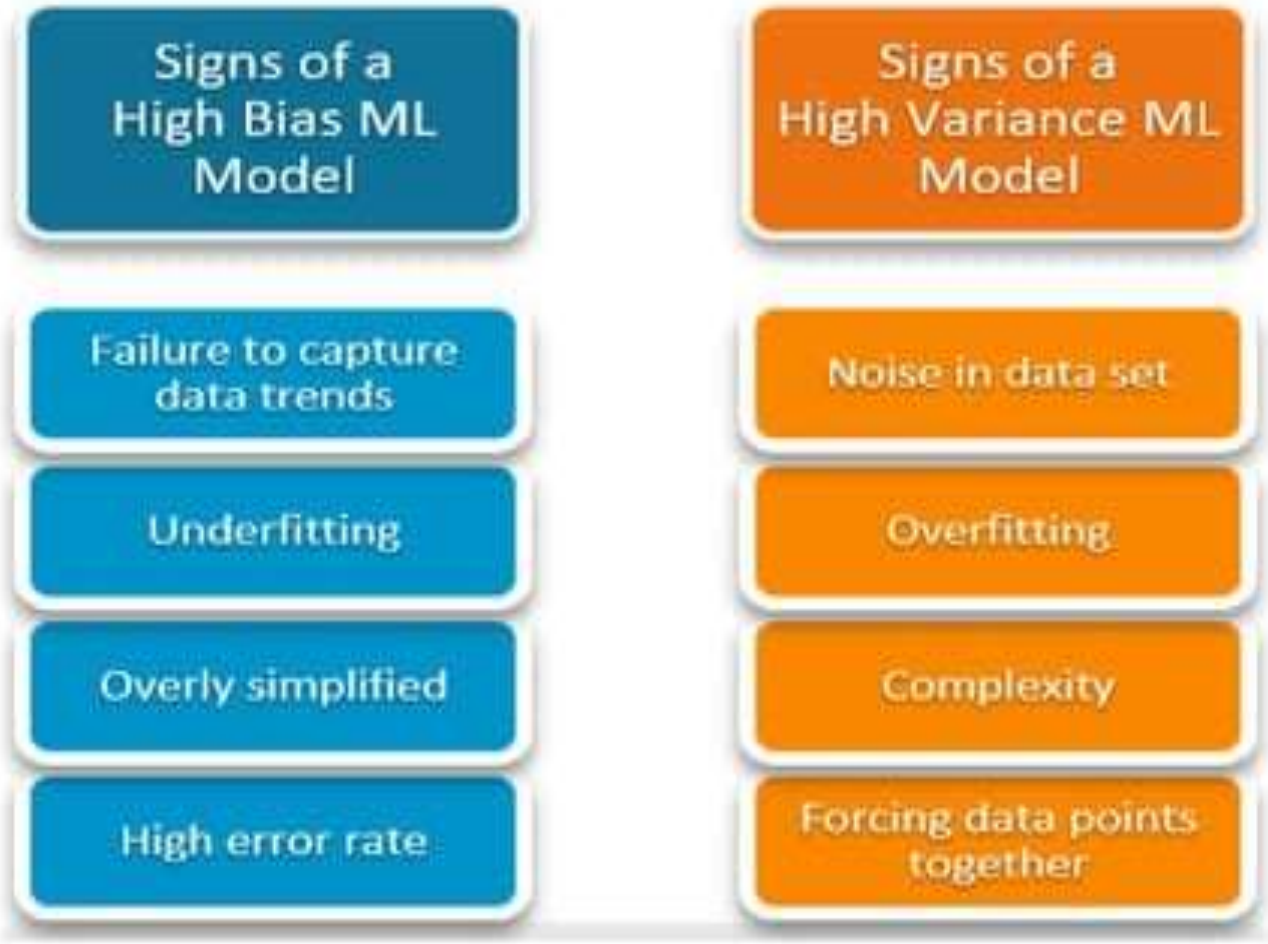
Bias-variance trade-off





Increasing the bias will decrease the variance

High Bias Vs High Variance



Parametric algorithms Vs non-parametric algorithms

Parametric algorithms

- Fixed number of parameters in the objective function or target functions.
- Any model that captures all the information about its predictions within a finite set of parameters
- A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples
- high bias but low variance.

Non -Parametric algorithms

- Models do not rely on any specific parameter
- Based on the training data, parameters are getting changed.
- low bias and high variance.
- Supervised algorithm k-Nearest Neighbors or kNN
- the user configurable parameter **'k'** can be used to do a trade-off between bias and variance. In one hand, when the value of 'k' is decreased, the model becomes simpler to fit and bias increases.
- On the other hand, when the value of 'k' is increased, the variance increases

EVALUATING PERFORMANCE OF A MODEL : Supervised learning - Classification

one major task is classification.

The responsibility of the classification model is to assign class label to the target feature based on the value of the predictor features.

For example, in the problem of predicting the win/loss in a cricket match,

the classifier will assign a class value win/loss to target feature based on the values of other features like

- whether the team won the toss,
- number of spinners in the team,
- number of wins the team had in the tournament, etc

Based on the number of correct and incorrect classifications or predictions made by a model, the accuracy of the model is calculated.

If 99 out of 100 times the model has classified correctly,

e.g. if in 99 out of 100 games what the model has predicted is same as what the outcome has been,

then the model accuracy is said to be 99%.

1% incorrect prediction

Model Performance Measures

There are four possibilities with regards to the cricket match win/loss prediction:

1. the model predicted win and the team won
2. the model predicted win and the team lost
3. the model predicted loss and the team won
4. the model predicted loss and the team lost

1. True Positive (TP)

The model has correctly classified data instances as the class of interest.

A model correctly classifies a positive sample as Positive?

2. False Positive (FP)

The model incorrectly classified data instances as the class of interest.

A model incorrectly classifies a negative sample as Positive?

3. False Negative (FN)

The model has incorrectly classified as not the class of interest.

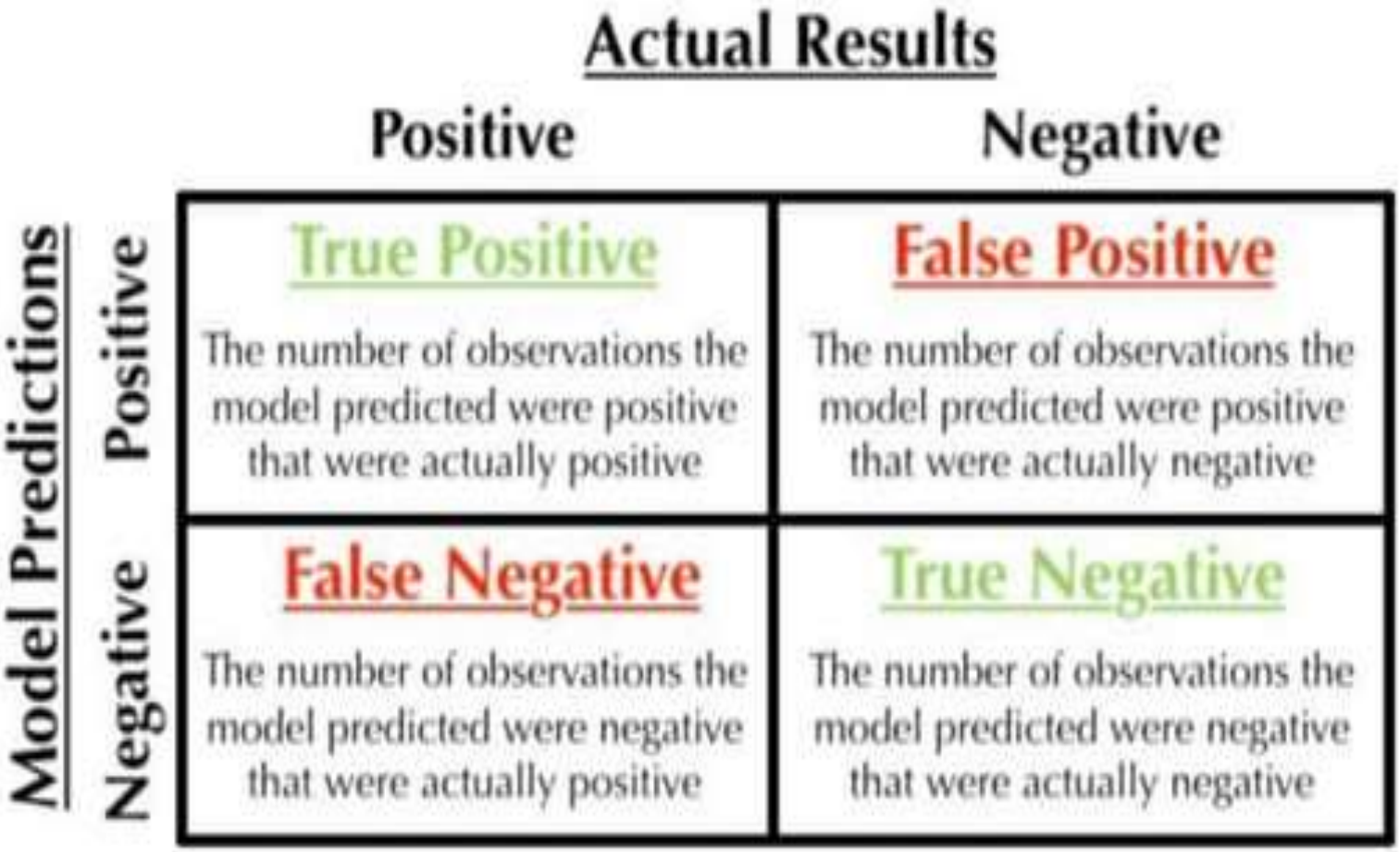
A model incorrectly classifies a positive sample as Negative?

4. True Negative (TN)

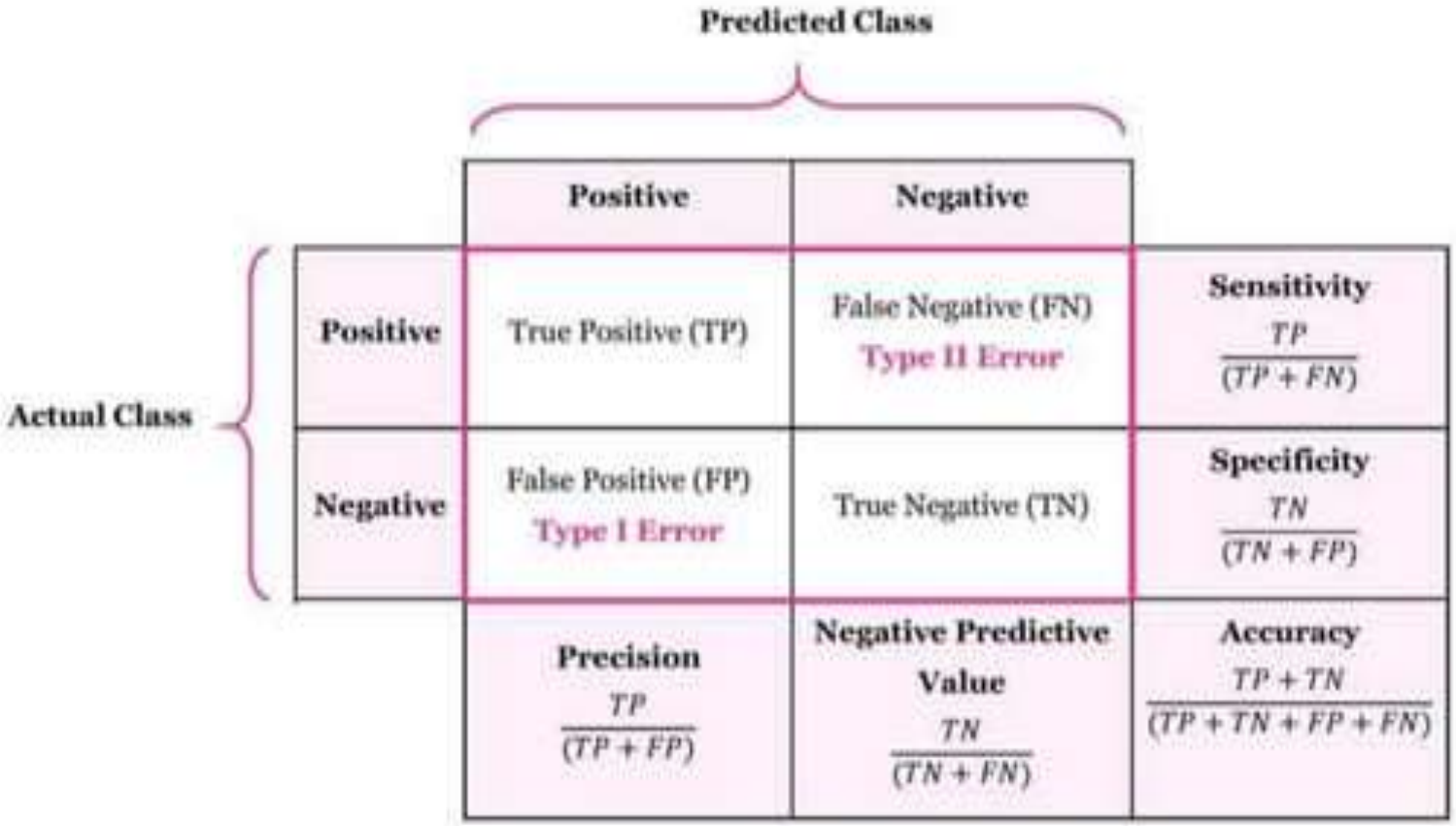
The model has correctly classified as not the class of interest.

A model correctly classifies a negative sample as Negative?

Confusion Matrix



The confusion matrix for a binary classification model


Predicted Class
Positive
Negative
Actual Class
Positive
True Positive (TP)
False Negative (FN)
Type II Error
Sensitivity
$\frac{TP}{(TP + FN)}$
Negative
False Positive (FP)
Type I Error
True Negative (TN)
Specificity
$\frac{TN}{(TN + FP)}$
Precision
$\frac{TP}{(TP + FP)}$
Negative Predictive Value
$\frac{TN}{(TN + FN)}$
Accuracy
$\frac{TP + TN}{(TP + TN + FP + FN)}$

Confusion Matrix

A matrix containing correct and incorrect predictions in the form of TPs, FPs, FNs and TNs

The win/loss prediction of cricket match has two classes of interest – win and loss.

For that reason it will generate a 2 × 2 confusion matrix.

For a classification problem involving three classes, the confusion matrix would be 3 × 3

	ACTUAL WIN	ACTUAL LOSS
Predicted Win	85	4
Predicted Loss	2	9

$$\therefore \text{Model accuracy} = \frac{TP + TN}{TP + FP + FN + TN} = \frac{85 + 9}{85 + 4 + 2 + 9} = \frac{94}{100} = 94\%$$

In context of the above confusion matrix, total count of TPs= 85, FPs=4

FNs = 2 and count of TNs = 9.

error rate

The percentage of misclassifications is indicated

$$\text{Error rate} = \frac{FP + FN}{TP + FP + FN + TN} = \frac{4 + 2}{85 + 4 + 2 + 9} = \frac{6}{100} = 6\%$$

$$= 1 - \text{Model accuracy}$$

Kappa

Kappa value can be 1 at the maximum,

which represents perfect agreement between model's prediction and actual values.

$$\text{Kappa value (k)} = \frac{P(a) - P(p_r)}{1 - P(p_r)}$$

P(a) = Proportion of observed agreement between actual and predicted in overall data set

$$= \frac{TP + TN}{TP + FP + FN + TN}$$

$P(p_r)$ = Proportion of expected agreement between actual and predicted data both in case of class of interest as well as the other classes

$$= \frac{TP + FP}{TP + FP + FN + TN} \times \frac{TP + FN}{TP + FP + FN + TN} + \frac{FN + TN}{TP + FP + FN + TN} \times \frac{FP + TN}{TP + FP + FN + TN}$$

Kappa

In context of the above confusion matrix,

total count of TPs = 85,

count of FPs = 4,

count of FNs = 2 and

count of TNs = 9.

$$\therefore P(a) = \frac{TP + TN}{TP + FP + FN + TN} = \frac{85 + 9}{85 + 4 + 2 + 9} = \frac{94}{100} = 0.94$$

$$P(p_r) = \frac{85 + 4}{85 + 4 + 2 + 9} \times \frac{85 + 2}{85 + 4 + 2 + 9} + \frac{2 + 9}{85 + 4 + 2 + 9} \times \frac{4 + 9}{85 + 4 + 2 + 9}$$

$$= \frac{89}{100} \times \frac{87}{100} + \frac{11}{100} \times \frac{13}{100} = 0.89 \times 0.87 + 0.11 \times 0.13 = 0.7886$$

$$\therefore k = \frac{0.94 - 0.7886}{1 - 0.7886} = 0.7162$$

Sensitivity

A measure of how well a machine learning model can detect positive instances.

It is also known as the true positive rate (TPR) or **Recall.**

Used to evaluate model performance because it allows us to see how many positive instances the model was able to correctly identify.

Recall indicates the proportion of correct prediction of positives to the total number of positives.

Ex:

Sensitivity or true positive rate is a measure of the proportion of people suffering from the disease who got predicted correctly as the ones suffering from the disease.

In other words, the person who is unhealthy (positive) actually got predicted as unhealthy.

$$\text{Recall} = \frac{TP}{TP + FN} \qquad \text{Sensitivity} = \frac{TP}{TP + FN} = \frac{85}{85 + 2} = \frac{85}{87} = 97.7\% \qquad \text{Recall} = \frac{TP}{TP + FN} = \frac{85}{85 + 2} = \frac{85}{87} = 97.7\%$$

Specificity

A model measures the proportion of **negative examples** which have been correctly classified.

It informs us about the proportion of actual negative cases that have gotten predicted as negative by our model.

It is the ratio of true negatives to all negatives.

$$\text{Specificity} = \frac{TN}{TN + FP} = \frac{9}{9 + 4} = \frac{9}{13} = 69.2\%$$

Precision

The ratio of correctly classified positive samples (True Positive) to a total number of classified positive samples (either correctly or incorrectly).

precision helps us to visualize the reliability of the machine learning model in classifying the model as positive.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{85}{85 + 4} = \frac{85}{89} = 95.5\%$$

F-measure/ F1 score/ F Score

- Measure of model performance
- **combines the precision and recall** into account using a single score
- It takes the **harmonic mean of precision and recall** as calculated as

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

- F1 Score is best if there is some **sort of balance** between precision (p) & recall (r) in the syste

$$F\text{-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

$$F\text{-measure} = \frac{2 \times 0.955 \times 0.977}{0.955 + 0.977} = \frac{1.866}{1.932} = 96.6\%$$

Receiver operating characteristic (ROC) curves

- Method to measure the performance of Model through visualization
- **An evaluation metric for binary classification problems.(comparing the efficiency of two** models).
- **A graph** showing the **performance of a classification model** at all classification thresholds.
- It is a probability curve that plots the **TPR against FPR**
- It shows the **efficiency of a model** in the **detection of true positives** while **avoiding the occurrence of false positives.**

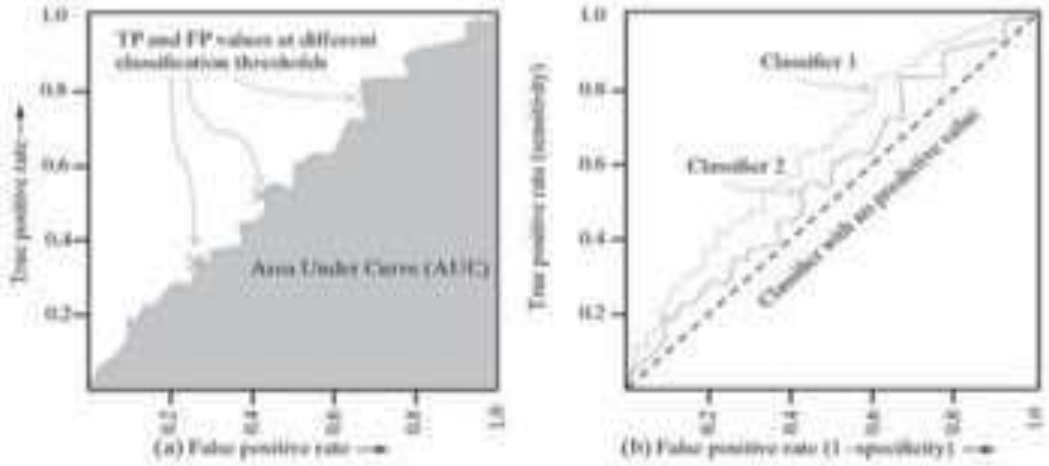


FIG. 3.8 ROC curve

Supervised learning: Regression

A regression model which ensures that the difference between predicted and actual values is low can be considered as a good model

The distance between the actual value and the fitted or predicted value, i.e. $\hat{y}$ is known as residual.

The regression model can be considered to be fitted well if the difference between actual and predicted value, i.e.

the residual value is less.

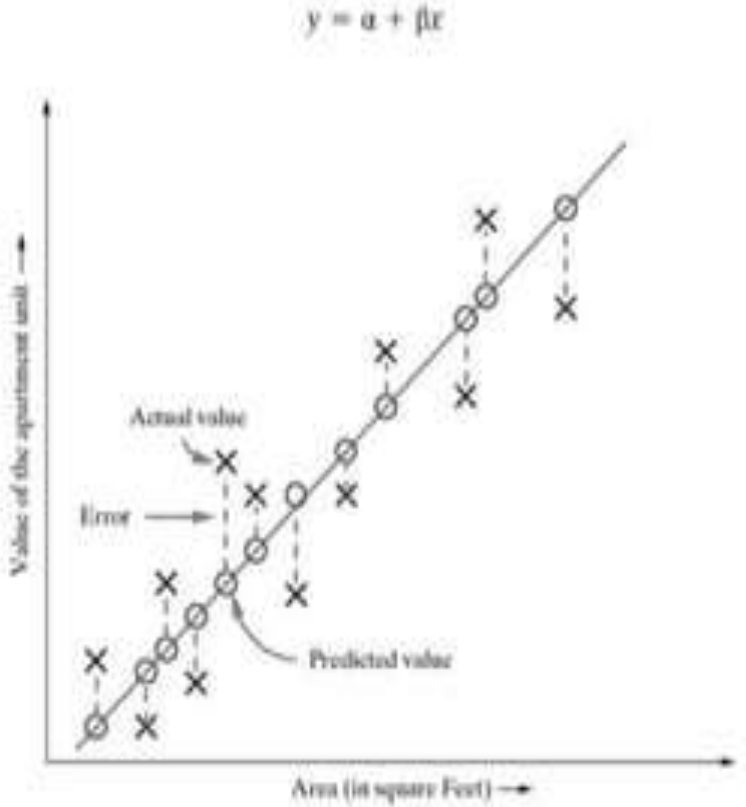


FIG. 3.9 Error – Predicted vs. actual value

R-squared

- **R-squared** is a good measure to evaluate the model fitness.
- known as the coefficient of determination, or for multiple regression
- The R-squared value lies between 0 to 1 (0%–100%) with a larger value representing a better fit

$$R^2 = \frac{SST - SSE}{SST}$$

- **Sum of Squares Total (SST)** = squared differences of each observation from the overall mean $= \sum_{i=1}^{n} (y_i - \bar{y})^2$

where $\bar{y}$ is the mean.

- **Sum of Squared Errors (SSE)** (of prediction) = sum of the squared residuals $\sum_{i=1}^{n} (Y_i - \hat{y})^2$
- where is the predicted value of *y and Y is the actual value of* y_i

Unsupervised learning : Clustering

Two inherent challenges which lie in the process of clustering

It is generally not known how many clusters can be formulated from a

1. Depends on the data set. It is completely open-ended in most cases and provided as a user input to a clustering algorithm.
2. Even if the number of clusters is given, the same number of clusters can be formed with different groups of data instances.

Internal evaluation

Measure cluster quality based on homogeneity of data belonging to the same cluster and heterogeneity of data belonging to different clusters.

The homogeneity/heterogeneity is decided by some similarity measure.

Silhouette coefficient

uses distance (Euclidean or Manhattan distances most commonly used) between data elements as a similarity measure

The value of silhouette width ranges between –1 and +1, with a high value indicating high intra-cluster homogeneity and inter-cluster heterogeneity.

For a data set clustered into 'k' clusters, silhouette width is calculated as:

$$\text{Silhouette width} = \frac{b(i) - a(i)}{\max\{a(i),\ b(i)\}}$$

a(i) is the average distance between the i th data instance and all other data instances belonging to the same cluster

b(i) is the lowest average distance between the i-the data instance and data instances of all other clusters.

Silhouette coefficient

There are four clusters namely cluster 1, 2, 3, and 4.

data element 'i' in cluster 1, resembled by the asterisk. a(i) is the average of the distances a ,

$a_{i1}, a_{i2}, a_{i3}, \ldots a_{in}$

a of the different data elements from the i^{th} data element in cluster 1

assuming there are n data elements in cluster 1.

$$b_{14}(\text{average}) = \frac{b_{14}(1) + b_{14}(2) + \ldots + b_{14}(n_4)}{(n_4)}$$

$b(i)$ = minimum [b (average), b (average), b (average)]

where n is the total number of elements in cluster 4. In the same way, we can calculate the values of b (average) and b (average).

b (i) is the minimum of all these values.

$$a(i) = \frac{a_{i1} + a_{i2} + \ldots + a_{in_i}}{n_i}$$

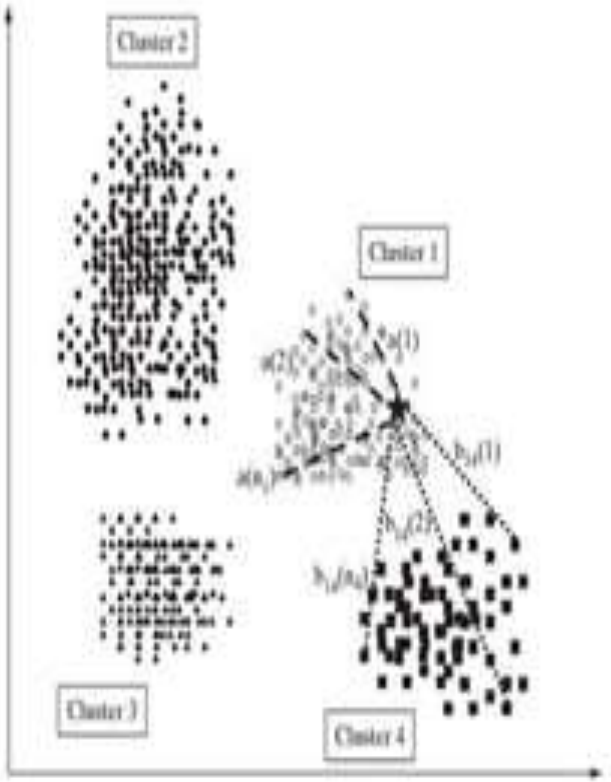


FIG. 3.10 Silhouette width calculation

External Evaluation

- the known class label is known for the data set subjected to clustering
- The cluster algorithm is assessed based on how close the results are compared to those known class labels
- **purity** is one of the most popular measures of cluster algorithms –
- evaluates the extent to which clusters contain a single class.Is are not a part of the data used in clustering
- For a data set having '*n*' data instances and '*c*' known class labels which generates '*k*' clusters, purity is measured as

$$\text{Purity} = \frac{1}{n}\sum_{k}\max(k \cap c)$$

IMPROVING PERFORMANCE OF A MODEL

- Model selection is done one several aspects:
  1. Type of learning the task in hand, i.e. supervised or unsupervised
  2. Type of the data, i.e. categorical or numeric
  3. Sometimes on the problem domain
  4. Above all, experience in working with different models to solve problems of diverse domains

IMPROVING PERFORMANCE OF A MODEL

- Model parameter tuning is the process of adjusting the model fitting options. For example, in the popular classification model k-Nearest Neighbour (kNN),
- using different values of 'k' or the number of nearest neighbours to be considered, the model can be tuned.
- In the same way, a number of hidden layers can be adjusted to tune the performance in neural networks model.

Ensemble

- Several models may be combined together
- This approach of combining different models with diverse strengths is known as **ensemble**
- Models in such combination are complimentary to each other
- one model may learn one type data sets well while struggle with another type of data set.
- Another model may perform well with the data set which the first one struggled with.
- Ensemble methods combine weaker learners to create stronger ones.
- EX: bootstrap aggregating or bagging,

  Random forest

Ensemble

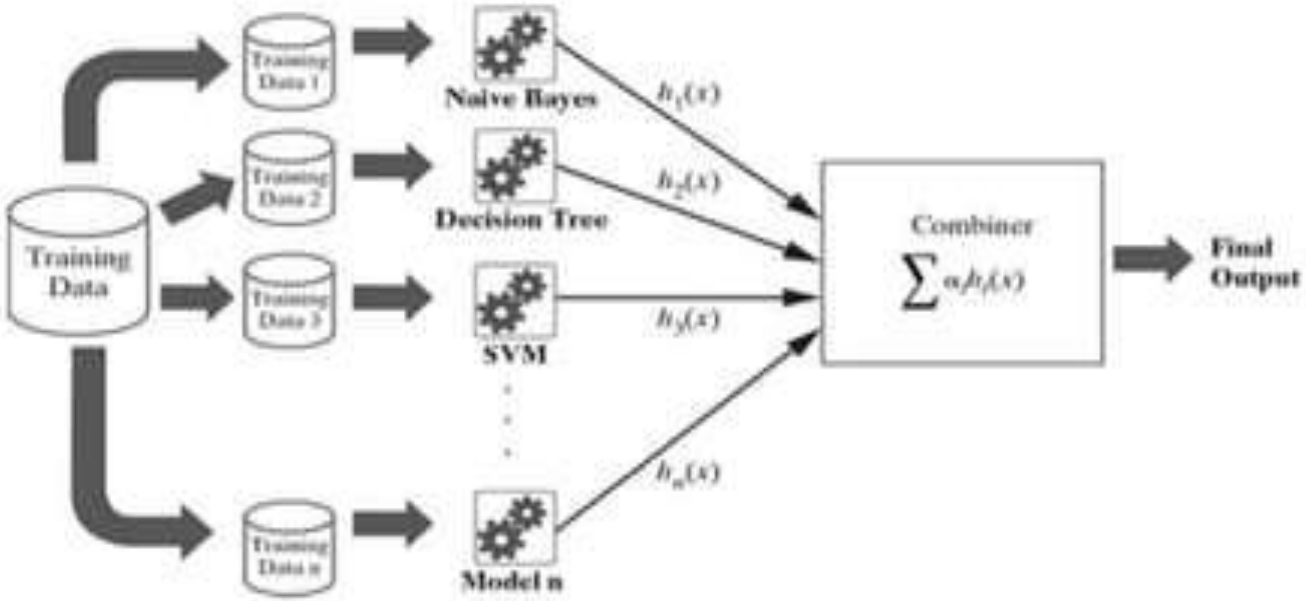


FIG. 3.11 Ensemble

Steps in Ensemble Process

- Build a number of models based on the training data
- For diversifying the models generated, the training data subset can be varied using the allocation function.

Sampling techniques like bootstrapping may be used to generate unique training data sets.

- Alternatively, the same training data may be used but the models combined are quite varying, e.g, SVM, neural network, kNN, etc.
- The outputs from the different models are combined using a combination function. A very simple strategy of combining, say in case of a prediction task using ensemble, can be majority voting of the different models combined.

For example, 3 out of 5 classes predict 'win' and 2 predict 'loss' – then the final outcome of the ensemble using majority vote would be a 'win'.

Ensemble Methods

Bagging

- Uses bootstrap sampling method (to generate multiple training data sets. These training data sets are used to generate (or train) a set of models using the same learning algorithm.
- Then the outcomes of the models are combined by majority voting (classification) or by average (regression).
- Bagging is a very simple ensemble technique which can perform really well for unstable learners like a decision tree, in which a slight change in data can impact the outcome of a model significantly

Boosting

- Weaker learning models are trained on resampled data and the outcomes are combined using a weighted voting approach based on the performance of different models.

Algorithms: Adaptive boosting or **AdaBoost**

Random forest

Another ensemble-based technique. It is an ensemble of decision trees – hence the name random forest to indicate a forest of decision trees.