UNIT-V

UNIT-V

Introduction to ROC Curves: Performance of diagnostic tests, confusion Matrix, true and false positives, precession and recall measures. Roc curves, Area Under the Curve, simple applications and algorithms in machine learning

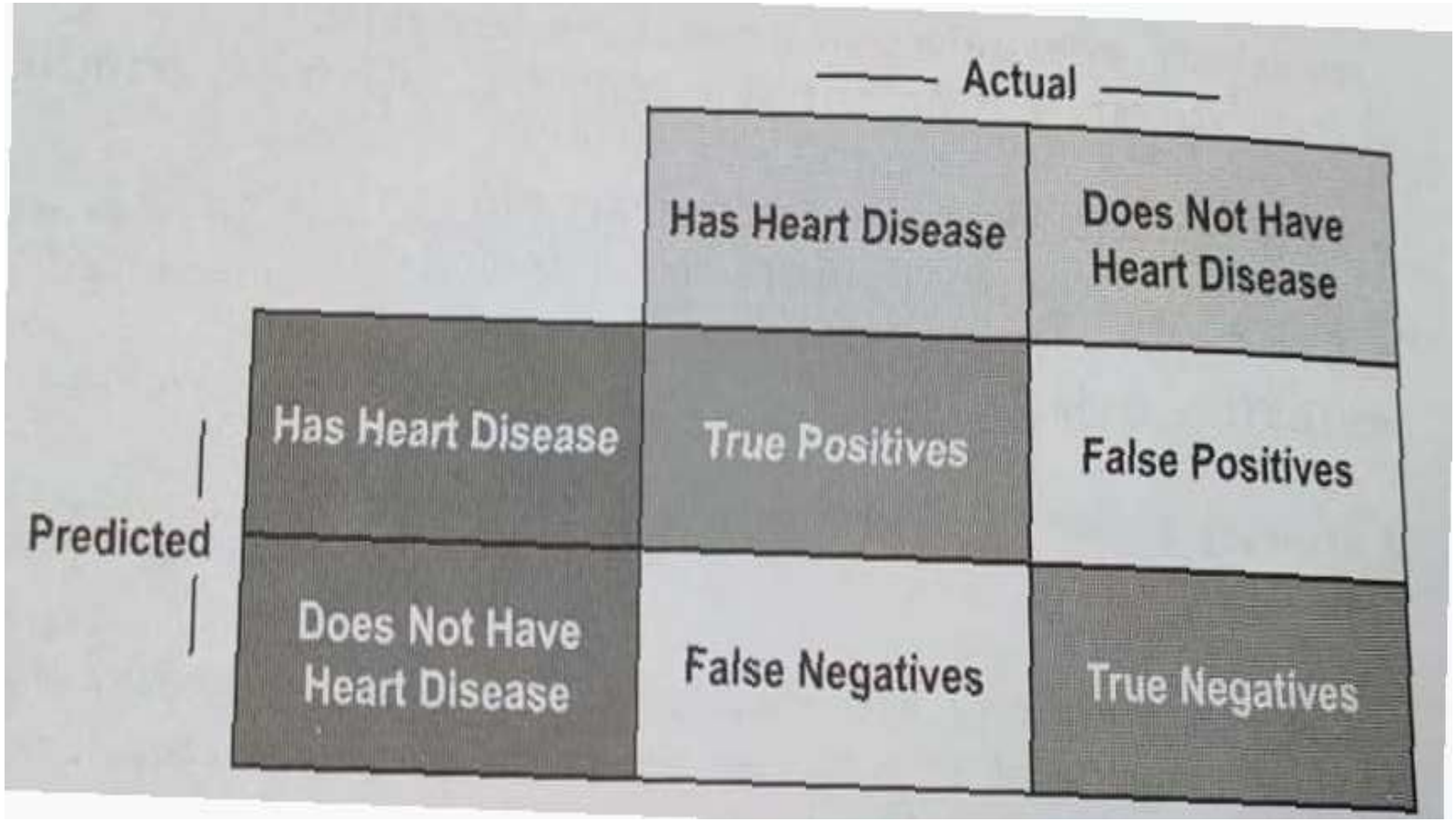
Cluster Analysis: Introduction to Clustering, types of clustering, CART algorithm

Reference Books:

1. McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.".
2. Swaroop, C. H. (2003). A Byte of Python. Python Tutorial.
3. Ken Black, sixth Editing. Business Statistics for Contemporary Decision Making. “John Wiley & Sons, Inc”.
4. Anderson Sweeney Williams (2011). Statistics for Business and Economics. “Cengage Learning”.

Introduction to ROC Curves - Receiver Operating Characteristic Curves

- ROC curves, short for **Receiver Operating Characteristic Curves**, are a graphical representation of the performance of a classification model.
- They are commonly used in machine learning and statistics to assess the ability of a model to discriminate between different classes or categories.
- **The ROC curve is used to assess the overall diagnostic performance of a test and to compare the performance of two or more diagnostic tests.**
- ROC curves provide a visual representation of the trade-off between the true positive rate (TPR) and the false positive rate (FPR) as the discrimination threshold of the model varies.
- **To understand ROC curves, it's helpful to first define a few key terms:**
- **True Positive (TP):** The model correctly predicts the positive class.
- **False Positive (FP):** The model incorrectly predicts the positive class when it is actually negative.
- **True Negative (TN):** The model correctly predicts the negative class.
- **False Negative (FN):** The model incorrectly predicts the negative class when it is actually positive.


Actual
Has Heart Disease
Does Not Have Heart Disease
Predicted
Has Heart Disease
True Positives
False Positives
Does Not Have Heart Disease
False Negatives
True Negatives

Understanding True Positive, True Negative, False Positive and False Negative in a Confusion Matrix

True Positive (TP)

The predicted value matches the actual value

The actual value was positive and the model predicted a positive value

True Negative (TN)

The predicted value matches the actual value

The actual value was negative and the model predicted a negative value

False Positive (FP) – Type 1 error

The predicted value was falsely predicted

The actual value was negative but the model predicted a positive value

Also known as the Type 1 error

False Negative (FN) – Type 2 error

The predicted value was falsely predicted

The actual value was positive but the model predicted a negative value

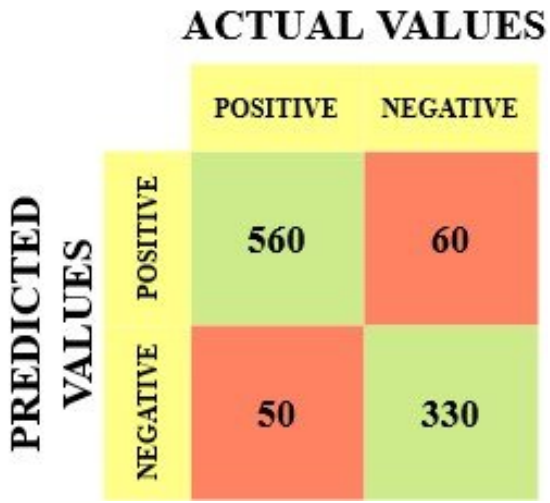
Also known as the Type 2 error

Example to better understand.

❖ Suppose we had a classification dataset with 1000 data points. We fit a classifier on it and get the below confusion matrix:

The different values of the Confusion matrix would be as follows:

- ❖ **True Positive (TP)** = 560; meaning 560 positive class data points were correctly classified by the model
- ❖ **True Negative (TN)** = 330; meaning 330 negative class data points were correctly classified by the model
- ❖ **False Positive (FP)** = 60; meaning 60 negative class data points were incorrectly classified as belonging to the positive class by the model
- ❖ **False Negative (FN)** = 50; meaning 50 positive class data points were incorrectly classified as belonging to the negative class by the model



Why Do We Need a Confusion Matrix?

- Before we answer this question, let's think about a hypothetical classification problem.
- Let's say you want to predict how many people are infected with a **contagious virus** in times before they show the symptoms, and isolate them from the healthy population.
- The two values for our target variable would be: **Sick and Not Sick.**
- Now, you must be wondering – why do we need a confusion matrix when we have our all-weather friend – Accuracy? Well, let's see where accuracy falters.
- Our dataset is an example of an imbalanced dataset. There are 947 data points for the negative class and 3 data points for the positive class. This is how we'll calculate the accuracy:

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

Let’s see how our model performed:

ID	Actual Sick?	Predicted Sick?	Outcome
1	1	1	TP
2	0	0	TN
3	0	0	TN
4	1	1	TP
5	0	0	TN
6	0	0	TN
7	1	0	FP
8	0	1	FN
9	0	0	TN
10	1	0	FP
:	:	:	:
1000	0	0	FN

The total outcome values are:

TP = 30, TN = 930, FP = 30, FN = 10

So, the accuracy for our model turns out to be:

$$Accuracy = \frac{30 + 930}{30 + 30 + 930 + 10} = 0.96$$

960/1000 = .96

- 96%! , But it is giving the wrong idea about the result. Think about it.

- Our model is saying "I can predict sick people 96% of the time". However, it is doing the opposite. It is predicting the people who will not get sick with 96% accuracy while the sick are spreading the virus!
- Do you think this is a correct metric for our model given the seriousness of the issue? Shouldn't we be measuring how many positive cases we can predict correctly to arrest the spread of the contagious virus? Or maybe, out of the correctly predicted cases, how many are positive cases to check the reliability of our model?
- This is where we come across the dual concept of Precision and Recall.

Precision vs. Recall

- Precision tells us how many of the correctly predicted cases actually turned out to be positive.
- Precision is how good the model is at predicting a specific category.

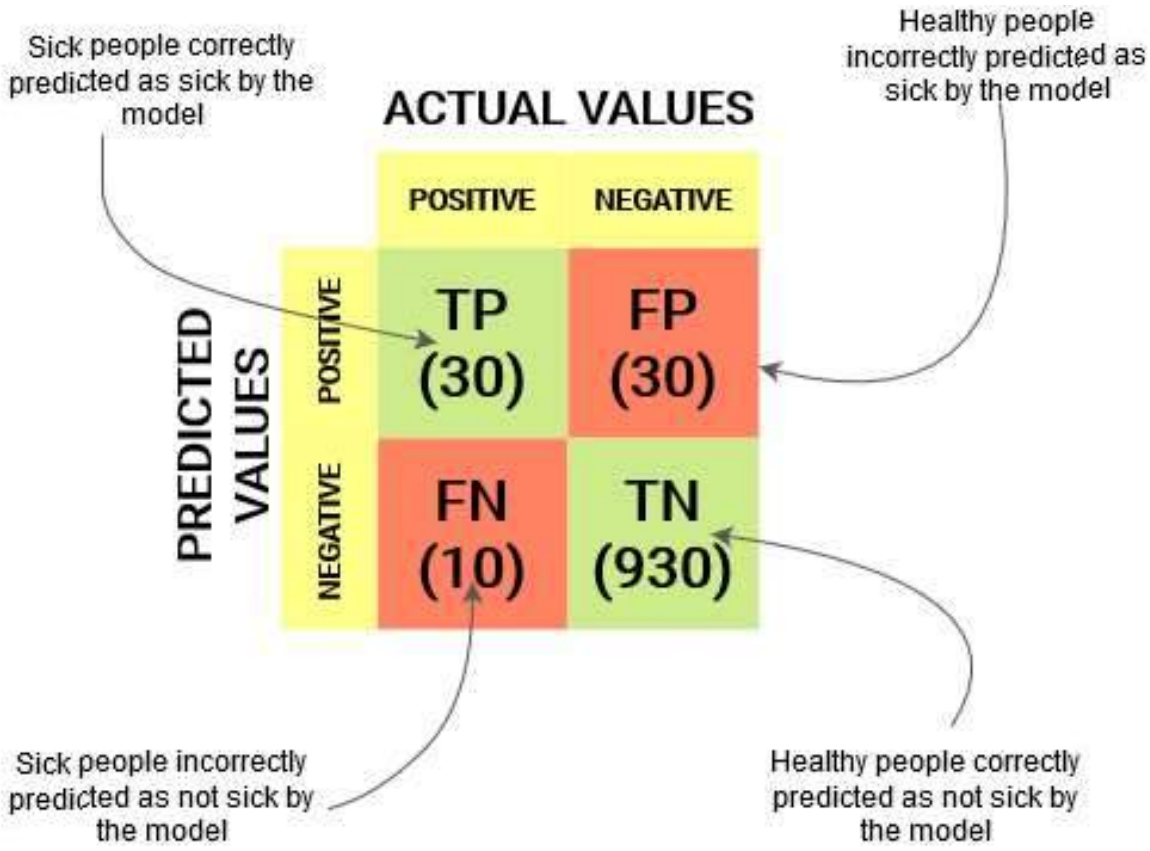
Here's how to calculate Precision:

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

- This would determine whether our model is reliable or not.
- Recall tells us how many of the actual positive cases we were able to predict correctly with our model.
- Recall tells you how many times the model was able to detect a specific category.

And here's how we can calculate Recall:

$$Recall = \frac{TP}{TP + FN}$$


Sick people correctly predicted as sick by the model
Healthy people incorrectly predicted as sick by the model
ACTUAL VALUES
POSITIVE
NEGATIVE
PREDICTED VALUES
POSITIVE
NEGATIVE
TP (30)
FP (30)
FN (10)
TN (930)
Sick people incorrectly predicted as not sick by the model
Healthy people correctly predicted as not sick by the model

- We can easily calculate Precision and Recall for our model by plugging in the values into the above questions:

$$Precision = \frac{30}{30 + 30} = 0.5$$

$$Recall = \frac{30}{30 + 10} = 0.75$$

- 50% percent of the correctly predicted cases turned out to be positive cases. Whereas 75% of the positives were successfully predicted by our model.
- **Precision** is a useful metric in cases where False Positive is a higher concern than False Negatives.
- **Precision** is important in music or video recommendation systems, e-commerce websites, etc. Wrong results could lead to customer churn and be harmful to the business.

- Recall is a useful metric in cases where False Negative outplays False Positive.
- **Recall is important in medical cases where it doesn't matter whether we raise a false alarm but the actual positive cases should not go undetected!**
- In our example, Recall would be a better metric because we don't want to accidentally discharge an infected person and let them mix with the healthy population thereby spreading the contagious virus. Now you can understand why accuracy was a bad metric for our model.
- **But there will be cases where there is no clear distinction between whether Precision is more important or Recall. What should we do in those cases? We combine them!**

F1-Score

- In practice, when we try to increase the precision of our model, the recall goes down, and vice-versa. The F1-score captures both the trends in a single value:

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

- F1-score is a harmonic mean of Precision and Recall, and so it gives a combined idea about these two metrics. It is maximum when Precision is equal to Recall.
- But there is a catch here.
- The interpretability of the F1-score is poor. This means that we don't know what our classifier is maximizing – precision or recall? So, we use it in combination with other evaluation metrics which gives us a complete picture of the result.

Confusion Matrix using scikit-learn in Python

- You know the theory – now let's put it into practice. Let's code a confusion matrix with the Scikit-learn (sklearn) library in Python.
- Sklearn has two great functions: confusion_matrix() and classification_report().
- **Sklearn confusion_matrix()** returns the values of the Confusion matrix. The output is, however, slightly different from what we have studied so far. It takes the **rows as Actual values** and the **columns as Predicted values**. The rest of the concept remains the same.
- **Sklearn classification_report()** outputs precision, recall and f1-score for each target class. In addition to this, it also has some extra values: micro avg, macro avg, and weighted avg

Mirco average is the precision/recall/f1-score calculated for all the classes.

$$\text{Micro avg Precision} = \frac{TP1 + TP2}{TP1 + TP2 + FP1 + FP2}$$

Macro average is the average of precision/recall/f1-score.

$$\text{Macro avg Precision} = \frac{P1 + P2}{2}$$

Weighted average is just the weighted average of **precision/recall/f1-score.**

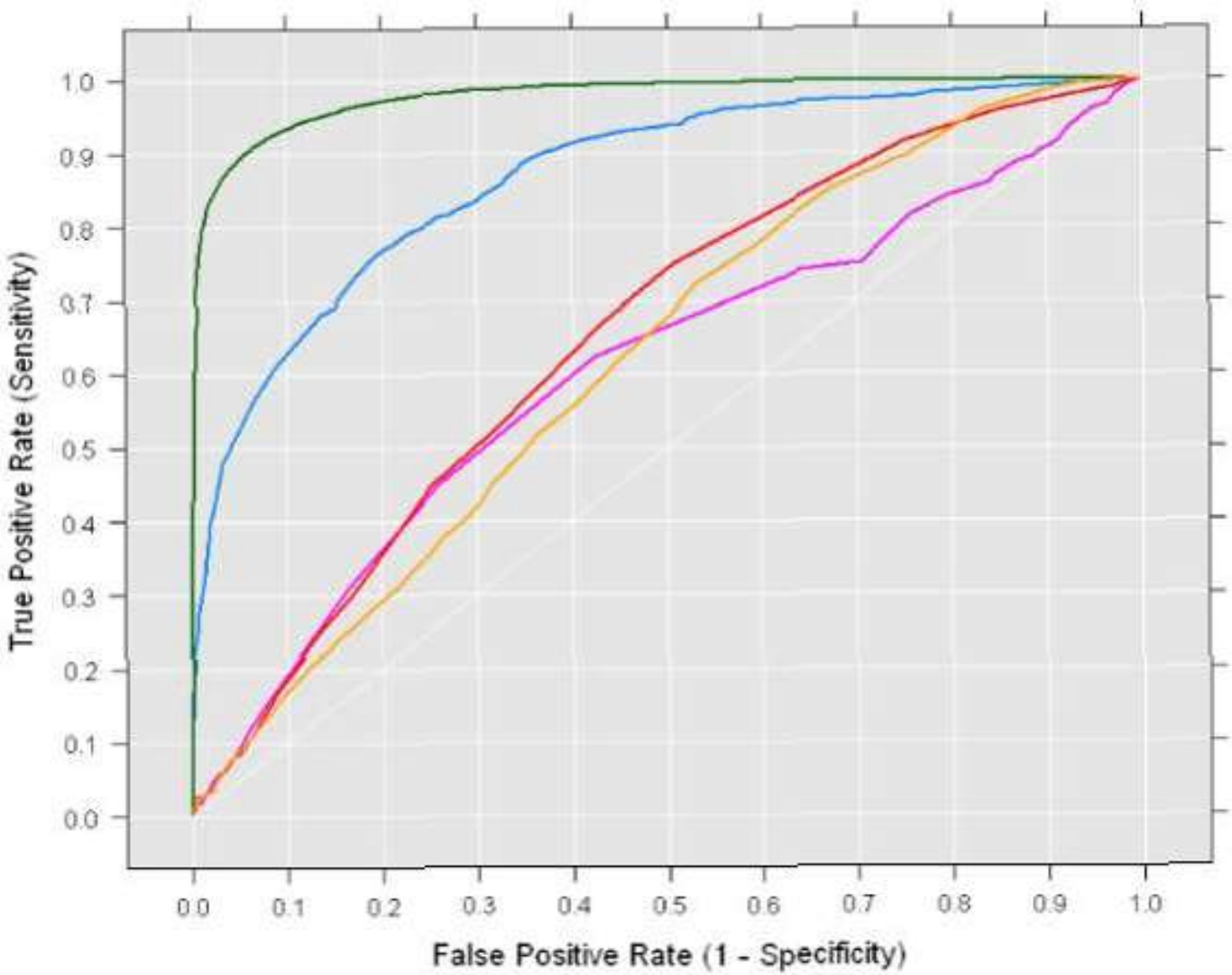
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y-test, y-predicted)
cm
```

- The Confusion matrix can be shown graphically using a Heatmap by passing the Confusion matrix object (cm) to heatmap() method of seaborn package, as:

  sns.heatmap(cm, annot=True)

- In the above statement, annot=True will show the values in the heatmap.
- If this attribute is not mentioned, then the heatmap will be created without any values in it.
- We will see how to draw a heatmap that represents Confusion matrix in the next section.

- To construct an ROC curve, the classification model is typically trained on a dataset with known true labels.
- The model's predictions are then compared to the true labels to calculate the TPR and FPR at various classification thresholds. By varying the threshold, you can adjust the model's trade-off between true positives and false positives.
- The ROC curve is created by plotting the TPR (sensitivity) on the y-axis against the FPR (1 - specificity) on the x-axis.
- Each point on the ROC curve represents a specific classification threshold. The curve itself is formed by connecting these points.
- A perfect classifier would have an ROC curve that passes through the top-left corner of the plot, indicating a high TPR and a low FPR for all threshold values.
- In practice, most classifiers produce curves that lie between the diagonal (representing random guessing) and the top-left corner. The closer the ROC curve is to the top-left corner, the better the model's discriminatory power.


True Positive Rate (Sensitivity)
False Positive Rate (1 - Specificity)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

❖ **Performance of diagnostic tests in ROC**

❖ When evaluating the performance of diagnostic tests using ROC curves, several key measures are assessed.

❖ These measures provide insights into the accuracy and effectiveness of the test in correctly classifying individuals with and without the condition of interest. The main performance measures include:

❖ **Sensitivity:** Sensitivity, also known as the true positive rate, measures the ability of the test to correctly identify individuals who have the condition.

❖ It is calculated as the number of **true positives divided by the sum of true positives and false negatives.**

❖ Sensitivity represents the **proportion of individuals with the condition who are correctly identified by the test.**

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **Specificity:** Specificity, also known as the true negative rate, measures the ability of the test to correctly identify individuals without the condition.
- It is calculated as the number **of true negatives divided by the sum of true negatives and false positives**.
- Specificity represents the **proportion of individuals without the condition who are correctly classified as negative by the test.**

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

- **Positive Predictive Value (PPV):** PPV measures the probability that an individual with a positive test result actually has the condition.
- It is calculated as the number of **true positives divided by the sum of true positives and false positives.** PPV is influenced not only by the test's sensitivity and specificity but also by the prevalence of the condition in the population being tested.

$$\textit{Positive predictive value, PPV} = \frac{\textit{True Positives}}{\textit{True Positives} + \textit{False Positives}}$$

- **Negative Predictive Value (NPV):** NPV measures the probability that an individual with a negative test result is truly free of the condition.
- It is calculated as the number of true negatives divided by the sum of true negatives and false negatives. NPV is influenced by the test's sensitivity and specificity as well as the prevalence of the condition.

$$Negative\ predictive\ value, NPV = \frac{True\ Negatives}{True\ Negatives + False\ Negatives}$$

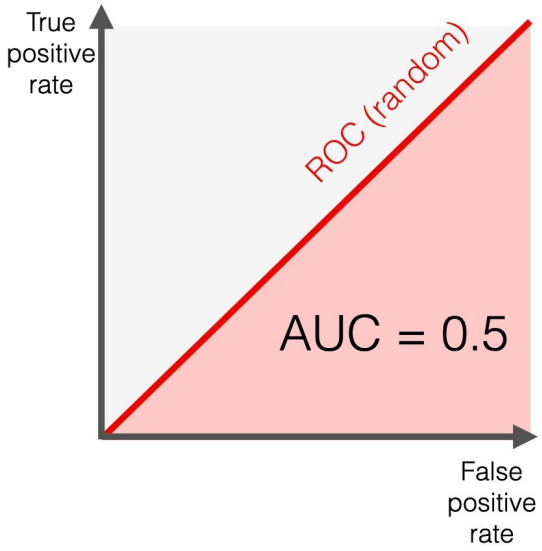
- **Accuracy:** Accuracy represents the overall correctness of the test and is calculated as the sum of true positives and true negatives divided by the total number of individuals tested.

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

- It reflects the proportion of correctly classified individuals among the total population.
- ROC curves provide a graphical representation of the trade-off between sensitivity and specificity for different thresholds of the diagnostic test.
- By examining the curve, you can determine the sensitivity and specificity values at various thresholds, enabling you to assess the overall performance of the test.

- **Area Under the ROC Curve :** The A**rea Under the ROC Curve (AUC)** is a commonly used metric to summarize the discriminatory power of a diagnostic test.
- AUC values range from 0 to 1, with a higher value indicating better discriminative ability.
- An AUC of 0.5 suggests that the test performs no better than random chance, while an AUC of 1 represents a perfect test with no false positives or false negatives.

- When evaluating the performance of diagnostic tests using ROC curves, sensitivity, specificity, PPV, NPV, accuracy, and AUC are important measures that provide insights into the test's ability to correctly classify individuals with and without the condition of interest.
- These measures help inform healthcare decisions and the appropriate utilization of the diagnostic test in clinical practice.



AUC-ROC Curve in Machine Learning - Area under the ROC Curve

- In Machine Learning, only developing an ML model is not sufficient as we also need to see whether it is performing well or not.
- It means that after building an ML model, we need to evaluate and validate how good or bad it is, and for such cases, we use different Evaluation Metrics.
- AUC-ROC curve is such an evaluation metric that is used to visualize the performance of a classification model.
- It is one of the popular and important metrics for evaluating the performance of the classification model.
- **AUC-ROC Curve**
- AUC-ROC curve is a performance measurement metric of a classification model at different threshold values.

- **ROC Curve**
- ROC or Receiver Operating Characteristic curve represents a probability graph to show the performance of a classification model at different threshold levels. The curve is plotted between two parameters, which are:
- **True Positive Rate or TPR**
- **False Positive Rate or FPR**
- In the curve, TPR is plotted on Y-axis, whereas FPR is on the X-axis
- **TPR:**
- TPR or True Positive rate is a synonym for Recall, which can be calculated as:

$$TPR = \frac{TP}{TP + FN}$$

- FPR or False Positive Rate can be calculated as:

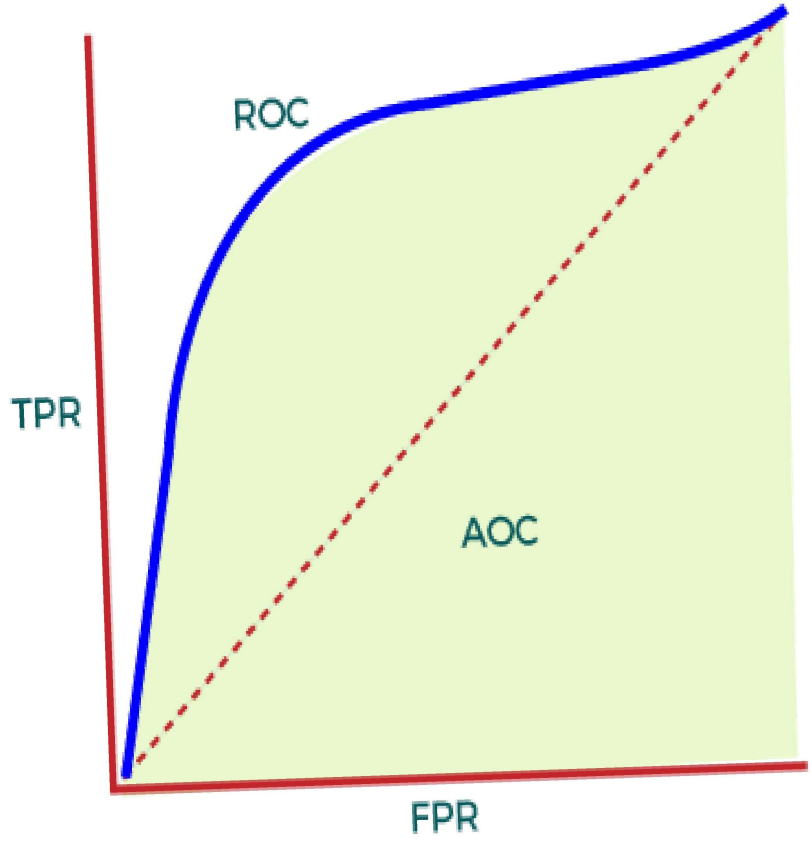
$$TPR = \frac{FP}{FP + TN}$$

- Here,
- TP: True Positive
- FP: False Positive
- TN: True Negative
- FN: False Negative
- Now, to efficiently calculate the values at any threshold level, we need a method, which is AUC.

❖ **AUC: Area Under the ROC curve**

❖ AUC is known for Area Under the ROC curve. As its name suggests, AUC calculates the two-dimensional area under the entire ROC curve ranging from (0,0) to (1,1), as shown below image:

- ❖ In the ROC curve, AUC computes the performance of the binary classifier across different thresholds and provides an aggregate measure.
- ❖ The value of AUC ranges from 0 to 1, which means an excellent model will have AUC near 1, and hence it will show a good measure of Separability.



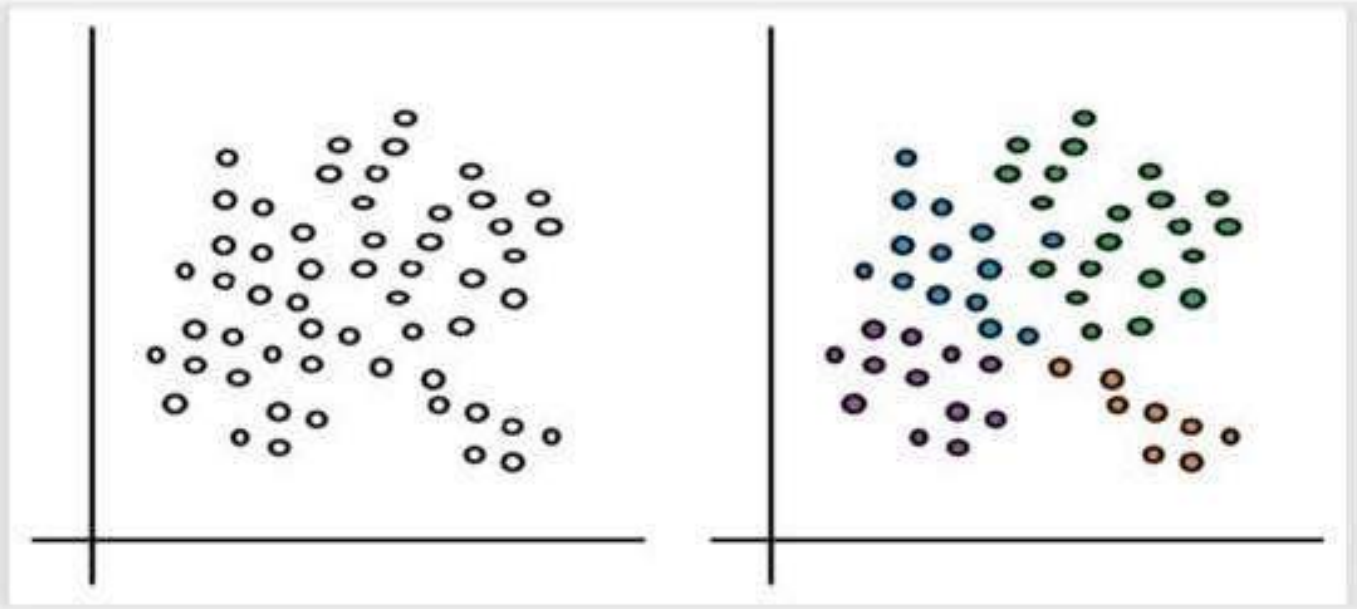
- **When to Use AUC-ROC**
- AUC is preferred due to the following cases:
- AUC is used to measure how well the predictions are ranked instead of giving their absolute values. Hence, we can say AUC is Scale-Invariant.
- It measures the quality of predictions of the model without considering the selected classification threshold. It means AUC is classification-threshold-invariant.

- **When not to use AUC-ROC**
- AUC is not preferable when we need to calibrate probability output.
- Further, AUC is not a useful metric when there are wide disparities in the cost of false negatives vs false positives, and it is difficult to minimize one type of classification error.

- **How AUC-ROC curve can be used for the Multi-class Model?**
- Although the AUC-ROC curve is only used for binary classification problems, we can also use it for multiclass classification problems.
- For multi-class classification problems, we can plot N number of AUC curves for N number of classes with the One vs ALL method.
- For example, if we have three different classes, X, Y, and Z, then we can plot a curve for X against Y & Z, a second plot for Y against X & Z, and the third plot for Z against Y and X.
- **Applications of AUC-ROC Curve**
- Although the AUC-ROC curve is used to evaluate a classification model, it is widely used for various applications. Some of the important applications of AUC-ROC are given below:

- **Classification of 3D model**
- The curve is used to classify a 3D model and separate it from the normal models. With the specified threshold level, the curve classifies the non-3D and separates out the 3D models.
- **Healthcare**
- The curve has various applications in the healthcare sector. It can be used to detect cancer disease in patients. It does this by using false positive and false negative rates, and accuracy depends on the threshold value used for the curve.
- **Binary Classification**
- AUC-ROC curve is mainly used for binary classification problems to evaluate their performance.

- **What is Clustering?**
- Clustering or Cluster analysis is the method of grouping the entities based on similarities. Defined as an unsupervised learning problem that aims to make training data with a given set of inputs but without any target values.
- It is the process of finding similar structures in a set of unlabeled data to make it more understandable and manipulative.
- It reveals subgroups in the available heterogeneous datasets such that every individual cluster has greater homogeneity than the whole.
- In simpler words, these clusters are groups of like objects that differ from the objects in other clusters.
- In clustering, the machine learns the attributes and trends by itself without any provided input-output mapping.
- The clustering algorithms extract patterns and inferences from the type of data objects and then make discrete classes of clustering them suitably.



❖ **Types of Clustering Methods**

❖ Clustering helps in performing surface-level analyses of the unstructured data.

❖ The cluster formation depends upon different parameters like shortest distance, graphs, and density of the data points.

❖ Grouping into clusters is conducted by finding the measure of similarity between the objects based on some metric called the similarity measure. It is easier to find similarity measures in a lesser number of features.

❖ Creating similarity measures becomes a complex process as the number of features increases.

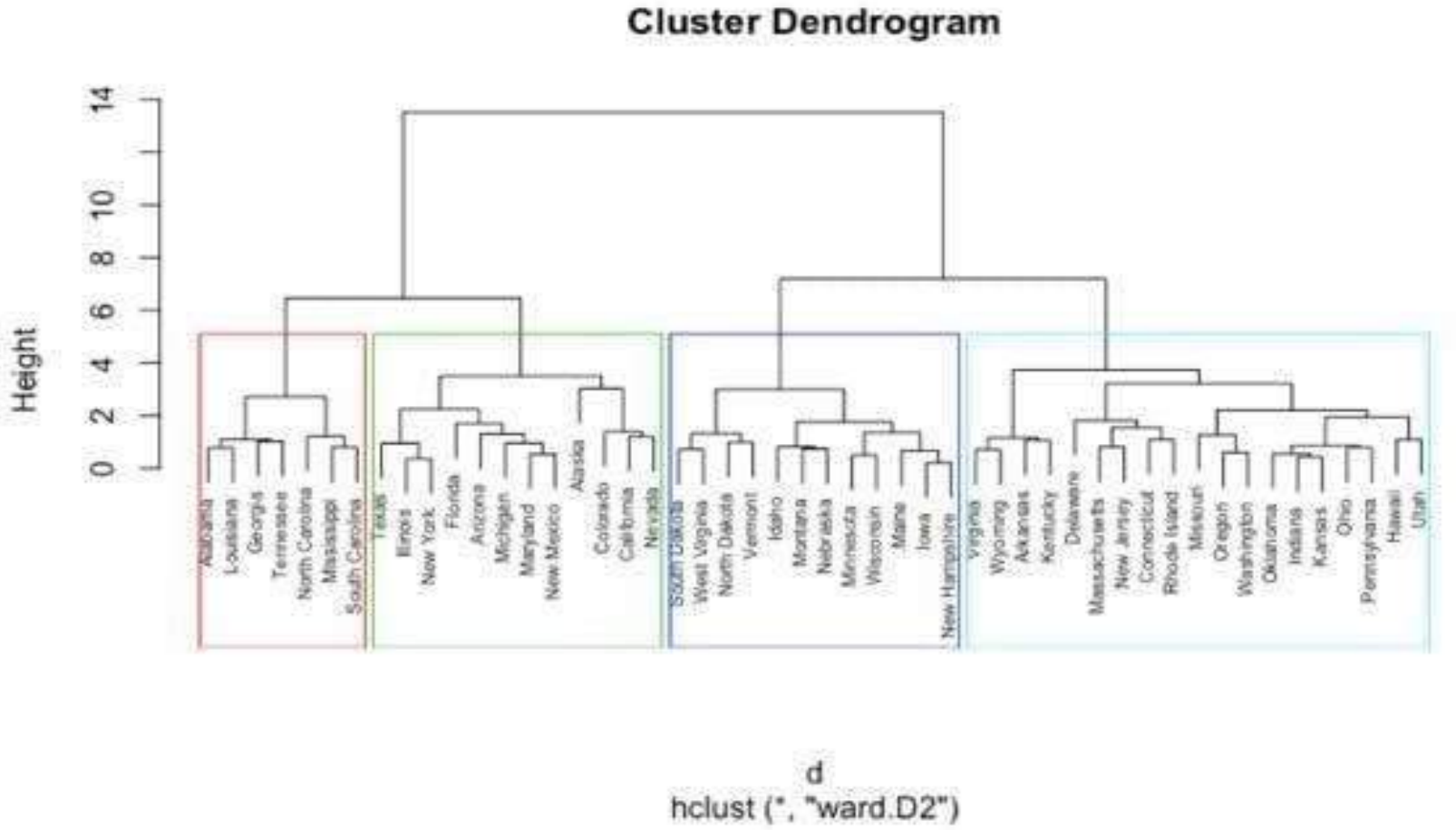
❖ Different types of clustering approaches in data mining use different methods to group the data from the datasets. This section describes the clustering approaches.

❖ The various types of clustering are:

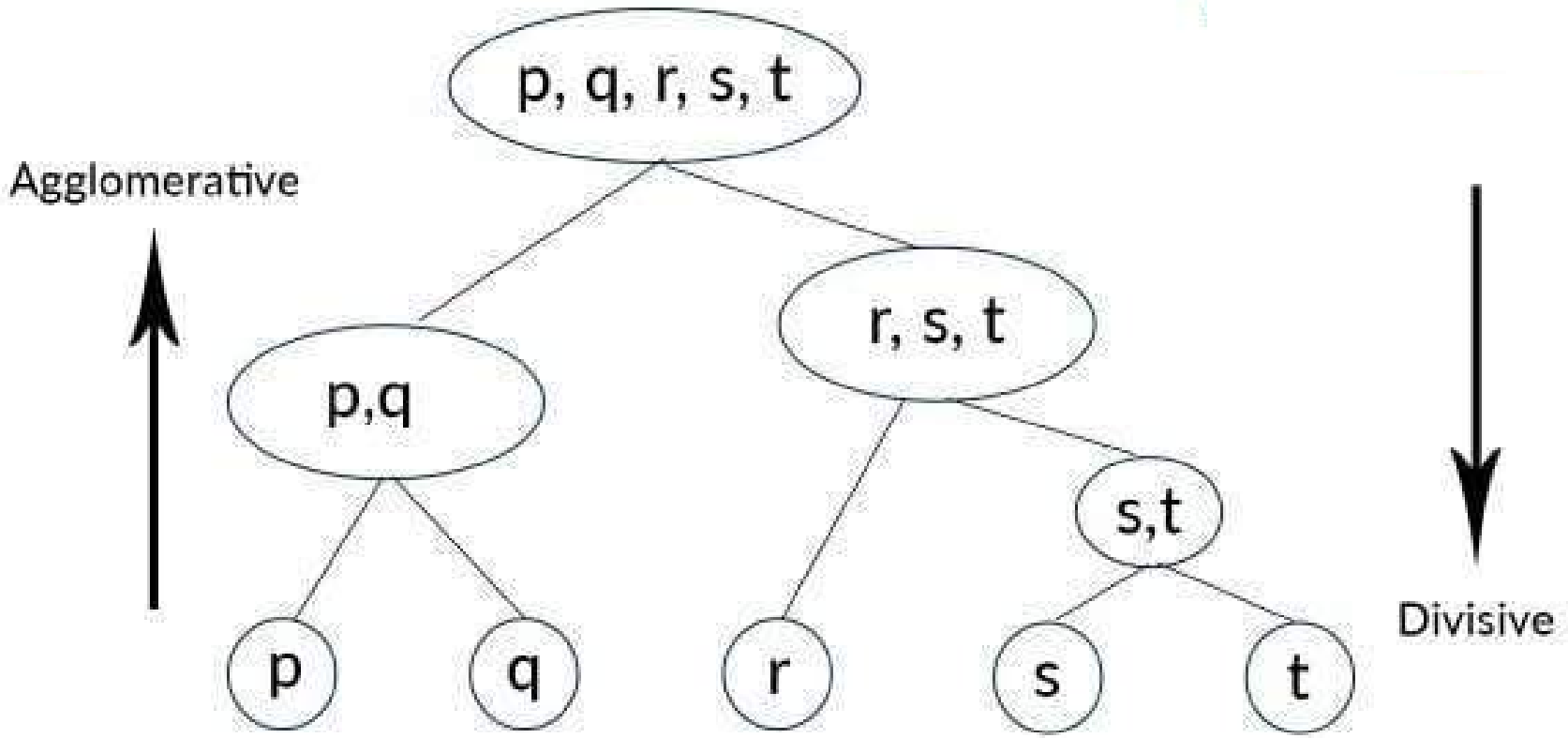
1. Connectivity-based Clustering (Hierarchical clustering)
2. Centroids-based Clustering (Partitioning methods)
3. Distribution-based Clustering
4. Density-based Clustering (Model-based methods)
5. Fuzzy Clustering
6. Constraint-based (Supervised Clustering)

1. Connectivity-based Clustering (Hierarchical Clustering)

- Hierarchical clustering, also known as connectivity-based clustering, is based on the principle that every object is connected to its neighbors depending on their proximity distance (degree of relationship).
- The clusters are represented in extensive hierarchical structures separated by a maximum distance required to connect the cluster parts.
- The clusters are represented as Dendrograms, where X-axis represents the objects that do not merge while Y-axis is the distance at which clusters merge.
- The similar data objects have minimal distance falling in the same cluster, and the dissimilar data objects are placed farther in the hierarchy.
- Mapped data objects correspond to a Cluster amid discrete qualities concerning the multidimensional scaling, quantitative relationships among data variables, or cross-tabulation in some aspects.
- The hierarchical clustering may vary in the data flow chosen in the following categories.


Cluster Dendrogram
Height
0
2
4
6
8
10
14
Alabama
Louisiana
Georgia
Tennessee
North Carolina
Mississippi
South Carolina
Texas
Illinois
New York
Florida
Arizona
Michigan
Maryland
New Mexico
Alaska
Colorado
California
Nevada
South Dakota
West Virginia
North Dakota
Vermont
Idaho
Montana
Nebraska
Minnesota
Wisconsin
Maine
Iowa
New Hampshire
Virginia
Wyoming
Arkansas
Kentucky
Delaware
Massachusetts
New Jersey
Connecticut
Rhode Island
Missouri
Oregon
Washington
Oklahoma
Indiana
Kansas
Ohio
Pennsylvania
Hawaii
Utah
d
hclust (*, "ward.D2")

- **Divisive Approach**
- This approach of hierarchical clustering follows a top-down approach where we consider that all the data points belong to one large cluster and try to divide the data into smaller groups based on a termination logic or a point beyond which there will be no further division of data points.
- This termination logic can be based on the minimum sum of squares of error inside a cluster, or for categorical data, the metric can be the GINI coefficient inside a cluster.
- Hence, iteratively, we are splitting the data, which was once grouped as a single large cluster, into “n” number of smaller clusters to which the data points now belong.
- It must be taken into account that this algorithm is highly “rigid” when splitting the clusters – meaning, once a clustering is done inside a loop, there is no way that the task can be undone.


p, q, r, s, t
Agglomerative
p,q
r, s, t
s,t
p
q
r
s
t
Divisive

❖ **Agglomerative Approach**

❖ Agglomerative is quite the contrary to Divisive, where all the "N" data points are considered to be a single member of "N" clusters that the data is comprised into.

❖ We iteratively combine these numerous "N" clusters to a fewer number of clusters, let's say "k" clusters, and hence assign the data points to each of these clusters accordingly.

❖ This approach is a bottom-up one, and also uses a termination logic in combining the clusters.

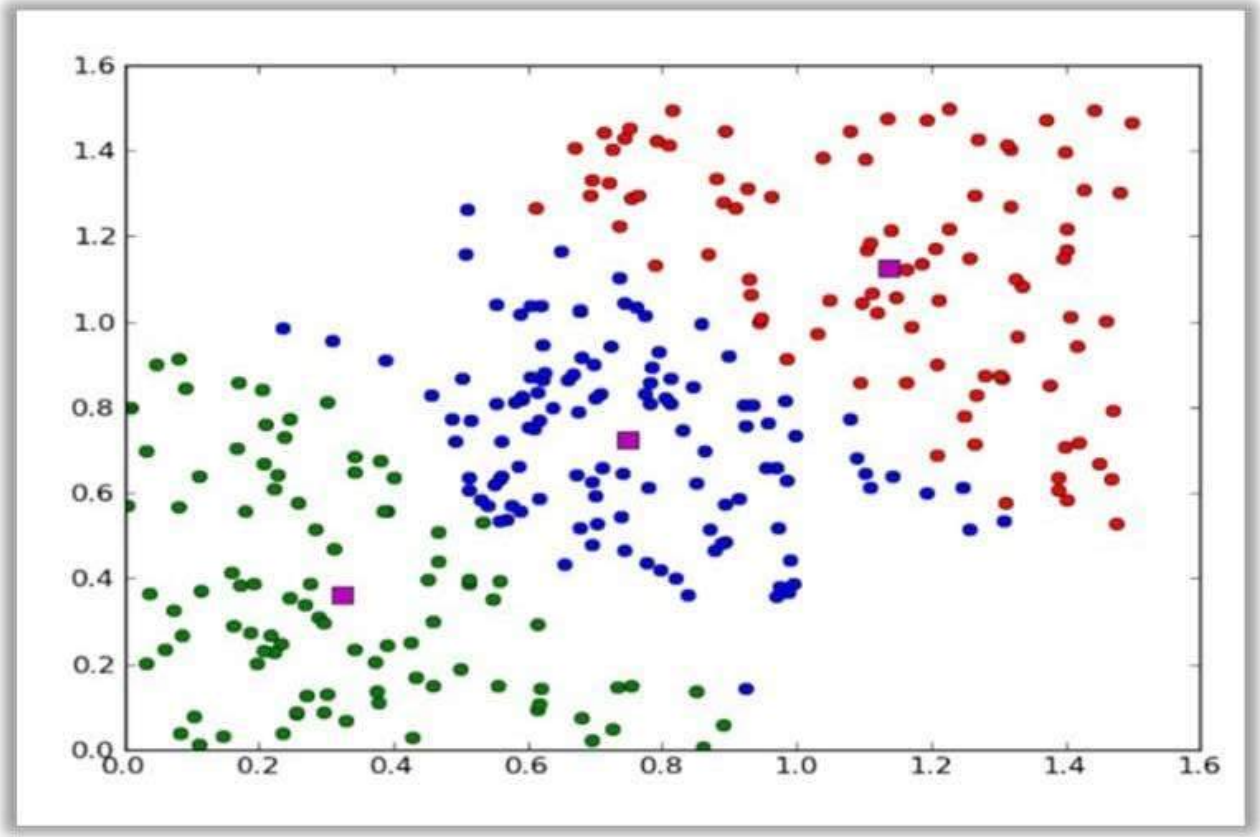
❖ This logic can be a number-based criterion (no more clusters beyond this point) or a distance criterion (clusters should not be too far apart to be merged) or a variance criterion (increase in the variance of the cluster being merged should not exceed a threshold, Ward Method)

2. Centroid-based or Partition Clustering

❖ Centroid-based clustering is the easiest of all the clustering types in data mining. It works on the closeness of the data points to the chosen central value.

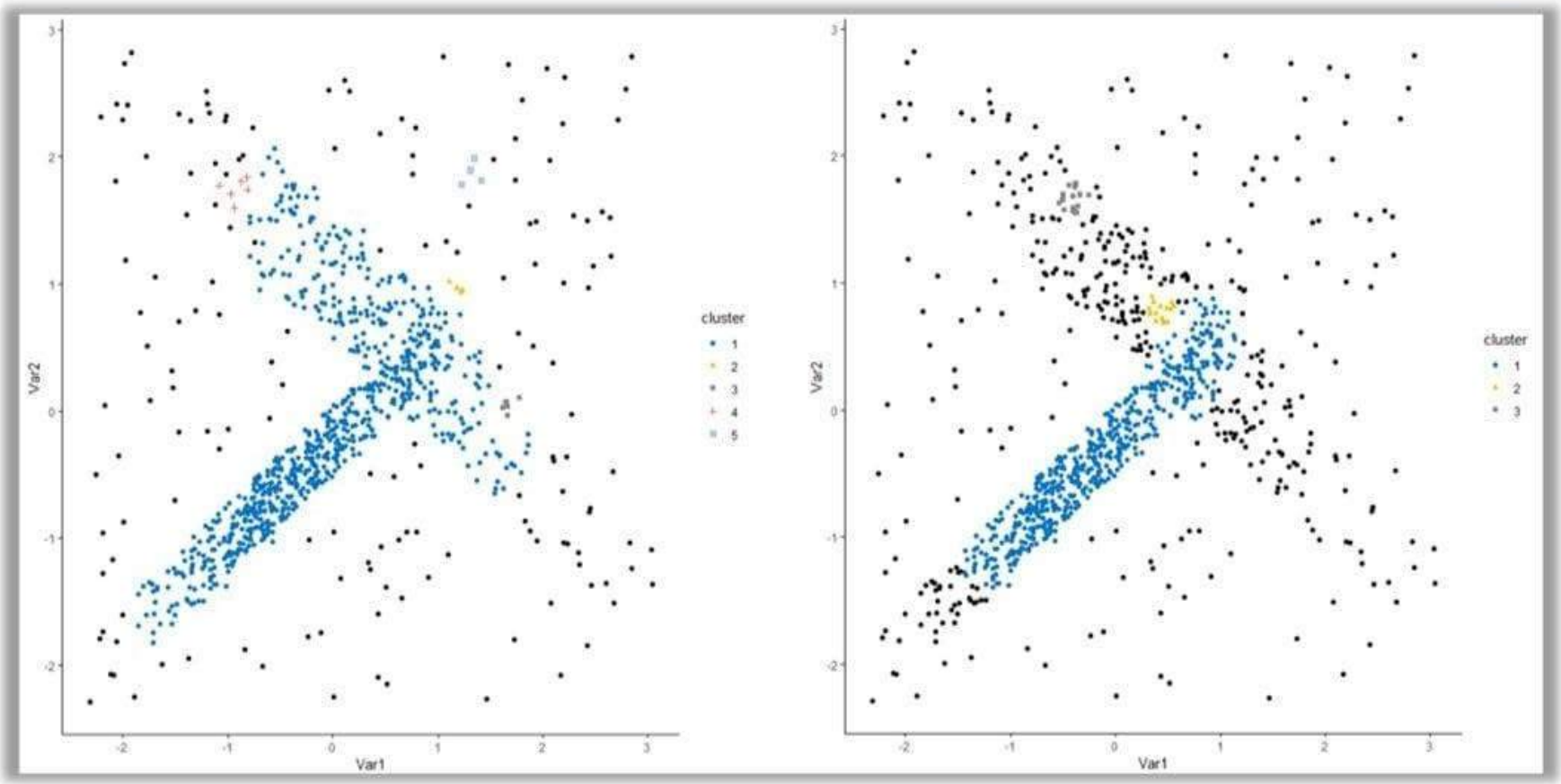
❖ The datasets are divided into a given number of clusters, and a vector of values references every cluster. The input data variable is compared to the vector value and enters the cluster with minimal difference.

- Pre-defining the number of clusters at the initial stage is the most crucial yet most complicated stage for the clustering approach.
- Despite the drawback, it is a vastly used clustering approach for surfacing and optimizing large datasets. The K-Means algorithm lies in this category.
- These groups of clustering methods iteratively measure the distance between the clusters and the characteristic centroids using various distance metrics.
- These are either Euclidian distance, Manhattan Distance or Minkowski Distance.
- The major setback here is that we should either intuitively or scientifically (Elbow Method) define the number of clusters, “k”, to begin the iteration of any clustering machine learning algorithm to start assigning the data points.
- Also, owing to their simplicity in implementation and also interpretation, these algorithms have wide application areas viz., market segmentation, customer segmentation, text topic retrieval, image segmentation, etc.


1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6

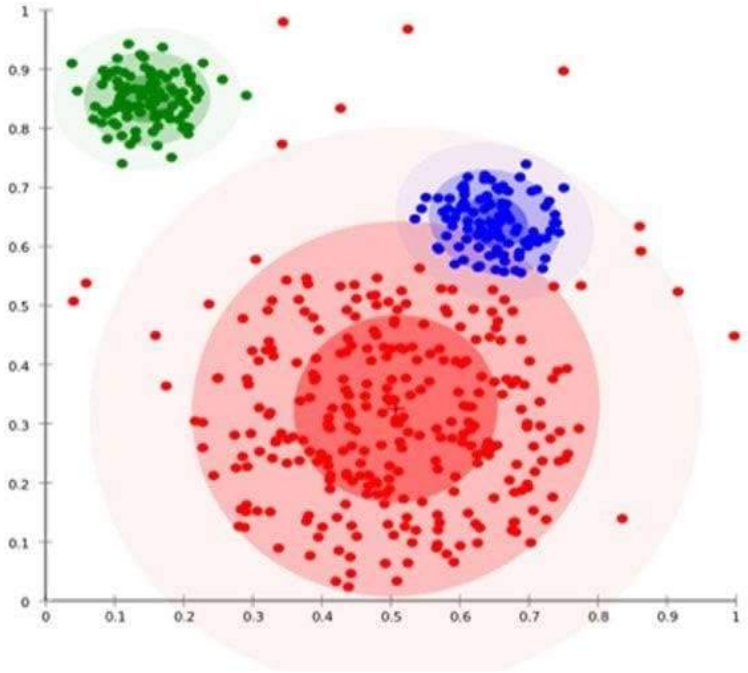
3. Density-based Clustering (Model-based Methods)

- The first two methods discussed above depend on a distance (similarity/proximity) metric. The very definition of a cluster is based on this metric.
- Density-based clustering method considers density ahead of distance. Data is clustered by regions of high concentrations of data objects bounded by areas of low concentrations of data objects. The clusters formed are grouped as a maximal set of connected data points.
- The clusters formed vary in arbitrary shapes and sizes and contain a maximum degree of homogeneity due to similar density. This clustering approach includes the noise and outliers in the datasets effectively.
- When performing most of the clustering, we take two major assumptions: the data is devoid of any noise and the shape of the cluster so formed is purely geometrical (circular or elliptical).
- The fact is, data always has some extent of inconsistency (noise) which cannot be ignored. Added to that, we must not limit ourselves to a fixed attribute shape.
- It is desirable to have arbitrary shapes to not to ignore any data points. These are the areas where density-based algorithms have proven their worth.


Var2
Var1
cluster
1
2
3
4
5
Var2
Var1
cluster
1
2
3

4. Distribution-Based Clustering

- Until now, the clustering techniques as we know them are based on either proximity (similarity/distance) or composition (density). There is a family of clustering algorithms that take a totally different metric into consideration – probability.
- Distribution-based clustering creates and groups data points based on their likely hood of belonging to the same probability distribution (Gaussian, Binomial, etc.) in the data



- A major drawback of density and boundary-based approaches is in specifying the clusters apriori to some of the algorithms and mostly the definition of the shape of the clusters for most of the algorithms.
- There is at least one tuning or hyper-parameter which needs to be selected and not only that is trivial but also any inconsistency in that would lead to unwanted results.
- Distribution-based clustering has a vivid advantage over the proximity and centroid-based clustering methods in terms of flexibility, correctness, and shape of the clusters formed.
- The major problem however is that these clustering methods work well only with synthetic or simulated data or with data where most of the data points most certainly belong to a predefined distribution, if not, the results will overfit.

5. Fuzzy Clustering

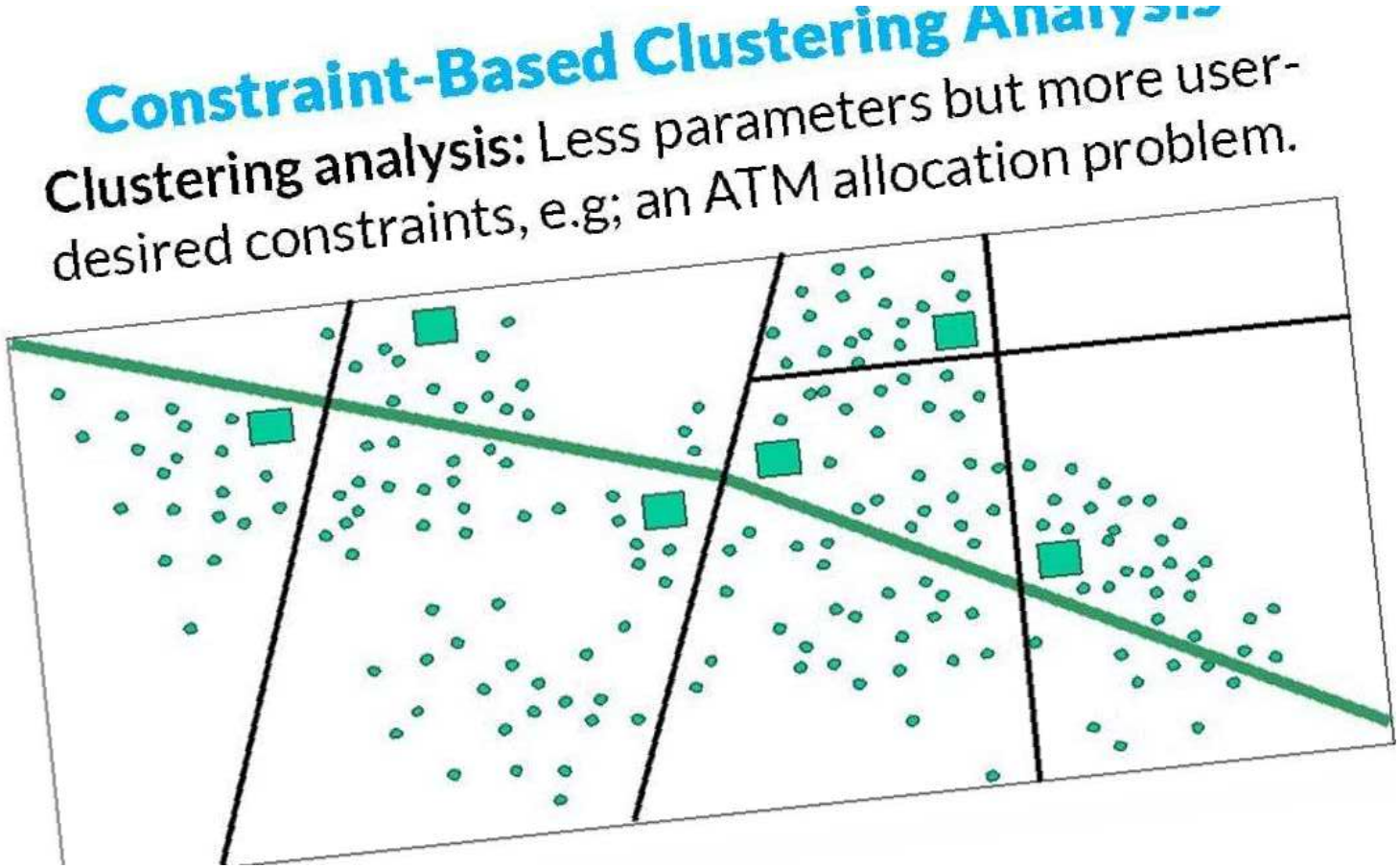
- Fuzzy clustering generalizes the partition-based clustering method by allowing a data object to be a part of more than one cluster.
- The process uses a weighted centroid based on the spatial probabilities.

- The algorithm works by assigning membership values to all the data points linked to each cluster center. It is computed from a distance between the cluster center and the data point. If the membership value of the object is closer to the cluster center, it has a high probability of being in the specific cluster.
- At the end iteration, associated values of membership and cluster centers are reorganized. Fuzzy clustering handles the situations where data points are somewhat in between the cluster centers or ambiguous. This is done by choosing the probability rather than distance.

6. Constraint-based (Supervised Clustering)

- The clustering process, in general, is based on the approach that the data can be divided into an optimal number of "unknown" groups.
- The underlying stages of all the clustering algorithms are to find those hidden patterns and similarities without intervention or predefined conditions.
- However, in certain business scenarios, we might be required to partition the data based on certain constraints.
- Here is where a supervised version of clustering machine learning techniques comes into play.

❖ A constraint is defined as the desired properties of the clustering results or a user's expectation of the clusters so formed – this can be in terms of a fixed number of clusters, the cluster size, or important dimensions (variables) that are required for the clustering process.

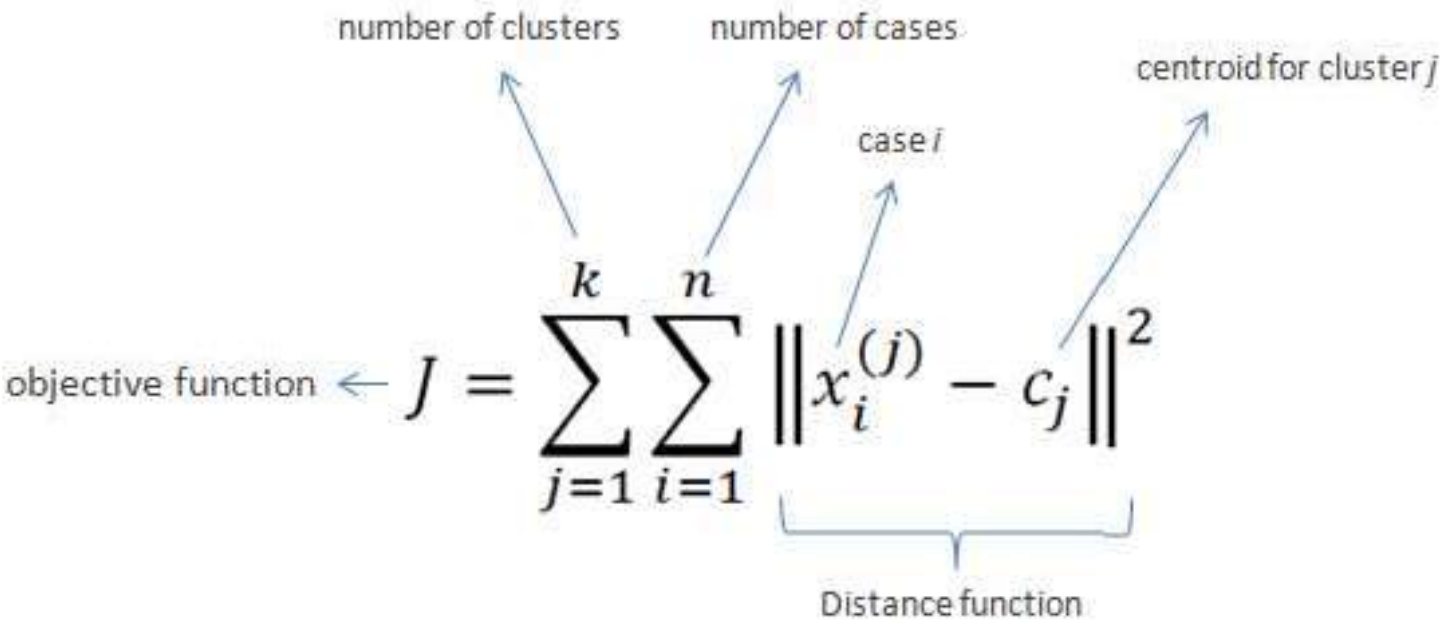


Types of Clustering Algorithms

- Clustering algorithms are used in exploring data, anomaly detection, finding outliers, or detecting patterns in the data.
- Clustering is an unsupervised learning technique like neural network and reinforcement learning. The available data is highly unstructured, heterogeneous, and contains noise. So the choice of algorithm depends upon how the data looks like.
- A suitable clustering algorithm helps in finding valuable insights for the industry. Let's explore the different types of clustering in machine learning in detail.
  - K-Means clustering
  - Mean Shift
  - Gaussian Mixture Model

- **K-Means clustering**
- K-Means is a partition-based clustering technique that uses the distance between the Euclidean distances between the points as a criterion for cluster formation.
- Assuming there are ‘n’ numbers of data objects, K-Means groups them into a predetermined ‘k’ number of clusters.
- Each cluster has a cluster center allocated and each of them is placed at farther distances.
- Every incoming data point gets placed in the cluster with the closest cluster center. This process is repeated until all the data points get assigned to any cluster. Once all the data points are covered the cluster centers or centroids are recalculated.
- After having these ‘k’ new centroids, a new grouping is done between the nearest new centroid and the same data set points.
- Iteratively, there may be a change in the k centroid values and their location this loop continues until the cluster centers do not change or in other words, centroids do not move anymore.
- The algorithm aims to minimize the objective function

- The correct value of K can be chosen using the Silhouette method and Elbow method.
- The Silhouette method calculates the distance using the mean intra-cluster distance along with an average of the closest cluster distance for each data point.
- While the Elbow method uses the sum of squared data points and computes the average distance.

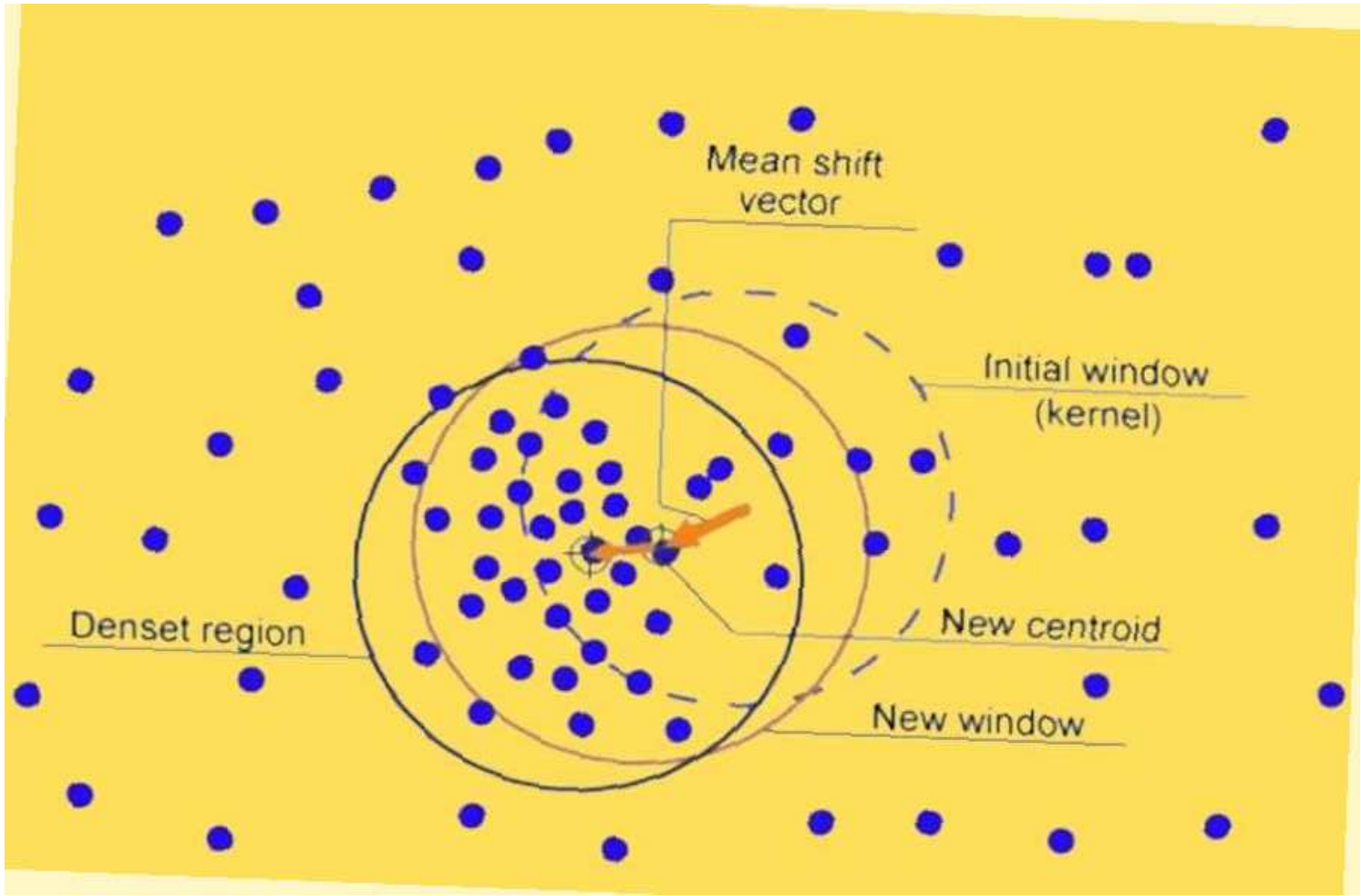


Implementation: K-Means clustering algorithm

- Select ‘k’ number of clusters and centroids for each cluster.
- Shuffle the data points in the dataset and initialize the selected centroid.
- Assign the clusters to the data points without replacement.
- Create new centroids by calculating the mean value of the samples.
- Reinitialize the cluster centers until there is no change in the clusters.

Mean Shift Clustering algorithm

- Mean shift clustering is a nonparametric, simple, and flexible clustering technique.
- It is based upon a method to estimate the essential distribution for a given dataset known as kernel density estimation.
- The basic principle of the algorithm is to assign the data points to the specified clusters recursively by shifting points towards the peak or highest density of data points.
- It is used in the image segmentation process.


Mean shift
vector
Initial window
(kernel)
Denset region
New centroid
New window

Algorithm:

Step 1 – Creating a cluster for every data point

Step 2 – Computation of the centroids

Step 3 – Update the location of the new centroids

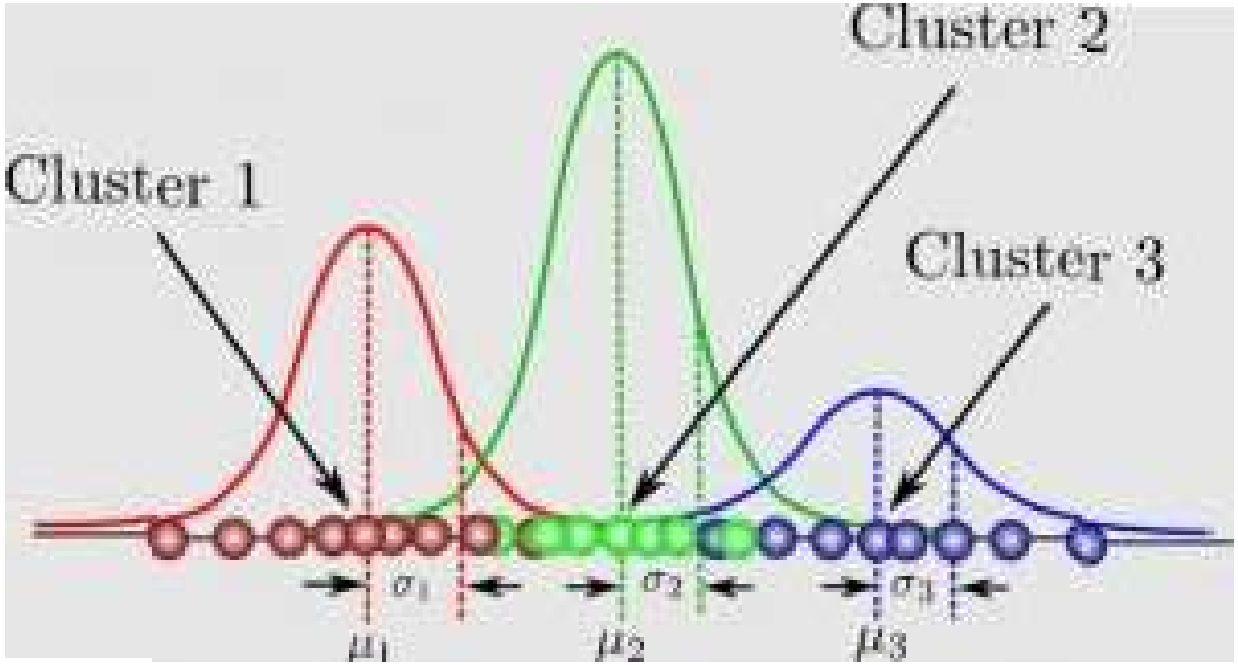
Step 4 – Moving the data points to higher density regions, iteratively.

Step 5 – Terminates when the centroids reach a position where they don't move further.

Gaussian Mixture Model

- The gaussian mixture model (GMM) is a distribution-based clustering technique. It is based on the assumption that the data comprises Gaussian distributions. It is a statistical inference clustering technique.
- The probability of a point being a part of a cluster is inversely dependent on distance, as the distance from distribution increases, the probability of a point belonging to the cluster decreases.
- The GM model trains the dataset and assumes a cluster for every object in the dataset. Later, a scatter plot is created with data points with different colors assigned to each cluster.

- GMM determines probabilities and allocates them to data points in the ‘K’ number of clusters.
- Each of which has three parameters: Mean, Covariance and mixing probability. To compute these parameters GMM uses the Expectation Maximization technique.



What are the Applications of Clustering?

- **Image Recognition and Object Detection:** Clustering is used in computer vision tasks to group similar features in images, enabling object recognition, image segmentation, and object detection.
- **Image Segmentation and Computer Vision Applications:** Clustering is used to segment images into meaningful regions or objects, aiding in image editing, object recognition, and computer vision tasks.
- **Image Compression and Video Analysis:** Clustering techniques are applied in image and video processing to reduce the size of images or video frames, improving storage efficiency and transmission speed.
- **Genetic Analysis and Genomic Clustering:** Clustering algorithms aid in analyzing genetic data, identifying genetic variations, gene expression patterns, and grouping patients based on their genetic profiles.

- **Environmental Data Analysis and Pattern Recognition:** Clustering techniques are applied to analyze environmental data, identify patterns, and aid in climate modeling, pollution monitoring, and ecological studies.

- **Speech and Audio Clustering:** Clustering techniques are utilized to group similar speech or audio segments, aiding in speech recognition, speaker identification, and audio content organization.

- **Natural Language Processing (NLP) and Text Mining:** Clustering is used to group similar textual data, aiding in text classification, sentiment analysis, document summarization, and information retrieval.

- **Document Clustering and Topic Modeling:** Clustering is applied to categorize and organize documents based on their textual content, aiding in information retrieval, topic extraction, and document recommendation.

- **Social Network Analysis and Community Detection:** Clustering aids in identifying communities or groups within social networks, enabling the analysis of social structures, influential users, and targeted advertising.
- **Social Media Sentiment Analysis:** Clustering is utilized to analyze social media data, group similar posts or users, and perform sentiment analysis to understand public opinion and trends.
- **Portfolio Optimization and Financial Asset Allocation:** Clustering aids in grouping similar financial assets, facilitating portfolio optimization, risk assessment, and asset allocation strategies.
- **Customer Segmentation and Personalization:** Clustering helps identify distinct groups of customers based on their preferences, behavior, or demographics, enabling personalized marketing and recommendations.
- **Market Segmentation and Targeted Marketing:** Clustering is utilized to divide a market into distinct segments based on customer preferences, behavior, or demographics, allowing businesses to optimize marketing strategies for different segments.
- **Anomaly Detection and Fraud Detection:** Clustering aids in identifying abnormal patterns or outliers in data, allowing for the detection of fraudulent activities, network intrusions, or unusual behaviors.

- **Recommendation Systems:** Clustering helps group users or items with similar characteristics, facilitating personalized recommendations and improving the accuracy of recommendation systems.
- **Data Compression and Dimensionality Reduction:** Clustering algorithms can be used for data compression and dimensionality reduction, reducing storage requirements and processing time while retaining important patterns in the data.
- **Network Traffic Analysis and Intrusion Detection:** Clustering helps analyze network traffic patterns, identify anomalies, and detect potential network intrusions or security breaches.

Explain different machine learning algorithms with applications.

Machine Learning Algorithms

- Machine Learning algorithms are the programs that can learn the hidden patterns from the data, predict the output, and improve the performance from experiences on their own. Different algorithms can be used in machine learning for different tasks, such as simple linear regression that can be used for prediction problems like stock market prediction, and the KNN algorithm can be used for classification problems.
- In this topic, we will see the overview of some popular and most commonly used machine learning algorithms along with their use cases and categories.
- Types of Machine Learning Algorithms
- Machine Learning Algorithm can be broadly classified into three types:
    1. Supervised Learning Algorithms
    2. Unsupervised Learning Algorithms
    3. Reinforcement Learning algorithm
- The below diagram illustrates the different ML algorithm, along with the categories:

Supervised Learning Algorithm

- Supervised learning is a type of Machine learning in which the machine needs external supervision to learn. The supervised learning models are trained using the labeled dataset.
- Once the training and processing are done, the model is tested by providing a sample test data to check whether it predicts the correct output.
- The goal of supervised learning is to map input data with the output data.
- Supervised learning is based on supervision, and it is the same as when a student learns things in the teacher's supervision. The example of supervised learning is spam filtering.
- Supervised learning can be divided further into two categories of problem
  - Classification
  - Regression
- Examples of some popular supervised learning algorithms are Simple Linear regression, Decision Tree, Logistic Regression, KNN algorithm

Unsupervised Learning Algorithm

- It is a type of machine learning in which the machine does not need any external supervision to learn from the data, hence called unsupervised learning.
- The unsupervised models can be trained using the unlabelled dataset that is not classified, nor categorized, and the algorithm needs to act on that data without any supervision.
- In unsupervised learning, the model doesn't have a predefined output, and it tries to find useful insights from the huge amount of data.
- These are used to solve the Association and Clustering problems. Hence further, it can be classified into two types:
  - Clustering
  - Association
- Examples of some Unsupervised learning algorithms are K-means Clustering, Apriori Algorithm, Eclat, etc

Reinforcement Learning

- In Reinforcement learning, an agent interacts with its environment by producing actions, and learn with the help of feedback.
- The feedback is given to the agent in the form of rewards, such as for each good action, he gets a positive reward, and for each bad action, he gets a negative reward.
- There is no supervision provided to the agent. Q-Learning algorithm is used in reinforcement learning

Best Machine Learning Applications and Examples

- With an understanding of the common machine learning uses, let's explore some examples of the popular applications in the market that rely heavily on machine learning.

1. Social Media (Facebook)

- Automatic friend tagging suggestions on Facebook are one of the best machine-learning applications. Facebook automatically locates a face that matches its database using face detection and image recognition and then advises us to tag that individual using DeepFace (a project of Facebook's Deep Learning division).

2. Transportation (Uber)

- Uber is a customized cab application that relies on machine learning to automatically locate a rider, and offer options to travel home, to work, or to any other regular location based on the rider's history and patterns.
- Moreover, the app further uses ML algorithms to make precision predictions around the Estimated Time of Arrival (ETA) to a particular destination by analyzing traffic conditions.

3. Language Translation (Google Translate)

- To break all language barriers and make traveling to foreign countries easy, Google Translate employs Google Neural Machine Translation (GNMT) which relies on Natural Language Processing(NLP) to translate words across thousands of languages and dictionaries.
- It also makes use of POS Tagging, Named Entity Recognition (NER), and Chunking to maintain the words' tonality.

4. Image Recognition:

- Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc.
- The popular use case of image recognition and face detection is, Automatic friend tagging suggestion:
- Facebook provides us a feature of auto friend tagging suggestion.
- Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's face detection and recognition algorithm.
- It is based on the Facebook project named "Deep Face," which is responsible for face recognition and person identification in the picture

5. Speech Recognition

- While using Google, we get an option of "Search by voice," it comes under speech recognition, and it's a popular application of machine learning.
- Speech recognition is a process of converting voice instructions into text, and it is also known as "Speech to text", or "Computer speech recognition."
- At present, machine learning algorithms are widely used by various applications of speech recognition. Google assistant, Siri, Cortana, and Alexa are using speech recognition technology to follow the voice instructions

6. Traffic prediction

- If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.
- It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways
  - Real Time location of the vehicle form Google Map app and sensors
  - Average time has taken on past days at the same time.
- Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends back to its database to improve the performance.

CART (Classification And Regression Tree) in Machine Learning

- CART(Classification And Regression Trees) is a variation of the decision tree algorithm. It can handle both classification and regression tasks. Scikit-Learn uses the Classification And Regression Tree (CART) algorithm to train Decision Trees (also called "growing" trees). CART was first produced by Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone in 1984.

CART(Classification And Regression Tree) for Decision Tree

- CART is a predictive algorithm used in Machine learning and it explains how the target variable's values can be predicted based on other matters. It is a decision tree where each fork is split into a predictor variable and each node has a prediction for the target variable at the end.
- The term CART serves as a generic term for the following categories of decision trees:
- Classification Trees: The tree is used to determine which "class" the target variable is most likely to fall into when it is continuous.
- Regression trees: These are used to predict a continuous variable's value

- In the decision tree, nodes are split into sub-nodes based on a threshold value of an attribute. The root node is taken as the training set and is split into two by considering the best attribute and threshold value. Further, the subsets are also split using the same logic.
- This continues till the last pure sub-set is found in the tree or the maximum number of leaves possible in that growing tree.

CART Algorithm

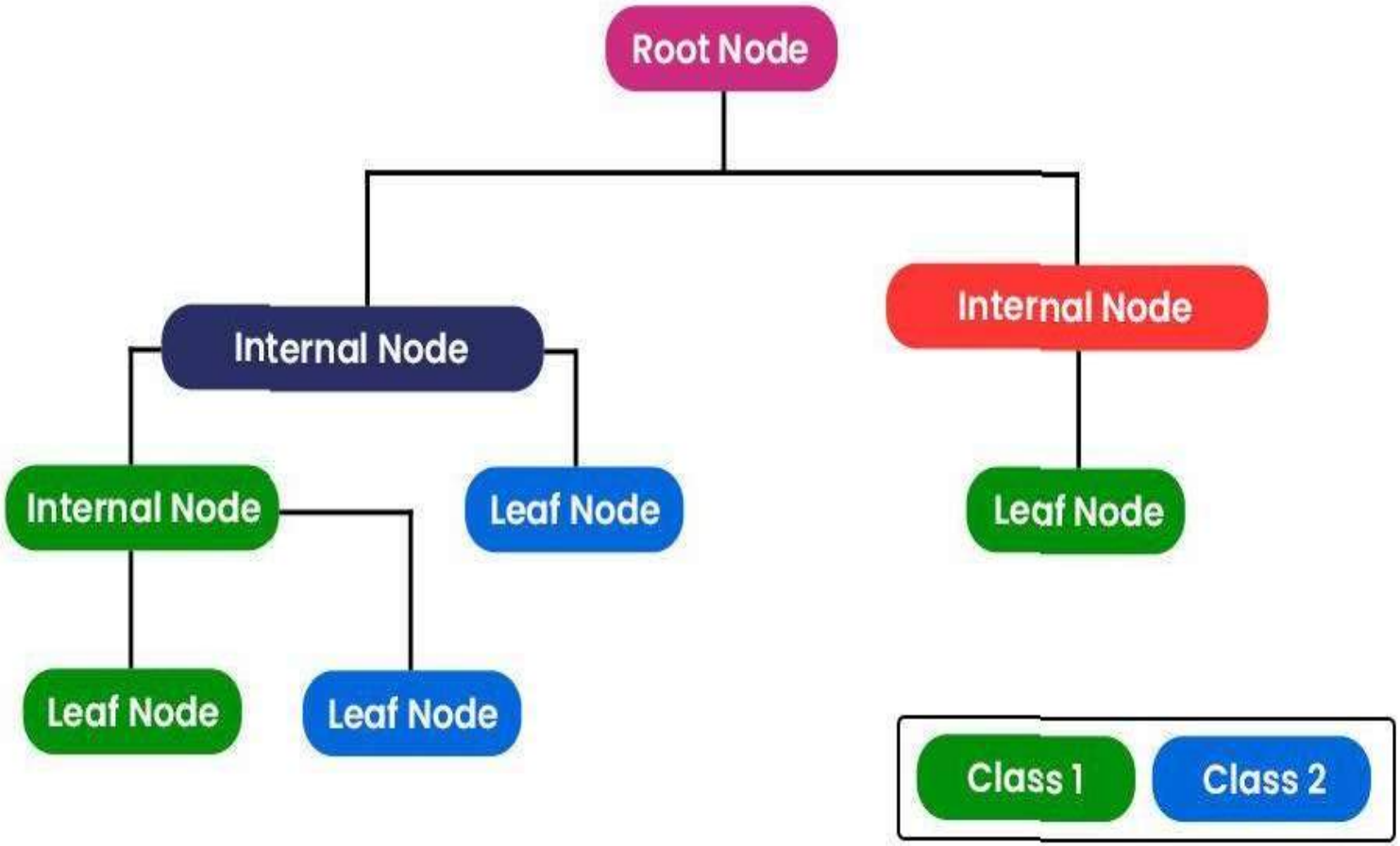
- Classification and Regression Trees (CART) is a decision tree algorithm that is used for both classification and regression tasks. It is a supervised learning algorithm that learns from labelled data to predict unseen data.
- **Tree structure:** CART builds a tree-like structure consisting of nodes and branches. The nodes represent different decision points, and the branches represent the possible outcomes of those decisions. The leaf nodes in the tree contain a predicted class label or value for the target variable.
- **Splitting criteria:**
- CART uses a greedy approach to split the data at each node. It evaluates all possible splits and selects the one that best reduces the impurity of the resulting subsets. For classification tasks, CART uses Gini impurity as the splitting criterion.

- The lower the Gini impurity, the more pure the subset is. For regression tasks, CART uses residual reduction as the splitting criterion. The lower the residual reduction, the better the fit of the model to the data.
- **Pruning:**
- To prevent overfitting of the data, pruning is a technique used to remove the nodes that contribute little to the model accuracy.
- Cost complexity pruning and information gain pruning are two popular pruning techniques.
- Cost complexity pruning involves calculating the cost of each node and removing nodes that have a negative cost. Information gain pruning involves calculating the information gain of each node and removing nodes that have a low information gain.

How does CART algorithm works?

The CART algorithm works via the following process:

- The best-split point of each input is obtained.
- Based on the best-split points of each input in Step 1, the new "best" split point is identified.
- Split the chosen input according to the "best" split point.
- Continue splitting until a stopping rule is satisfied or no further desirable splitting is available.


Root Node
Internal Node
Internal Node
Internal Node
Leaf Node
Leaf Node
Leaf Node
Leaf Node
Class 1
Class 2

- CART algorithm uses Gini Impurity to split the dataset into a decision tree .It does that by searching for the best homogeneity for the sub nodes, with the help of the Gini index criterion.

Gini index/Gini impurity

- The Gini index is a metric for the classification tasks in CART. It stores the sum of squared probabilities of each class. It computes the degree of probability of a specific variable that is wrongly being classified when chosen randomly and a variation of the Gini coefficient. It works on categorical variables, provides outcomes either "successful" or "failure" and hence conducts binary splitting only.
- The degree of the Gini index varies from 0 to 1,
- Where 0 depicts that all the elements are allied to a certain class, or only one class exists there.
- The Gini index of value 1 signifies that all the elements are randomly distributed across various classes, and
- A value of 0.5 denotes the elements are uniformly distributed into some classes.

❖ Mathematically, we can write Gini Impurity as follows:

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2$$

❖ where p_i is the probability of an object being classified to a particular class.

CART for Classification

❖ A classification tree is an algorithm where the target variable is categorical. The algorithm is then used to identify the "Class" within which the target variable is most likely to fall. Classification trees are used when the dataset needs to be split into classes that belong to the response variable(like yes or no)

❖ For classification in decision tree learning algorithm that creates a tree-like structure to predict class labels. The tree consists of nodes, which represent different decision points, and branches, which represent the possible result of those decisions. Predicted class labels are present at each leaf node of the tree.

How Does CART for Classification Work?

- CART for classification works by recursively splitting the training data into smaller and smaller subsets based on certain criteria. The goal is to split the data in a way that minimizes the impurity within each subset. Impurity is a measure of how mixed up the data is in a particular subset. For classification tasks, CART uses Gini impurity
- Gini Impurity- Gini impurity measures the probability of misclassifying a random instance from a subset labeled according to the majority class. Lower Gini impurity means more purity of the subset.
- Splitting Criteria- The CART algorithm evaluates all potential splits at every node and chooses the one that best decreases the Gini impurity of the resultant subsets. This process continues until a stopping criterion is reached, like a maximum tree depth or a minimum number of instances in a leaf node.

CART for Regression

- A Regression tree is an algorithm where the target variable is continuous and the tree is used to predict its value. Regression trees are used when the response variable is continuous. For example, if the response variable is the temperature of the day.
- CART for regression is a decision tree learning method that creates a tree-like structure to predict continuous target variables.
- The tree consists of nodes that represent different decision points and branches that represent the possible outcomes of those decisions. Predicted values for the target variable are stored in each leaf node of the tree.

How Does CART works for Regression?

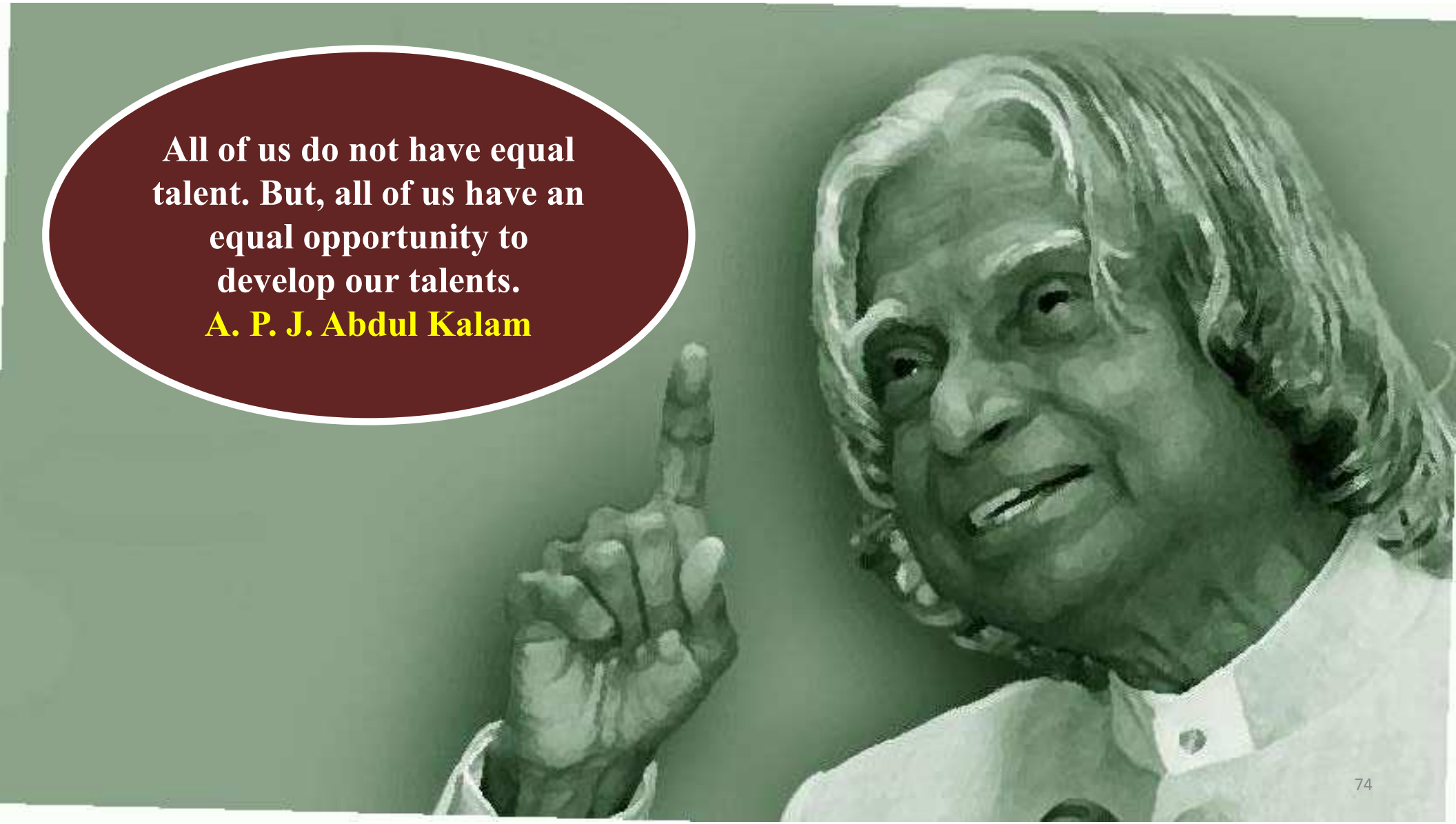
- Regression CART works by splitting the training data recursively into smaller subsets based on specific criteria. The objective is to split the data in a way that minimizes the residual reduction in each subset.
- Residual Reduction- Residual reduction is a measure of how much the average squared difference between the predicted values and the actual values for the target variable is reduced by splitting the subset. The lower the residual reduction, the better the model fits the data.
- Splitting Criteria- CART evaluates every possible split at each node and selects the one that results in the greatest reduction of residual error in the resulting subsets. This process is repeated until a stopping criterion is met, such as reaching the maximum tree depth or having too few instances in a leaf node.

Advantages of CART

- ❖ Results are simplistic.
- ❖ Classification and regression trees are Nonparametric and Nonlinear.
- ❖ Classification and regression trees implicitly perform feature selection.
- ❖ Outliers have no meaningful effect on CART.
- ❖ It requires minimal supervision and produces easy-to-understand models.

Limitations of CART

- ❖ Overfitting.
- ❖ High Variance.
- ❖ low bias.
- ❖ the tree structure may be unstable.


All of us do not have equal talent. But, all of us have an equal opportunity to develop our talents.
A. P. J. Abdul Kalam


Thank
You!