Data Science & AI

Machine Learning

Supervised Learning

Lecture No.- 06

By- Krish Naik Sir

Recap of Previous Lecture

Topic

Topic

Topic

Topic

Topic

Feature Transformation
naive baye's

Topics to be Covered

- **Topic** Random forest classifier
- **Topic** KNN classifier & regressor
- **Topic** K-means
- **Topic**
- **Topic**

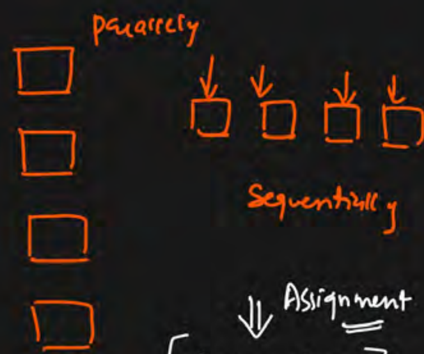
Machine Learning

1. Random Forest Classifier And Regressor [Bagging]
2. KNN Classifier And Regressor [KNN].
3. K Means ✓
4. Hierarichal Mean ✓
5. DBScan ✓.
6. Silhoutte Scoring. ✓

SVM ← 45-1hr.

↑

Tomorrow.

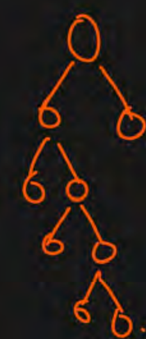


⇓ Assignment

[ROC - AUC]

Curve

1. Random Forest Classifier And Regressor [Bagging]

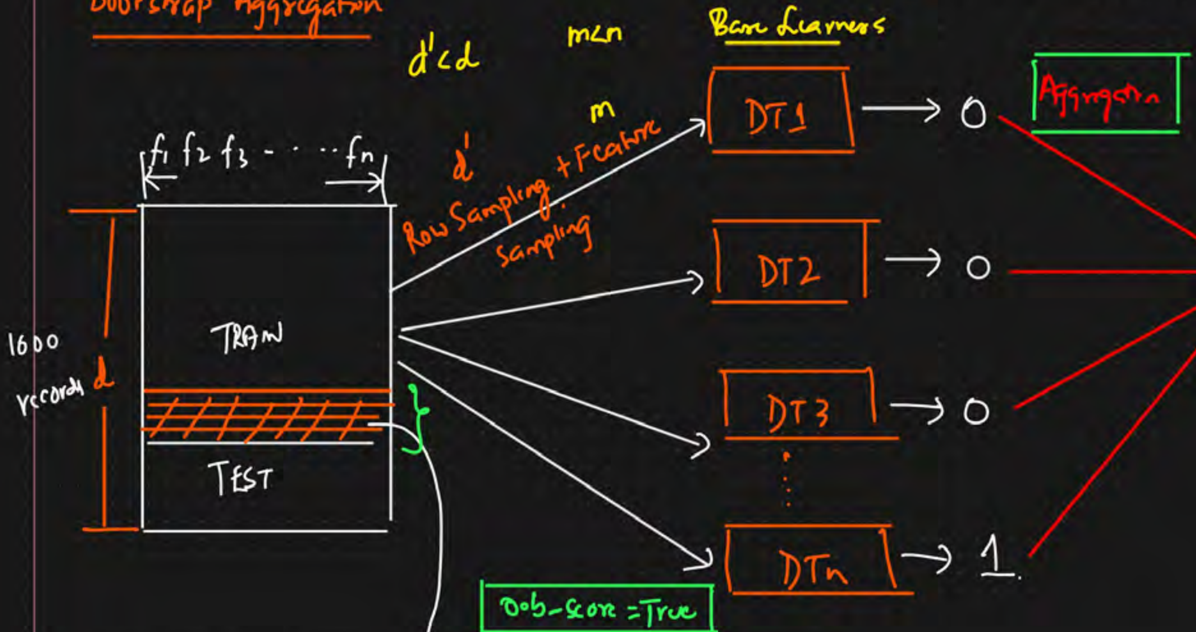


Decision Tree → Overfitting → Low Bias → Low Bias

High Variance → Low Variance

Bagging

Bootstrap Aggregation



Majority Voting Classifier

Regression ⇓ Average

Oob- Score = False

Out of Bag Score

Out of Bag Data ⇒ Validation DATA → CROSS Validation

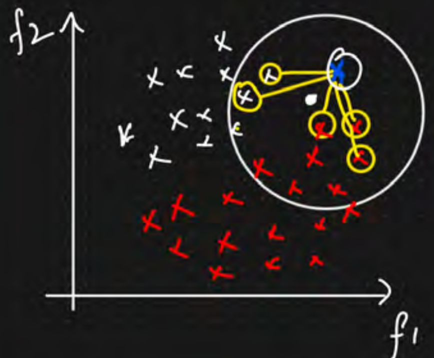
⇓

Validation accuracy → {oob-score}

② KNN Classifier And Regressor [K Nearest Neighbor]

① Classifier
② Regressor

① Classification



$\boxed{K = 5}$

$\boxed{K = 15}$

f_1 f_2 O/p (Yes/No)

① We have to initialize the K value
K = 1, 2, 3, 4, ... ⇒ Hyperparameter

② Find the K Nearest Neighbor from the new Test data

③ Maximum no. of points is in red category

Test data ⟶ Red category point.

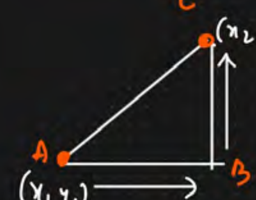
In Regression we Consider the Average of all data point.

① Euclidean Distance

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)}$$

② Manhattan Distance



⇒ AB + BC

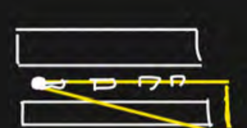
$$\boxed{|x_2 - x_1| + |y_2 - y_1|}$$

Application :

Air Traffic Control

Euclidean Distance

Manhattan



Manhattan Distance

A

B

Categorical ⟶ Numerical

$f_1 \longrightarrow f_1'$

Yes

No

Yes.

One hot Encoding

Yes	~~No~~
1	0
0	1
1	0

Location

New York

Florida

Texas.

New Y	F	~~T~~
1	0	0
0	1	0
0	0	1

Variants of KNN

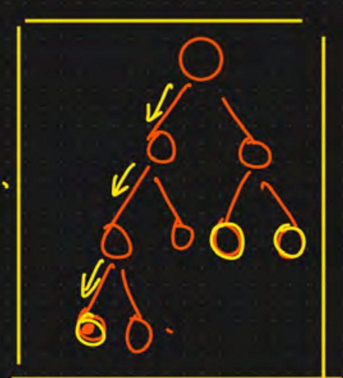
Disadvantage :

Time Complexity is more to search Nearest Neighbour.

Brut search

$O(n)$

$\boxed{K=3}$

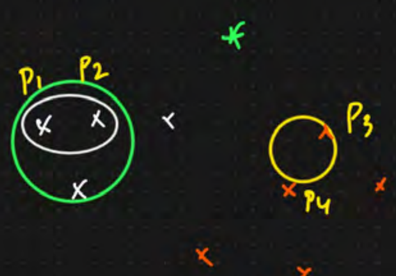
⇒ 

$O(\log n)$ ⟵

Variants of KNN

① KD Tree ⇒ K Dimension Tree

② Ball tree ⟵



$O(\log n)$

Feature Scaling ?? ⇒ Distance Formula

① Impact of Outliers?

② Feature Scaling?

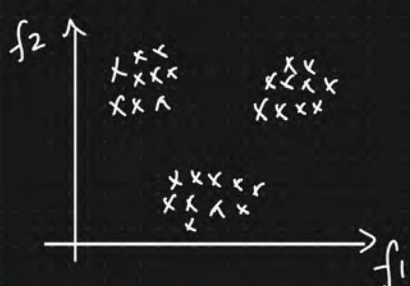
③ Unsupervised Machine Learning

① K Means

② Hierarichal Clustering

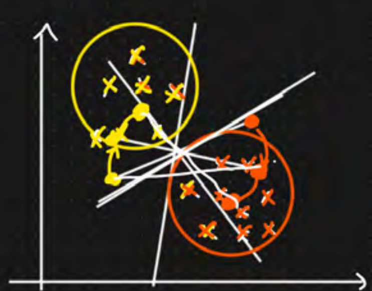
③ DBScan Clustering.

① K Means Clustering



⇒

K=3 K=4



Steps K = ?

① Intialize some K Centroids.

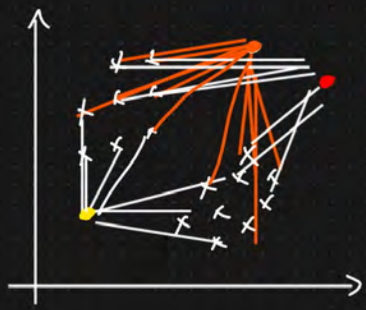
② Points nearest to the Centroid → will be grouped.

③ Move the Centroid → Mean

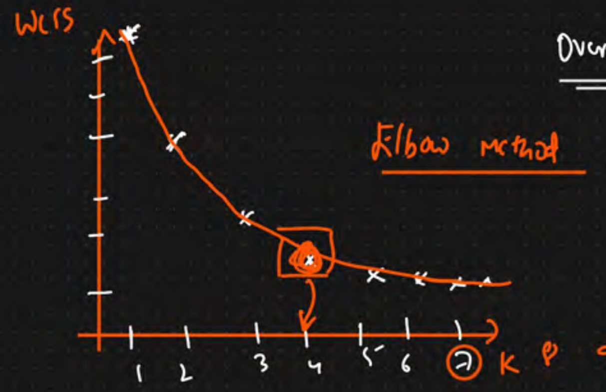
How do we select the K value

Initialize K=1 to 20

Within Cluster Sum of square



$$WCSS = \sum_{i=1}^{K} \left(\text{Distance between points to the nearest centroid}\right)^2$$

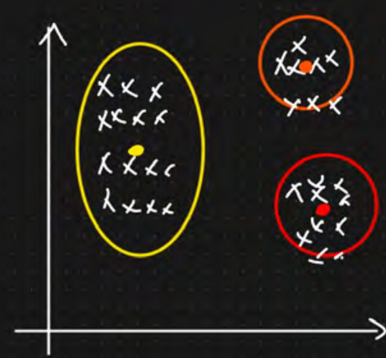


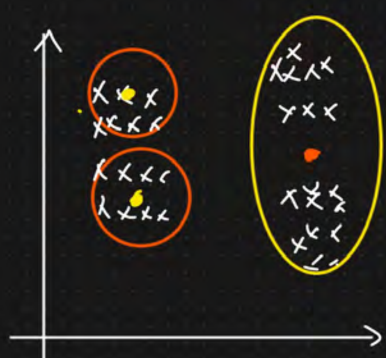
Elbow method

Overfitting
⇓
Generalize Model.

Random Initialization Trap (K Means ++)

KMeans ⇒



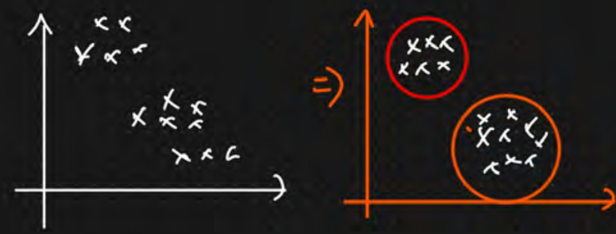


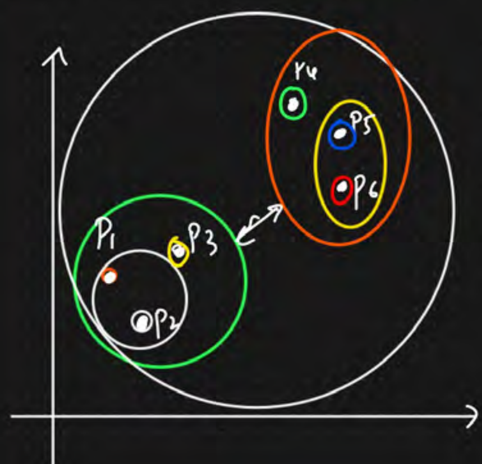
④ Hierarichal Clustering

Two Types

① Agglomerative
② Divisive

⇒ Geometric Intuition





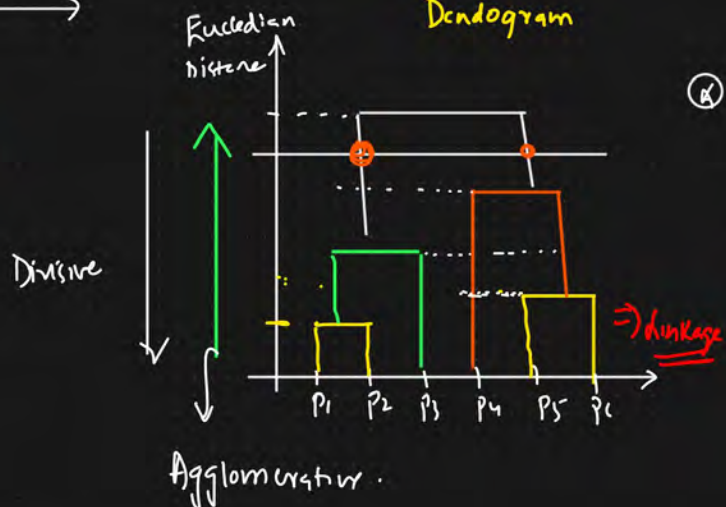
Steps

① For each point initially we consider it as a seperate cluster

② Find the nearest point and create a new cluster

2 centroids

Dendogram



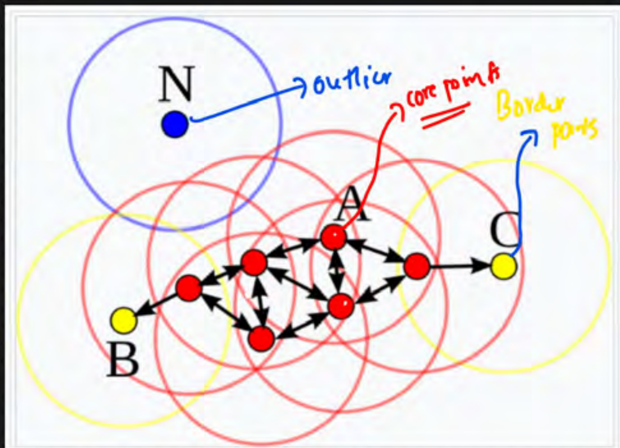
④ Select the longest vertical such that no horizontal line passes through it

KMeans Vs Hierarichal Clustering

① Data size
- → Huge ⇒ K Means
- → Small ⇒ Hierarichal clustering.

② Types of Data
- → Numerical ⇒ K Means
- → Variety of data ⇒ Hierarichal Means.

(*) Density Based Spatial Clustering . [DBScan Clustering]



→ Core point
→ border points
→ Noise / outliers .

} Non Linear Clustering

Hyperparameter

(1) minpts = 4 (2) ϵ = radius .

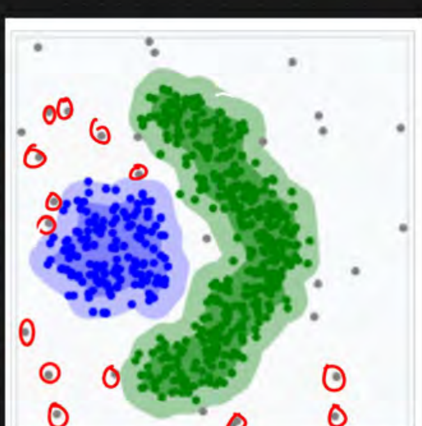
(1) Core point

No. of points within the ϵ should be $\geq$ minpts

(2) Border point

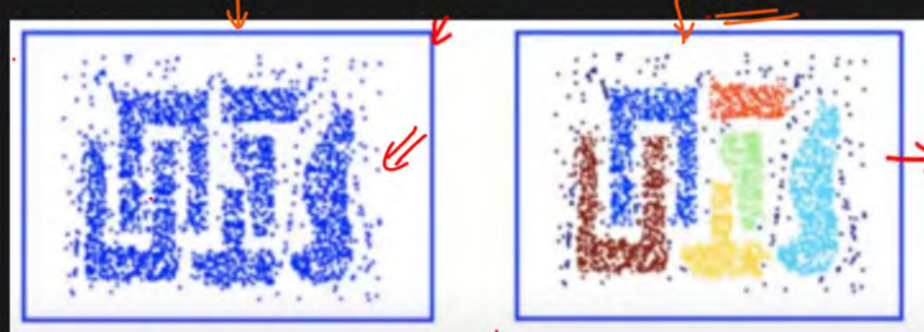
No. of data points within the radius ϵ but is less than minpts = 4.

(3) Outliers [DBScan is Robust to Outliers]



K Means

DBScan



The left image depicts a more traditional clustering method that does not account for multi-dimensionality. Whereas the right image shows how DBSCAN can contort the data into different shapes and dimensions in order to find similar clusters.

(*) Silhoutte Clustering → Performance metrics of clustering

(1)

Assume the data have been clustered via any technique, such as k-medoids or k-means, into k clusters.

For data point $i \in C_I$ (data point i in the cluster C_I), let

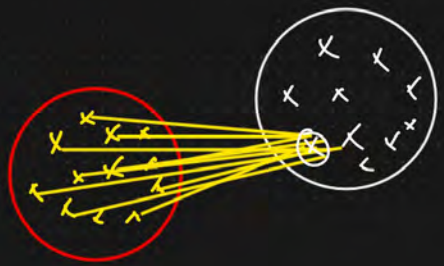
$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i,j)$$

be the mean distance between i and all other data points in the same cluster, where $|C_I|$ is the number of points belonging to cluster C_I, and d(i,j) is the distance between data points i and j in the cluster C_I (we divide by $|C_I| - 1$ because we do not include the distance d(i,i) in the sum). We can interpret a(i) as a measure of how well i is assigned to its cluster (the smaller the value, the better the assignment).

a(i) :

(2)

b(i) ⇒.



We then define the mean dissimilarity of point i to some cluster C_J as the mean of the distance from i to all points in C_J (where $C_J \neq C_I$).

For each data point $i \in C_I$, we now define

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$$

(3)

We now define a *silhouette* (value) of one data point i

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_I| > 1$$

$\boxed{-ve}$

⇓

Clustering is not done well

and

$$s(i) = 0, \text{ if } |C_I| = 1$$

Which can be also written as:

$$s(i) = \begin{cases} 1 - \underbrace{a(i)}_{} /b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

0 to 1

-1 to 0%.

From the above definition it is clear that

$$-1 \leq s(i) \leq 1$$

≤ 1

THANK - YOU