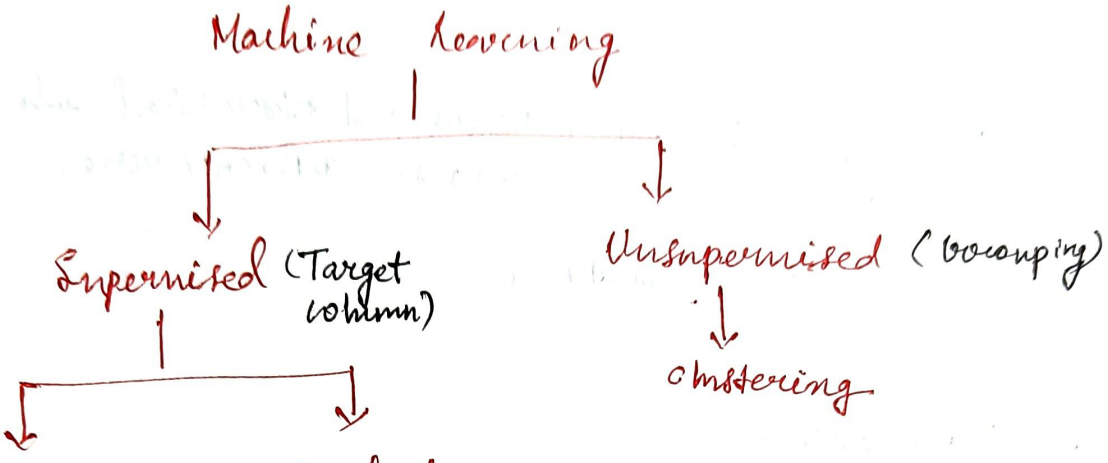
05-NOV-2022

(KNN) K-Nearest-Neighbour

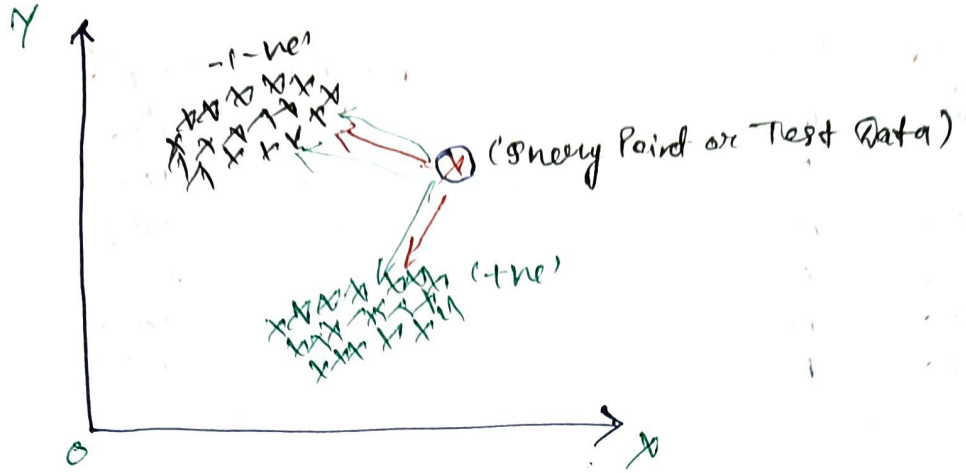


Regression
1. Linear Regression
2. Ridge Regression
3. Lasso Regression
4. Elastic Net
5. SVM/SVR
6. Decision Trees classifier
7. Random Forest classifier
8. Gradient Boosting classifier
9. Xgboost Regressor

Classification
1. Logistic Regression
2. SVM/SVC
3. Decision Trees classifier
4. Random Forest classifier
5. Gradient Boosting classifier
6. Xgboost classifier
7. Naive Bayes → classification → test data
   ↓
   NLP with ML
   ↓
   test data
   TF/IDF
   NB
   Evaluation

10. KNN → Distance based approach.
- there is no learning, it is a simple distance approach.
- It is called as lazy learner.
- We can solve classification as well as regression problem.
- Any data we can take like 2D, 3D etc.

classification



Ques x belong to which class?

Ans ⇒ ① Find distance w.r.t each and every point.

② Now, choose K and it is a hyperparameter.

K = 2, 3, 5, 6, 7.

Let's assume [K=3].

3-Nearest Neighbour. after calculating distance

+ne
+ne
-ne

} - Based on a majority note, we will select query point will belong to which class.

- My Here, Query point belong to '+ve' class.

5- Nearest Neighbour

+ne
+ne
-ne
-ne
-ne

} - Here, Query point belong to '-ve' class

[Note: - We don't consider K as even, because some cont of '+ve' and '-ve' class can occur. Hence we cannot decide majority.

- Take K value as odd.]

[- In classification we take majority note and in regression we take mean instead of majority note.]

Not ML algorithm

Classification

weight	Height	gender
50	150	M
60	160	F
70	175	M
80	185	F
90	190	M
85	150	M

KNN - classifier

Regression

weight	Height
50	150
60	160
70	175
80	185
90	190
85	150

KNN - Regressor

① train ② testing (Not ml algo / Lazy learner)

> Here no, training happens. whenever new data come, it will calculate the distance and assign the value.

> Time complexity will be high.

To reduce time complexity:

① Ball tree

② Kd tree

K-Means sklearn Parameters

init = initialization of centroid

n_cluster = number of cluster

Kmeans.labels // shows the clusters tagging.

Kmeans.cluster_centers_ // shows the center of cluster.