K-Nearest Neighbour (KNN)

* KNN is Instance Base Learning

* Instance Base Learning

1) When we get the traning example we donot Process them and learn a model instead we just store example.

2) Algorithm not train rather when it gets the test instance, it uses the stored instance in memory in order to find possible y.

3) In space we have instance with x & y value if new instance is given find out close x_i & that $x_i \Rightarrow y_i$
Find similar instance or find same neighboring instance, most nearest Instances.

KNN:- Algorithm.

1) Training Phase - Save training example.
Stroe the example in structure so that searching through this example become faster.

2) Predication :- Get test instance - x_t. Find training example (x_1, y_1) ie closest to x_t, & predict y_1 as the output y_t.

3) Insted find one ~~example~~ traing example.
find K - No of traiming example.

* Classification

Predict majority class from $\{y_1, y_2, \ldots y_K\}$

* Regression :- we will get different values of $y_1, y_2, y_3, \ldots y_K$ Predict the average.

1) Regression take average of K values for averaging under circumstances where there are.
   1) Noise in Attributes.
   2) Noise in class Labels.
   3) Classes may partially overlap.

* How to decide K value.

1) Small K capture fine structure of problem for small training set.

2) Large K :-
   1) Use large K, use weighted distance function.
   2) less ~~less~~ sensitive to noise (particularly class noise)
   3) Better probability estimate for discrete classes.
   4) large training set allows use large K value.

* weighted Euclidian Distance.

$$D(x_i, x_j) = \sqrt{\sum_{m=1}^{N} w_m (x_{im} - x_{jm})^2}$$

* Why Feature Reduction is important in KNN

1) If we have instance based large features it pose a problem, because some features may be more important than other and some features may be irretevent.
This specially impacts K-nearest neighbor istance based learning algorithm.

2) So it is important to remove extra features because for high dimensional phase two items which are similar may still differ in some unimportant attributes & the difference in distance may similar.

3) So it is important to find good representative training example for given test example. So feature reduction is important.

* Distance weighted KNN

1) There is treadoff between small & large K can be difficult.

2) Use large K, but more emphasis on nearer neighbor

3) Predication Test = $\dfrac{\sum_{i=1}^{K} w_i * \text{classes}}{\sum_{i=1}^{K} w_i}$

KNN :- K Nearest Neighbors. (Lazy Learning)

(*) → Instances Based Learning (distance function)

1) KNN is based on features similarity, we can do classification using KNN classifier.

2) KNN - K Nearest Neighbors is one of the simplest Supervised Machine Learning algorithm. It classifies a data point based on how its neighbors are classified

3) KNN algorithm is based on feature similarity. Choosing the right value of K is a process called parameter tuning & important for better accuracy.

* A very simple classification & Regression
a) In case of classification new data points gets classified in a particular class.
b) In case of regression, new data gets labeled based on the avg. value of K-nearest neighbour

* It is a lazy learner because it doesnot learn much from training data, learn from live data.

* Default method is Euclidean distance.

* Requirement for K-NN
1) Generally K gets decided based on the square root of data points (K is generally odd)
for ex:- if 1000 data points the to K=100 with class
2) Data Normalization
3) Installation of "class" ~~to~~ library.

breast cancer.

* A case is classified by a majority vote of its neighbors with the case being assigned to the class most common amongest its K Nearest neighbors measured by a distance function

Distance measure

Euclidean $\sqrt{\sum_{i=1}^{K}(x_i-y_i)^2}$

manhattan $\sum_{i=1}^{K}|x_i-y_i|$

Minkowski $\left(\sum_{i=1}^{K}(|x_i-y_i|)^q\right)^{1/q}$

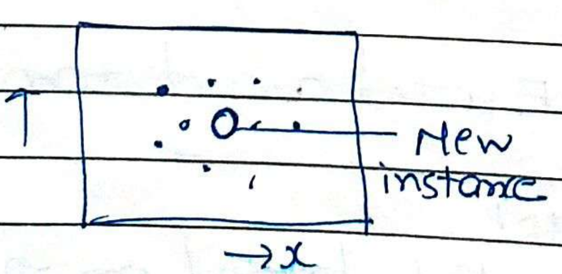
Hamming Distance $D_H=\sum_{i=1}^{K}|x_i-y_i|$

$x=y \Rightarrow D=0$ x y Male Male 0

$x\neq y \Rightarrow D=1$ Male female 1

* Instance Based Learning (Lazy Learning)

In this it won't learn the model it stores the value & used it.



find closest instance to $(\hat{y}_x)$ ie. of that x find y.

1) Find most similar instance (similarity) distance function.

* Tranning Phase :- Save the example (Store example) in such way that it will helpful latter

* Prediction time :- Get the test instance x_t
Find the training example (x_1, y_1) ie closeast to x_t & then predict y_t as the output

· Instead of find single tranning example we find K- tranning example
$\{(x_1, y_1)(x_2, y_2)(x_K, y_K)\}$ which are closest to x_t & predict ast output y_t we y_1, y_2, y_K in

Classification classification :- predict majority of class as O/P (most frequent class) from $(y_1, y_2 .. y_K)$

Regression :-
i) Predict $y_1, y_2 .. y_K$ & take average of it & predict it as y_t (estimate)

* Vioronoi Diagram

* Improvements
- Weighting examples from the distance
- measuring "closeness".
- Finding "Close" examples in a large training set quikly

$x_i = (x_{i1}, x_{i2} .. x_{iM})$ $x_j = (x_{j1}, x_{j2} .. x_{jM})$

$$\text{Distance Eculiden} = \sqrt{\sum_{K=1}^{M} (x_{i_K} - x_{j_K})^2}$$

Find Ecuiliden distance from a test point to all the point of tranning & Select the smallest distance point.

Noise in attributes.
Noise in class labels.
classes may partially overlap.

Small K :- Capture fine structure of pattern space better. may be necessary for small training set.
Large K :- Less sensitive noise (class Noise) better probability estimate for discrete classes

· For larger training set allows use to large K.

$x\,(P_1 = 3\ \&\ P_2 = 7)$ K = 3 nearest neighbour

	P_1	P_2	Class
i	7	7	False
ii	7	4	False
iii	3	4	True
iv)	1	4	True

Euclidean Distance = $\sqrt{(x_H - H_1)^2 + (y_H - W)^2 + \ldots}$

observed   Actual.

Here observed is 3 & 7

$D(x, i) = \sqrt{(3-7)^2 + (7-7)^2} = 4 \leftarrow N_4$ False

$D(x, ii) = \sqrt{(3-7)^2 + (7-4)^2} = 5$

$D(x, iii) = \sqrt{(3-3)^2 + (7-4)^2} = 3 \leftarrow N_1$ True

$D(x, iv) = \sqrt{(3-1)^2 + (7-4)^2} = 3.6 \leftarrow N_2$ True

$\therefore$ K-NN = N_1 2 True > 1 False

$\therefore$ Answer is $X(P_1 = 3, P_2 = 7)$ belong to ~~False~~ True

Predict the type of fruit or food type.
Tomato (Sweet = 6, Crunch = 4) belongs.

Ingradient	Sweet	Chrunch	food Type
Grape	8	5	fruit
Green bean	3	7	Vegetable
Nuts	3	6	Protein
Orange	7	3	fruit.

D(Tomato, Grape)

$D(x,i) = \sqrt{(6-8)^2+(4-5)^2} = 2.2$ F

$D(x,ii) = \sqrt{(6-3)^2+(4-7)^2} = \cancel{3.6}\ 4.2$ V

$D(x,iii) = \sqrt{(6-3)^2+(4-6)^2} = \cancel{3.31}\ 3.6$ P

$D(x,iv) = \sqrt{(6-7)^2+(4-3)^2} = 1.4$ fruit

Since distance of tomato from Organge is minimum
∴ tomato will belong to fruit.

Eager Vs Lazy Learner.

1) Eager

a) Generalized model from training data set is constructed
b) using the model the class of test data set is predicted
c) Decision Tree
ex –

a) Training datasets stored
b) on querying similarity b/w test data & training set records is calculated to predict the class of test data

ex – KNN

K-NN

1) Non-parametric method used for classific
2) Prediction for test data is done on the basis of it neighbor
3) K is an integer (small) K=1 k is assigned to class of single nearest neighbor

	Acid Durability	Strength	class
1	7	7	Bad
2	7	4	Bad
3	3	4	Good
4	1	4	Good

Test data → acid durability = 3 & Strength = 7 Class?

$D = \sqrt{(3-7)^2+(7-7)^2} = 4$

$D = \sqrt{(3-7)^2+(7-4)^2} = 5$

$D = \sqrt{(3-3)^2+(7-4)^2} = 3$ ← Min^m N

$D = \sqrt{(3-1)^2+(7-4)^2} = 3.6$

∴ (Durability = 3, Strength = 7 − Good

* Instance based Learning includes neargest neighor & locally weighted regression ~~space~~ methods that assume instances can be
* represented as point in a Eculidean space.

* Instance based method are sometimes referred to as "lazy" learning method because the delay processing until a new instance must be classified.

A Key advantage of this kind of delayed or lazy learning is that instead of estimating the target function once for the entire instance space these methods can estimate it locally & differently for each new instance to be classified