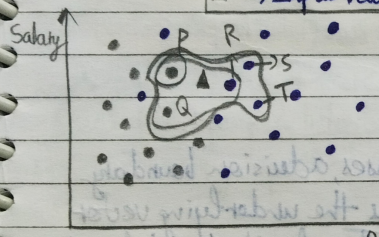
Date.....

KNN [K Nearest Neighbors]

Geometric Intuition

S.No	Age	Salary (in Thousands)	Purchased
1	25	20	N
2	63	120	Y
3	33	75	N
4	42	100	Y
...	...	...	...

▲ → Input Values Set



Scale:-
● → Purchased
● → Not Purchased
▲ → New Input

Q. Determine whether this input, Purchase or Not Purchased the product?

Ans Case 1:- K=1 [It means, it selects the Nearest Point]

So, as Point 'P' is labelled as Not Purchased. i.e, the input Point is also labelled as Not Purchased...

Case 2:- K=3

[It means, it firstly selects the 3 Nearest points and then, the Majority Count of the point is followed].

i.e, ⇒ The Nearest Points are:- P, Q and R. But According to Majority Count..

● → 2
● → 1

⇒ The New Input point will be labelled as Not Purchased...

Case:-3 K=5

⇒ In this Case, the Nearest Points are:- 'P', 'Q', 'R', 'S', 'T'.

According to Majority Count ⇒

● → 3
● → 2

So:- The New Input point is labelled as Purchased...

Spiral

Date.....

Points to consider

- Although in this example we are talking about the 2D example but the concept holds true for higher dimensions as well.

- In this example we have taken Euclidean Distances into consideration but other distances are used as well like Manhatten Distance or MinKowski Distance.

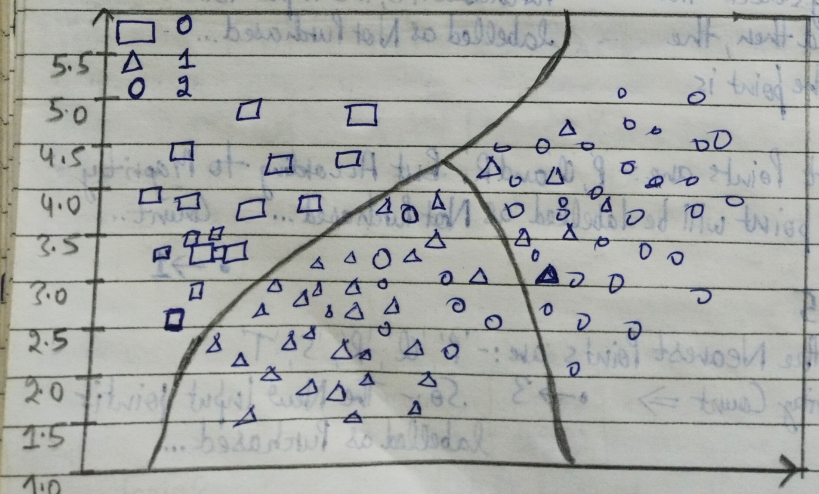
Euclidean Distance Formula:-

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

To Calculate the K=?
K=Square Root(Total No. of Rows in Table)

Decision Boundary

In a classification problem with two or more classes, a decision boundary or a decision surface is a hypersurface that partitions the underlying vector space into two or more sets, one for each class. The classifier will classify all the points on one side of the decision boundary to one class and all those on the other side as belonging to the other class.



Spiral

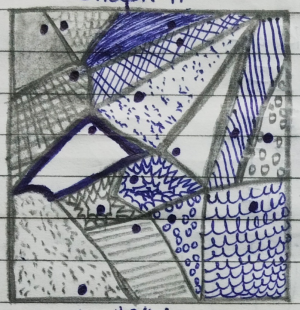
Date.....

Vornoi Diagram

Q What are the Vornoi Diagrams?

Ans In mathematics, a Vornoi Diagram is a partitionary of a plane into regions based on distance to points in a specific subset of the plane.

Graph "A"



Graph "A" is a Vornoi Diagram

According to the "A" there are many regions. The points or set of points [called Seeds] is specified beforehand and for each seed there is corresponding region consisting of all the points closer to that seed than to any other.

These regions are termed as Vornoi Cells. The Vornoi diagram of a set of points is dual to its Delaunay triangulation.

* Steps to plot Decision Boundary for Knn.

1. Train the classifier on the training set.
2. Create a uniform grid (with the help of numpy Meshgrids) of points that densely cover the region of input space containing the training set.
3. Classify each point on the grid. Store the results in the Array A, where Aij contains the predicted class for the point at row i, column j on the grid.
4. Plot the array as an image, where each pixel corresponding to grid point and its color represents the predicted class. The decision boundary can be seen as contours where the image changes color.

Spiral

Step-1:- Train the data with a classifier.

Date.....

Step-2: Creating a Meshgrid....

X ⇒ [1,2,3,4], Y ⇒ [5,6,7]

XX, YY = np.meshgrid(X, Y)

15	25	35	45
16	26	36	46
17	27	37	47

>>

Output :-

1	2	3	4
1	2	3	4
1	2	3	4

XX

5	5	5	5
6	6	6	6
7	7	7	7

YY

Background Working:-

X.Length ⇒ 4

Y.Length ⇒ 3

XX.YY ⇒ 4X3 ⇒ 12

Array 1 ⇒ [-4,-3,-2,-1,0,1,2,3,4]

Array 2 ⇒ [-5,-4,-3,-2,-1,0,1,2,3,4,5]

99 Points [9*11]

Purpose of Meshgrid...

Meshgrid is very useful to evaluate functions on a grid. We can apply any function to the points of a meshgrid to plot a function.

Step-3: Classifying every point on the meshgrid

It means all the classified points of the Array is now passed as an prediction to the classifier.

For example:- X_train ← Includes the points.

Y_train ← Includes the points.

K=2

Spiral

Date.....

```
Knn = KNeighborsClassifier(n_neighbors = K)
Knn.fit(X_train)
```

To Plot the Decision Boundary:-

```
contourf(X_Value, Y_Value, Z)
```

```
Input_Data = np.array([XX.ravel(), YY.ravel()])
labels = Knn.predict(Input_Data)

plt.contourf(XX, YY, labels.reshape(XX.shape))
plt.show
```

Effects on Decision Boundary by Changes on "K" VALUE...

According to the summerization, I have Concluded the following:-

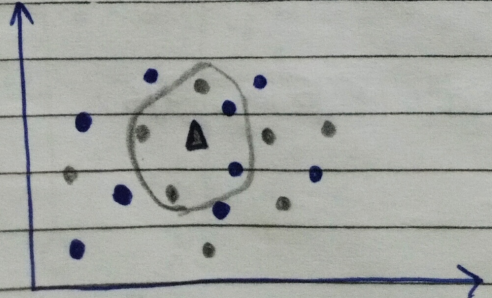
(i) For Smaller "K" Value like K=1, model generally overfits and shows high Variance.

(ii) For bigger "K" value like K=n, where 'n' is the number of rows in the training set. the Model underfits and shows high bias.

(iii) "But, as we Increase the Value of "K", the smoothness of the Decision Boundary Increases.."

Weighted KNN

Weighted KNN is a variant of K Nearest Neighbors where we take an yet elegant assumption that the impact of the nearest point is more than the points that are far away on the New point....



Δ → New Point..

The point will label as "Blue" Point or "Black" Point. Let's do this as per the Weighted KNN.

Spiral

Date.....

[1/Distance]

Point	Label	Distance	Weights
x1, y1	Blue	0.2	5
x2, y2	Blue	0.5	2
x3, y3	Black	0.7	1.4
x4, y4	Black	1.2	0.8
x5, y5	Black	1.5	0.6

- Blue Points ⇒ 5+2 = 7
- Black Points ⇒ 1.4+0.8+0.6 ⇒ 2.8

Weight = 1/Distance (Inverse Proposition al)

Sum of Weight of Blue Points > Sum of Weight of Black Points

Hence, ▲ New Point will be labelled as Blue Point....

- As per the Weighted KNN, the results produced by the Weighted KNN is far better than the results produced by the Normal KNN.
- The normal Intitution of Weighted KNN is :-

★ AS THE DISTANCES BETWEEN POINT ⇧

WEIGHT ⇩ $W = 1/d$