Unit 4

SUPERVISED LEARNING II

- **Distance Based Models:**
  - Neighbors and Examples,
  - Nearest Neighbor Classification
  - Finding values of K,
  - Distance Measures.
- **Kernel Based Models:**
  - Support Vector Machines,
  - Linear SVM,
  - RBF SVM,
  - Sigmoid SVM,
  - Polynomial SVM.
- **Probability Based Models:**
  - Conditional Probability,
  - Bayes Theorem,
  - Naive Bayes Classification,
  - Bayesian Regression.
- **Case Studies:** classification algorithm for student learning capacity

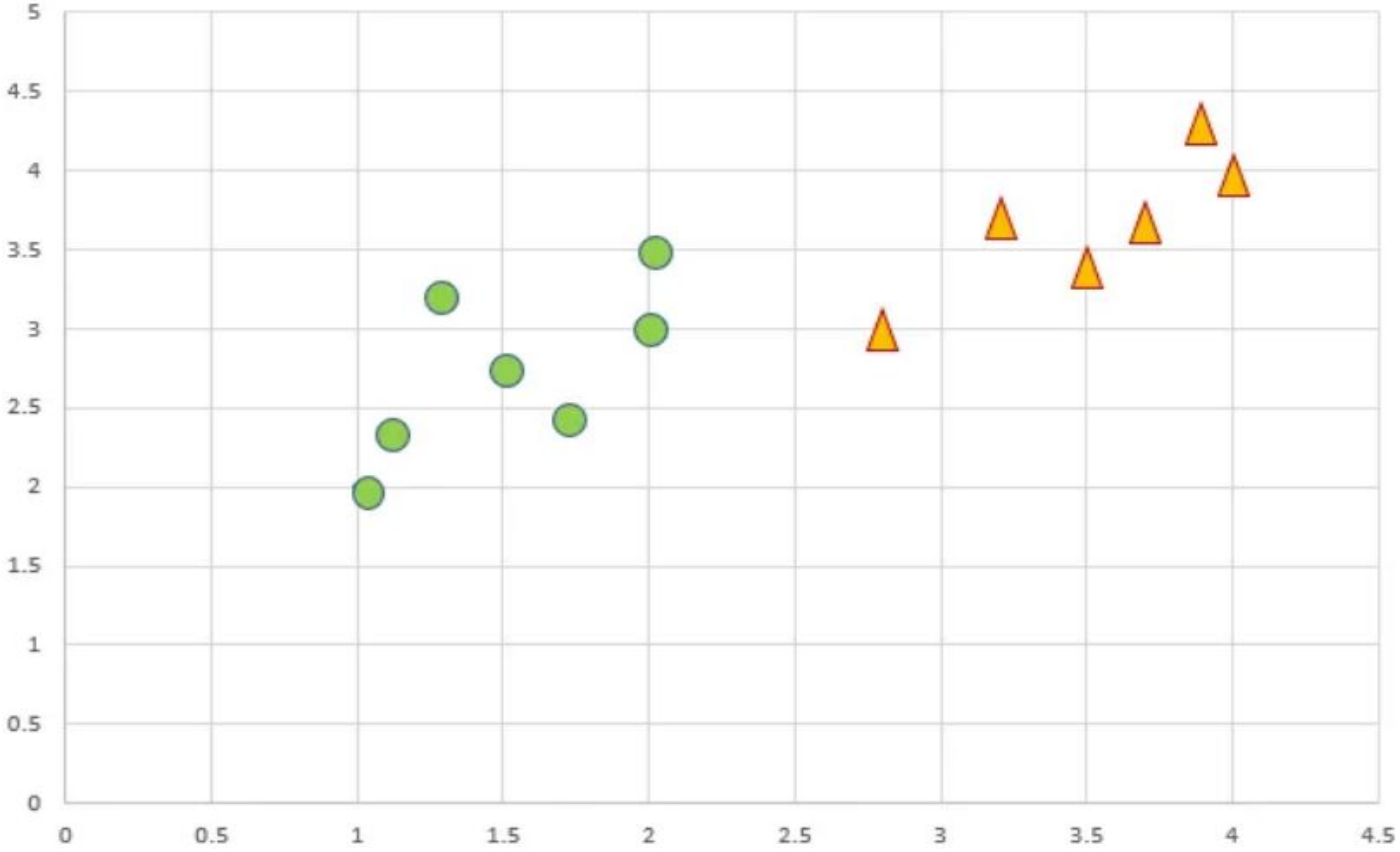
KNN

- The K-Nearest Neighbours (KNN) algorithm is one of the simplest supervised machine learning algorithms that is used to solve both classification and regression problems.
- For **classification problems**, it will find the k nearest neighbors and predict the class by the majority vote of the nearest neighbors.
- For **regression problems**, it will find the k nearest neighbors and predict the value by calculating the mean value of the nearest neighbors.

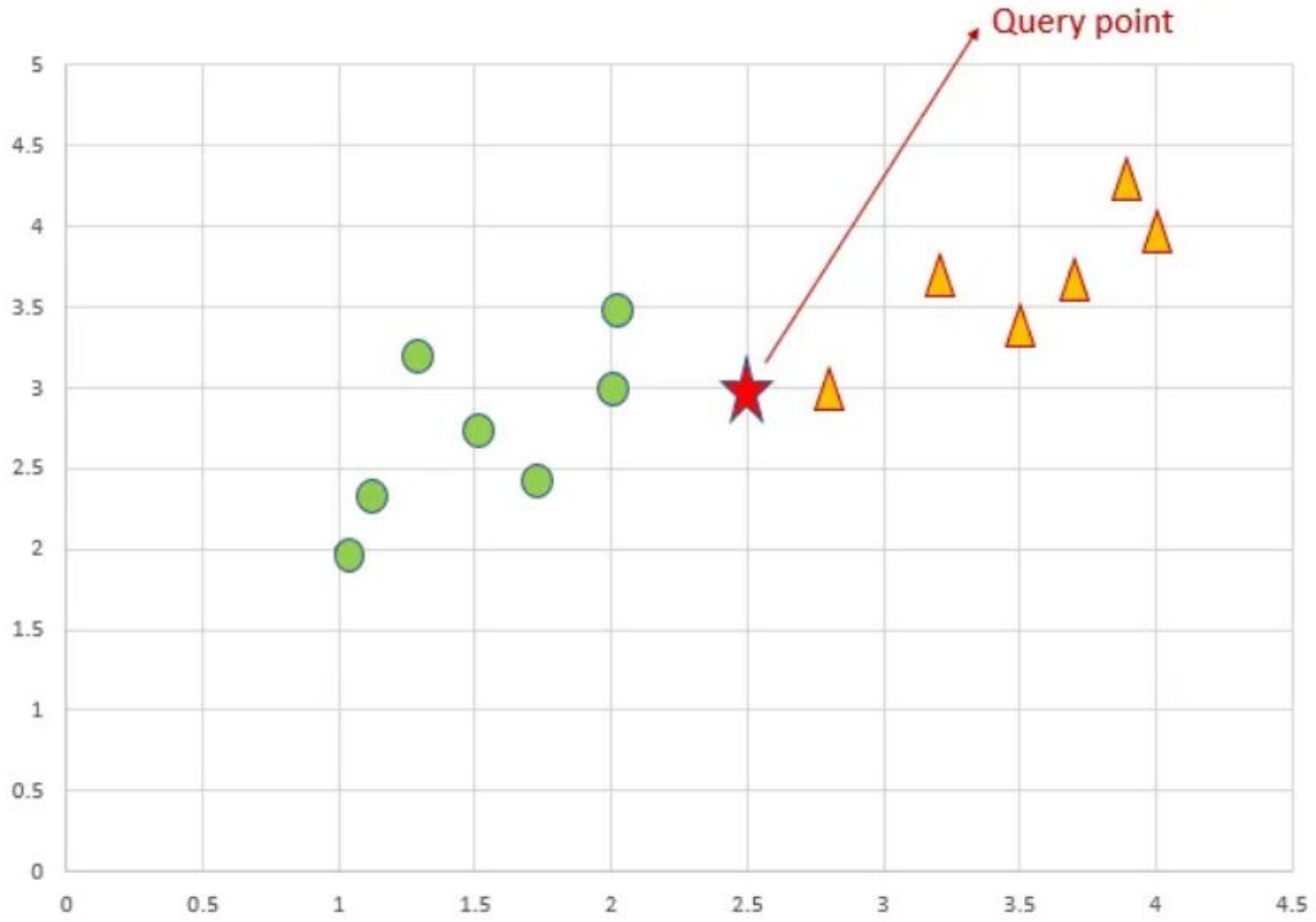
The main concept behind KNN is

Birds of same feather flock together

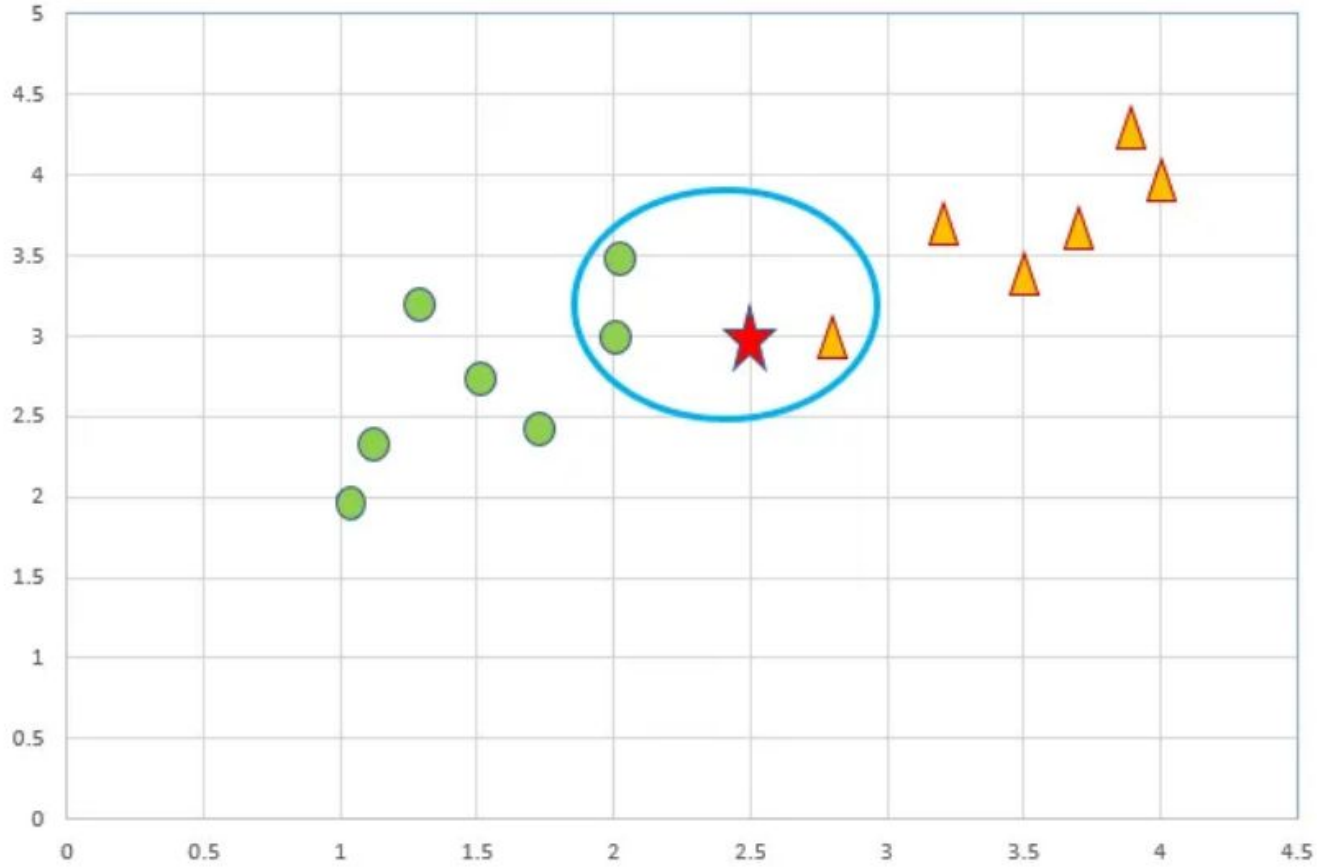
- **KNN Classification**
- Let’s learn how to classify data using the KNN algorithm. Suppose we have two classes circle and triangle.
- Below is the representation of data points in the feature space.



- Now we have to predict the class of new query point (star shape shown in the figure). We have to predict whether the new data point (star shape) belongs to class circle or triangle.



- First, we have to determine k value. k denotes the number of neighbors.
- Second, we have to determine the nearest k neighbors based on distance.
- This algorithm finds the k nearest neighbor, and classification is done based on the majority class of the k nearest neighbors.
- Here in this example, the nearest neighbors are shown inside the blue oval shape.
- So the majority class belongs to “Circle”, so the query point belongs to class circle.



- Now, comes the important point.
  1. How to find the optimum k value?
  2. How to find the k nearest neighbors?

- **How to find the optimum k value?**
- Choosing the k value plays a significant role in determining the efficiency of the model.

1. If we choose k =1 means the algorithm will be sensitive to outliers.
2. If we choose k= all (means the total number of data points in the training set), the majority class in the training set will always win. Since KNN classifies class based on majority voting mechanism. So all the test records will get the same class which is the majority class in the training set.
3. Generally, k gets decided based on the square root of the number of data points.
4. Always use k as an odd number. Since KNN predicts the class based on the majority voting mechanism, the chances of getting into a tie situation will be minimized.
5. We can also use an error plot or accuracy plot to find the most favorable K value. Plot possible k values against error and find the k with minimum error and that k value is chosen as the favorable k value.

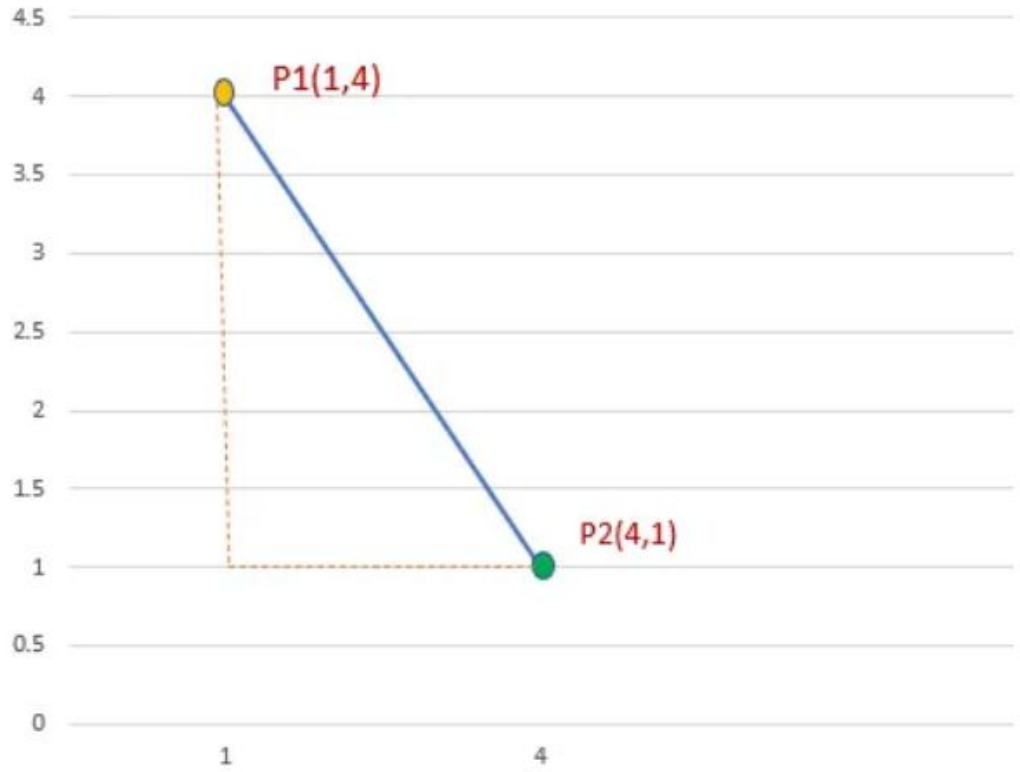
How to find the k nearest neighbors?

- There are different techniques to find the k nearest neighbors.
- Euclidean distance
- Manhattan distance
- Minkowski distance

One of the most used techniques is the Euclidean distance.

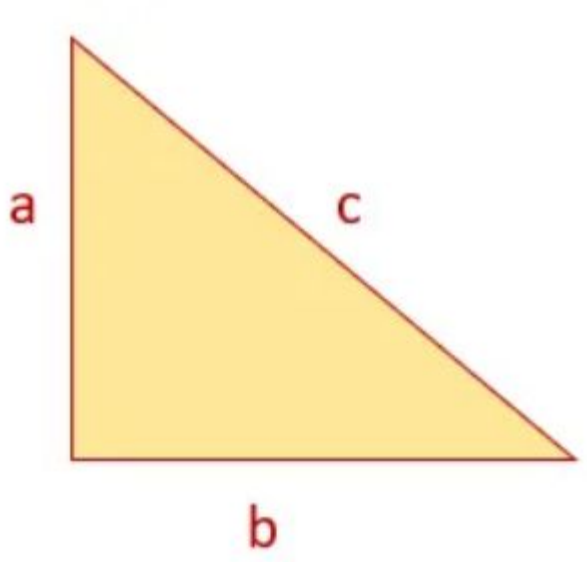
Euclidean distance

- Euclidean distance is used to calculate the distance between two points in a plane or a three-dimensional space. Euclidean distance is based on the Pythagoras theorem.
- Let’s calculate the distance between P1 and P2
- P1(1,4) and P2(4,1)



Pythagoras theorem

- In any right-angled triangle, the square of the hypotenuse(longest side of a triangle) is equal to the sum of the other two sides of the triangle.



$a^2 + b^2 = c^2$

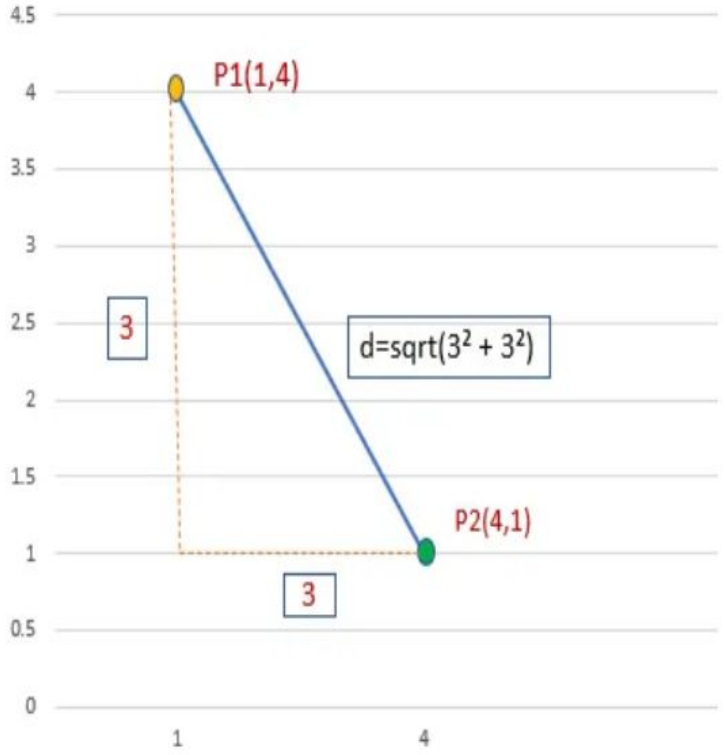
$\rightarrow c^2 = a^2 + b^2$

$\rightarrow c = \text{sqrt}(a^2+b^2)$

The Euclidean distance formula is derived from Pythagoras theorem

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

- Now we can calculate the distance between two points P1(1,4) and P2(4,1) using the Euclidean distance formula



$$d = \sqrt{(4-1)^2 + (1-4)^2}$$

$$d = \sqrt{9+9}$$

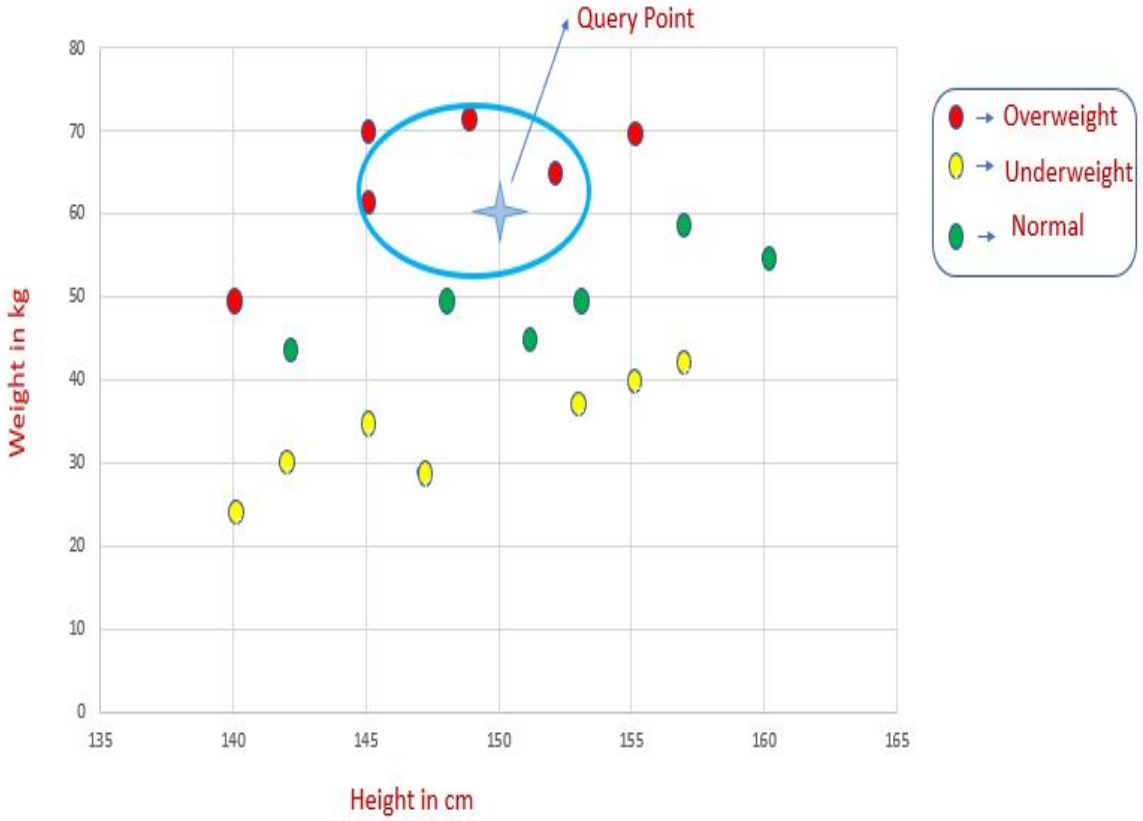
$$d = 4.24$$

1. KNN algorithm calculates the distance of all data points from the query points using techniques like Euclidean distance.
2. Then, it will select the k nearest neighbors.
3. Then based on the majority voting mechanism, KNN algorithm will predict the class of the query point.

- **Example Dataset**
- I have taken a small data set that contains the features Height and Weight and the target variable BMI.

Height	Weight	BMI	Distance
151	45	Normal	16.03
155	70	Overweight	45
145	35	Underweight	130
160	55	Normal	60
140	50	Overweight	110
142	30	Underweight	248
152	65	Overweight	8
157	42	Underweight	133
142	44	Normal	136
153	50	Normal	33
145	62	Overweight	5
155	40	Underwieght	105
148	50	Normal	22
145	70	Overweight	45
140	25	underweight	100
147	29	Underweight	30
149	71	Overweight	10
157	59	Normal	14
153	37	Underweight	72
Query point			
150	61	Overweight	

- **Nearest neighbors**



- We have taken k=3
- Out of 3 nearest neighbors, all 3 of them belong to the “Overweight” category So, the person whose height=150 cm and weight= 61 kg belongs to the “Overweight” category.
- If two of them belong to the “Overweight” and one belongs to the “Normal” category means the majority wins.