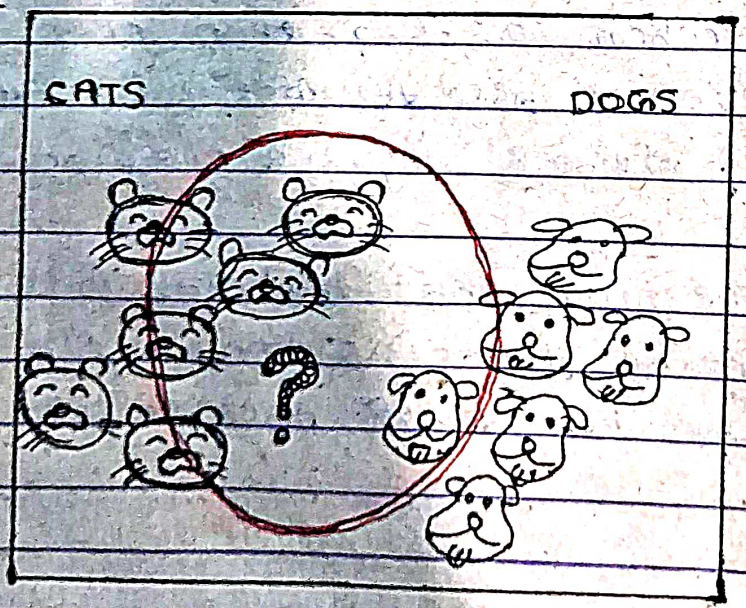
★K-nearest Neighbour :

- The distance based machine learning algorithm.
- Non-parametric, supervised ML algorithm.
- Can be used to solve both classification & regression problem statements.
- Eg, We often notified that you share many characteristics with your peers, whether it be your thinking process, working etiquette, philosophies or other factors.
  As a result we build friendship with people we deem similar to us.

The K-NN employs the same principle it aims to locate all of the closest neighbours around a new unknown data point in order to figure out what class it belongs to.

Eg



In this value of k is 3, since, there are 2 cat and just one dog

In the proximity of 3 closest neighbours, the algorithm would predict that it's a cat base on proximity of the three closest neighbours in red circle boundary.

Here, 'k' is the hyperparameter for KNN, for proper classification/prediction, the value of 'k' must be fine tuned.

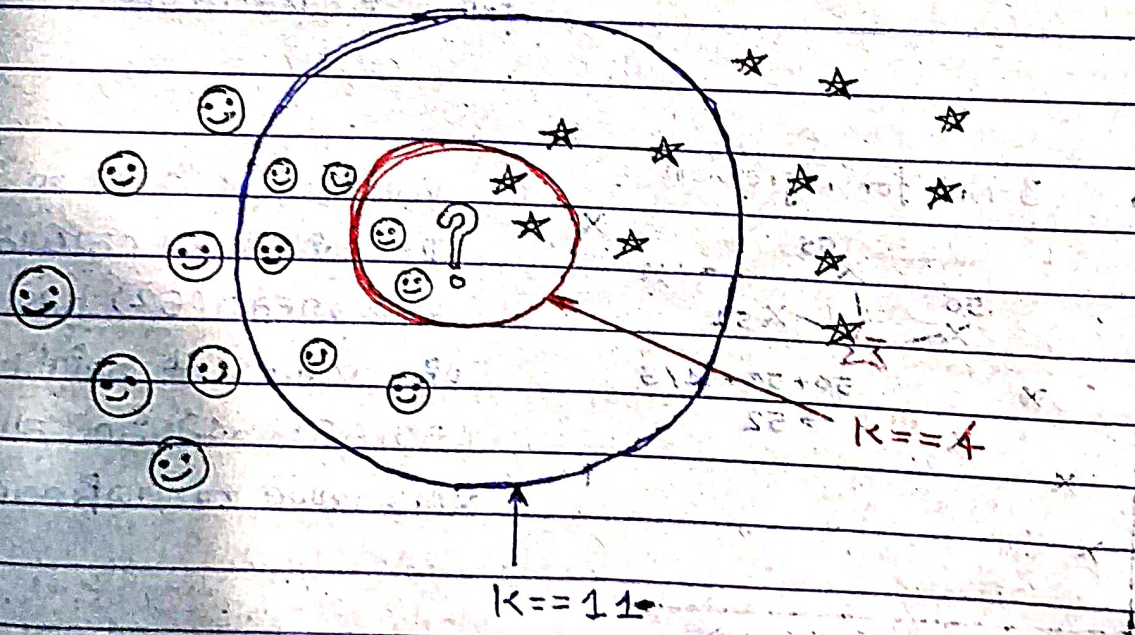
But, How do we select right value of 'k'?

We don't have a particular method to determine value of 'k'. Here we will try to test the models accuracy for different 'k' values.

The value of 'k' that delivers best Accuracy for both training & testing data is selected.

Note: It's recommended to always select an odd-value of 'k'.

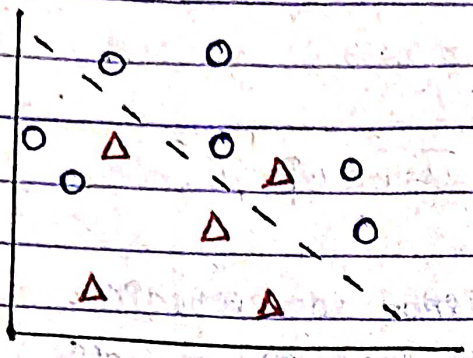
When the value of 'k' is even a situation may arise in which elements from both groups are equal



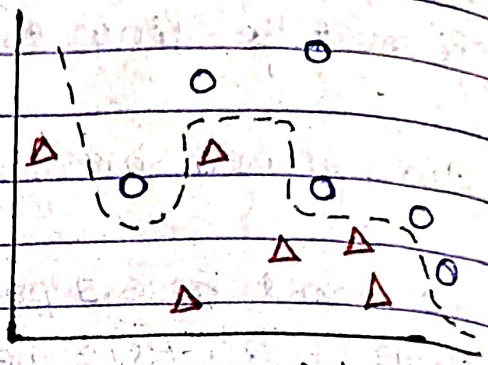
KNN for classification statements

KNN tends to use the concept of "majority voting" within the given range of k values, the class with the most votes is chosen

★ Impact of selecting smaller & larger k-value :



large 'k'



smaller 'k'

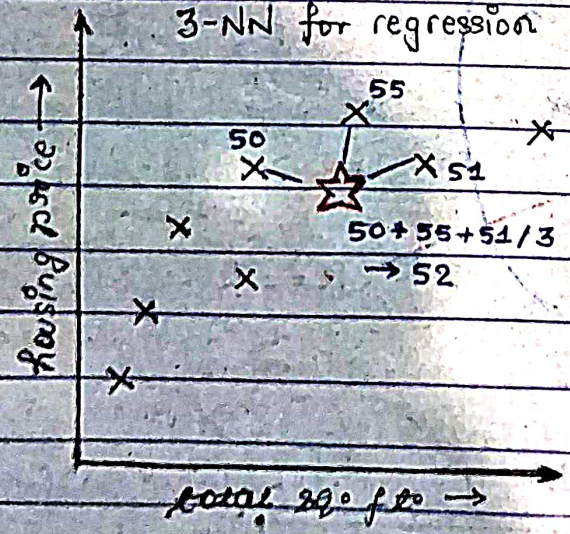
The case of "Underfitting" occur when the value of k is increased.

In this case model would be unable to correctly learn on training data.

The case of "Overfitting" occur when the value of k is smaller.

The model will capture all of the training data including noise.

The model will perform poorly for test data in this scenario.



Value of k is set to 3, It will now calculate the mean (52) based on the values of neighbors (50, 55 & 51) and allocate this value to unknown data

KNN works for regression statement

- K-NN is non-parametric, meaning it doesn't make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM which assume gaussian dist° of given data.)

- Intuition behind K-NN Algorithm?

Distance metric used in KNN algorithm.

Euclidean Distance $$d(p,q) = \sum_{i=1}^{n} (q_i - p_i)^2$$

cartesian dist° b/w two points which are in plane / hyperplane.

Manhattan Distance $$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

summing the absolute difference b/w the coordn° of the points in n-dimension.

(1) Choose value of 'k',

If i/p data has more outlier / noise ↓
higher value of k would be better.

use cross-validation method.

selecting the optimal value of k, k represents nos of nearest neighbours that needs to be considered while making prediction.

(2) calculating distance

(3) finding nearest neighbour

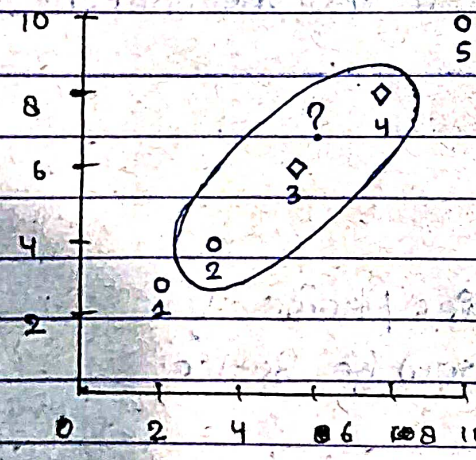
(4) voting for classification or taking average for regression

Exercise

(1)

Data Point	X	Y	Label
1	2	3	A
2	3	4	A
3	5	6	B
4	7	8	B
5	10	10	A

Now, consider a new data point with $x1 = 6$ & $y1 = 7$
Using KNN with $k = 3$ predict label for this new data point



$\sqrt{(2-6)^2 + (3-7)^2}$
$\Rightarrow \sqrt{16+16} \Rightarrow \sqrt{32}$

$\sqrt{(3-6)^2 + (4-7)^2}$
$\Rightarrow \sqrt{9+9} \Rightarrow \sqrt{18} \approx$

$\sqrt{(5-6)^2 + (6-7)^2}$
$\Rightarrow \sqrt{1+1} \Rightarrow \sqrt{2} \approx$

$\sqrt{(7-6)^2 + (8-7)^2}$
$\Rightarrow \sqrt{1+1} \Rightarrow \sqrt{2} \approx$

$\sqrt{(10-6)^2 + (10-7)^2}$
$\Rightarrow \sqrt{(4)^2 + (3)^2} \Rightarrow \sqrt{16+9} = \sqrt{25}$

choose $k = 3$ min dist

predicted through majority voting
class = B

(2) consider a set of five training examples given as ((xi, yi), ci) values, where xi & yi are the attribute values (positive integers) and ci is the binary class label.

classify a test eg at coordn (3,6) using k-NN classifier with k=3, and manhattan dist defined by $d((u,v),(p,q)) = |u-p| + |v-q|$.

(a) ((1,1),-1) → (|1-3| + |1-6|) → |-2| + |-5| → 7

(b) ((1,7),+1) (|1-3| + |7-6|) → |-2| + |1| → (3)

(c) ((3,3),+1) (|3-3| + |3-6|) → |0| + |-3| → (3)

(d) ((5,4),-1) (|3-5| + |6-4|) → |-2| + |2| → 4

(e) ((2,5),-1) (|2-3| + |5-6|) → |-1| + |-1| → (2)

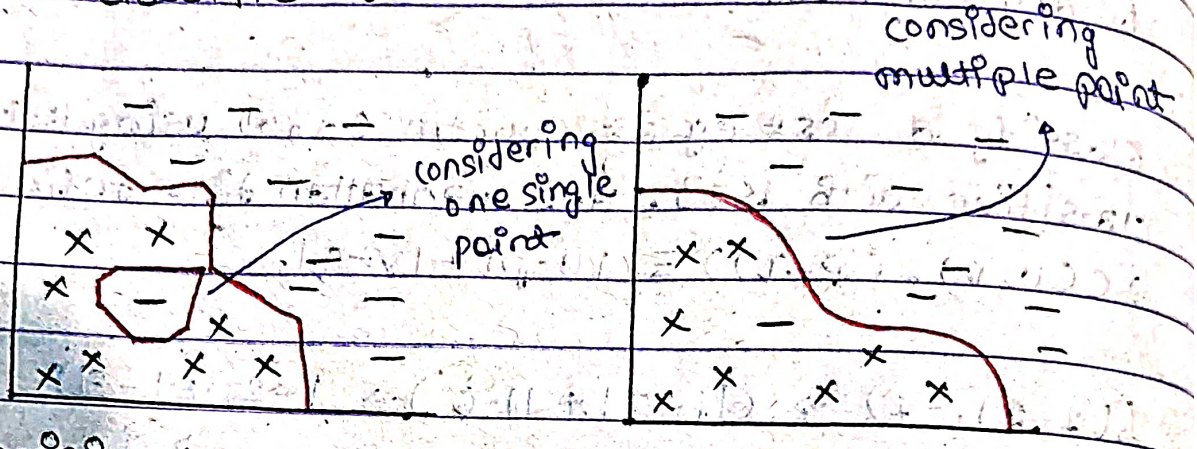
major voting → b, c, e

-1 +1 +1

predicted class label → +1

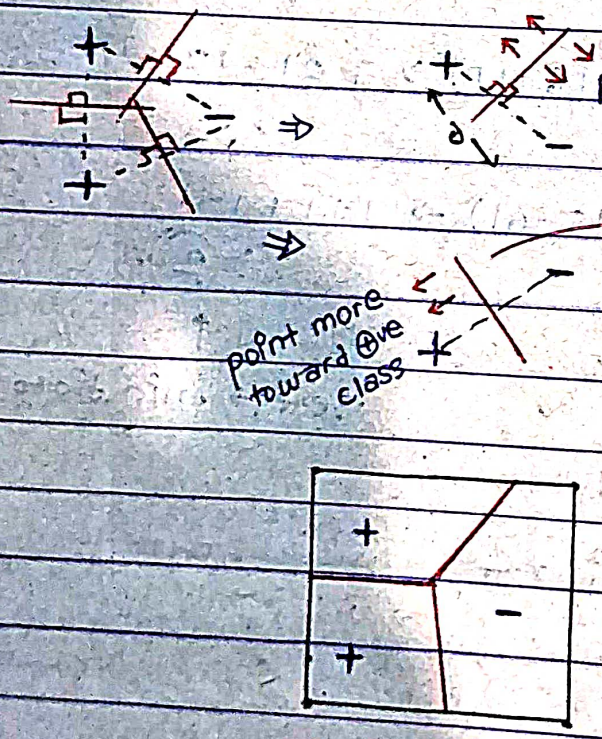
select min dist

(3) fig illustrates decision boundaries for two-nn classifiers. Determine which one of the boundaries belongs to 1-NN classifier & which one belong to 3-NN classifier ?

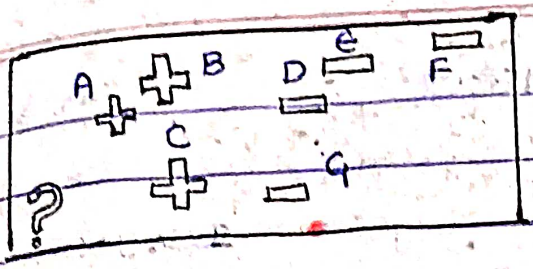


decision boundary 1 decision boundary 2

ie, 1-NN decision boundary
(using euclidean distance metric)



(4) Given following labelled dataset
For what (minimal) value of k will the query point "?" be negative.



1NN → NN → A → ⊕ve

2NN → NN → A, C → ⊕ve

3NN → NN → A, C, B → ⊕ve (max vote)

4NN → NN → A, C, B, D → ⊕ve (max vote)

5NN → NN → A, C, B, D, G → ⊕ve

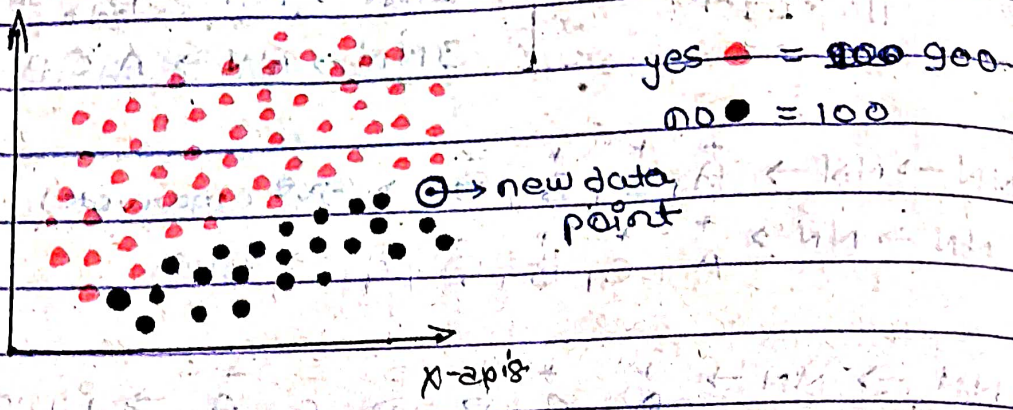
6NN → NN → A, C, B, D, G, E → ignore

7NN → NN → A, C, B, D, G, E, F → ⊖ve (max vote)

pros- ○ easily implementation
○ few hyperparmeters
(K → which is selected by validation)

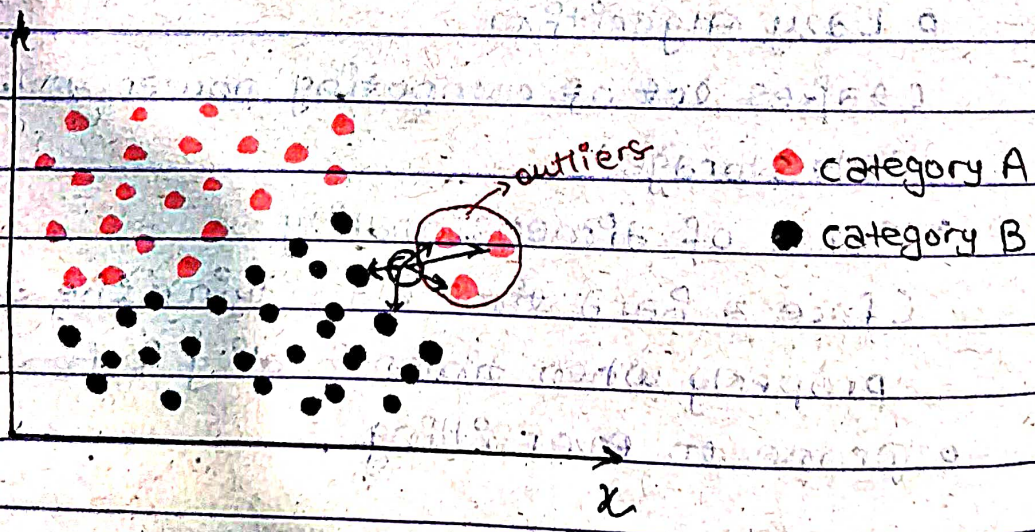
cons- ○ Lazy Algorithm
(takes lot of computing power as well as data storage)
○ Curse of dimensionality
(face a hard time classifying the data-points properly when dimensionality is too high)
○ prone to overfitting

- Impact of imbalanced dataset



→ the model will be biased
→ As a consequence, the buff of the closest
→ neighbours to this new point will be from "dominant class".
→ we have to balance our dataset using "upscaling" or "Downscaling" strategy.

- Impact of outliers on dataset.



→ new datapoint should belong to category B, but due to outliers it will have category A.