



# Machine Learning







Host

## K-Nearest Neighbors (K-NN)







[www.goclasses.in](http://www.goclasses.in)



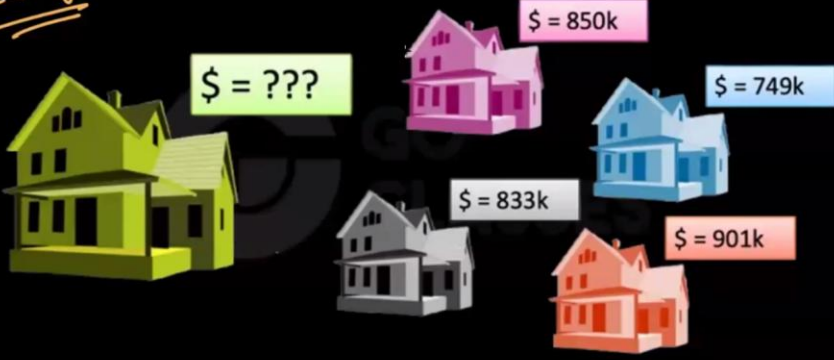
# Machine Learning



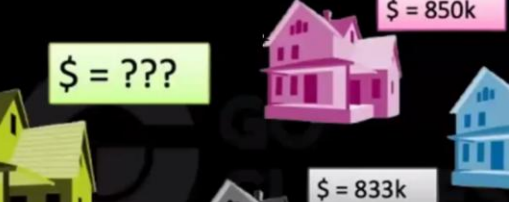


Host

Colony



House Color	Price
Yellow	\$ = ???
Pink	\$ = 850k
Blue	\$ = 749k
Grey	\$ = 833k
Orange	\$ = 901k



[www.goclasses.in](http://www.goclasses.in)

03:33 -1:31:41

GO CLASSES
Machine Learning
GO Classes

## 5-Nearest Neighbor (kNN) classifier

*Handwritten note: odd no.*

● Whales  
● Seals  
● Sharks

It will be whale

GO CLASSES
Machine Learning
GO Classes

## k-Nearest Neighbors (kNN)

**To predict category label  $y$  of a new point  $x$  (classification):**

- Find  $k$  nearest neighbors (according to some distance metric)
- Assign the majority label to the new point

**To predict numeric value  $y$  of a new point  $x$  (regression):**

- Find  $k$  nearest neighbors
- "Average" the values associated with the neighbors

**Note:** Changing  $k$  may result in a different prediction.

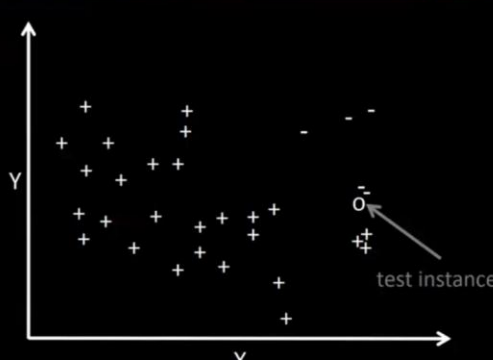
[www.goclasses.in](http://www.goclasses.in)

GO CLASSES

Machine Learning


GO Classes

Question:



(a) What would be the class assigned to this test instance for  $K=1$  [5 points]

Host



09:37
||
-1:25:37

GO CLASSES


Machine Learning

GO Classes

(a) What would be the class assigned to this test instance for  $K=1$  [5 points]



KNN assigns a test instance the target class associated with the majority of the test instance's  $K$  nearest neighbors. For  $K=1$ , this test instance would be predicted negative because it's single nearest neighbor is negative.

Host





09:51
||
-1:25:23

[https://ils.unc.edu/courses/2013\\_fall/inls613\\_001/INLS\\_613\\_midterm\\_Fall2013.pdf](https://ils.unc.edu/courses/2013_fall/inls613_001/INLS_613_midterm_Fall2013.pdf)



# Machine Learning




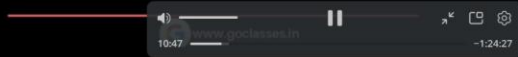


Host



(b) What would be the class assigned to this test instance for  $K=3$  [5 points]

KNN assigns a test instance the target class associated with the majority of the test instance's  $K$  nearest neighbors. For  $K=3$ , this test instance would be predicted negative. Out of its three nearest neighbors, two are negative and one is positive.







10:47 [www.goclasses.in](http://www.goclasses.in) -1:24:27



# Machine Learning





Host


## Distance Functions

► Euclidean Distance:

$$d_{\text{Euclidean}}(x, y) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

► Manhattan Distance:

$$d_{\text{Manhattan}}(x, y) = \sum_{i=1}^k |x_i - y_i|$$



17:03 -1:27:58

## Question:

Consider a set of five training examples given as  $((x_i, y_i), c_i)$  values, where  $x_i$  and  $y_i$  are the two attribute values (positive integers), and  $c_i$  is the binary class label. The training examples are listed in the table below:

Training Example (x, y)	Class Label (c)
(1, 1)	-1
(1, 7)	+1
(3, 3)	+1
(5, 4)	-1
(2, 5)	-1

Handwritten calculations for Manhattan distance from (3, 6) to training examples:

- $2+5=7$  (to (1, 1))
- $3+1=4$  (to (1, 7))
- $0+3=3$  (to (3, 3))
- $2+2=4$  (to (5, 4))
- $1+1=2$  (to (2, 5))

Classify a test example at coordinates (3, 6) using a  $k$ -Nearest Neighbors ( $k = NN$ ) classifier with  $k = 3$  and Manhattan distance defined by:

$$d((u, v), (p, q)) = |u - p| + |v - q|$$

Your answer should be either +1 or -1.

<https://napps.cs.wisc.edu/~dvr/cs540/exams/exam1-s18-sol.pdf>

Here nearest(lower) values are 2,3,3 with respective their(x,y) values which are +1,+1,-1 ( $k=3$ ) so +1 majority

+1 because the Manhattan distances from (3, 6) to each training example are:

- (1, 1): distance =  $2 + 5 = 7$
- (1, 7): distance =  $2 + 1 = 3$
- (3, 3): distance =  $0 + 3 = 3$
- (5, 4): distance =  $2 + 2 = 4$
- (2, 5): distance =  $1 + 1 = 2$

The 3 nearest neighbors are:

- (2, 5), -1
- (1, 7), +1
- (3, 3), +1

Since the majority class is +1, classify (3, 6) as class +1.

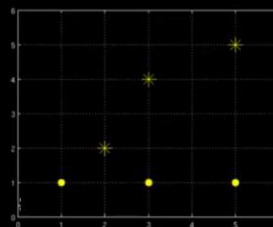
	train time cost	test time cost
① <u>knn</u>	$\sim 0$	high
② linear regression	high	$\hat{y} = wx + b$ quite low



[2 points] What is the training error (fraction labeled wrong) in the above picture using 1-nearest neighbor and the  $L_1$  distance norm between points? If there is more than one equally close nearest neighbor, use the majority label of all the equally close nearest neighbors. Please make sure you really are computing training error, not testing error.

- (a) 0
- (b) 1/6
- (c) 2/6
- (d) 3/6
- (e) 4/6

in 1NN TRAINING ERROR is always zero



★ SOLUTION: A



itself is shortest



in 1NN training error is Always zero.

→ overfit



(kNN) Consider a 2D dataset where training points are on the integer grid  $(x_1, x_2)$ , and both dimensions  $x_1$  and  $x_2$  range from 1 to 100 (inclusive). The binary label for point  $(x_1, x_2)$  is  $(-1)^{(x_1+x_2)}$ . What is the label for test point  $(51.3, 62.1)$  with a 3-NN classifier?

The 3NNs are  $(51, 62)$ ,  $(51, 63)$ ,  $(52, 62)$ . The labels of those are  $-1, 1, 1$ . So 3NN majority vote has label 1.



i. A  $k$ -nearest-neighbor classifier with  $k = 1$  will always have 100% training accuracy on this dataset.

**Solution:** True. For any training data point, the 1 nearest neighbour is always itself. Therefore, the predicted class label will by construction be correct. So the training accuracy is 100%.

ii. A  $k$ -nearest-neighbor classifier with  $k > 1$  will always have 100% training accuracy on this dataset.

**Solution:** False. The data points in the training dataset do not necessarily have the same label values. Some data points may have labels different from the particular training data point in consideration. When taking a majority vote of class labels from any point's  $k$  neighbours, the majority vote may differ from this point's true label. Therefore, no guarantee of 100% training accuracy.

iii. In general, using a  $k$ -nearest-neighbors classifier with  $k > 1$  as opposed to 1-nearest-neighbor can effectively reduce the tendency of the model to overfit to training data.

**Solution:** True. With  $k = 1$ , the classifier's decision boundary is heavily influenced by each individual data point. Any potential noise in the training data set is significantly impacting the decision boundary - classical symptom of over-fitting. When  $k$  increases, the decision boundary is smoothed out, meaning that some local noises are getting ignored, therefore higher  $k$  reduces the tendency to overfit to the local data.



Zero error==100 accuracy

Question:

You are using K-Nearest Neighbors (KNN) regression with  $K = 3$ . Below is the dataset of the nearest neighbors and their corresponding target values:

Handwritten note:  $q \rightarrow 1.2$  with an arrow pointing to the first row of the table.

Neighbor	Distance from $q$	Target Value
N1	1.2	15
N2	1.5	20
N3	1.7	25
N4	2.0	30
N5	2.5	35

Using simple averaging (mean) of the target values of the 3 nearest neighbors, what will be the predicted value for the new data point?

- ▶ A) 20
- ▶ B) 22
- ▶ C) 25
- ▶ D) 23

Handwritten calculation:  $15 +$



GO CLASSES

Machine Learning

GO Classes

Host

**Explanation:** To predict the value using KNN regression with  $K = 3$ , we take the mean of the target values of the 3 nearest neighbors (N1, N2, and N3):

$$\text{Predicted Value} = \frac{15 + 20 + 25}{3} = \frac{60}{3} = 20$$

Thus, the predicted value is 20.

53:06

www.goclasses.in

~42:08

GO CLASSES

Machine Learning

GO Classes

Host


Training Data

What does the classifier look like?

www.goclasses.in


1:00:03

~35:11




Machine Learning

GO Classes




Host




Training Data

What does the classifier look like?




1:00:31 -34:43

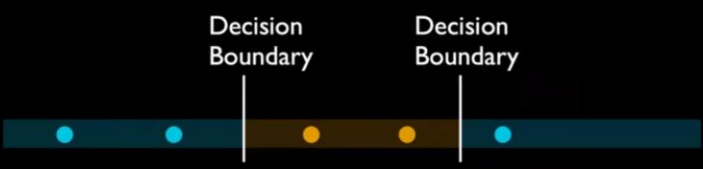


Machine Learning

GO Classes




Host



Decision Boundary

Decision Boundary

Training Data



1:00:40 -34:34

Machine Learning

GO Classes

Host

## The choice of K

1. What if we set  $K$  very large?
 

*underfit* ← Considering whole India

Top K-neighbors will include examples that are very far away...
2. What if we set  $K$  very small ( $K=1$ )?
 

label has noise (easily **overfit** to the noise)

(What about the training error when  $K = 1$ ?)

[www.goclasses.in](http://www.goclasses.in)

1:20:49

14:25

Machine Learning

GO Classes

Host

## Question:

In the image below, which would be the best value for  $k$ , assuming you are using the  $k$ -nearest neighbor algorithm?


- a. 3
- ☒ b. 10
- c. 20
- d. 50


model is becoming simpler

[https://www.cs.rhodes.edu/welshc/COMP345\\_F18/InClassActivity7.pdf](https://www.cs.rhodes.edu/welshc/COMP345_F18/InClassActivity7.pdf)

1:23:00

12:14

Machine LearningGO Classes

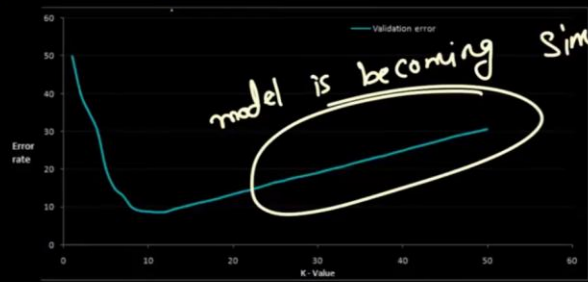


**Question:**

In the image below, which would be the best value for  $k$ , assuming you are using the  $k$ -nearest neighbor algorithm?

- a. 3
- ☒ b. 10
- c. 20
- d. 50


*K=10 gives lowest validation error*




[https://www.cs.rhodes.edu/welshc/COMP345\\_F18/InClassActivity7.pdf](https://www.cs.rhodes.edu/welshc/COMP345_F18/InClassActivity7.pdf)

1:23:16

11:58

Machine LearningGO Classes



**Question:**

Which of the following option is true about  $k$ -NN algorithm?

- A) It can be used for classification
- B) It can be used for regression
- ☒ C) It can be used in both classification and regression

1:25:15

09:59

GO CLASSES
Machine Learning
GO Classes

## The K-NN Algorithm

**Input:** classification training **dataset**  $\{x_i, y_i\}_{i=1}^n$ , and parameter  $K \in \mathbb{N}^+$ , and a **distance metric**  $d(x, x')$  (e.g.,  $\|x - x'\|_2$  euclidean distance)

**K-NN Algorithm:**

Store all training data

For any test point  $x$ :

Find its top K nearest neighbors (under metric  $d$ )

Return the most common label among these K neighbors

(If for regression, return the average value of the K neighbors)

take test point  
compare with  
ALL training  
points

GO CLASSES
Machine Learning
GO Classes

## K-Nearest Neighbor: Properties

- What's nice
  - Simple and intuitive; easily implementable
- What's not so nice..
  - Store all the training data in memory even at test time
    - Can be memory intensive for large training datasets
    - An example of non-parametric, or memory/instance-based methods
    - Different from parametric, model-based learning models
  - Expensive at test time:  $O(ND)$  computations for each test point
    - Have to search through all training data to find nearest neighbors
    - Distance computations with  $N$  training points ( $D$  features each)
  - Sensitive to noisy features

you have to have  
training data all  
time with you

$(x_1 - x_1)^2 + (x_2 - x_2)^2 + \dots$