Module: Machine Learning

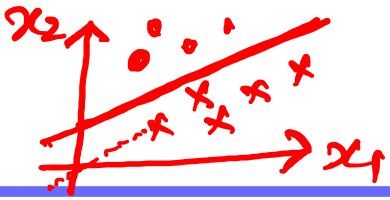
Live Session-4

Agenda:

Logistic Regression cont..

KNN Classifier

Logistic Regression: Binary Classes



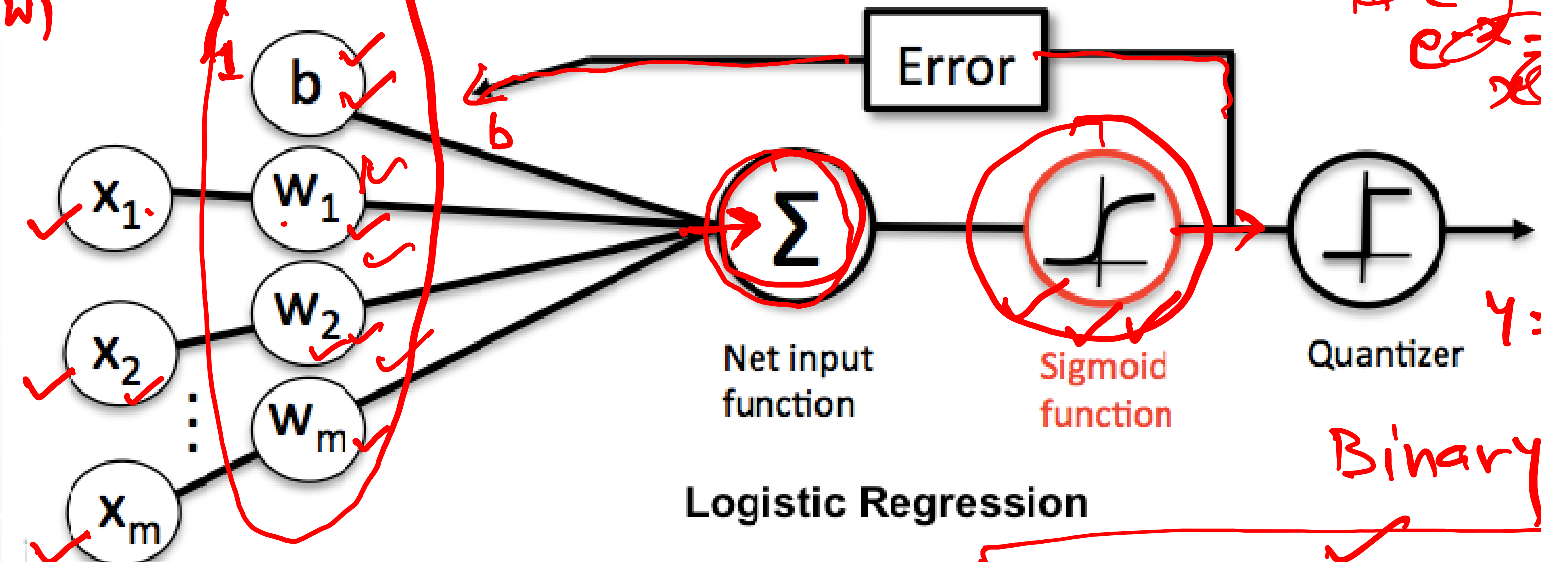
$$\frac{1}{1+e^{-w_1x_1+w_2x_2+b}}$$

$\frac{w_2}{w_1}$

$\frac{1}{1+e^{-x}}$ & ● 1

$e^{-x} = 0$

$x \to \infty$



Logistic Regression

$y = \frac{1}{1+e^{-x}}$

Binary class.

$$\boxed{w_1x_1+w_2x_2+\dots+w_mx_m+b}$$

Class = w^Tx+b Net input

S_1 $[x_{11}$ … $x_{m1}]$ ~~class~~

S_2 $[x_{21}$ -- $x_{m2}]$

⋮

S_n

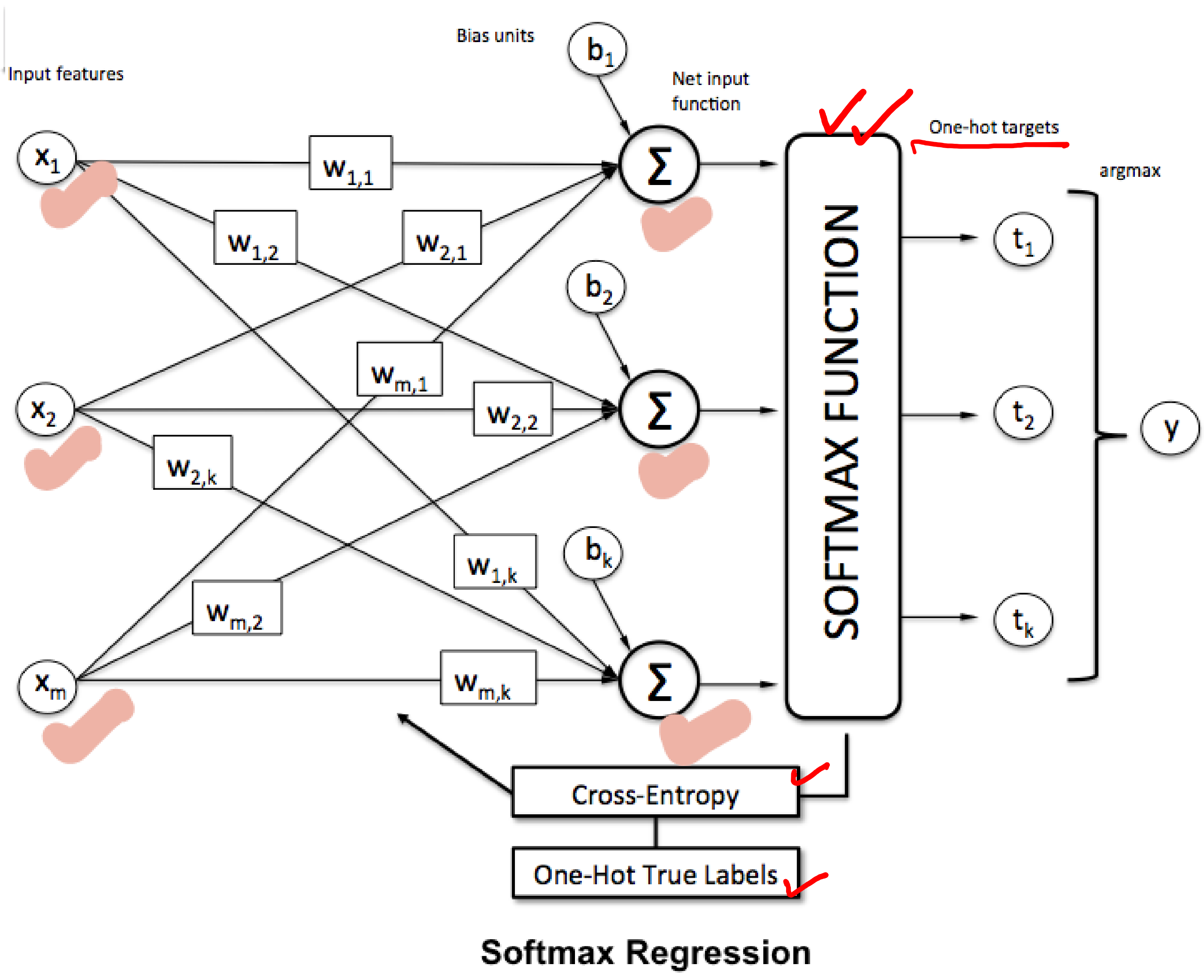
Gradient descent ⇒ $\frac{1}{1+e^{-(w^Tx+b)}}$

$\geq .5$ ✓

$< .5$ ✓

metric ⟹ $\boxed{\text{ROC-AUC}}$

Logistic Regression: Multiple classes



Softmax Regression

Softmax Reg.

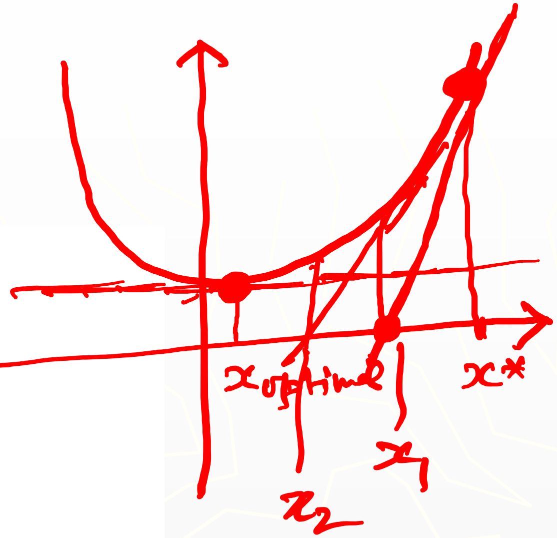
Multinomial Logistic Reg.

Logistic Regression: One-hot encoding & Softmax

One-hot-encoding

	Class	
p1	Class-0	[ ]
p2	Class-1	[
p3	Class-2	
p4	class-2	

Total=3



Gradient. Descent.

Logistic Regression: Pros

- The predicted parameters give inference about the importance of each feature.
- Performs well on low-dimensional data.
- Very efficient when the dataset has features that are linearly separable.
- Outputs well-calibrated probabilities along with classification results.

Logistic Regression: Cons

- Issue of Overfitting on high dimensional data.

- Non linear problems can't be solved with logistic regression since it has a linear decision surface.

- Fails to capture complex relationships.

- Only important and relevant features should be used otherwise model's predictive value will degrade.

$\int \quad \frac{1}{1+e^{-(w^T x+b)}}$

$w^T x + b$

$> .5$

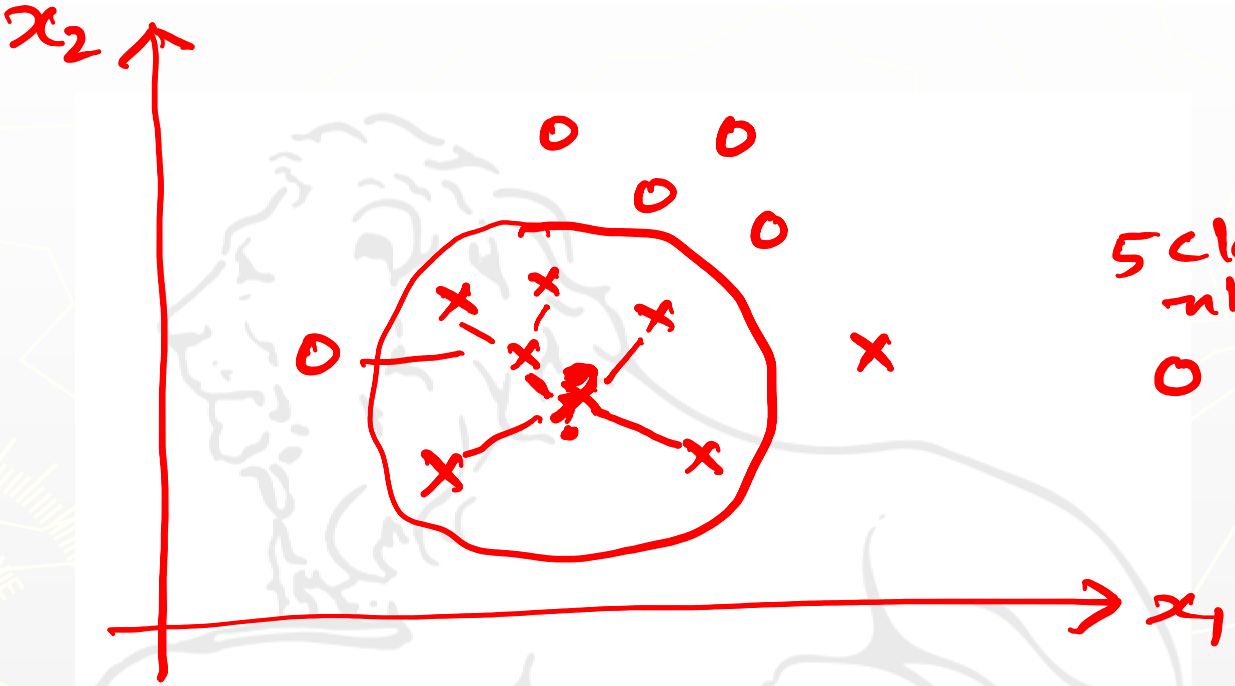
$\leq .5$

$0 - 1$

K-Nearest Neighbour: Idea

KNN

Lazy classifn



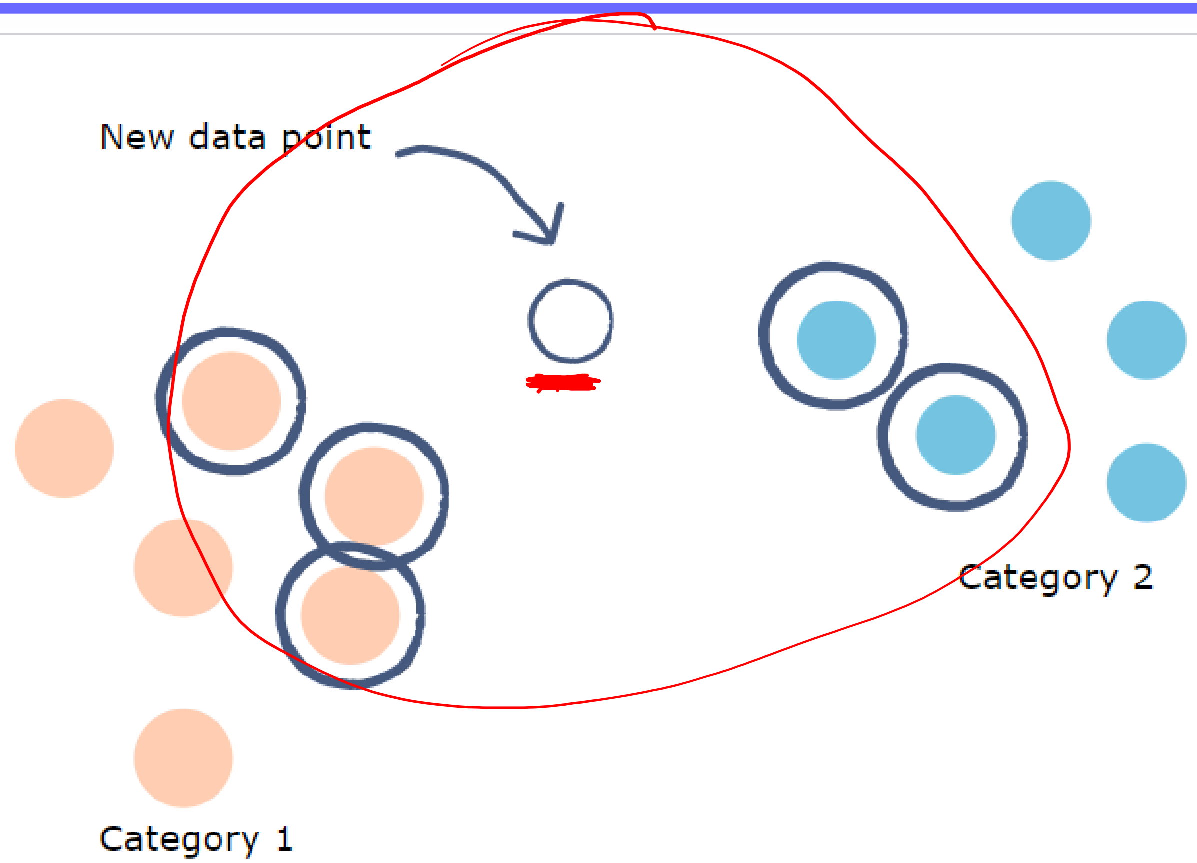
K=5

5 closest nhbr.

o ×

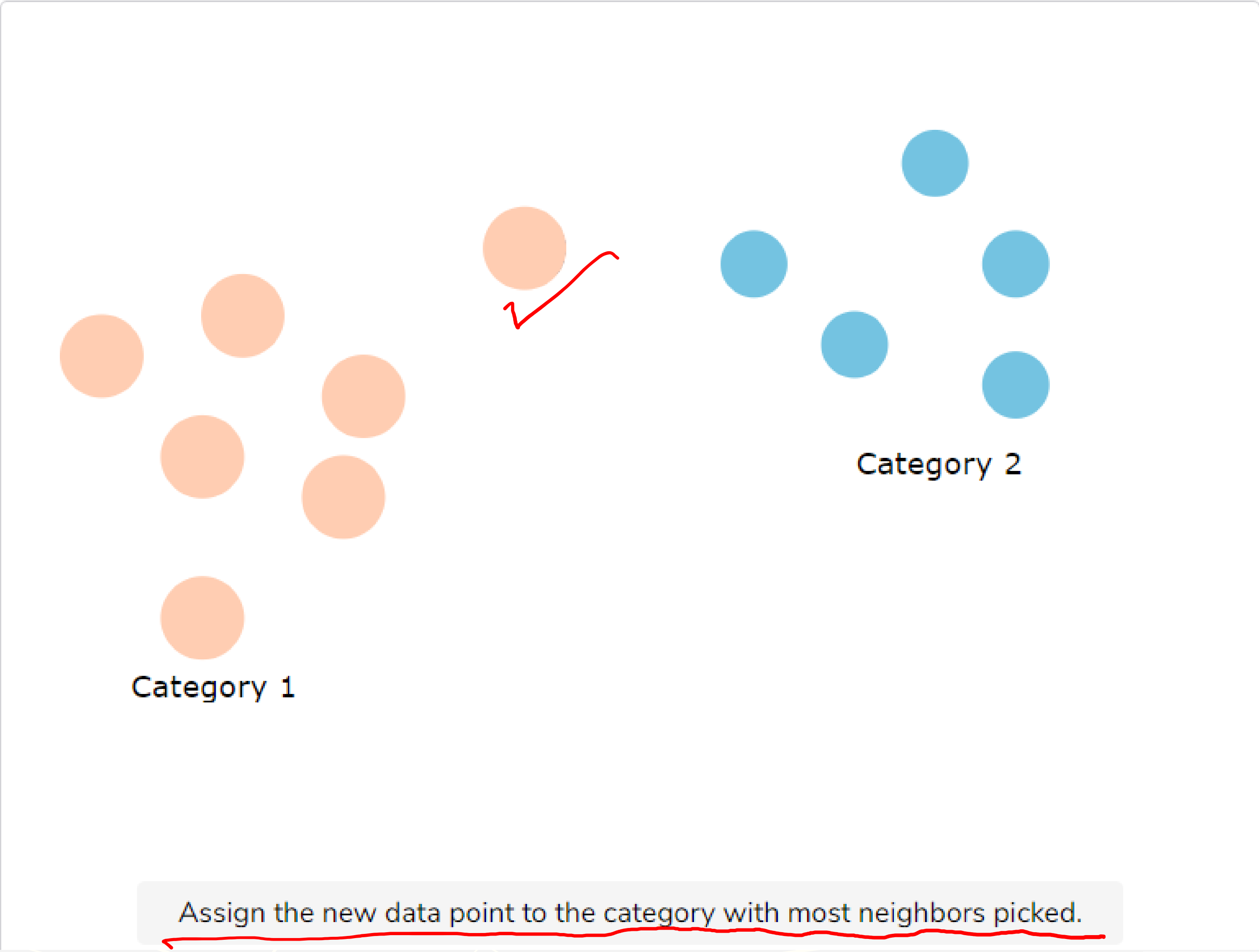
$$\hat{y}(x) = y_{n^*} \text{ where } n^* = \arg\min_{n \in D} dist(x, x_n)$$

KNN



Take the K nearest neighbors of the new data point based on their Euclidean distance.

KNN



Assign the new data point to the category with most neighbors picked.

Algorithm:

Step 1: Determine the value for K   Set

Step 2: Calculate the distances between the new input (test data) and all the training data. The most commonly used metrics for calculating distance are Euclidean, Manhattan and Minkowski

Step 3: Sort the distance and determine k nearest neighbors based on minimum distance values

Step 4: Analyze the category of those neighbors and assign the category for the test data based on majority vote

Step 5: Return the predicted class

Distance Metric in KNN

Let A and B are represented by feature vectors

$$A = (x_1, x_2, \dots, x_n) \text{ and } B = (y_1, y_2, \dots, y_n),$$

where n is the dimensionality of the feature space. To calculate the distance between A and B, the Euclidean metric is generally used by

$$D_e(A,B) = \sqrt{\sum(x_i - y_i)^\wedge 2}$$

$(\underline{1}, \underline{2})$, $(\underline{2}, \underline{3})$ $De = \sqrt{(1-2)^2 + (2-3)^2} = \sqrt{2}$

To calculate the distance between A and B, the Manhattan metric is generally used by

$$D_m(A,B) = \sum(|x_i - y_i|)$$

$= |1-2| + |2-3| = 2$

$D_p(A,B) = \left(\sum (x_i - y_i)^p\right)^{1/p}$

KNN metrics

We choose the distance function according to the types of data we're handling.

So for quantitative data (example: weight, wages, size, shopping cart amount, etc.) of the same type, Euclidean distance is a good candidate.

Taxicab geometry (Manhattan) is a good measure to use when the data (input variables) are not of the same type (example: age, sex, length, weight, etc.).

Choosing the value of k:

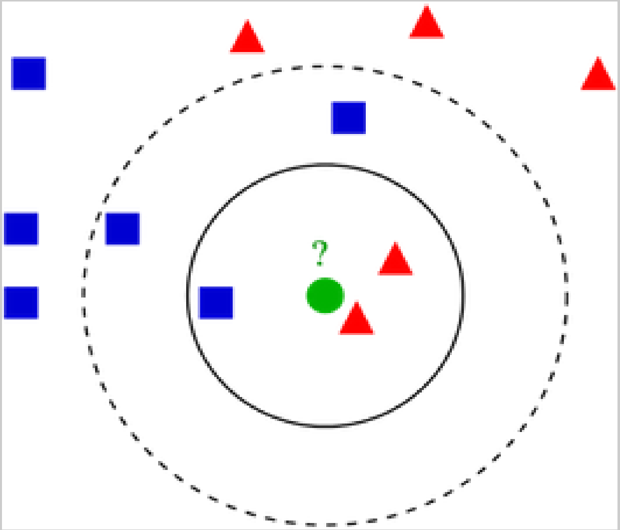
- Square root of the total number of observations
- Trial and Run

Shows

K=3
K=5

K=
100 data pts.
10

In the case of a small number of neighbors, the noise will have a higher influence on the result, and a large number of neighbors make it computationally expensive.



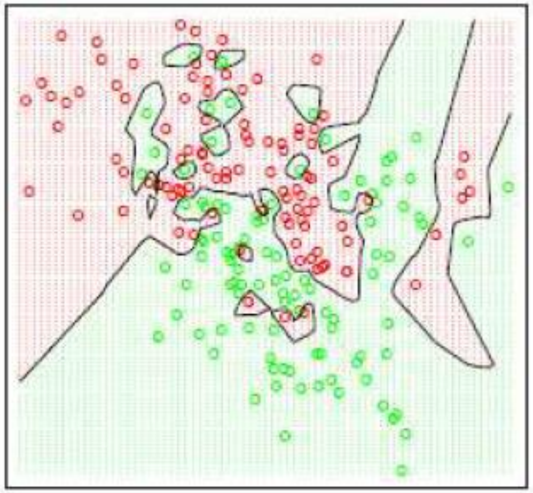
Example of *k*-NN classification. The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If *k* = 3 (solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If *k* = 5 (dashed line circle) it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle).

Effect of K:

- K yields smoother predictions, since we average over more data
- K=1 yields y=piecewise constant labeling
- K = N predicts y=globally constant (majority) label

K=1

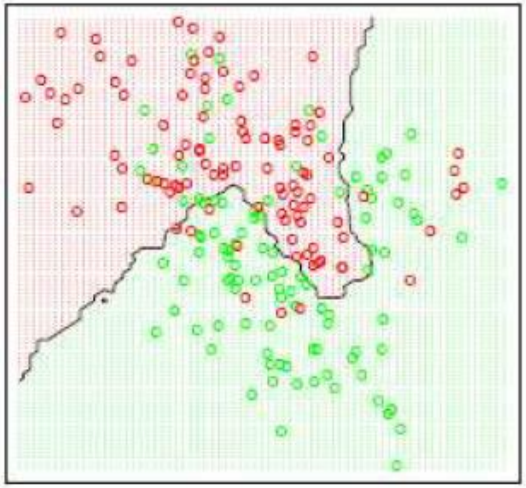
1-Nearest Neighbor Classifier



K=15

✓

15-Nearest Neighbor Classifier



Positive Aspects:

No Training Period: KNN is called **Lazy Learner (Instance based learning)**. It does not learn anything in the training period. It does not derive any discriminative function from the training data.

w, b

Since the KNN algorithm requires no training before making predictions, **new data can be added seamlessly** which will not impact the accuracy of the algorithm.

Online classifi

KNN is very **easy to implement**. There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc.)

Negative Aspects:

- Can be slow to find nearest nbr in high dim space ✓
- Need to store all the training data, so takes a lot of memory ✓
- Need to specify the distance function
- Does not give probabilistic output