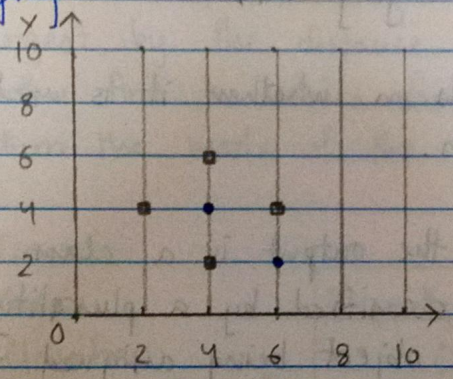
Page:
Date:

ASSIGNMENT A3

Title: k-NN Classification

Problem Statement: In the following diagram, let blue circles indicate positive examples and orange (black) squares indicate negative examples. We want to use the k-NN algorithm for classifying the points. If k=3, find the class of the point (6,6). Extend the same example for Distance-Weighted k-NN and locally weighted averaging.



Objectives: To learn how to apply kNN classification to classify positive and negative points

Outcomes: Students will be able to implement kNN classification on positive and negative points, and extend the same for Distance-Weighted k-NN and locally weighted averaging

Requirements: python3, jupyter, pandas, numpy, sklearn

Theory:
K-nearest neighbors (KNN) algorithm is a type of supervised Machine learning algorithm which can be used for both classification and regression predictive problems.

Page:
Date: / /

However, it is mainly used for classification predictive problems in the industry. The following properties define KNN well:

- Lazy learning algorithm - It does not have a specialized training phase and uses all the data for training while classification

- Non-parametric learning algorithm - It doesn't assume anything about the underlying data

The output of k-NN depends on whether it is used for classification or regression.

- In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k-nearest neighbors (k is a positive integer, typically small). If k=1, the object is simply assigned to the class of the single nearest neighbor.

- In k-NN regression, the output is the property value of the object. The value is the average of the values of k-nearest neighbors.

Page:
Date: / /

Algorithm

1. Load the data
2. Initialize k to chosen number of neighbors (in this case, 3)
3. For each example in the data.
   (a) Calculate the distance between the query example and the current example from the data
   (b) Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest by the distance.
5. Pick the first k entries from the sorted collection
   - Return the mode of the k labels

Test Cases

Test Case	Expected output	Actual Output
KNN Classifier: Test data: [(6,2),(6,4)]	0, 1	1, 1
Weighted KNN Classifier: Test data: [(6,2),(6,4)]	0, 1	0, 0
KNN Classifier: Test data: [(6,6)]	0	0 or 1
Weighted KNN Classifier: Test data: [(6,6)]	0	0

Conclusion: Succesfully implemented KNN Classifier (weighted and non-weighted) and predicted values of given test data