**K-Nearest Neighbors (KNN)**

**INTUITION**

|  |  |
| --- | --- |
| Sketch    Hyper-Parameter: k | Requires 3 things to make a judgement:   1. The set of stored records 2. Distance Metric (How to define "nearest" mathematically) 3. The value of , the number of nearest neighbors to retrieve.   COMMENT on Hyper-Parameter   * If is too small, sensitive to noise points. * If is too large, neighborhood may include points from other classes.   **Explain**: In Fig.(a) the model targeted ONE neighbor of the under-recognizing data point . While in Fig.(b) the model targeted TWO and THREE in Fig.(c) correspondingly. |

**COMMENTS**

* Simplicity and ease of implementation.
* Does not require any assumptions about the underlying data distribution.
* Able to also handle both numerical and categorical data.
* Less sensitive to outliers(异常值) compared to other algorithms

**IMPLEMENTATION**

Distance Metrics

* Euclidean Distance
* Manhattan Distance
* Minkowski Distance

**WORKING of KNN**

Step-1: Selecting the optimal value of

Step-2: Calculate the distance

Step-3: Finding Nearest Neighbors

Step-4: Voting for Classification or Taking Average for Regression

Explaination of STEP-4

* In the **CLASSIFICATION** problem, the class labels of K-nearest neighbors are determined by performing majority voting. The class with the most occurrences among the neighbors becomes the predicted class for the target data point.
* In the **REGRESSION** problem, the class label is calculated by taking average of the target values of K nearest neighbors. The calculated average value becomes the predicted output for the target data point.