KNNs Algorithm

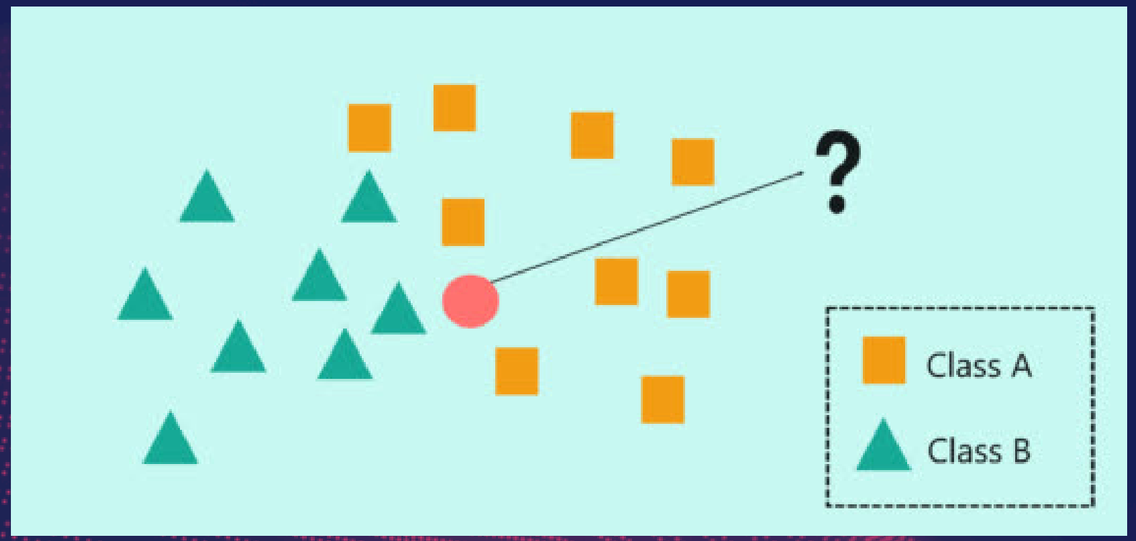
KNNs -- K NEAREST NEIGHBORS

- It was formalized in 1967 by Cover and Hart, and was the begining of basic pattern recognition.
- KNN is an supervised learning algorithm used in both classification and regression.
- It is a simple, easy to implement and understand algorithm

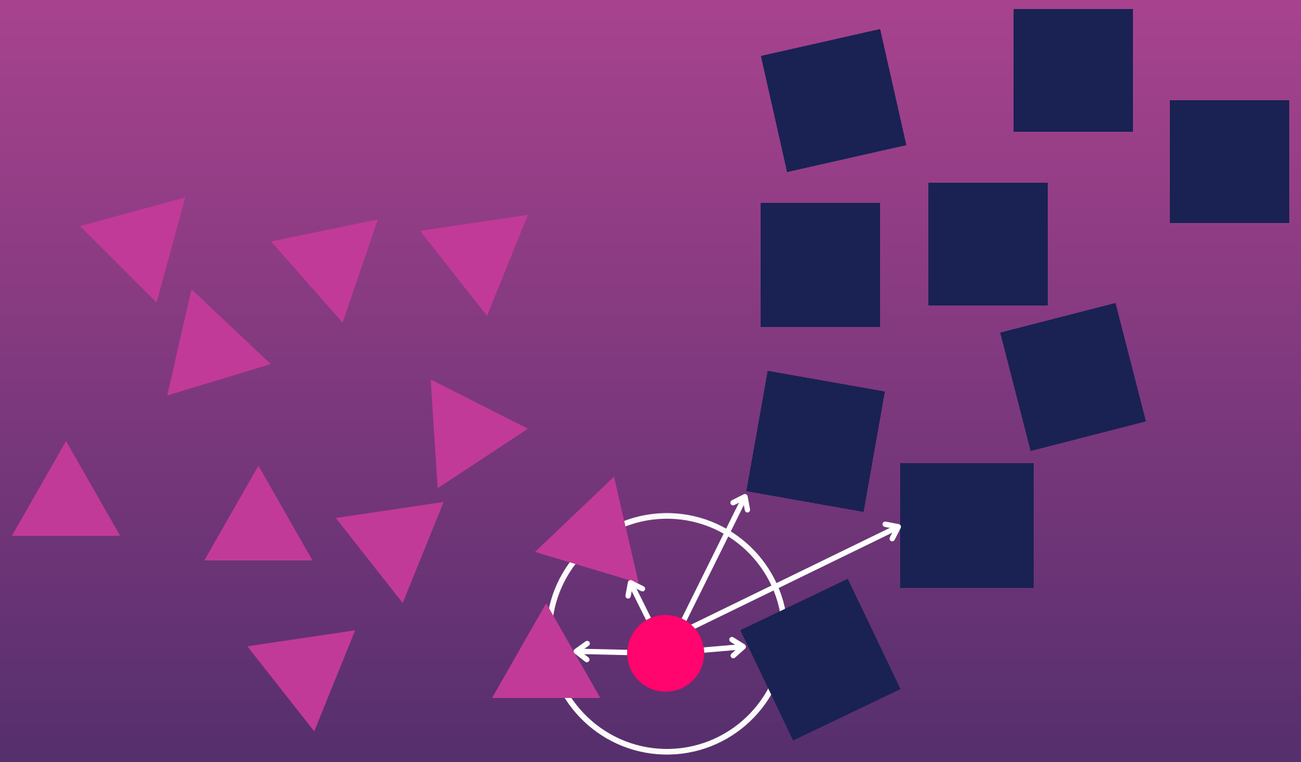
- As the name suggests, it considers the k nearest data points and classifies the test data point in the majority class.

- Its working principle - “ Birds of a feather flock together”

WORKING:



- In this algorithm the majority class label among its ‘k’ nearest datapoints in the feature space determines the class label of a new data point.
- It is like a voting system, except the voters are only the neighbors
- Intuitively, it classifies based on the neighbors because it believes datapoints belonging to the same class lie near each other.



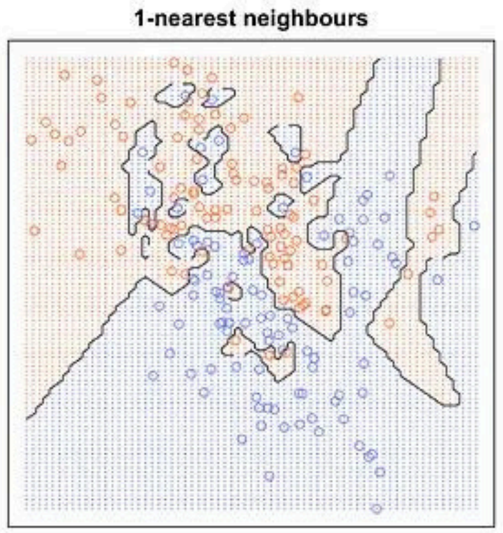
- But for a regression task, it takes the average value of the target values of the K nearest neighbors and that becomes the predicted output .

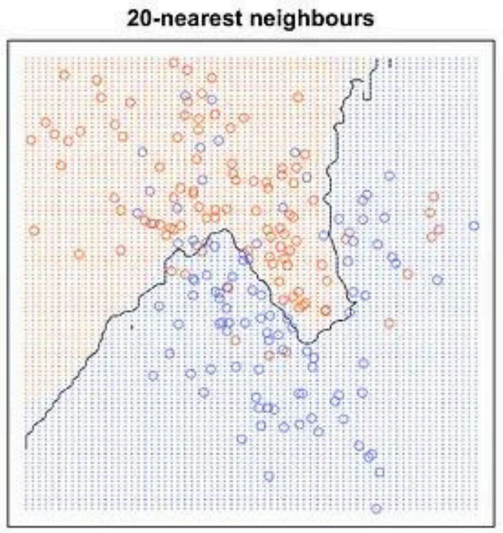
TUNING THE HYPERPARAMETERS:

- The hyperparameters… are ‘K’ and the distance.

‘K” :

- ‘k’ is the number of nearest neighbors we consider.
- The parameter ‘k’ can be played around with and the best one can be chosen.
- A rule of thumb is to choose k as the sqrt(n), where n is the number of datapoints.
- Usually odd values of k are preferred to avoid ties in classification,





Higher value of k:

ADVANTAGES:

- Robust to noise and outliers: By considering a large number of datapoints, we reduce the possibility of only including a few outliers of one class, thus making the prediction less affected by random variations in the data.

DISADVANTAGES:

- A simple model will have a smooth decision boundary and it'll make overly generalized predictions , failing to capture intricate details and potentially overlooking important patterns in the data.

Smaller value of k :

ADVANTAGES:

- The decision boundary becomes more complex and captures every intricate detail and pattern, but it comes at the risk of overfitting.

DISADVANTAGES:

- A small value of k can lead to overfitting, as the model will be too sensitive to noise in the training data and capture too much of the local variation, resulting in poor generalization to unseen data.

Distance Metric

- What do we mean by nearest?
- Why are there different distance metrics?
- What are the different distance metrics?
- Which one to use?

- There are different distance metrics used to measure the closeness or similarity between two datapoints

- Reasons being:

1. Data Characteristics
2. Nature of the problem
3. Dimensionality

- Different distance metrics capture different aspects of the data and are chosen based on the above reasons.

What are the different distance metrics?

$$Minkowski = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$$

- Manhattan Distance **p = 1**

$$\text{Manhattan Distance in n-dimensions} = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_N - y_N| = \sum_{i=1}^{N} |x_i - y_i|$$

- Euclidean Distance **p = 2**

$$Euclidean = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- Chebyshev Distance

$$Chebyshev = max(|x_i - y_i|)$$

p is very large or tends to inf

- These are few distance metrics and there are many more!
- ‘i’ here goes from 1 to n , where n is the number of features or dimensions

→ Minkowski distance

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

when $p \to \infty$

$$\left(\ldots (0.05)^p + 1^p + 32^p + 15^p + \ldots \right)^{1/p}$$

$p \to \infty$, say $p = 1000$ then $\to 0$

finite quantity

(a little smaller) than 32^p

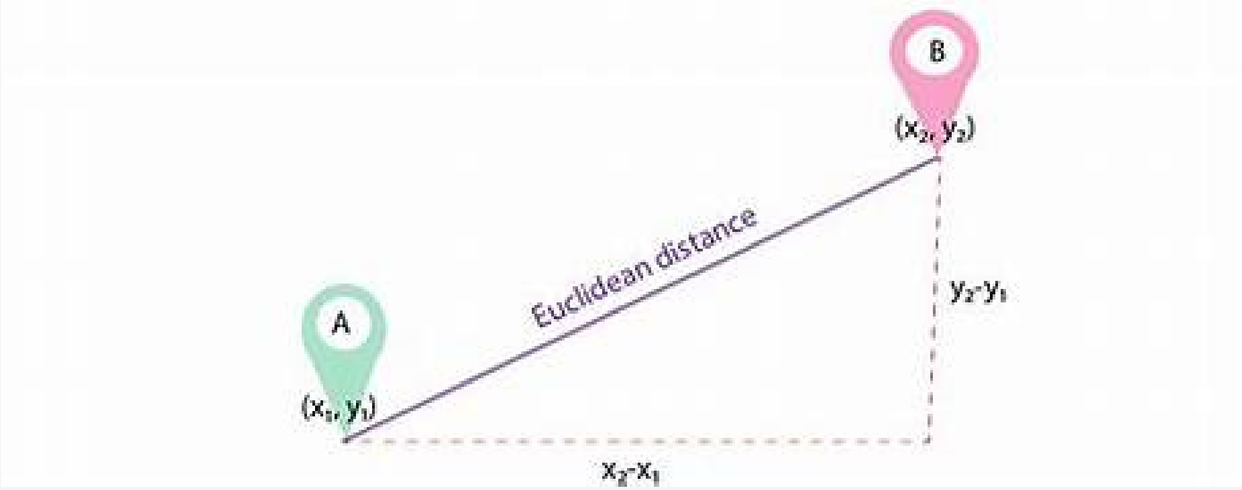
now $32^{1000} >> 15^{1000}$, $\therefore$ $32^{1000} + 15^{1000} \approx 32^{1000}$ itself

$$\therefore \text{Summation} = \lim_{p \to \infty} (32^p)^{1/p} = \underbrace{32}$$

or in general the max difference among all dimensions.

$$\therefore \text{Summation} = \lim_{p \to \infty} \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} = \max\left(|x_i - y_i|\right)$$

i goes from 1 to n



Some other distance measures:

- Hamming Distance - used for categorical data (strings)
- Cosine similarity
- Jaccard similarity

Features of KNNs

INSTANCE BASED LEARNING

- Entire Instance of dataset is used to predict output, there is no seperate training.

NON PARAMETRIC

- No preconceived notions about the functional form of the hypotheses (mapping function) or number of parameters or the data distribution.
- The principle “closeness implies similarity” is just a guiding principle to make a prediction, not a assumption of the data

LAZY LEARNING

- The model doesn’t explicitly ‘learn’ anything from the training data but uses it to make pred on the test data

Disadvantages and Drawbacks of KNNs

01

The curse of dimensionality

When the number of features increase then we have a higher dimension space, this can degrade the performance of the KNN as all the points tend to become equidistant from each other.

02

What if there is class imbalance??

Yeah, if there is class imbalance or too many outliers then KNN will fail to perform.

03

Space complexity and Time complexity

We measure many distances and all these have to be stored and compared against to find the nearest. This can increase the space and time complexity of the algo, especially when dealing with large datasets.

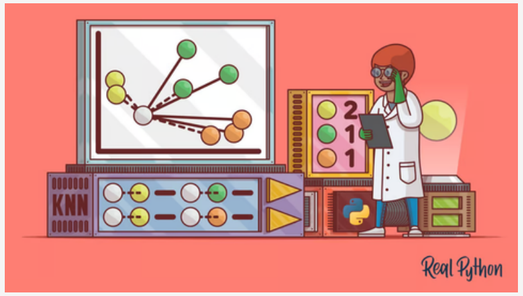
QUESTION :

In K-Nearest Neighbors (KNN), how does increasing the value of k affect the bias and variance of the model?

A. Increases bias and decreases variance.
B. Decreases bias and increases variance.
C. Increases both bias and variance.
D. Decreases both bias and variance.

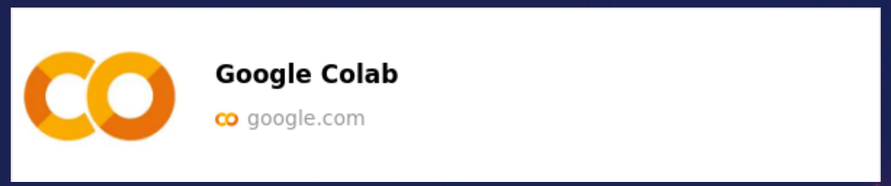
3377078

IMPLEMENTATION



1. Load the dataset, split it into test and train and scale if req
2. Provide ‘k’ (or a set of k values), the distance metric and weights if any.
3. Provide input test data
4. The model will calculate all distances between feature vectors and selects the k nearest neighbors.
5. Determines class as the majority class of the k nearest neighbors or regression value as the avg or weighted avg of the target values of the knns.

CODE IMPLEMENTATION



ATTENDANCE