K-Nearest Neighbors (KNN) algorithm

An algorithm which finds the nearest neighbors

Vaibhav Jayaswal Aug 25, 2020 · 6 min read ★

Image by Joshua J. Cotten, from Unsplash

Table of Contents:

1. What is KNN?
2. Working of KNN algorithm
3. What happens when K changes?
4. How to select appropriate K?
5. Limitation of KNN
6. Real-world application of KNN
7. Conclusion

1. What is KNN?

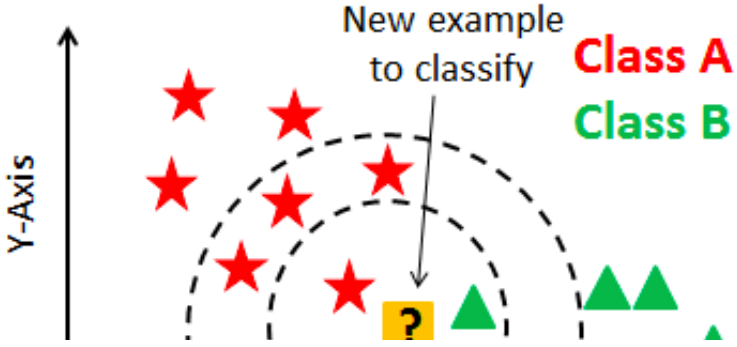
K nearest neighbors (KNN) is a supervised machine learning

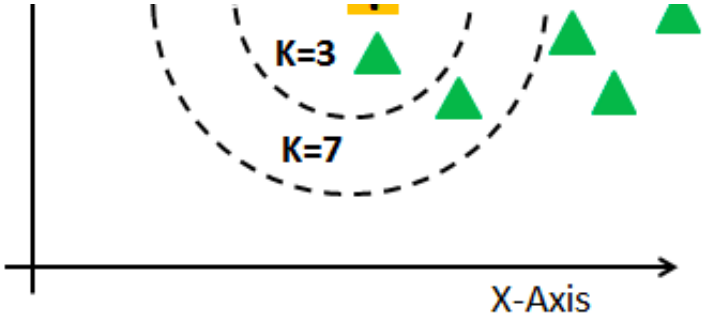
algorithm. A supervised machine learning algorithm's goal is to learn a function such that f(X) = Y where X is the input, and Y is the output. KNN can be used both for classification as well as regression. In this article, we will only talk about classification. Although for regression, there is just a minute change.

The properties of KNN is that it is a lazy learning algorithm and a non-parametric method.

Lazy learning means the algorithm takes almost zero time to learn because it only stores the data of the training part (no learning of a function). The stored data will then be used for the evaluation of a new query point.

The non-parametric method refers to a method that does not assume any distribution. Therefore, KNN does not have to find any parameter for the distribution. While in the parametric method, the model finds new parameters, which in turn will be used for the prediction purpose. The only hyperparameter (provided by the user to the model) KNN has is K, which is the number of points that needs to be considered for comparison purpose.





K-Nearest Neighbors. Source

In the above image, yellow is the query point, and we want to know which class it belongs to (red or green).

With K=3, the 3 nearest neighbors of the yellow point are considered, and the class is assigned to the query point based on the majority (e.g., 2 green and 1 red — then it is of green class). Similarly, for K=5, 5 nearest neighbors are considered for the comparison, and the majority will decide which class the query point belongs to. One thing to notice here, if the value of K is even, it might create problems when taking a majority vote because the data has an even number of classes (i.e., 2). Therefore, choose K as an odd number when the data has an even number of classes and even number when the data has an odd number of classes.

2. Working of KNN algorithm

In the training phase, the model will store the data points. In the testing phase, the distance from the query point to the points from the training phase is calculated to classify each point in the test dataset. Various distances can be calculated, but the most popular one is the Euclidean distance (for smaller

dimension data).

Euclidean distance between a query point (q) and a training data point (p) is defined as

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

Euclidean distance between point p and q (n-dimensional points). Source

Other distance measures such as Manhattan, Hamming, and Chebyshev distance can also be used based on the data, which is out of the scope of this article.

Let's learn it with an example:

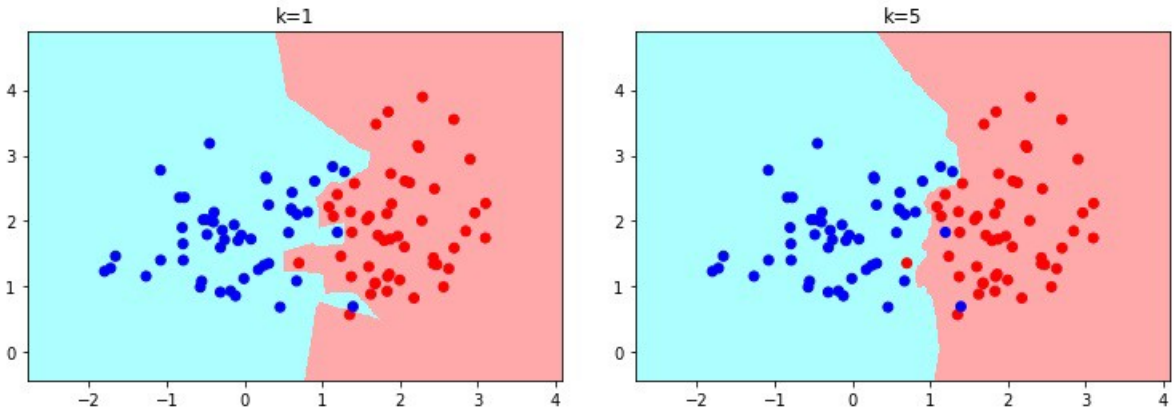
We have 500 N-dimensional points, with 300 being class 0 and 200 being class 1.

The procedure for calculating the class of query point is:

1. The distance of all the 500 points is calculated from the query point.
2. Based on the value of K, K nearest neighbors are used for the comparison purpose.
3. Let's say K=7, 4 out of 7 points are of class 0, and 3 are of

class 1. Then based on the majority, the query point p is assigned as class 0.

3. What happens when K changes?



Decision Surface separating the red and blue class with k=1 (left) and k=5 (right). Image by author

K=1 means that it will take one nearest neighbor and classify the query point based on that. The surface that divides the classes will be very uneven (many vertices).

The problem that arises here is if an outlier is present in the data, the decision surface considers that as a data point. Due to this, KNN will perform exceptionally well on the training dataset but will misclassify many points on the test dataset (unseen data). This is considered as overfitting, and therefore, KNN is sensitive to outliers.

As the value of K increases, the surface becomes smooth and will not consider the outliers as data points. This will better generalize the model on the test dataset also.

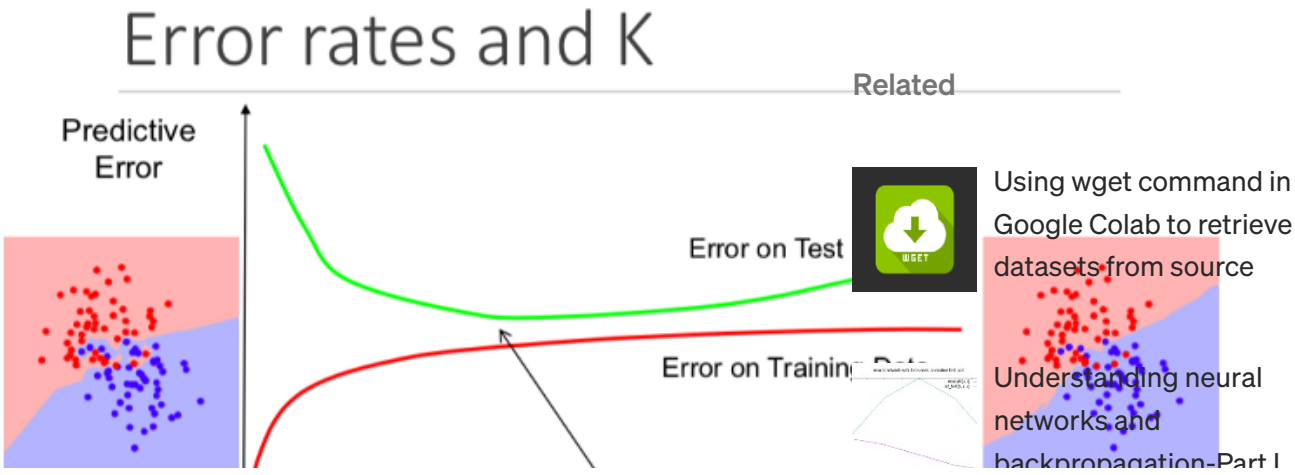
If K value is extremely large, the model will underfit and will be unable to classify the new data point. For example, if K is equal to the total number of data points, no matter where the query point lies, the model will always classify the query point based on the majority class of the whole dataset.

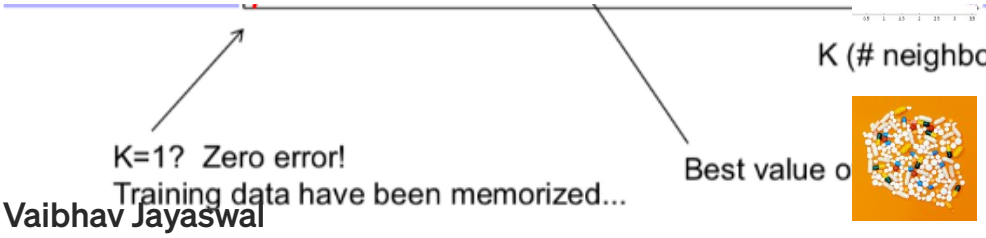
Choosing a correct value K will give accurate results. But how to choose that?

4. How to select appropriate K?

In real-world problems, the dataset is separated into three parts, namely, training, validation, and test data. In KNN, the training data points get stored, and no learning is performed. Validation data is to check the model performance, and the test data is used for prediction.

To select optimal K, plot the error of model (error = 1 — accuracy) on training as well as on the validation dataset. The best K is where the validation error is lowest, and both training and validation errors are close to each other.





Error rate and K. Source

Vaibhav Jayaswal

A machine learning enthusiast. I have pursued my masters from Indian Institute of Technology, Bombay.

5. Limitation of KNN

The time complexity and space complexity is enormous, which is a major disadvantage of KNN. Time complexity refers to the time model takes to evaluate the class of the query point. Space complexity refers to the total memory used by the algorithm. If we have n data points in training and each point is of m dimension. Then time complexity is of order O(nm), which will be huge if we have higher dimension data. Therefore, KNN is not suitable for high dimensional data.

Another disadvantage is if the data point is far away from the classes present (no similarity), KNN will classify the point even if it is an outlier. In order to overcome the problem of time complexity, algorithms such as KD-Tree and Locality Sensitive Hashing (LSH) can be used, which is not covered in this article.

6. Real-world application of KNN

1. KNN can be used for Recommendation Systems. Although in the real world, more sophisticated algorithms are used for the recommendation system. KNN is not suitable for high dimensional data, but KNN is an excellent baseline approach for the systems. Many companies make a

personalized recommendation for its consumers, such as Netflix, Amazon, YouTube, and many more.

2. KNN can search for semantically similar documents. Each document is considered as a vector. If documents are close to each other, that means the documents contain identical topics.
3. KNN can be effectively used in detecting outliers. One such example is Credit Card fraud detection.

7. Conclusion

K- Nearest Neighbors (KNN) identifies the nearest neighbors given the value of K. It is lazy learning and non-parametric algorithm. KNN works on low dimension dataset while faces problems when dealing with high dimensional data.

Knn Nearest Neighbors Real World Examples Knn