**Modern Education Society’s**

**College of Engineering, Pune-01**

|  |
| --- |
| **NAME OF STUDENT: CLASS:** |
| **SEMESTER/YEAR: ROLL NO:** |
| **DATE OF PERFORMANCE: DATE OF SUBMISSION:** |
| **EXAMINED BY: EXPERIMENT NO:** |

# TITLE: ASSIGNMENT ON DIABETES PREDICATION SYSTEM WITH KNN ALGORITHM

**Problem Statement**:

Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusion

matrix, accuracy, error rate, precision and recall on the given dataset.

Dataset link : https://www.kaggle.com/datasets/abdallamahgoub/diabetes

**Objectives:**

* Understand and evaluate unsupervised machine learning algorithms.
* Analyze performance of an algorithm.

**Pre-requisites:**

1. Knowledge of python programming.
2. Knowledge of Data Pre-processing.
3. Knowledge of evaluation metrics of unsupervised algorithms.

**Description:**

K-Nearest Neighbor(KNN) Algorithm for Machine Learning

* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
* It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
* Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



## Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



## How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

* Step-1: Select the number K of the neighbors
* Step-2: Calculate the Euclidean distance of K number of neighbors
* Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
* Step-4: Among these k neighbors, count the number of the data points in each category.
* Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
* Step-6: Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:



* Firstly, we will choose the number of neighbors, so we will choose the k=5.
* Next, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



* By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



* As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.
* Hence, what k-means does is to miminize the objective function:



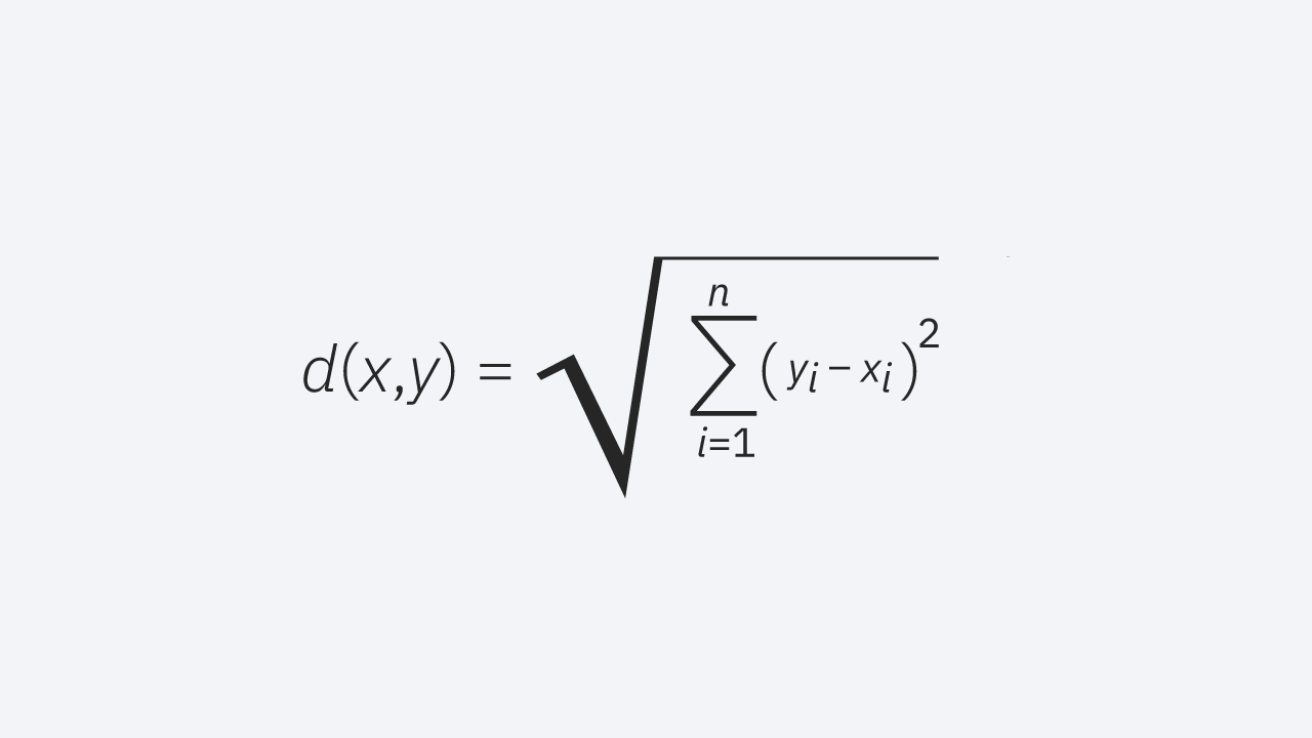
Distance metric for KNN:

the goal of the k-nearest neighbor algorithm is to identify the nearest neighbors of a given query point, so that we can assign a class label to that point. In order to do this, KNN has a few requirements:

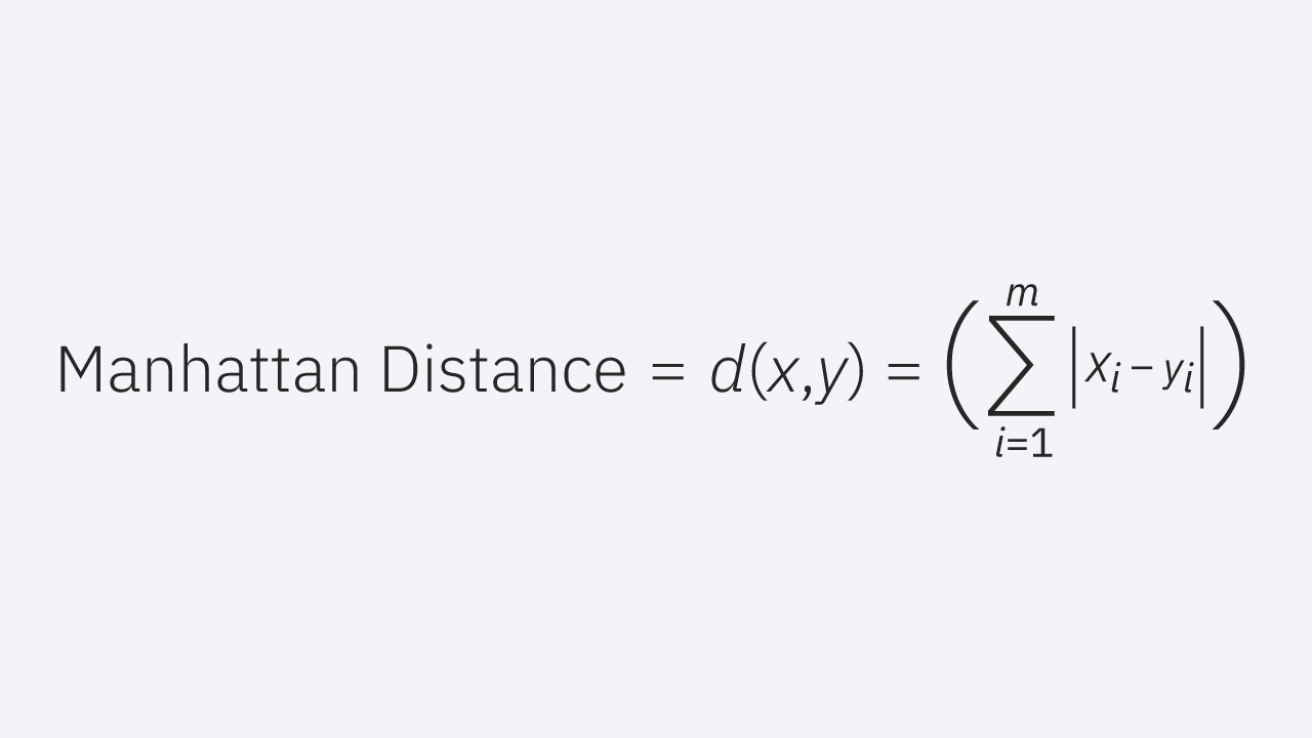
Determine your distance metrics

In order to determine which data points are closest to a given query point, the distance between the query point and the other data points will need to be calculated. These distance metrics help to form decision boundaries, which partitions query points into different regions. You commonly will see decision boundaries visualized with Voronoi diagrams.  
  
While there are several distance measures that you can choose from, this article will only cover the following:

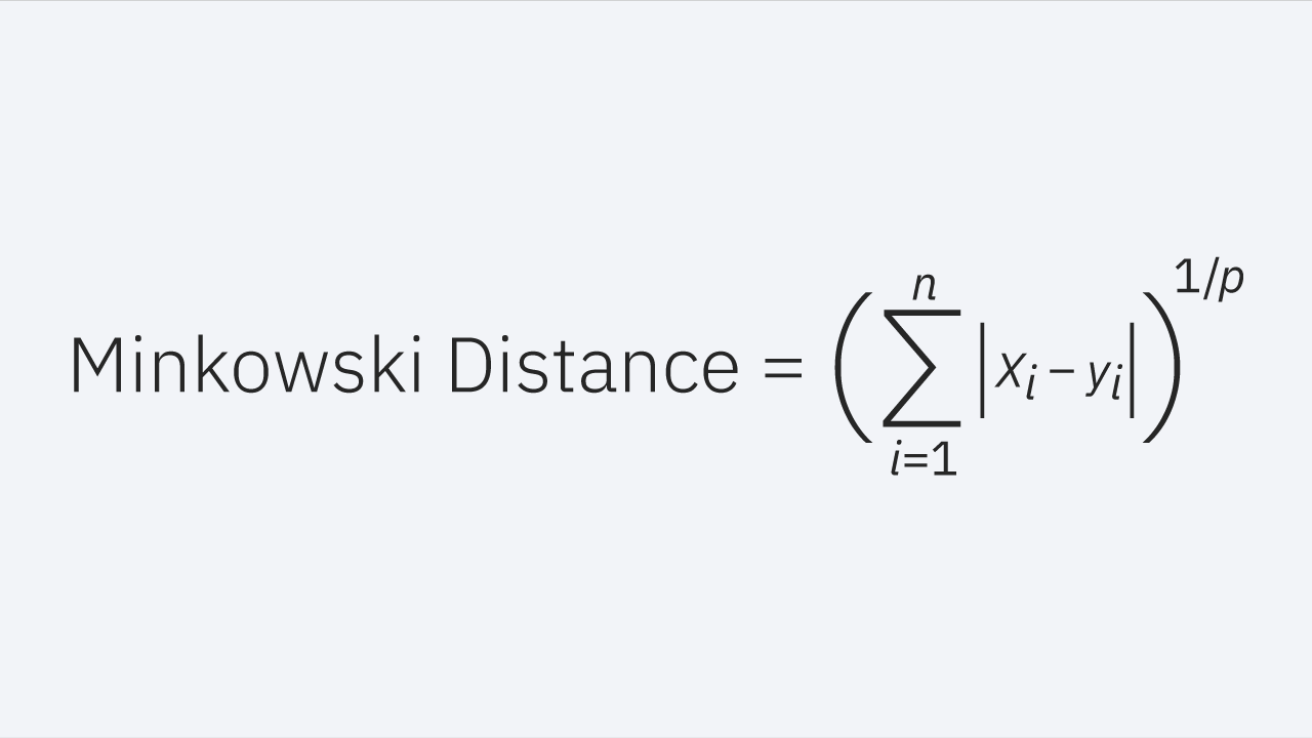
Euclidean distance (p=2): This is the most commonly used distance measure, and it is limited to real-valued vectors. Using the below formula, it measures a straight line between the query point and the other point being measured.



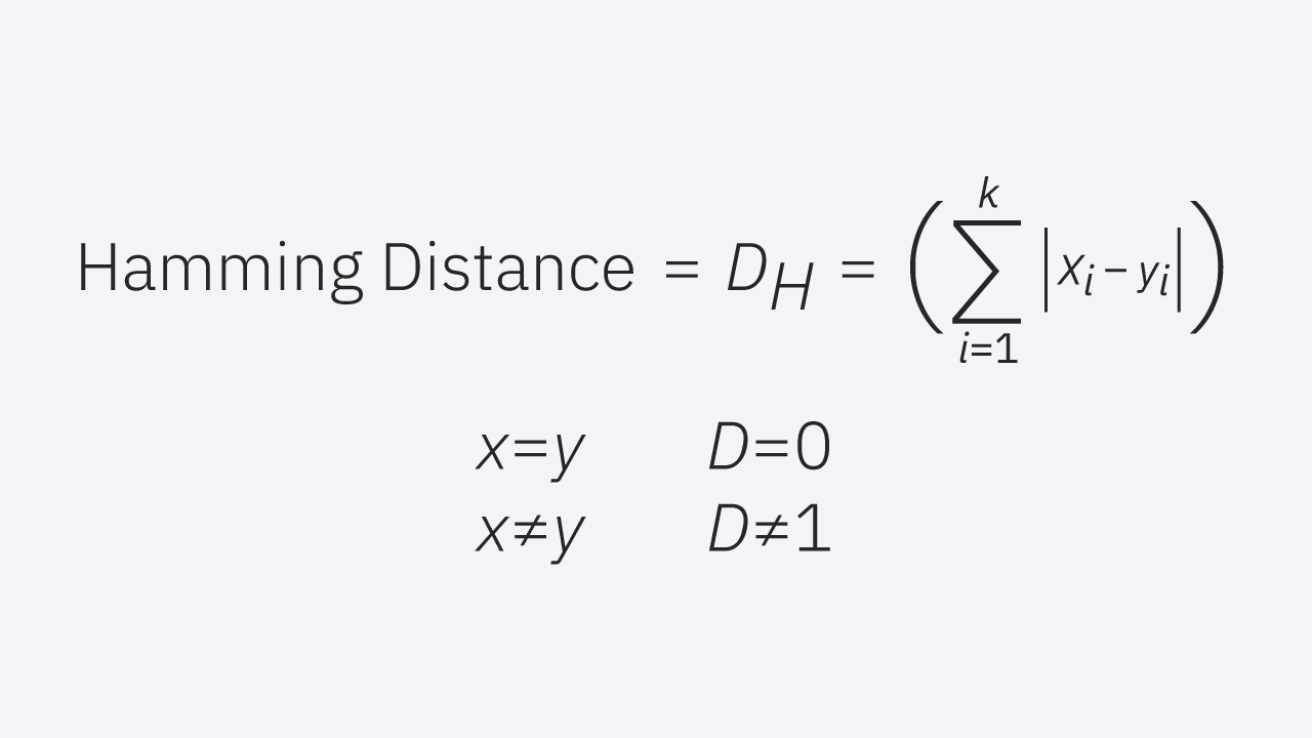
Manhattan distance (p=1): This is also another popular distance metric, which measures the absolute value between two points. It is also referred to as taxicab distance or city block distance as it is commonly visualized with a grid, illustrating how one might navigate from one address to another via city streets.



Minkowski distance: This distance measure is the generalized form of Euclidean and Manhattan distance metrics. The parameter, p, in the formula below, allows for the creation of other distance metrics. Euclidean distance is represented by this formula when p is equal to two, and Manhattan distance is denoted with p equal to one.



Hamming distance: This technique is used typically used with Boolean or string vectors, identifying the points where the vectors do not match. As a result, it has also been referred to as the overlap metric. This can be represented with the following formula:

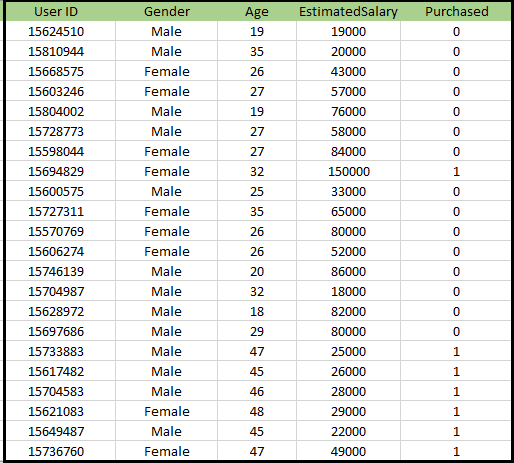


## How to select the value of K in the K-NN Algorithm?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

* There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
* A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
* Large values for K are good, but it may find some difficulties.

Problem for K-NN Algorithm: There is a Car manufacturer company that has manufactured a new SUV car. The company wants to give the ads to the users who are interested in buying that SUV. So for this problem, we have a dataset that contains multiple user's information through the social network. The dataset contains lots of information but the Estimated Salary and Age we will consider for the independent variable and the Purchased variable is for the dependent variable. Below is the dataset:



Steps to implement the K-NN algorithm:

* Data Pre-processing step
* Fitting the K-NN algorithm to the Training set
* Predicting the test result
* Test accuracy of the result(Creation of Confusion matrix)
* Visualizing the test set result.

1. Data Pre-Processing Step:

#importing libraries

#importing datasets

#Extracting Independent and dependent Variable

#Splitting the dataset into training and test set.

#feature Scaling

2. Fitting K-NN classifier to the Training data:

Now we will fit the K-NN classifier to the training data. To do this we will import the KNeighborsClassifier class of Sklearn Neighbors library. After importing the class, we will create the Classifier object of the class. The Parameter of this class will be

* n\_neighbors: To define the required neighbors of the algorithm. Usually, it takes 5.
* metric='minkowski': This is the default parameter and it decides the distance between the points.
* p=2: It is equivalent to the standard Euclidean metric.

And then we will fit the classifier to the training data.

3. Predicting the Test Result: To predict the test set result, we will create a y\_pred vector.

4. Creating the Confusion Matrix:  
Now we will create the Confusion Matrix for our K-NN model to see the accuracy of the classifier.

## 5. Visualizing the Training set result: Now, we will visualize the training set result for K-NN model.

6. Visualizing the Test set result:  
After the training of the model, we will now test the result by putting a new dataset, i.e., Test dataset. Code remains the same except some minor changes: such as x\_train and y\_train will be replaced by x\_test and y\_test.

## Applications of k-NN in machine learning

The k-NN algorithm has been utilized within a variety of applications, largely within classification. Some of these use cases include:

- Data preprocessing: Datasets frequently have missing values, but the KNN algorithm can estimate for those values in a process known as missing data imputation.

- Recommendation Engines: Using clickstream data from websites, the KNN algorithm has been used to provide automatic recommendations to users on additional content. However, given the scaling issues with KNN, this approach may not be optimal for larger datasets.

- Finance: It has also been used in a variety of finance and economic use cases. For example, one using KNN on credit data can help banks assess risk of a loan to an organization or individual. It is used to determine the credit-worthiness of a loan applicant. It is used in stock market forecasting, currency exchange rates, trading futures, and money laundering analyses.

- Healthcare: KNN has also had application within the healthcare industry, making predictions on the risk of heart attacks and prostate cancer. The algorithm works by calculating the most likely gene expressions.

- Pattern Recognition: KNN has also assisted in identifying patterns, such as in text and digit classification. This has been particularly helpful in identifying handwritten numbers that you might find on forms or mailing envelopes.

## Advantages of KNN Algorithm:

* It is simple to implement.
* It is robust to the noisy training data
* It can be more effective if the training data is large.

## Disadvantages of KNN Algorithm:

* Always needs to determine the value of K which may be complex some time.
* The computation cost is high because of calculating the distance between the data points for all the training samples.

Questions:

1. Is Feature Scaling required for the KNN Algorithm? Explain with proper justification.

2. How can you relate KNN Algorithm to the Bias-Variance tradeoff?

3. The k-NN algorithm does more computation on test time rather than train time.