

Example 1:

KNN using IRIS Dataset

Loading data

```
data(iris)
```

```
iris
```

```
# Structure
```

```
str(iris)
```

training and testing data set

```
train = iris[1:120,-5]
```

```
test = iris[121:150,-5]
```

```
library(class)
```

creating the model

```
model = knn(train,test,iris[1:120,5],k=6)
```

```
summary(model)
```

Confusion Matrix

```
cm = table(iris[121:150,5],model) # table(actual,predicted)
```

```
cm
```

```
accuracy = sum(diag(cm))/length(iris[121:150,5])
```

```
accuracy
```

```
sprintf("Accuracy: %.2f%%", accuracy*100)
```

Example 2:

study a Cancer dataset and build a Machine Learning model that predicts whether a patient

#can be diagnosed as Malignant or Benign.

#Logic: This problem statement can be solved using the **KNN algorithm** that will classify the diagnosis into two classes:

1. Malignant, M

#2. Benign, B

```
install.packages("class")
```

```
library(class)
```

#adding the dataset

```
wdbc <- read.csv("cancer_dataset.csv",header=T)
```

```
#wdbc <- read.csv(file.choose(),header=T)
```

#Data Cleaning

#Removing the first column ,id , which is unnecessary

```
wdbc<-wdbc[,-1]
```

#Normalize the data

#The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values

```
data_norm <-function(x){ (x-min(x))/(max(x)-min(x))}
```

```
wdbc_norm <- data.frame(lapply(wdbc[,-1],data_norm))
```

```
summary(wdbc[,2:5])
```

```
summary(wdbc_norm[,1:4])
```

#Creating Training and Testing dataset

```
wdbc_train <- wdbc_norm[1:450,]
```

```
wdbc_test <- wdbc_norm[451:569,]
```

#Applying KNN model

#The knn () function needs to be used to train a model for which we need to install a package 'class'. The knn() function identifies the k-nearest neighbours using Euclidean distance where k is a user-specified number.

#The value for k is generally chosen as the square root of the number of observations.

#knn() returns a factor value of predicted labels for each of the examples in the test data set which is then assigned to the data frame.

```
wdbc_pred <- knn(wdbc_train,wdbc_test,wdbc[1:450,1],k=21)
```

```
wdbc_pred
```

#confusion matrix or frequency table

#Table function in R table(), performs categorical tabulation of data with the variable and its frequency.c

```
cm = table(wdbc[451:569,1],wdbc_pred)
```

```
*****
```