MACHINE LEARNING

FINAL ASSIGNMENT

WRITE UP

By,
Ajayvishagan P
(110120004,ICE DEP)
Ram Ganesh K.R
(110120090,ICE DEPT)

Problem Statement:

Applying Clustering algorithm(KNN algorithm) in our dataset(water_potability)

Approach Adopted:

KNN Algoritham:

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.

Working of KNN Algoritham

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.

Advantages of KNN Algorithm:

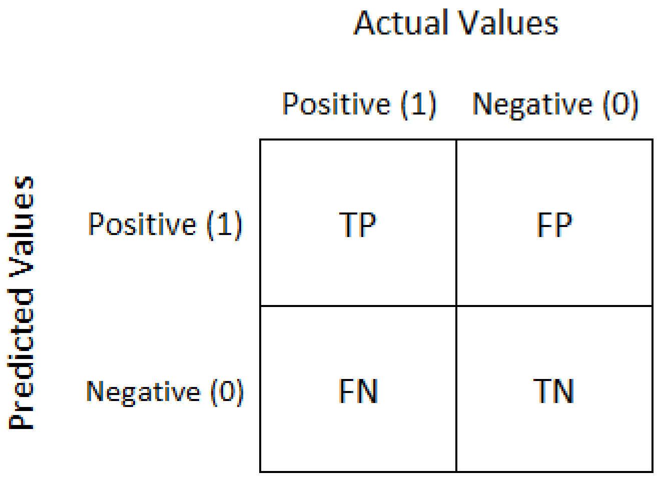
- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

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.

Confusion matrix -

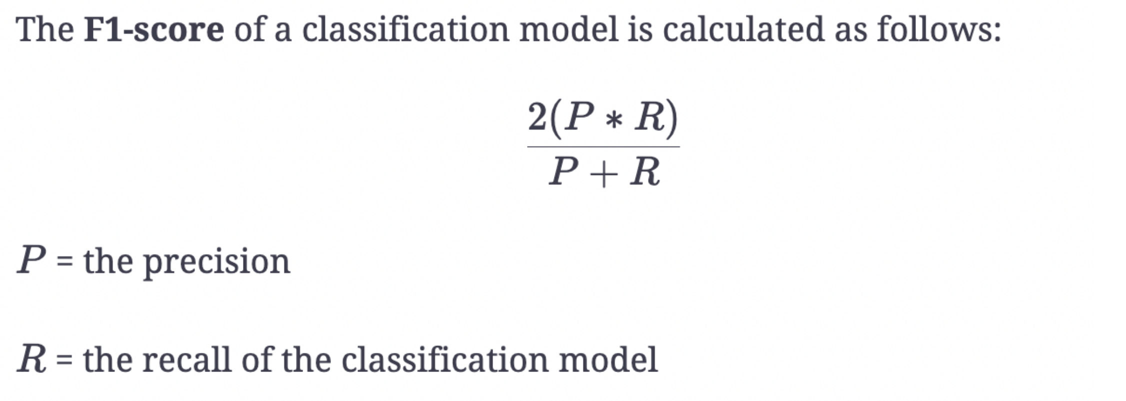
It is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

F1 score:

The **F1-score** combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of two classifiers. Suppose that classifier A has a higher recall, and classifier B has higher precision.



The **F1-score** of a classification model is calculated as follows:

$$\frac{2(P * R)}{P + R}$$

P = the precision

R = the recall of the classification model

Accuracy Score:

Model accuracy is a machine learning model performance metric that is defined as the ratio of true positives and true negatives to all positive and negative observations. In other words, accuracy tells us how often we can expect our machine learning model will correctly predict an outcome out of the total number of times it made predictions. For example: Let’s assume that you were testing your machine learning model with a dataset of 100 records and that your machine learning model predicted all 90 of those instances correctly. The accuracy metric, in this case, would be: (90/100) = 90%. The accuracy rate is great but it doesn’t tell us anything about the errors our machine learning models make on new data we haven’t seen before.

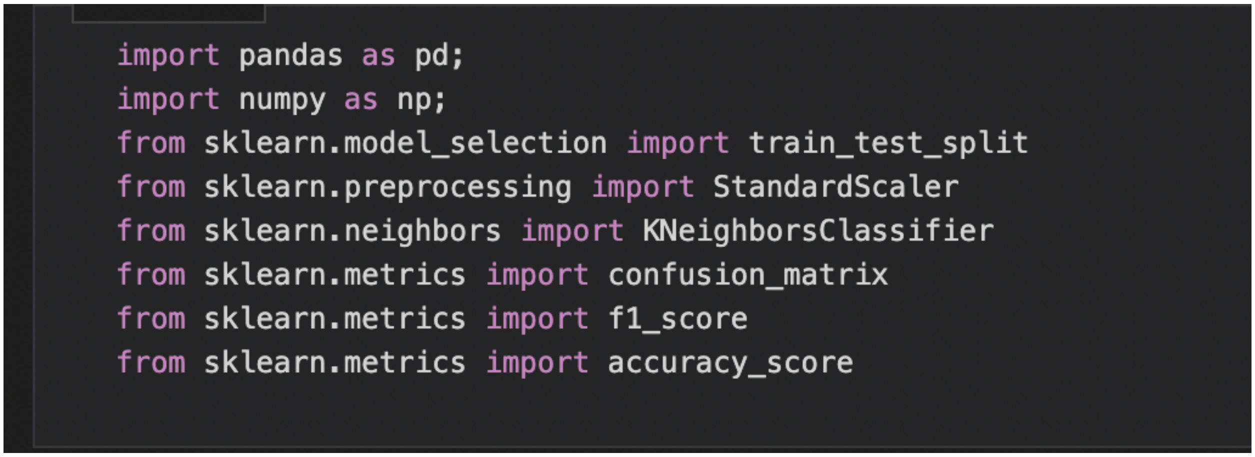
Mathematically, it represents the ratio of the sum of true positive and true negatives out of all the predictions.

Accuracy Score = (TP + TN)/ (TP + FN + TN + FP)

Dataset Used - water_potability

Step wise working Procedure:

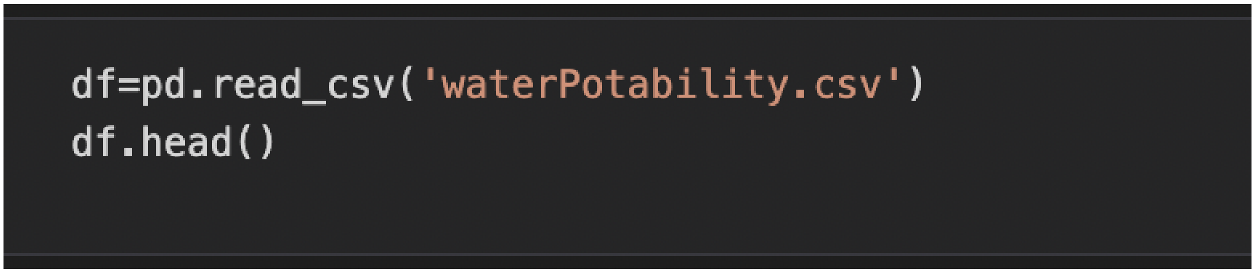
Step 1)



```
import pandas as pd;
import numpy as np;
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
```

Importing the required packages.
Pandas - To analyse the data and do the machine learning task
Numpy- to do variety of mathematical operation on the dataset
from sklearn.model_selection **import** train_test_split - Splitting the dataset for for training and testing with respect to the given value N.
from sklearn.neighbours import KNeighborsClassifier - Classifier implementing the k-nearest neighbors vote.

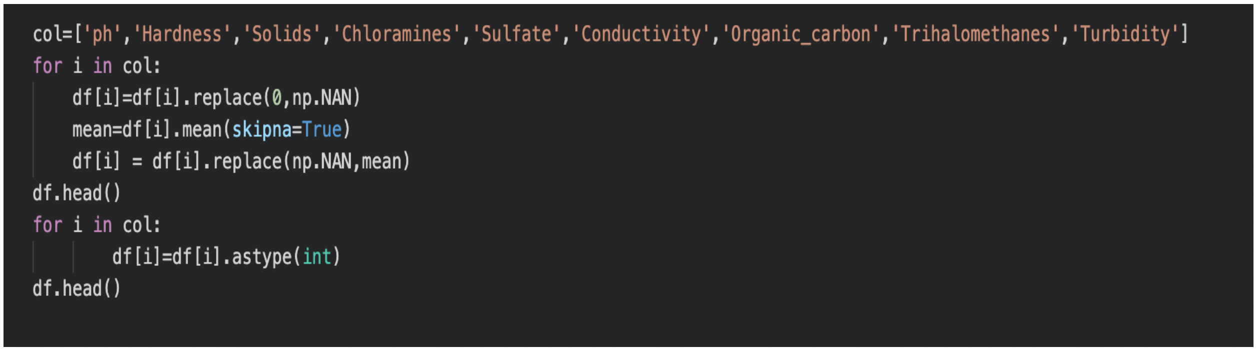
Step 2 -



```
df=pd.read_csv('waterPotability.csv')
df.head()
```

Displays the first 4 rows of the given dataset

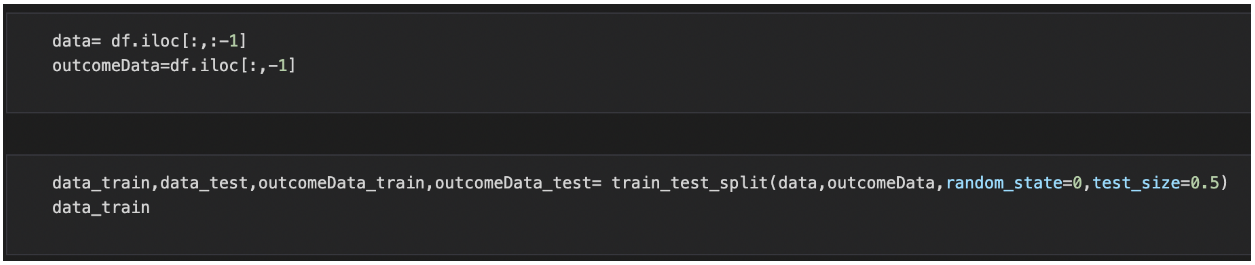
Step 3 -



```
col=['ph','Hardness','Solids','Chloramines','Sulfate','Conductivity','Organic_carbon','Trihalomethanes','Turbidity']
for i in col:
    df[i]=df[i].replace(0,np.NAN)
    mean=df[i].mean(skipna=True)
    df[i] = df[i].replace(np.NAN,mean)
df.head()
for i in col:
        df[i]=df[i].astype(int)
df.head()
```

Replacing all zeroes and not available values with the mean value of the given column and converting all floating values to integer values

Step 4 -



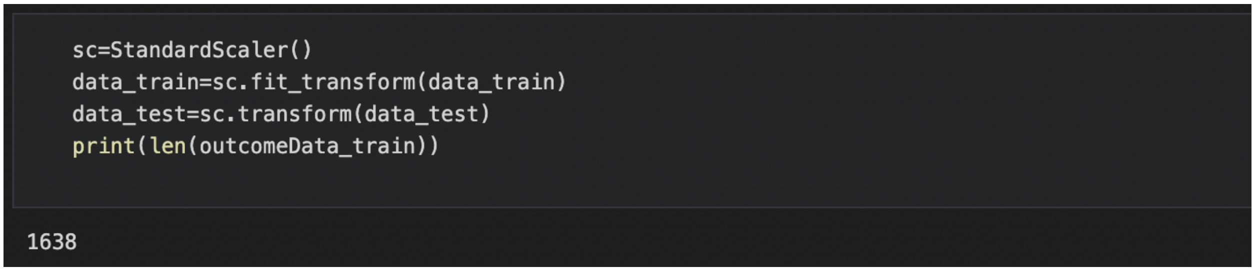
```
data= df.iloc[:,:-1]
outcomeData=df.iloc[:,-1]
```

```
data_train,data_test,outcomeData_train,outcomeData_test= train_test_split(data,outcomeData,random_state=0,test_size=0.5)
data_train
```

Excluding the last column of the dataset which should not be used for calculation as data and the outcome data as the last data.

Train_test_split is imported from the sklearn used to split , train and test dataset of size 50%

Step 5-

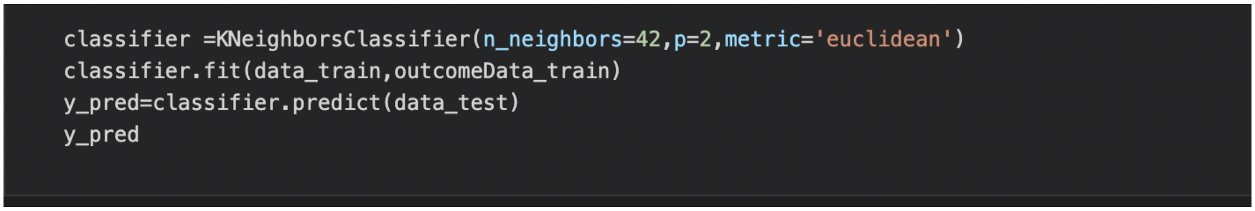


```
sc=StandardScaler()
data_train=sc.fit_transform(data_train)
data_test=sc.transform(data_test)
print(len(outcomeData_train))
```

```
1638
```

Display the length of the outcome data after fitting and transforming data

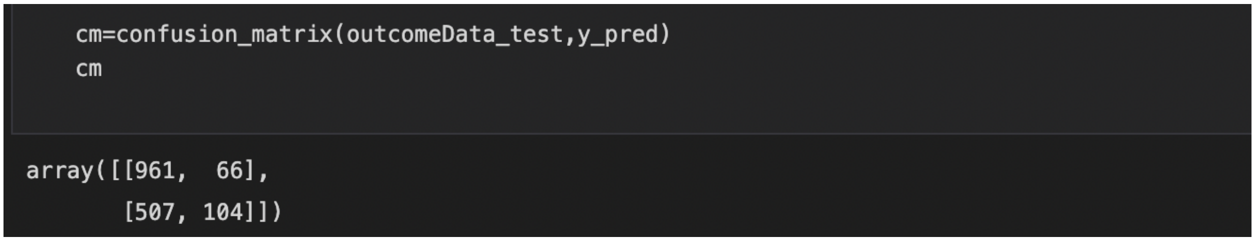
Step 6 -



```
classifier =KNeighborsClassifier(n_neighbors=42,p=2,metric='euclidean')
classifier.fit(data_train,outcomeData_train)
y_pred=classifier.predict(data_test)
y_pred
```

With KNeighbors Classifiers the number of neighbors to be taken to account is taken as 42 . the value of k is generally the square root of the N

Step 7 -

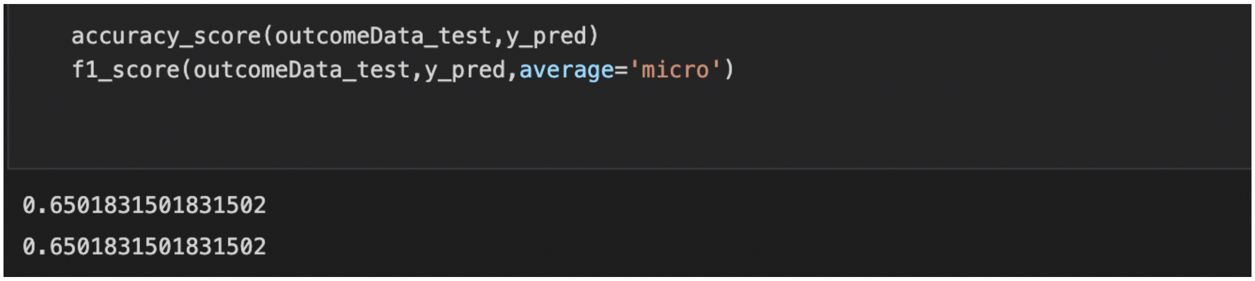


```
cm=confusion_matrix(outcomeData_test,y_pred)
cm
```

```
array([[961,  66],
       [507, 104]])
```

Displays the confusion matrix of the given training dataset

Step 8 -



```
accuracy_score(outcomeData_test,y_pred)
f1_score(outcomeData_test,y_pred,average='micro')
```

```
0.6501831501831502
0.6501831501831502
```

Displays the accuracy and f1score as 0.65

Result Analysis :

In this model an accuracy score of 0.65 is obtained for the given training dataset. The dataset is being trained over K-Nearest neighbor algorithm which trains data based on the number of nearest value to the point (k) .Depending upon the value of k the accuracy of the model varies , Choosing a largest value makes model more generalised and leads to underfitting and smaller value will lead to overfitting of the model thereby a optimised value of k should be chosen i.e. the value of k is square root of number of training datasets samples. KNN uses eucledian distances between reference point and other points to calculate and build model.It classifies itself into classes or groups and checks which class gets fitted for the obtained data.

Conclusion:

In this assignment a KNN model is implemented on the previous water-potablity dataset and calculated the accuracy and f1score out of it .

Git Hub Link of the project -
Ram-20062003/WaterPotability-Dataset

Contribution :

Ajayvishagan P -110120004- Written the write up for the code. Explained the basics and various terms associated with KNN algorithm in a detailed and legible manner.In assignment explained about the terms like bias and variance etc. Also helped through some code parts

Ram Ganesh K.R -110120090- Coded with KNN algorithm in this assignment using sklearn libraries and used linear regression for the previous assignment and calculated bias and variance and showed the difference. Helped in some parts of the writeups regarding working and analysis