Assignment 2

Student Name and ID of the member submitting the assignment:

Dumpa Bharat Kumar , ID - 1001870815

Student Name and ID of the remaining members:

Likhita Muddana, ID – 1001949141

1) Explain all the preprocessing done in detail.[4 points]

It is a process that is used to transform the data into the understandable format. To resolve some problems, we use the following steps.

- **Data Cleaning:**

It the process of cleaning datasets for missing values, correcting inconsistent data points, and smoothing noisy data.

Missing values: It arises when there is a missing data in the database. There are some ways in which we can solve these problems. This can be done either by ignoring the tuples or filling the missing values.

Noisy Data: It is meaningless data. It is caused due to incorrect data correction, or some errors in data entry. Binding method, regression, and clustering are the strategies which are used to solve the problems.

- **Data Transformation:**

It is a process of transforming of data from one format to another for a computer to understand.

There are some strategies in the transformation. Those are,

Normalization: It is used to convert the data variables in the specific range

Discretization: converting the attributes into a sets of similar intervals

Generalization: It is used to convert some features from low data to high data.

Concept hierarchy generation: It is used to create a hierarchy and convert the attributes in the hierarchy from low level to higher level.

- **Data Reduction:**

It is a process which is used to reduce the storage data and the costs. The various steps involved in Data Reduction are

Attribute Subset Selection: It is a process of selecting the subset features that contribute to the most important one

Numerosity reduction: It is a process of replacing the original data with a smaller form of data representation.

Dimensionality reduction: It is used to reduce the features in the dataset.

2) Explain all the parameters of KNN in details in your own words

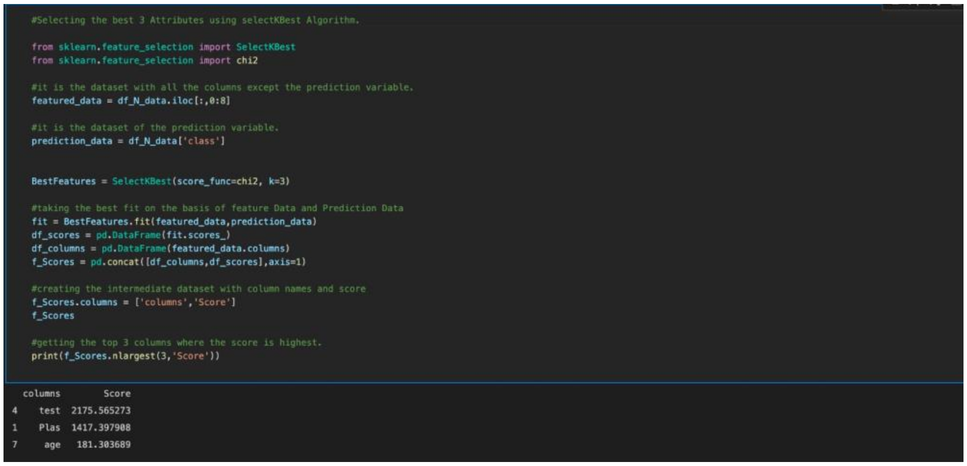
It is the essential classification algorithm in machine learning. This is simple and easy complement that can solve both classification and regression problems.

Here first we need to find the value of K. Then we need to calculate the distance between the new data and the training data. This can be done by using Euclidian Distance or Minkowski. From this result we need to find the closet K neighbors to the new data and have the highest quality. Then select the new training data that is close to new data with highest quality. Finally calculate the accuracy of the model.

3) Explain what your criteria for selecting the three attributes. [5 points]

We are using the Select Best algorithm. Here selection is a technique which helps us to choose those features in our data that contribute to the target variable. It selects the features based ono highest K scores.

Hereafter loading the dataset, we use the featured data to check the feature data dimensions. Prediction_data is used to separate the predictions variables. So, for getting the selected features we use f_scores.columns and finally we are printing the best 3 attributes.

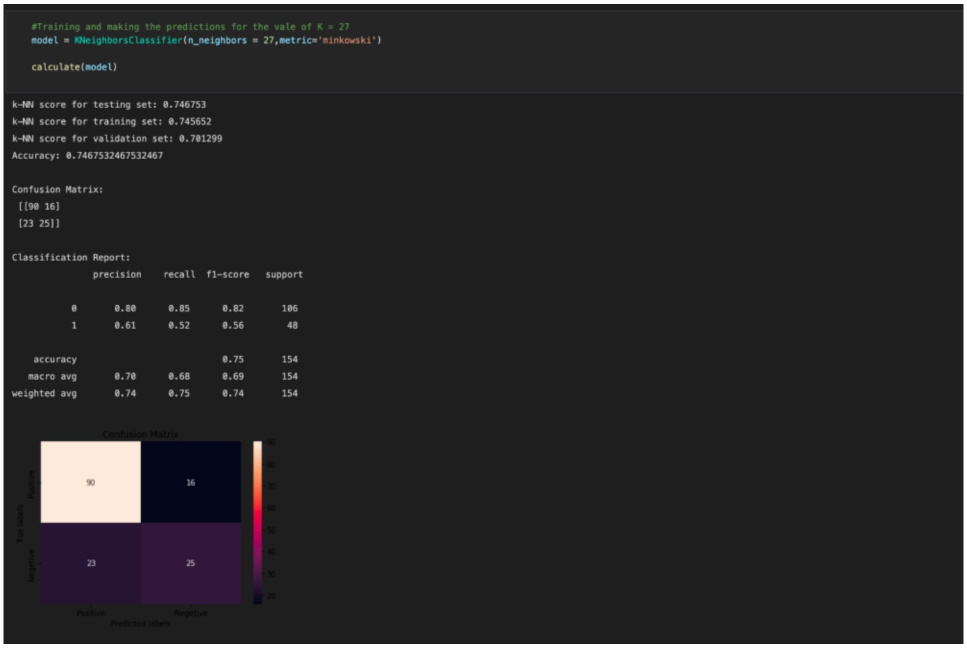


```
#Selecting the best 3 Attributes using selectKBest Algorithm.

from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

#it is the dataset with all the columns except the prediction variable.
featured_data = df_N_data.iloc[:,0:8]

#it is the dataset of the prediction variable.
prediction_data = df_N_data['class']


BestFeatures = SelectKBest(score_func=chi2, k=3)

#taking the best fit on the basis of feature Data and Prediction Data
fit = BestFeatures.fit(featured_data,prediction_data)
df_scores = pd.DataFrame(fit.scores_)
df_columns = pd.DataFrame(featured_data.columns)
f_Scores = pd.concat([df_columns,df_scores],axis=1)

#creating the intermediate dataset with column names and score
f_Scores.columns = ['columns','Score']
f_Scores

#getting the top 3 columns where the score is highest.
print(f_Scores.nlargest(3,'Score'))
```

```
  columns        Score
4    test  2175.565273
1    Plas  1417.397908
7     age   181.303689
```

4) Visualizations of the classifier in a 2D projection and write your observations.

First, we are loading the dataset. Then we are splitting the dataset into training, testing, and validation set. We are calculating the knn score for test, training, and validation and calculating the accuracy of the trained classifier and testing data set. creating the confusion matrix for the trained classifier over the test dataset and finally plotting the classification report in the 2D projection.

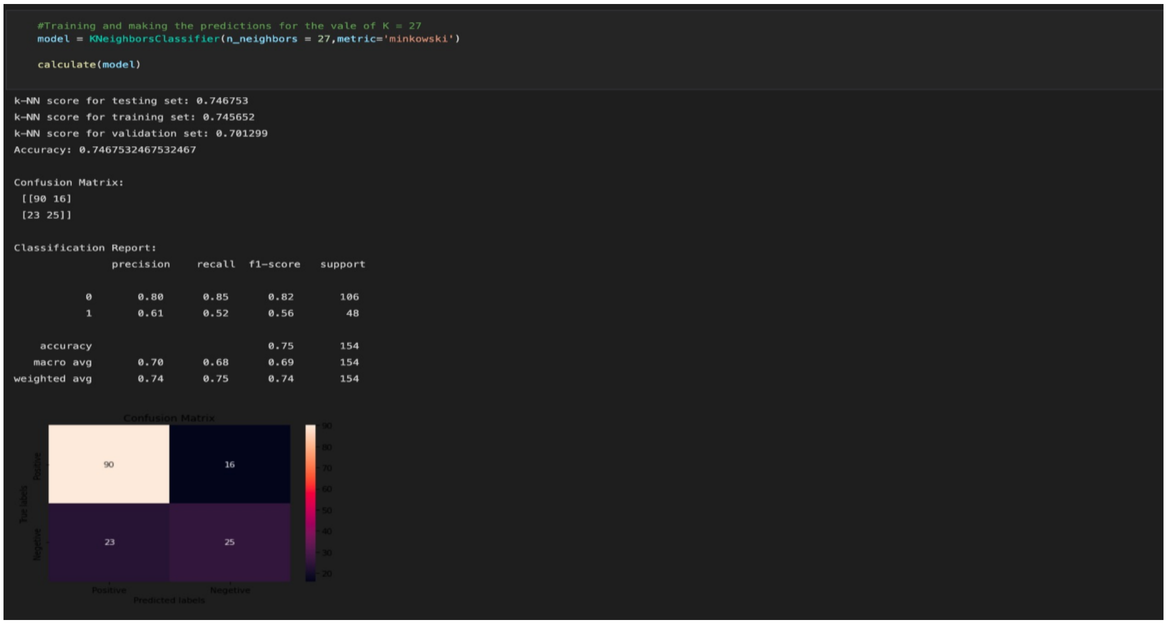


```
#Training and making the predictions for the vale of K = 27
model = KNeighborsClassifier(n_neighbors = 27,metric='minkowski')

calculate(model)
```

```
k-NN score for testing set: 0.746753
k-NN score for training set: 0.745652
k-NN score for validation set: 0.701299
Accuracy: 0.7467532467532467

Confusion Matrix:
 [[90 16]
 [23 25]]

Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.85      0.82       106
           1       0.61      0.52      0.56        48

    accuracy                           0.75       154
   macro avg       0.70      0.68      0.69       154
weighted avg       0.74      0.75      0.74       154
```

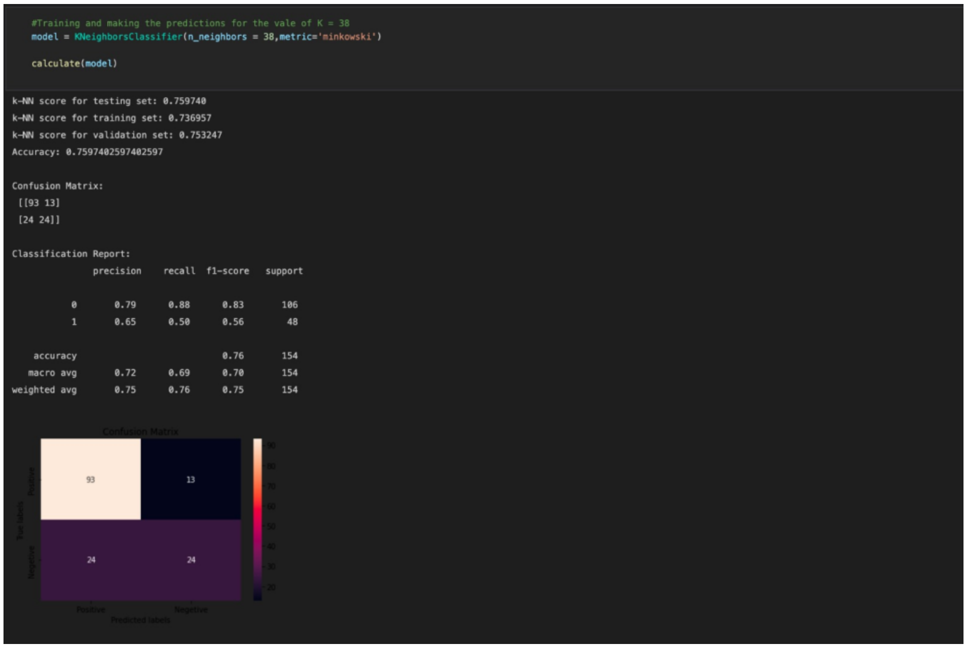
5) Interpreted and compare the results explain in detail.

So here we are comparing the results by changing the K values. From below we could see the whole result by comparing the k values.



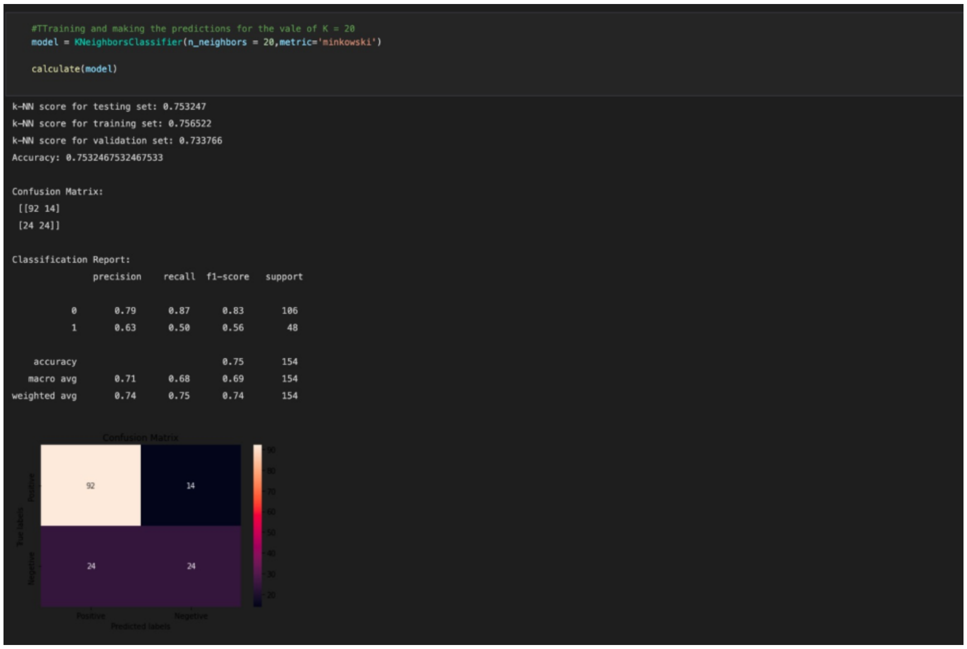
```
#Training and making the predictions for the vale of K = 27
model = KNeighborsClassifier(n_neighbors = 27,metric='minkowski')

calculate(model)
```

```
k-NN score for testing set: 0.746753
k-NN score for training set: 0.745652
k-NN score for validation set: 0.701299
Accuracy: 0.7467532467532467

Confusion Matrix:
 [[90 16]
 [23 25]]

Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.85      0.82       106
           1       0.61      0.52      0.56        48

    accuracy                           0.75       154
   macro avg       0.70      0.68      0.69       154
weighted avg       0.74      0.75      0.74       154
```



```
#Training and making the predictions for the vale of K = 38
model = KNeighborsClassifier(n_neighbors = 38,metric='minkowski')

calculate(model)
```

```
k-NN score for testing set: 0.759740
k-NN score for training set: 0.736957
k-NN score for validation set: 0.753247
Accuracy: 0.7597402597402597

Confusion Matrix:
 [[93 13]
 [24 24]]

Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.88      0.83       106
           1       0.65      0.50      0.56        48

    accuracy                           0.76       154
   macro avg       0.72      0.69      0.70       154
weighted avg       0.75      0.76      0.75       154
```



```
#TTraining and making the predictions for the vale of K = 20
model = KNeighborsClassifier(n_neighbors = 20,metric='minkowski')

calculate(model)
```

```
k-NN score for testing set: 0.753247
k-NN score for training set: 0.756522
k-NN score for validation set: 0.733766
Accuracy: 0.7532467532467533

Confusion Matrix:
 [[92 14]
 [24 24]]

Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.87      0.83       106
           1       0.63      0.50      0.56        48

    accuracy                           0.75       154
   macro avg       0.71      0.68      0.69       154
weighted avg       0.74      0.75      0.74       154
```