KUMAR SANATAN
ROLL NO – 2211AI24

CS 564
Foundations of Machine Learning
ASSIGMENT 1:

INDIAN INSTITUTE OF TECHNOLOGY
PATNA

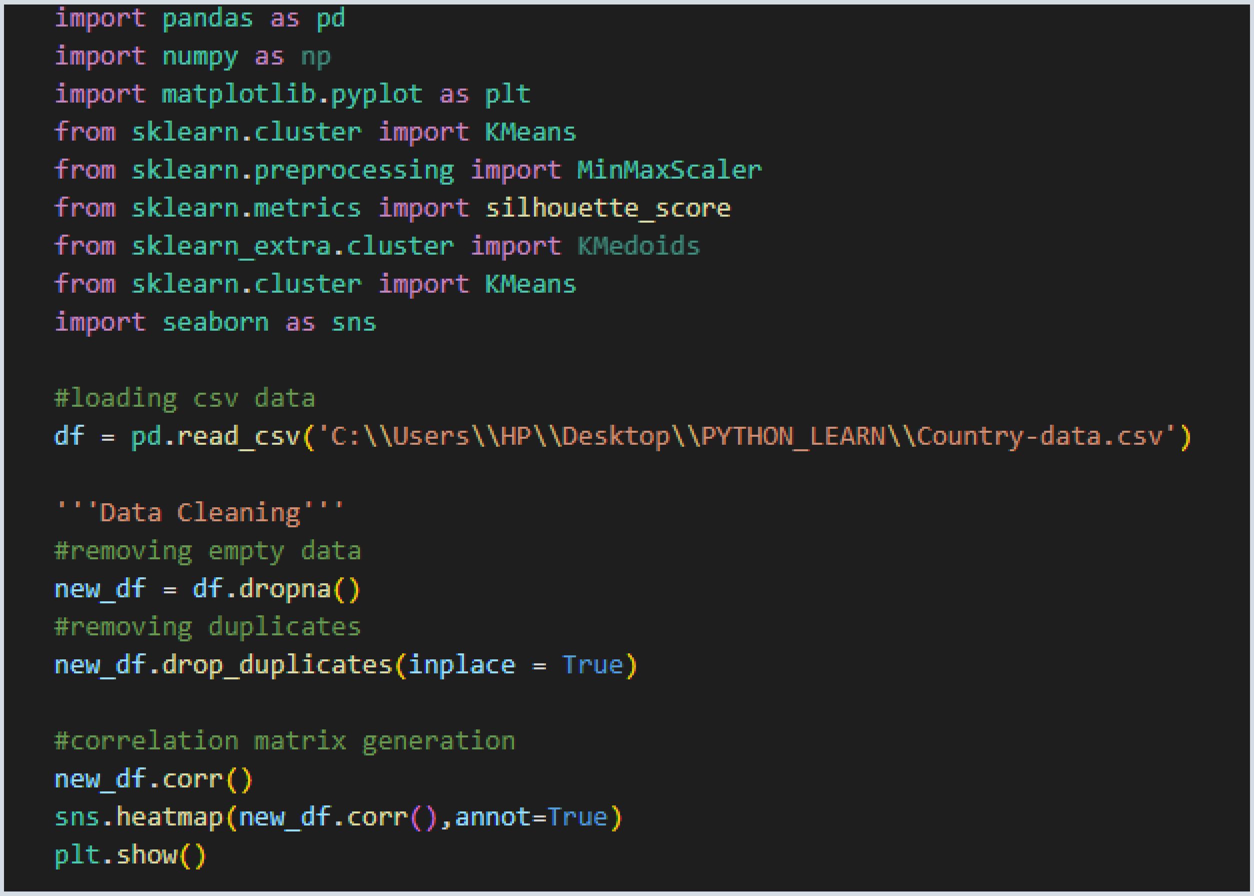
Date: 18th August 2022 **Deadline:** 26th August 2022

OBJECTIVE

The assignment targets to implement K-Means and K-Medoid algorithms to cluster the dataset consists of socio-economic and health factors of countries and determine the overall development of the country

<u>CODE SUMMARY :</u>

Importing required packages, and reading the csv file (data file).
Performing the Data Cleaning prerequisites by removal of empty spaces and duplicates if present. Generation of correlation matrix for getting overall summary of data and the correlation among attributes present.



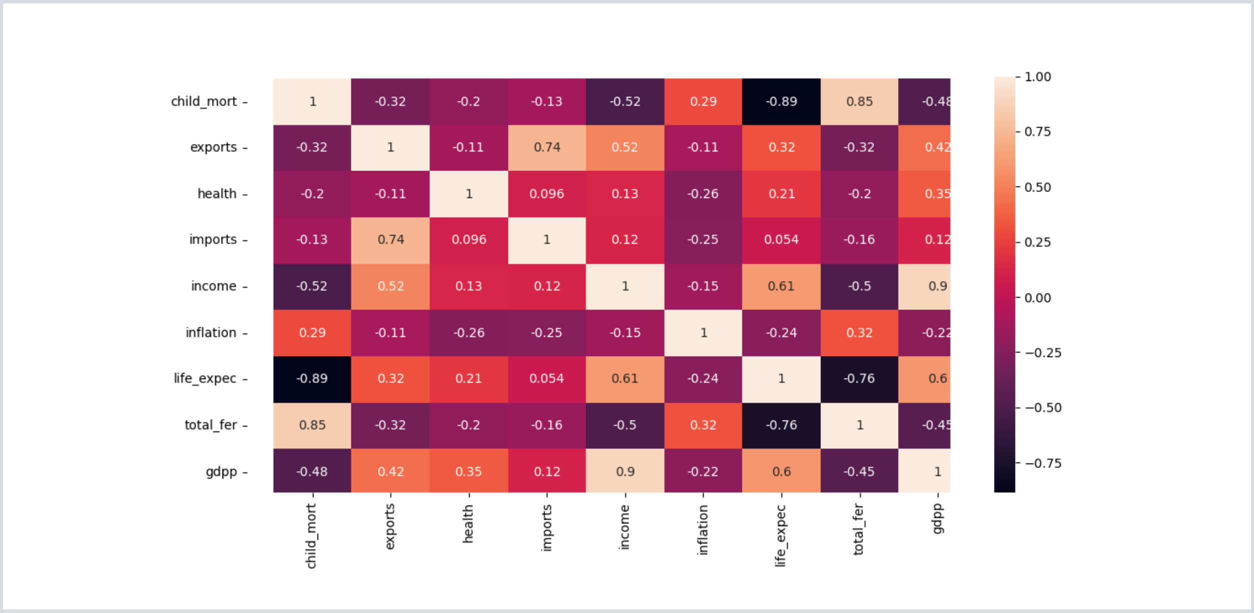
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import silhouette_score
from sklearn_extra.cluster import KMedoids
from sklearn.cluster import KMeans
import seaborn as sns

#loading csv data
df = pd.read_csv('C:\\Users\\HP\\Desktop\\PYTHON_LEARN\\Country-data.csv')

'''Data Cleaning'''
#removing empty data
new_df = df.dropna()
#removing duplicates
new_df.drop_duplicates(inplace = True)

#correlation matrix generation
new_df.corr()
sns.heatmap(new_df.corr(),annot=True)
plt.show()
```

Correlation Output:



	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
child_mort	1	-0.32	-0.2	-0.13	-0.52	0.29	-0.89	0.85	-0.48
exports	-0.32	1	-0.11	0.74	0.52	-0.11	0.32	-0.32	0.42
health	-0.2	-0.11	1	0.096	0.13	-0.26	0.21	-0.2	0.35
imports	-0.13	0.74	0.096	1	0.12	-0.25	0.054	-0.16	0.12
income	-0.52	0.52	0.13	0.12	1	-0.15	0.61	-0.5	0.9
inflation	0.29	-0.11	-0.26	-0.25	-0.15	1	-0.24	0.32	-0.22
life_expec	-0.89	0.32	0.21	0.054	0.61	-0.24	1	-0.76	0.6
total_fer	0.85	-0.32	-0.2	-0.16	-0.5	0.32	-0.76	1	-0.45
gdpp	-0.48	0.42	0.35	0.12	0.9	-0.22	0.6	-0.45	1

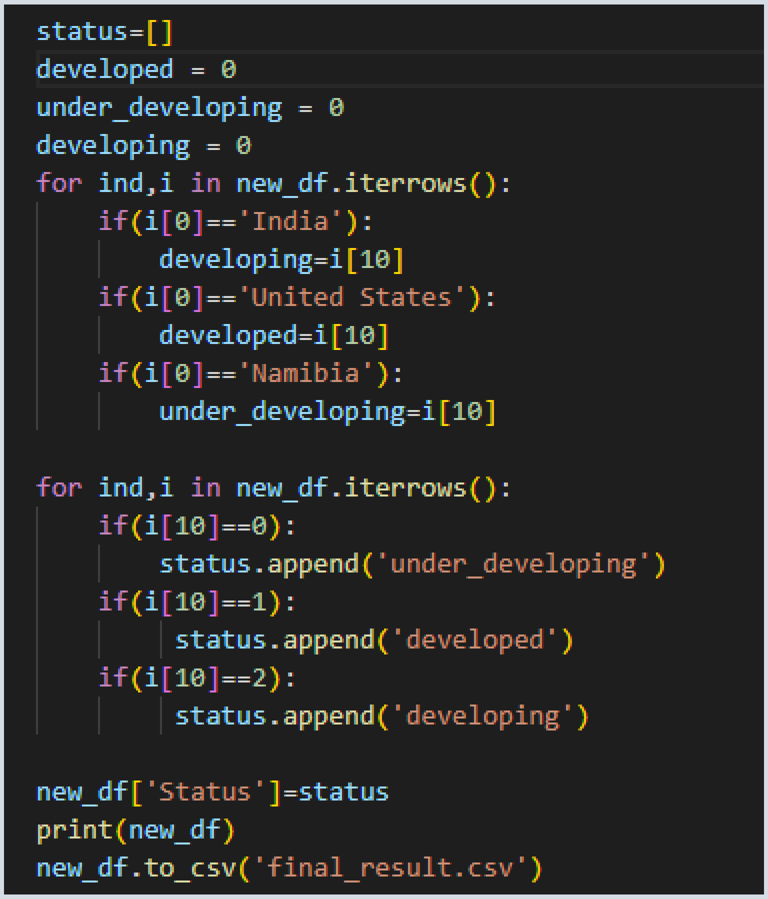
K-Means:

Scaling the present csv data via parsing it to data frame and selecting the number of clusters to be 3. A new column as ‘cluster’ is added to the data that depicts the cluster assigned to the row. The data is plotted for different pair of attributed selected among the ten given in the csv file and after that silhouette score is calculated for the cluster formed.

```
# transform data
scaler = MinMaxScaler()
scaled = pd.DataFrame(scaler.fit_transform(new_df.iloc[ :,1:10]))
kmeans = KMeans(3,random_state=3)
identified_clusters = kmeans.fit_predict(scaled)
new_df['clusters']=identified_clusters
#plotting clustering data
plt.scatter(scaled.iloc[:,4],scaled.iloc[:,8],c=new_df['clusters'],cmap='rainbow')
plt.show()

#calculating silhouettte score
score = silhouette_score(scaled.iloc[:,1:10],identified_clusters)
print("SilHouette Score = "+str(score))
```

Here we are trying to identify the clusters/countries as developed, under_developing and developing by initial assumption of United States being developed, India being developing and Namibia as under_developing. After that we are inserting the status of all countries in the status list and post that we are adding the new column as ‘Status’ to the csv data and finally appending the status corresponding to the particular row.



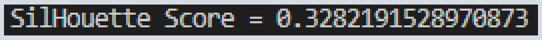
```python
status=[]
developed = 0
under_developing = 0
developing = 0
for ind,i in new_df.iterrows():
    if(i[0]=='India'):
        developing=i[10]
    if(i[0]=='United States'):
        developed=i[10]
    if(i[0]=='Namibia'):
        under_developing=i[10]

for ind,i in new_df.iterrows():
    if(i[10]==0):
        status.append('under_developing')
    if(i[10]==1):
         status.append('developed')
    if(i[10]==2):
         status.append('developing')

new_df['Status']=status
print(new_df)
new_df.to_csv('final_result.csv')
```

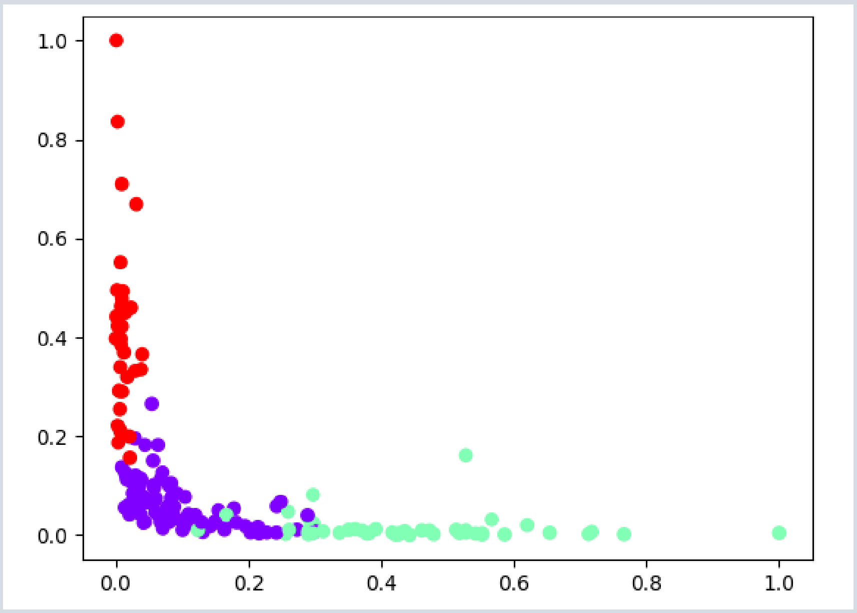
K Means Outputs:

Silhouette Score for the Cluster:

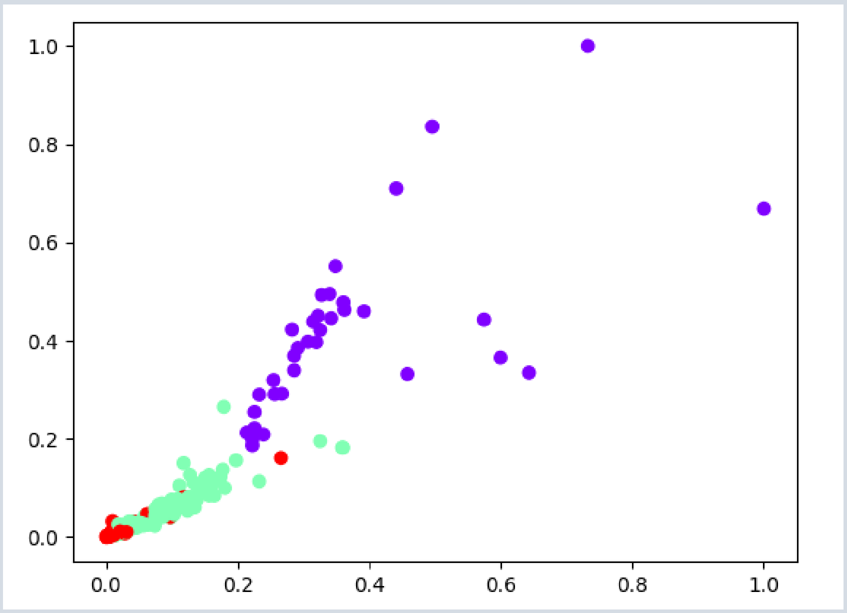


```
SilHouette Score = 0.3282191528970873
```

Child Mortality vs GDPP:

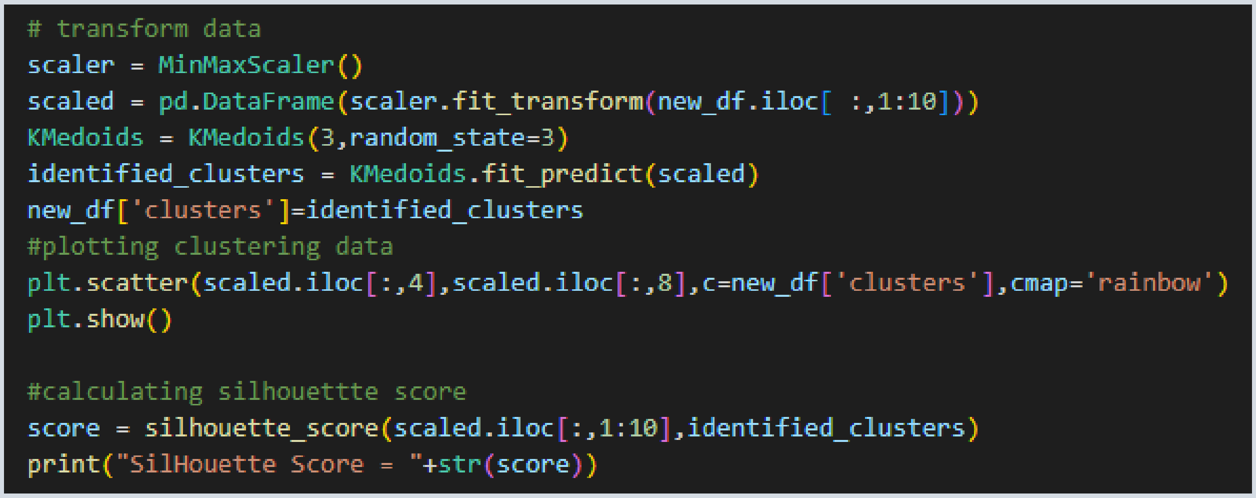


Income vs GDPP:



K-Medoid:

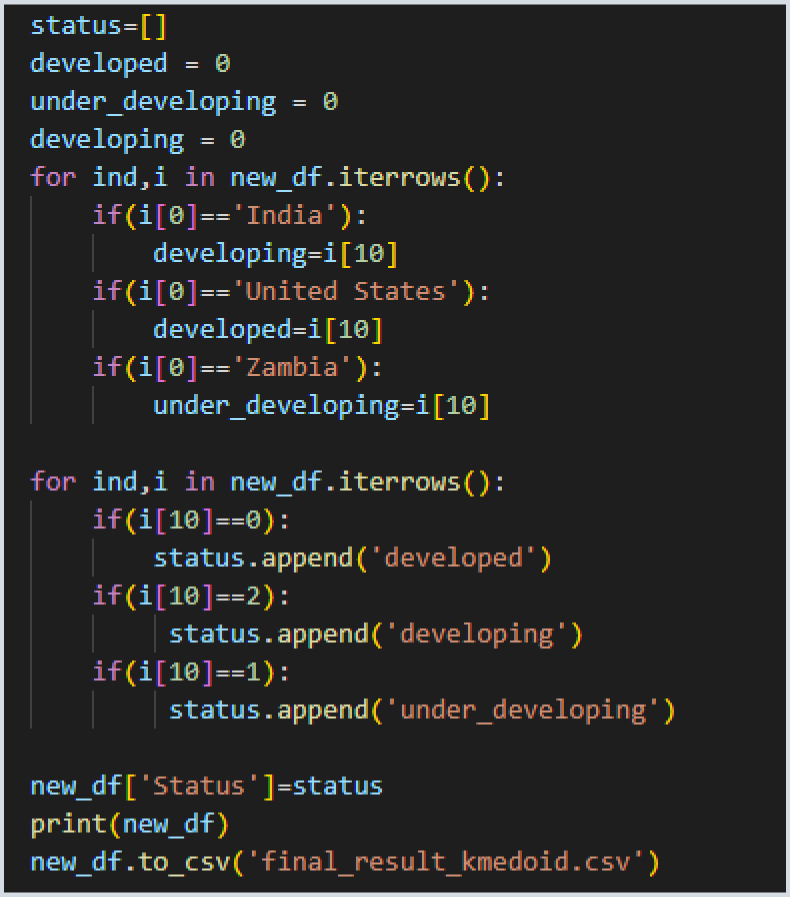
Scaling the present csv data via parsing it to data frame and selecting the number of clusters to be 3. A new column as ‘cluster’ is added to the data that depicts the cluster assigned to the row. The data is plotted for different pair of attributed selected among the ten given in the csv file and after that silhouette score is calculated for the cluster formed



```
# transform data
scaler = MinMaxScaler()
scaled = pd.DataFrame(scaler.fit_transform(new_df.iloc[ :,1:10]))
KMedoids = KMedoids(3,random_state=3)
identified_clusters = KMedoids.fit_predict(scaled)
new_df['clusters']=identified_clusters
#plotting clustering data
plt.scatter(scaled.iloc[:,4],scaled.iloc[:,8],c=new_df['clusters'],cmap='rainbow')
plt.show()

#calculating silhouettte score
score = silhouette_score(scaled.iloc[:,1:10],identified_clusters)
print("SilHouette Score = "+str(score))
```

Here we are trying to identify the clusters/countries as developed, under_developing and developing by initial assumption of United States being developed, India being developing and Zambia as under_developing. After that we are inserting the status of all countries in the status list and post that we are adding the new column as ‘Status’ to the csv data and finally appending the status corresponding to the particular row.



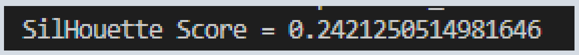
```python
status=[]
developed = 0
under_developing = 0
developing = 0
for ind,i in new_df.iterrows():
    if(i[0]=='India'):
        developing=i[10]
    if(i[0]=='United States'):
        developed=i[10]
    if(i[0]=='Zambia'):
        under_developing=i[10]

for ind,i in new_df.iterrows():
    if(i[10]==0):
        status.append('developed')
    if(i[10]==2):
         status.append('developing')
    if(i[10]==1):
         status.append('under_developing')

new_df['Status']=status
print(new_df)
new_df.to_csv('final_result_kmedoid.csv')
```

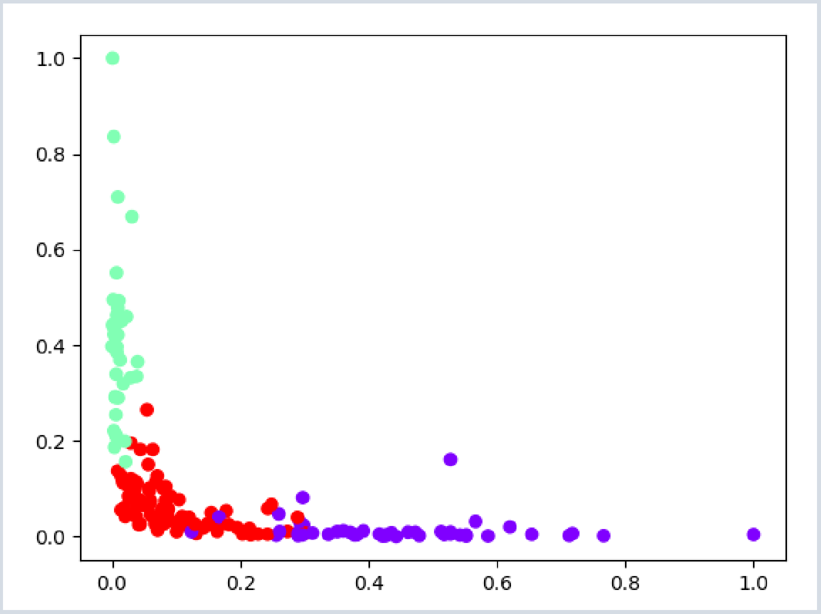
K Medoid Outputs:

Silhouette Score for the Cluster:

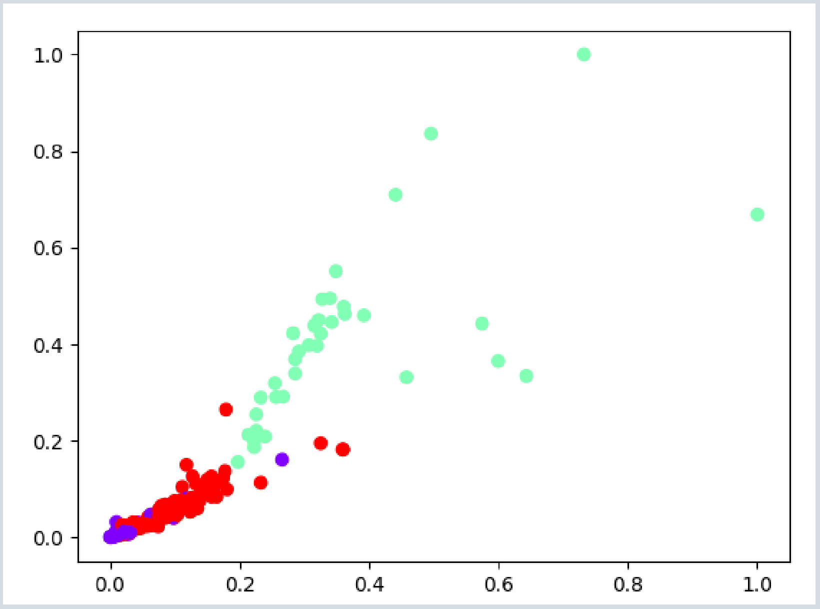


```
SilHouette Score = 0.2421250514981646
```

Child Mortality vs GDPP:



Income vs GDPP:



SUMMARY:

K-means Algorithm uses the euclidian distance to measure the goodness of the clustering and is designed to optimise (to minimise this distance)

K-medoids Algorithm uses the first Norm to compute the distance and so to measure the goodness of the clustering

Thus the K-medoids Algorithm is more robust/resistant to outliers, since the error is linear proportional to the distance, whereas the error we try to minimize while using the K-means Algorithm is quadratically proportional to the distance.

Silhouette coefficient is used to select the number of clusters in algorithms like K-Means where no_of_clusters is a hyper-parameters. It intuitively says how well you have clustered. The value ranges from -1 to 1 , 1 means very well clustered and that the maximum value you can attain . 0 is at the borderline that means the boundaries of different clusters are almost overlapping and -1 means that you have actually put some points wrongly to other clusters.