CS564 : Foundations of Machine Learning
Assignment 1

By Priyanka Sachan (1901CS43)

Problem Statement

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

Installation

Install the following dependencies either using pip or through conda in a Python 3.5+ environment:

```
python3 -m pip install pandas seaborn matplotlib scikit-learn-extra
```

Running the program

Use the following command to run the program :

```
python3 cluster_algos.py
```

Implementation

Code added in zip file or check Jupyter Notebook.

Import libraries and packages

```
# Import libraries and packages
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# scaling
from sklearn.preprocessing import Normalizer
# kmeans clustering
from sklearn.cluster import KMeans
# kmeans clustering
from sklearn_extra.cluster import KMedoids
# silhouette score
from sklearn.metrics import silhouette_score
```

Import data

```
# Read csv file
data = pd.read_csv(f'Country-data.csv')

# Get statistical information
print(data.describe())
```

We check the dataset for missing values and drop the country column.

```
# Check for missing values
data.isnull().sum()

country_list=data['country']
# Drop country column
data=data.drop(['country'],axis=1)
```

```
country       0
child_mort    0
exports       0
health        0
imports       0
income        0
inflation     0
life_expec    0
total_fer     0
gdpp          0
dtype: int64
```

Scaling data

Our dataset is not scaled some values are much bigger than others,if we do not scale our data our model will not going to perform well.So now we are going to scale our data using StandardScaler ↗
It standardises features by removing the mean and scaling to unit variance.

```
# Scaling data
scaled=StandardScaler().fit_transform(df)
scaled_df=pd.DataFrame(scaled,columns=df.columns)
print('Scaled data \n',scaled_df)
```

K-means algorithm

Fit data using K-means algorithm

```
# Number of clusters=3 from problem statement
print('\nUSING K-MEANS WITH K=3')
# Fitting kmeans model
kmeans = KMeans(n_clusters = 3,random_state = 0)
kmeans.fit(scaled_df)

# Predicting values
cluster_labels = kmeans.fit_predict(scaled_df)

kmeans_df=df.copy(deep=False)
kmeans_df.insert(0,'country',country_list)
kmeans_df['clusters']=cluster_labels
print('Result\n',kmeans_df)

# Save result to csv file
kmeans_df.to_csv('kmeans_results.csv', header=False, index=False)
```

Evaluation

```
# Calculate Silhouette Coefficient for K=3
print('\nEVALUATION')
kmeans_evaluation_score = silhouette_score(scaled_df, kmeans.labels_)
print('Silhouette Score: ',kmeans_evaluation_score)
```

```
EVALUATION
Silhouette Score:  0.28329575683463126
```

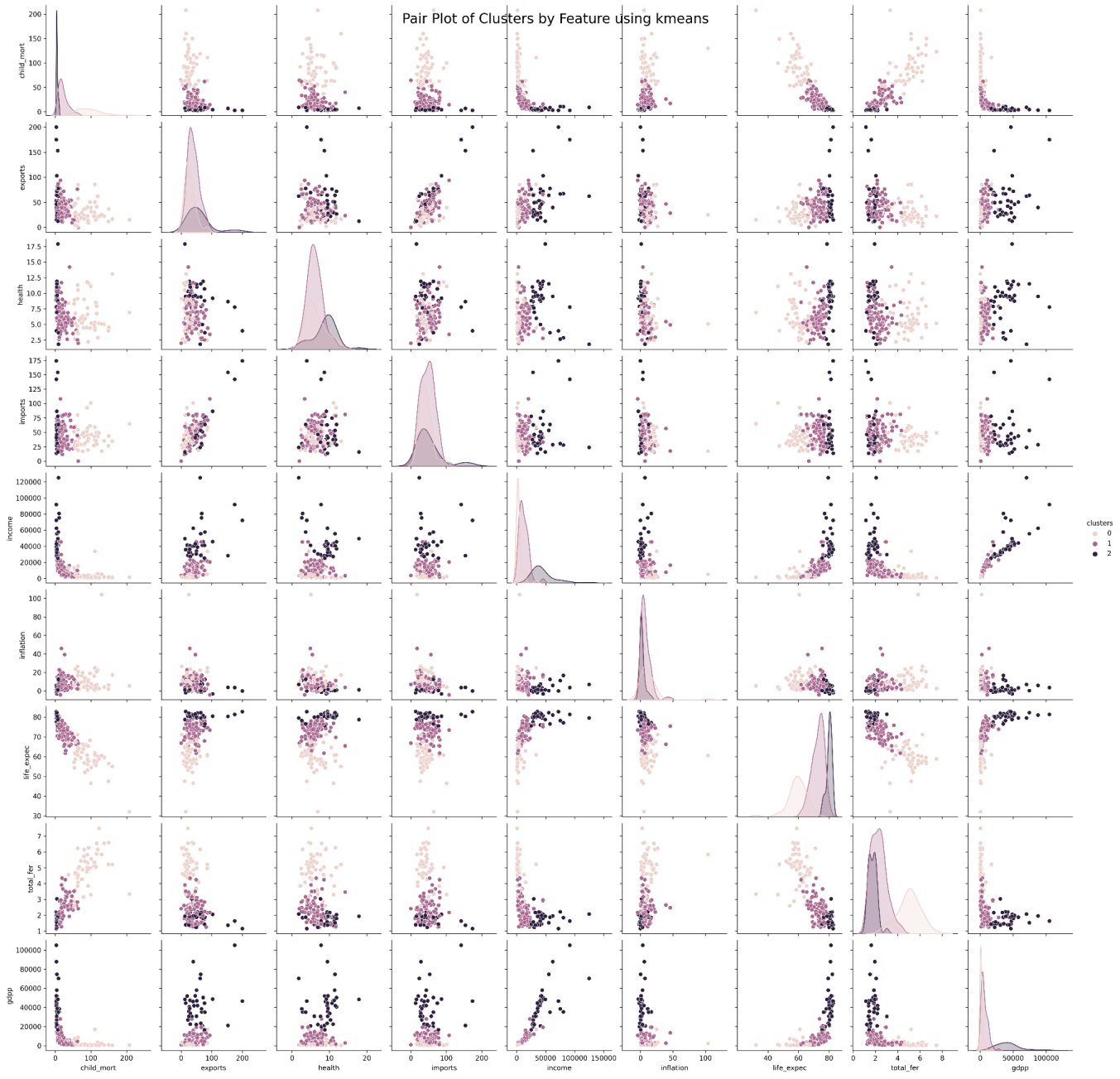
Cluster Mapping

```
# Mean feature values for each cluster
print('\nCLUSTER MAPPING')
kmeans_mean=kmeans_df.groupby('clusters')[['child_mort', 'exports','health'
,'imports', 'income', 'inflation',  'life_expec', 'total_fer',  'gdpp'  ]].mean()
print(' Mean feature values for each cluster \n',kmeans_mean)

# Decide which cluster maps to under_developed, developing or developed countries.
Here,
# 0 = under-developing country
# 1 = developing country
# 2 = developed country
print('\nJudging from mean values, \n0 = under-developing country  \n1 = developing
country \n2 = developed country \n')

#find number of developed country,developing country,under-developed country
```

```
under_developing=kmeans_df[kmeans_df['clusters']==0]['country']
developing=kmeans_df[kmeans_df['clusters']==1]['country']
developed=kmeans_df[kmeans_df['clusters']==2]['country']

print('Number of developed countries: ',len(under_developing))
print('Number of developing countries: ',len(developing))
print('Number of under-developing countries: ',len(developed))
```

Visualisation

```
# Visualization
print('\nVISUALIZATION')
sns.pairplot(kmeans_df, hue='clusters')
plt.suptitle('Pair Plot of Clusters by Feature using kmeans', size = 20);
plt.savefig('Kmeans_PairPlot.png',dpi=300, facecolor='w')
```



K-medoid algorithm

Fit data using K-medoid algorithm

```
# Number of clusters=3 from problem statement
print('\nUSING K-MEDOIDS WITH K=3')
# Fitting kmedoids model
kmedoids = KMedoids(n_clusters=3, random_state=0)
kmedoids.fit(scaled_df)

#predicting values
cluster_labels = kmedoids.fit_predict(scaled_df)

kmedoids_df=df.copy(deep=False)
kmedoids_df.insert(0,'country',country_list)
kmedoids_df['clusters']=cluster_labels
print('Result\n',kmedoids_df)

# Save result to csv file
kmedoids_df.to_csv('kmediods_results.csv', header=False, index=False)
```

Evaluation

```
# Calculate Silhouette Coefficient for K=3
print('\nEVALUATION')
kmedoids_evaluation_score = silhouette_score(scaled_df, kmedoids.labels_)
print('Silhouette Score: ',kmedoids_evaluation_score)
```

```
EVALUATION
Silhouette Score:  0.1562250700966545
```

Cluster Mapping

```
# Mean feature values for each cluster
print('\nCLUSTER MAPPING')
kmedoids_mean=kmedoids_df.groupby('clusters')[['child_mort', 'exports','health'
,'imports', 'income', 'inflation',  'life_expec', 'total_fer',  'gdpp'  ]].mean()
print(' Mean feature values for each cluster \n',kmedoids_mean)

# Decide which cluster maps to under_developed, developing or developed countries.
Here,
# 1 = under-developing country
# 2 = developing country
# 0 = developed country
print('\nJudging from mean values, \n1 = under-developing country  \n2 = developing
country \n0 = developed country \n')

#find number of developed country,developing country,under-developed country
under_developing=kmeans_df[kmeans_df['clusters']==1]['country']
```

```
developing=kmeans_df[kmeans_df['clusters']==2]['country']
developed=kmeans_df[kmeans_df['clusters']==0]['country']

print('Number of developed countries: ',len(under_developing))
print('Number of developing countries: ',len(developing))
print('Number of under-developing countries: ',len(developed))
```

Visualisation

```
# Visualization
print('\nVISUALIZATION')
sns.pairplot(kmedoids_df, hue='clusters')
plt.suptitle('Pair Plot of Clusters by Feature using kmedoids', size = 20);
plt.savefig('Kmedoids_PairPlot.png',dpi=300, facecolor='w')
```