In [4]:
```
import pandas as pd
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,roc_auc_score
```

EDA

In [7]:
```
pwd
```

Out[7]:
```
'/Users/sangitalamichhane/TAKEO Training/Week 2/Day 13 KNN'
```

In [30]:
```
##Loading or reading data#
df = pd.read_csv('gapminder.csv')
```

In [31]:
```
df.head()
```

Out[31]:

	population	fertility	HIV	CO2	BMI_male	GDP	BMI_female	life	child_mortality	Regi
0	34811059.0	2.73	0.1	3.328945	24.59620	12314.0	129.9049	75.3	29.5	Mid Eas No Afri
1	19842251.0	6.43	2.0	1.474353	22.25083	7103.0	130.1247	58.3	192.0	Su Sahar Afri
2	40381860.0	2.24	0.5	4.785170	27.50170	14646.0	118.8915	75.5	15.4	Ameri
3	2975029.0	1.40	0.1	1.804106	25.35542	7383.0	132.8108	72.5	20.0	Euro Cent A
4	21370348.0	1.96	0.1	18.016313	27.56373	41312.0	117.3755	81.5	5.2	Ea Asia Paci

In [32]:

```
df.describe()  ##Statistical summary of all your numerical variables
```

Out[32]:

	population	fertility	HIV	CO2	BMI_male	GDP	BMI_female
count	1.390000e+02	139.000000	139.000000	139.000000	139.000000	139.000000	139.000000
mean	3.549977e+07	3.005108	1.915612	4.459874	24.623054	16638.784173	126.701914
std	1.095121e+08	1.615354	4.408974	6.268349	2.209368	19207.299083	4.471997
min	2.773150e+05	1.280000	0.060000	0.008618	20.397420	588.000000	117.375500
25%	3.752776e+06	1.810000	0.100000	0.496190	22.448135	2899.000000	123.232200
50%	9.705130e+06	2.410000	0.400000	2.223796	25.156990	9938.000000	126.519600
75%	2.791973e+07	4.095000	1.300000	6.589156	26.497575	23278.500000	130.275900
max	1.197070e+09	7.590000	25.900000	48.702062	28.456980	126076.000000	135.492000

In [33]:

```
df.info() ##Data type of columns, Numbers of NA values in each columns, sha
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 139 entries, 0 to 138
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   population       139 non-null    float64
 1   fertility        139 non-null    float64
 2   HIV              139 non-null    float64
 3   CO2              139 non-null    float64
 4   BMI_male         139 non-null    float64
 5   GDP              139 non-null    float64
 6   BMI_female       139 non-null    float64
 7   life             139 non-null    float64
 8   child_mortality  139 non-null    float64
 9   Region           139 non-null    object 
dtypes: float64(9), object(1)
memory usage: 11.0+ KB
```

Train test split and Data processing

In [37]:

```
x = df.drop('Region',1)
y = df['Region']
```

In [75]:

```
X_train,X_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_
```

In [39]:
```
y_train.value_counts() ##Checking whether all classes have proportional rep
```

Out[39]:
```
Sub-Saharan Africa            32
Europe & Central Asia         28
America                       22
East Asia & Pacific           10
Middle East & North Africa     8
South Asia                     4
Name: Region, dtype: int64
```

In [52]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [51]:
```
scaler = MinMaxScaler()  ##(x-max(x))/(max(x))
```

In [67]:
```
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [69]:
```
x
```

Out[69]:

	population	fertility	HIV	CO2	BMI_male	GDP	BMI_female	life	child_mortality
0	34811059.0	2.73	0.1	3.328945	24.59620	12314.0	129.9049	75.3	29.5
1	19842251.0	6.43	2.0	1.474353	22.25083	7103.0	130.1247	58.3	192.0
2	40381860.0	2.24	0.5	4.785170	27.50170	14646.0	118.8915	75.5	15.4
3	2975029.0	1.40	0.1	1.804106	25.35542	7383.0	132.8108	72.5	20.0
4	21370348.0	1.96	0.1	18.016313	27.56373	41312.0	117.3755	81.5	5.2
...	...	...	...	...	...	...	...	...	...
134	3350832.0	2.11	0.5	2.489764	26.39123	15317.0	124.2604	76.0	13.0
135	26952719.0	2.46	0.1	4.476669	25.32054	3733.0	124.3462	68.7	49.2
136	86589342.0	1.86	0.4	1.479347	20.91630	4085.0	121.9367	75.4	26.2
137	13114579.0	5.88	13.6	0.148982	20.68321	3039.0	132.4493	52.0	94.9
138	13495462.0	3.85	15.1	0.654323	22.02660	1286.0	131.9745	49.0	98.3

139 rows × 9 columns

In [76]:
```
pd.DataFrame(X_train,columns = x.columns)
```

Out[76]:

	population	fertility	HIV	CO2	BMI_male	GDP	BMI_female	life	child_mortality
131	46028476.0	1.38	1.10	7.032359	25.42379	8762.0	131.4962	68.2	12.9
44	1473741.0	4.28	5.30	1.079539	24.07620	15800.0	130.3625	57.5	68.0
126	6052937.0	4.88	3.20	0.251983	21.87875	1219.0	131.0248	60.0	96.4
99	6047131.0	3.06	0.30	0.698582	25.54223	6684.0	123.6150	73.6	25.7
53	748096.0	2.74	1.20	2.073415	23.68465	5208.0	125.1512	63.0	41.9
...	...	...	...	...	...	...	...	...	...
129	70344357.0	2.15	0.06	4.021903	26.70371	16454.0	124.0675	75.1	22.2
79	406392.0	1.38	0.10	6.182771	27.68361	27872.0	124.1571	81.4	6.6
133	304473143.0	2.07	0.60	18.545992	28.45698	50384.0	118.4777	78.2	7.7
72	3219802.0	1.42	0.10	4.498483	26.86102	23223.0	130.8226	72.0	8.2
37	6004199.0	2.32	0.80	1.067765	26.36751	7450.0	119.9321	74.1	21.6

104 rows × 9 columns

In [77]:
```
pd.DataFrame(X_train,columns = x.columns).describe()
```

Out[77]:

	population	fertility	HIV	CO2	BMI_male	GDP	BMI_female
count	1.040000e+02	104.000000	104.000000	104.000000	104.000000	104.000000	104.000000
mean	2.592682e+07	3.044904	2.110962	4.274682	24.609105	16234.682692	126.682877
std	4.254939e+07	1.516544	4.464762	6.551241	2.194723	19331.699025	4.577898
min	2.773150e+05	1.280000	0.060000	0.008618	20.397420	588.000000	117.552800
25%	3.629256e+06	1.867500	0.100000	0.459517	22.512847	2881.750000	123.097100
50%	1.041772e+07	2.450000	0.450000	1.699095	25.132705	8527.500000	126.456800
75%	2.931917e+07	4.212500	1.625000	6.051902	26.411750	21538.500000	130.416600
max	3.044731e+08	6.810000	25.900000	48.702062	28.456980	126076.000000	135.492000

Model Building

In [61]:
```
clf =KNeighborsClassifier(n_neighbors = 5)    #k=5
```

In [62]:
```
clf.fit(X_train,y_train)
```

Out[62]:
```
KNeighborsClassifier()
```

In [63]:

```
y_pred = clf.predict(X_test)
```

In [81]:

```
print(classification_report(y_pred,y_test))
```

```
                            precision    recall  f1-score   support

                   America       1.00      0.56      0.71         9
       East Asia & Pacific       0.50      1.00      0.67         2
     Europe & Central Asia       0.92      0.80      0.86        15
Middle East & North Africa       0.00      0.00      0.00         0
                South Asia       0.33      1.00      0.50         1
        Sub-Saharan Africa       1.00      1.00      1.00         8

                  accuracy                           0.80        35
                 macro avg       0.63      0.73      0.62        35
              weighted avg       0.92      0.80      0.83        35


/Users/SUMAZ/opt/anaconda3/lib/python3.8/site-packages/sklearn/metrics/_c
lassification.py:1221: UndefinedMetricWarning: Recall and F-score are ill
-defined and being set to 0.0 in labels with no true samples. Use `zero_d
ivision` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [ ]: