In [126…

```
import warnings
warnings.filterwarnings("ignore")
import pandas as pd
import numpy as np
import wbgapi as wb
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from kneed import KneeLocator
from sklearn.preprocessing import normalize
```

Clustering

In [127…

```
country_codes = ['TUR','GHA','IND']
ind1=["EN.ATM.CO2E.KT"]  # C02 Emission
ind2=["EG.ELC.COAL.ZS"]  # Electricity production from coal source
```

In [128…

```
my_dataframe1  = wb.data.DataFrame(ind1, country_codes, mrv=50).T
my_dataframe1=my_dataframe1.fillna(my_dataframe1.mean())
my_dataframe1.head()
```

Out[128…

economy	GHA	IND	TUR
YR1990	2790.0	562480.0	139220.0
YR1991	2560.0	607340.0	143820.0
YR1992	3080.0	630260.0	149830.0
YR1993	3190.0	652470.0	155630.0
YR1994	3460.0	687950.0	153930.0

In [129…

```
my_dataframe2  = wb.data.DataFrame(ind2, country_codes, mrv=50).T*10000
my_dataframe2=my_dataframe2.fillna(my_dataframe2.mean())
my_dataframe2
```

Out[129…

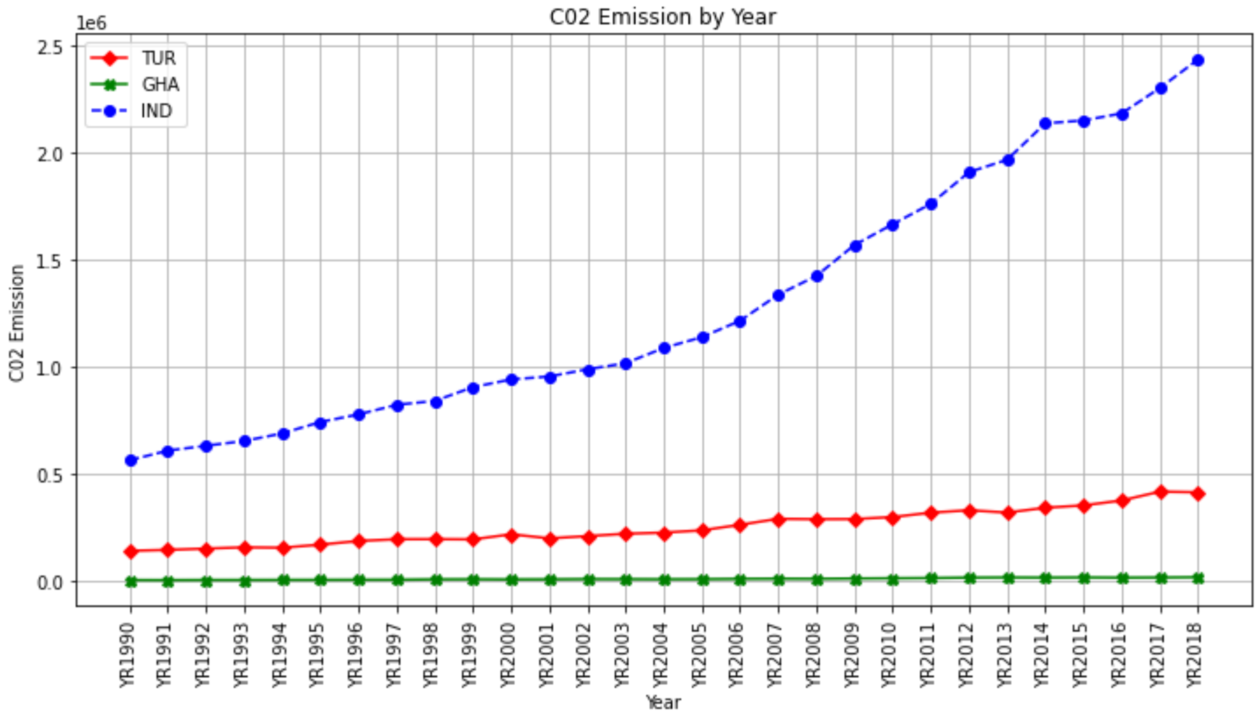
economy	GHA	IND	TUR
YR1966	0.0	634417.364517	471266.438480
YR1967	0.0	634417.364517	341651.907905
YR1968	0.0	634417.364517	321943.483276
YR1969	0.0	634417.364517	306072.977800
YR1970	0.0	634417.364517	327496.231010
YR1971	0.0	490916.485900	304774.562928

YR1972	0.0	520151.454989	259806.101574
YR1973	0.0	493928.237815	261086.519115
YR1974	0.0	535186.102924	287230.095719
YR1975	0.0	513348.695389	263265.698009
YR1976	0.0	541484.076766	236667.942898
YR1977	0.0	529940.967168	237879.893022
YR1978	0.0	483206.058350	256328.822609
YR1979	0.0	498763.528067	285853.831809
YR1980	0.0	510443.571494	256068.743287
YR1981	0.0	534595.329857	248733.433308
YR1982	0.0	568192.420237	242580.596565
YR1983	0.0	572773.042877	313672.432077
YR1984	0.0	581204.437954	330513.180675
YR1985	0.0	623577.270724	439171.220667
YR1986	0.0	636865.044541	489633.455095
YR1987	0.0	688641.915218	398011.408473
YR1988	0.0	674017.681729	259865.134865
YR1989	0.0	680510.488636	389478.133887
YR1990	0.0	654636.322643	350711.641729
YR1991	0.0	670678.076528	357882.681008
YR1992	0.0	680382.346574	364868.878263
YR1993	0.0	700533.470049	321929.872101
YR1994	0.0	677073.178221	360503.568647
YR1995	0.0	699362.696502	325193.919789
YR1996	0.0	704767.837917	320602.559507
YR1997	0.0	689321.271563	327795.848823
YR1998	0.0	667956.986686	321449.802742
YR1999	0.0	669578.760284	318026.451391
YR2000	0.0	684994.242463	305686.748531
YR2001	0.0	694170.572844	313025.056020
YR2002	0.0	698934.647225	248446.676971
YR2003	0.0	678314.232289	229419.338317
YR2004	0.0	676916.801537	228589.629590
YR2005	0.0	668596.364734	266689.718195

YR2006	0.0	666882.489170	264607.286485
YR2007	0.0	654820.578389	278928.575157
YR2008	0.0	684446.959115	290880.867663
YR2009	0.0	669680.214368	285834.548180
YR2010	0.0	671782.980878	260629.332222
YR2011	0.0	667599.782604	288661.816184
YR2012	0.0	716360.550608	283983.866119
YR2013	0.0	725227.381120	265604.570401
YR2014	0.0	744921.858695	302675.392816
YR2015	0.0	753093.989605	290950.902083

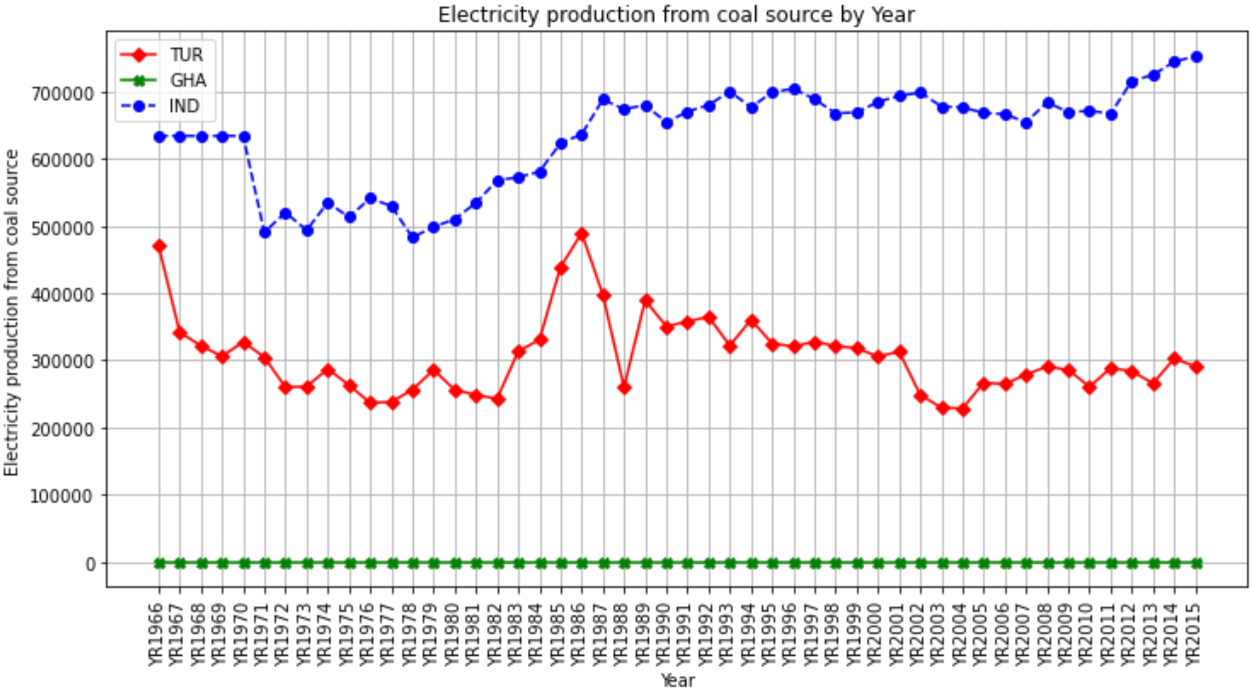
In [130…

```
plt.figure(figsize=(12,6))
plt.title('C02 Emission by Year')
plt.plot(my_dataframe1['TUR'],"rD-",label="TUR")
plt.plot(my_dataframe1['GHA'],"gX-",label="GHA")
plt.plot(my_dataframe1['IND'],"bo--",label="IND")
plt.xlabel("Year")
plt.xticks(rotation=90)
plt.ylabel("C02 Emission")
plt.legend(loc="best")
plt.grid()
plt.show()
```



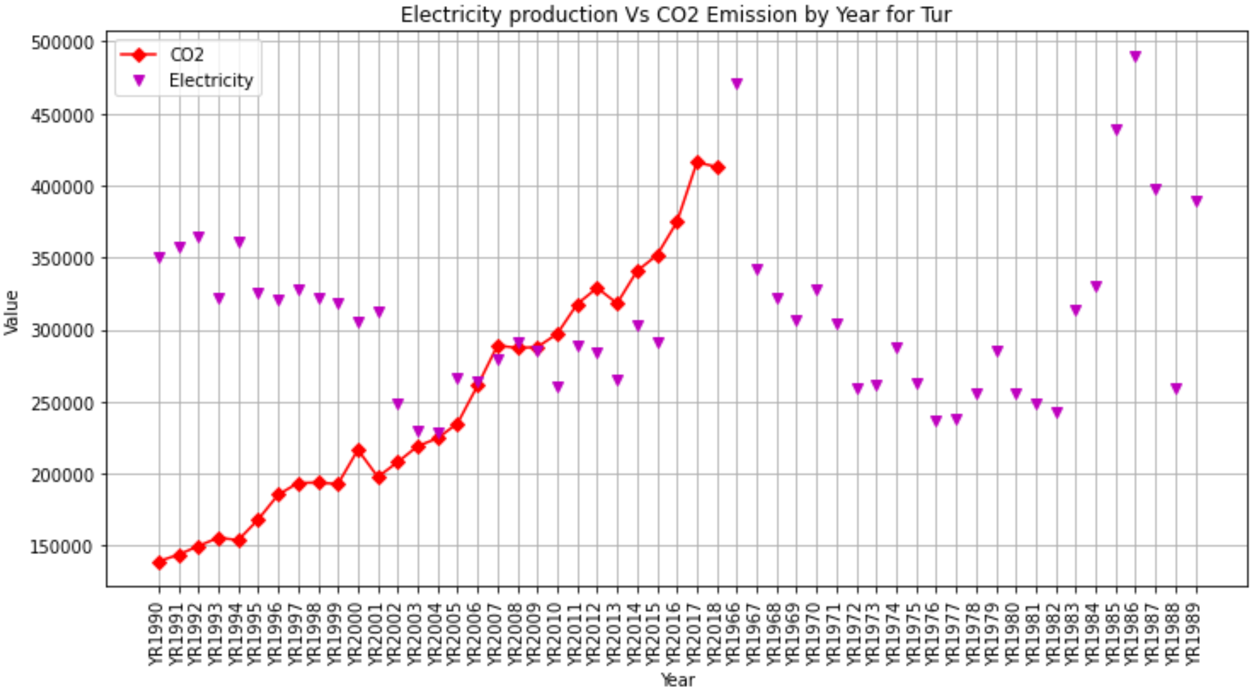
In [131…

```
plt.figure(figsize=(12,6))
plt.title('Electricity production from coal source by Year')
plt.plot(my_dataframe2['TUR'],"rD-",label="TUR")
plt.plot(my_dataframe2['GHA'],"gX-",label="GHA")
plt.plot(my_dataframe2['IND'],"bo--",label="IND")
plt.xlabel("Year")
plt.xticks(rotation=90)
plt.ylabel("Electricity production from coal source")
plt.grid()
plt.legend(loc="best")
plt.show()
```



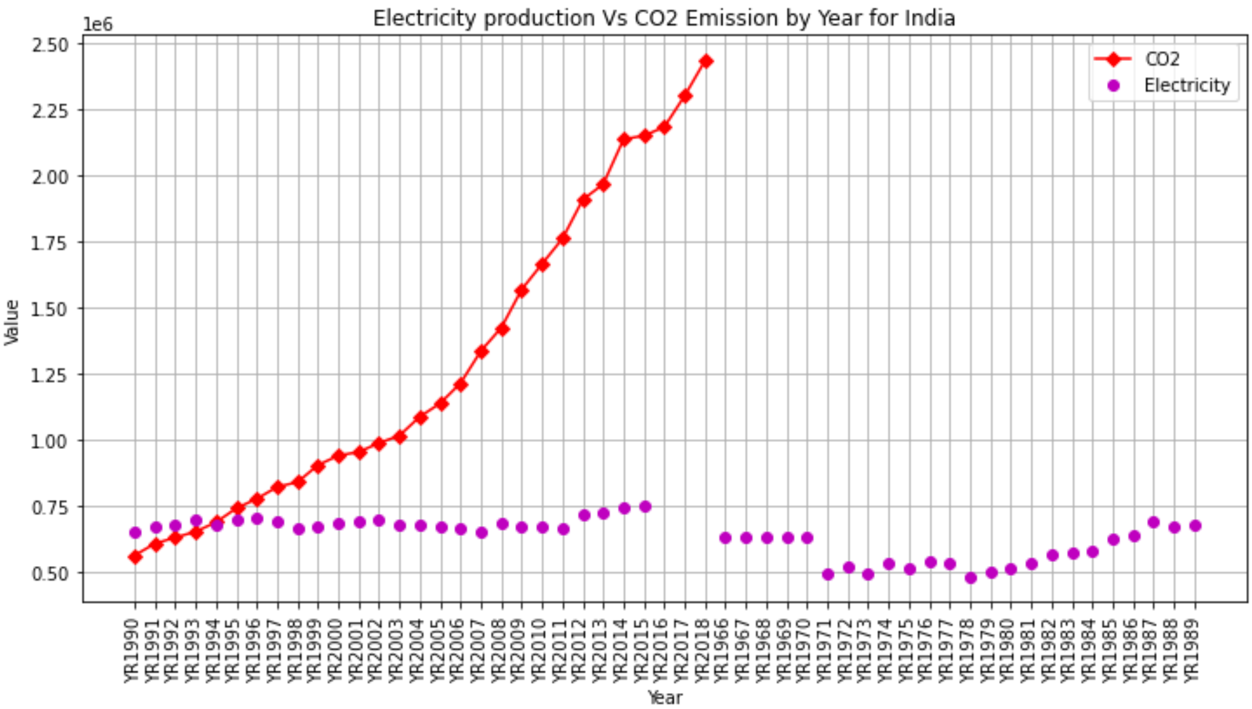
In [132…

```
plt.figure(figsize=(12,6))
plt.title('Electricity production Vs CO2 Emission by Year for Tur')
plt.plot(my_dataframe1['TUR'],"rD-",label="CO2")
plt.plot(my_dataframe2['TUR'],"mv",label="Electricity")
plt.xlabel("Year")
plt.xticks(rotation=90)
plt.ylabel("Value")
plt.legend(loc="best")
plt.grid()
plt.show()
```



In [133…

```
plt.figure(figsize=(12,6))
plt.title('Electricity production Vs CO2 Emission by Year for India')
plt.plot(my_dataframe1['IND'],"rD-",label="CO2")
plt.plot(my_dataframe2['IND'],"mo",label="Electricity")
plt.xlabel("Year")
plt.xticks(rotation=90)
plt.ylabel("Value")
plt.legend(loc="best")
plt.grid()
plt.show()
```



In [134…

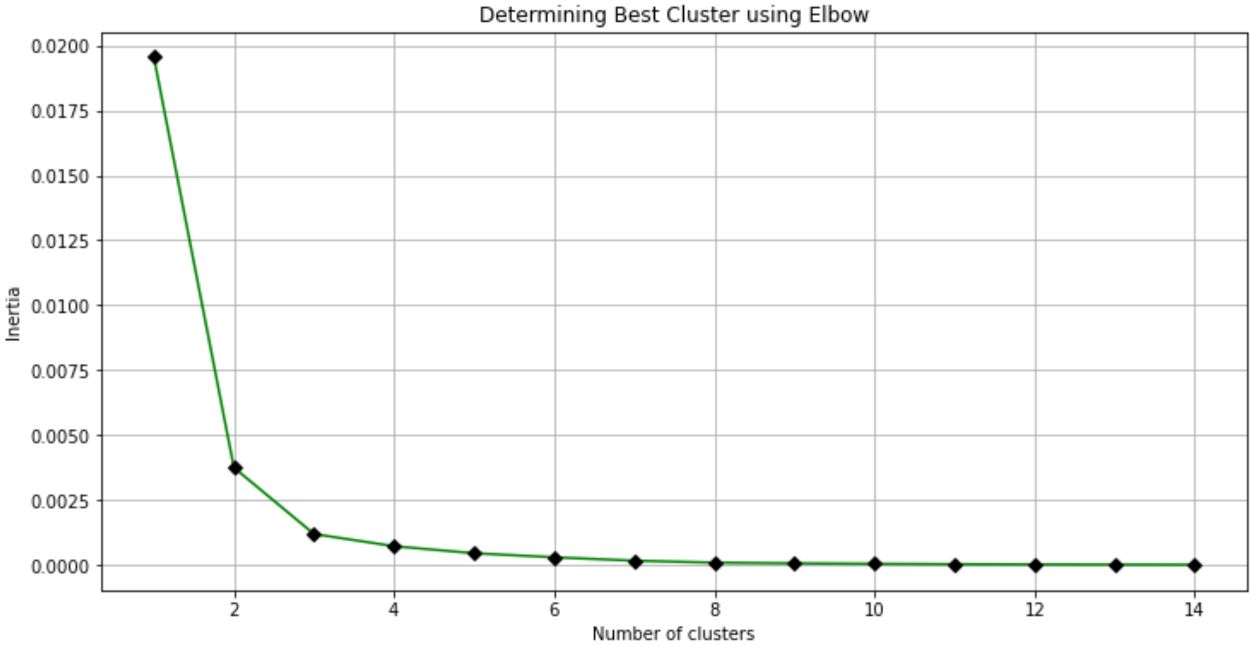
```
def normlz(data):
    nm=normalize(data)
    return nm
data=normlz(my_dataframe1.values)
data
```

Out[134…

```
array([[0.00481483, 0.97069707, 0.24025822],
       [0.00410163, 0.97308061, 0.23042851],
       [0.00475432, 0.97287582, 0.23127913],
       [0.00475565, 0.97270108, 0.23201292],
       [0.00490802, 0.97585819, 0.21834995],
       [0.00488177, 0.97517007, 0.22140348],
       [0.0051217 , 0.97264748, 0.23222973],
       [0.00508298, 0.97332513, 0.22937383],
       [0.00707711, 0.97435584, 0.22490133],
       [0.00711769, 0.97796862, 0.20863057],
       [0.0059493 , 0.97445163, 0.22451866],
       [0.00637454, 0.97915377, 0.20302033],
       [0.007352  , 0.97847958, 0.20621264],
       [0.00686048, 0.97748882, 0.21087564],
       [0.00596307, 0.97930299, 0.20231188],
       [0.00604246, 0.97930815, 0.20228454],
       [0.00678727, 0.97763225, 0.21021207],
       [0.00662023, 0.97738013, 0.21138652],
       [0.00589296, 0.98024142, 0.19771704],
       [0.00623922, 0.9835578 , 0.18048583],
       [0.00659122, 0.98443245, 0.17563971],
       [0.00682399, 0.9840847 , 0.17756894],
       [0.00734219, 0.98544944, 0.16981018],
       [0.00753349, 0.98713863, 0.15968899],
       [0.00664079, 0.98750948, 0.1574196 ],
       [0.0069441 , 0.98687011, 0.16136659],
       [0.00636955, 0.98557869, 0.16909783],
       [0.00631958, 0.98404112, 0.17782897],
       [0.00652398, 0.98589492, 0.16723832]])
```

In [135…

```
wcss = []
for i in range(1, 15):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300,  random_s
    kmeans.fit(data)
    wcss.append(kmeans.inertia_)
plt.figure(figsize=(12,6))
plt.title('Determining Best Cluster using Elbow')
plt.plot(range(1, 15), wcss,"g")
plt.plot(range(1, 15), wcss,"Dk")
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.grid()
plt.show()
```



In [136…

```
kmeans = KMeans(n_clusters=3, init='k-means++', max_iter=300, n_init=10, ra
pred_y = kmeans.fit(data)
```

In [137…

```
kmeans.cluster_centers_
```

Out[137…

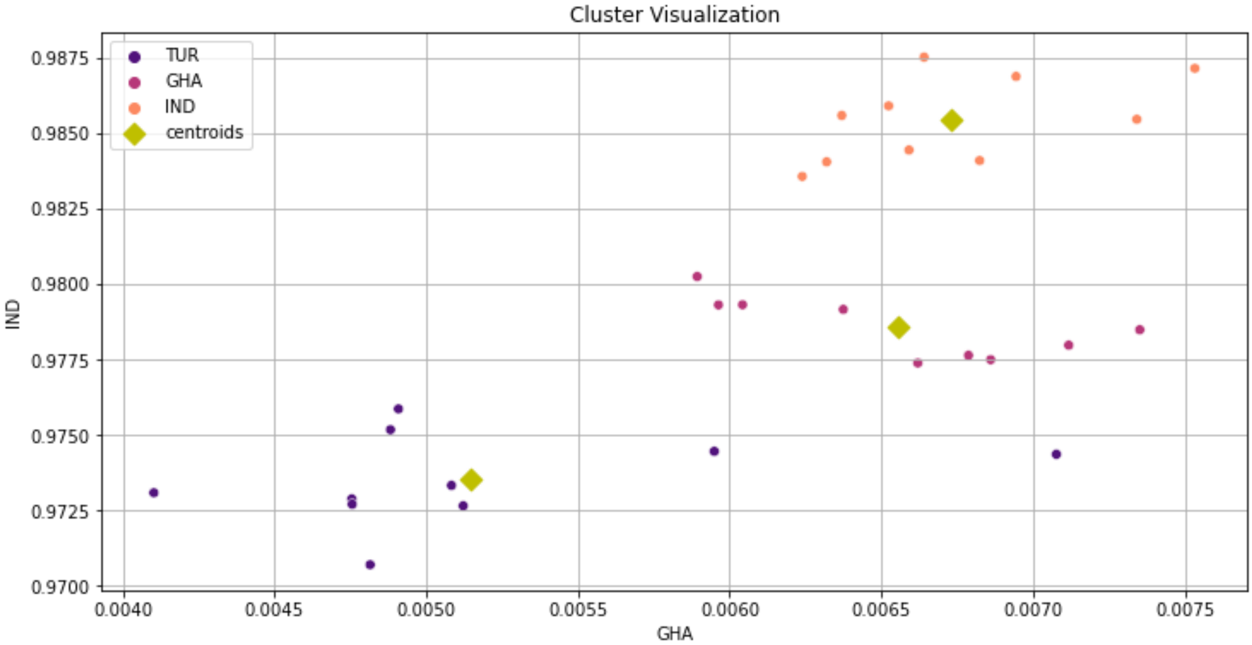
```
array([[0.00655674, 0.97855064, 0.20585014],
       [0.00673281, 0.98545573, 0.1696145 ],
       [0.00514473, 0.97351629, 0.22847558]])
```

In [138…

```
klb=[]
for i in kmeans.labels_:
    if i==0:
        klb.append(my_dataframe1.columns[0])
    elif i==1:
        klb.append(my_dataframe1.columns[1])
    elif i==2:
        klb.append(my_dataframe1.columns[2])
```

In [139…

```
df=pd.DataFrame(data,columns=my_dataframe1.columns)
plt.figure(figsize=(12,6))
plt.title('Cluster Visualization')
sns.scatterplot(data=df, x=my_dataframe1.columns[0], y=my_dataframe1.colum
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], ma
plt.grid()
plt.legend()
plt.show()
```



Curve Fitting

In [144…

```
from scipy.optimize import curve_fit
#!pip install lmfit
from lmfit import Model
```

In [145…

```
def gaussian(x, amp, cen, wid):
    return (amp / (np.sqrt(2*np.pi) * wid)) * np.exp(-(x-cen)**2 / (2*wid*
```

In [146…

```
norml2 = normlz(my_dataframe2.values)

y = gaussian(norml2[:,1], 2.33, 0.21, 1.51) + np.random.normal(0, 0.2, norm

init_vals = [1, 0, 1]
best_vals, covar = curve_fit(gaussian, norml2[:,1], y, p0=init_vals)
gmodel = Model(gaussian)
```

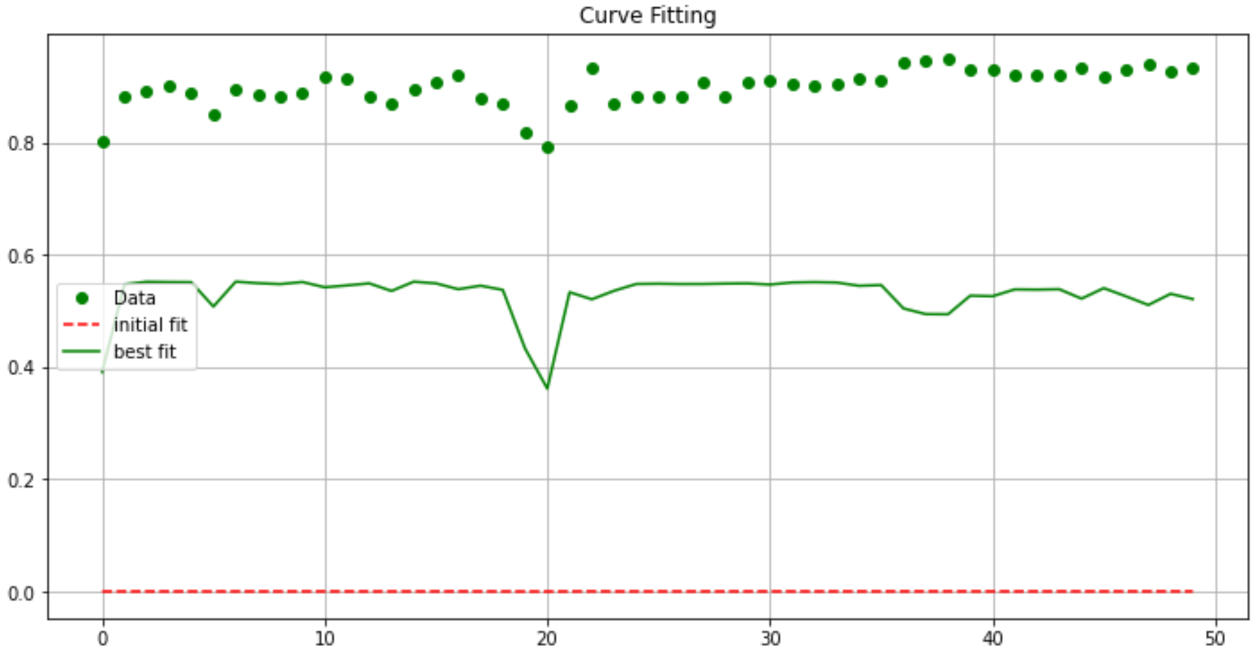
In [147…

```
result = gmodel.fit(y, x=norml2[:,1], amp=5, cen=5, wid=1)

print(result.fit_report())

plt.figure(figsize=(12,6))
plt.title('Curve Fitting')
plt.plot(norml2[:,1],"go",label="Data")
plt.plot(result.init_fit, 'r--', label='initial fit')
plt.plot(result.best_fit, 'g-', label='best fit')
plt.legend()
plt.grid()
plt.show()
```

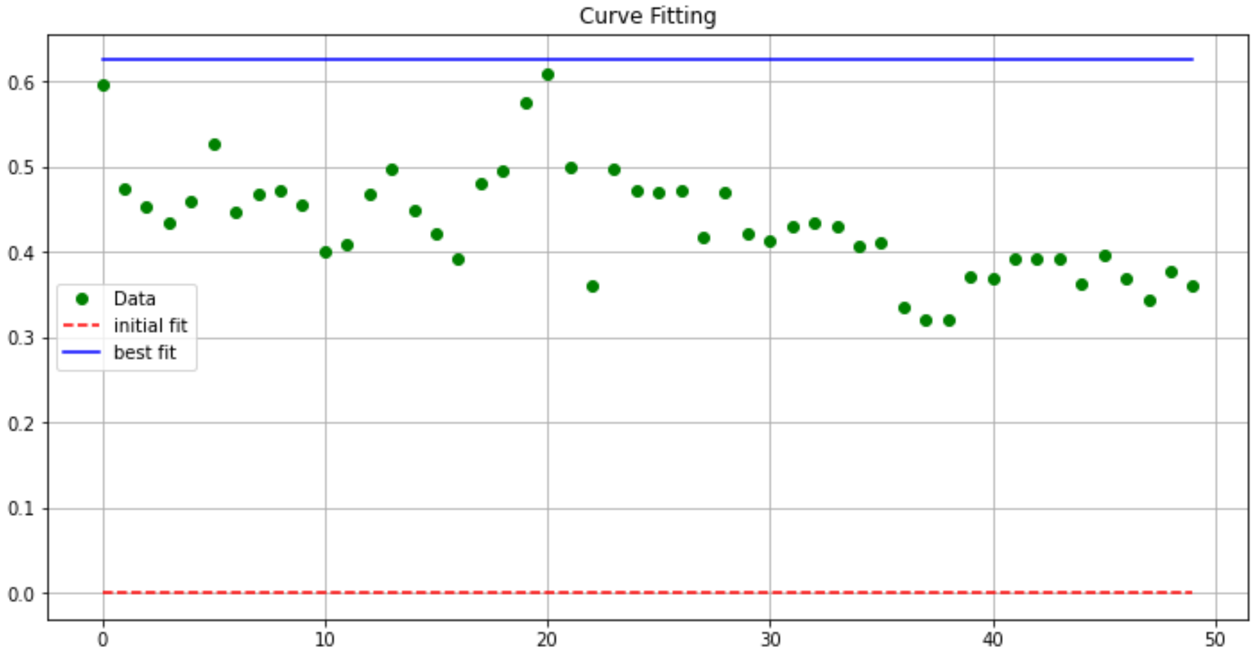
```
[[Model]]
    Model(gaussian)
[[Fit Statistics]]
    # fitting method   = leastsq
    # function evals   = 1551
    # data points      = 50
    # variables        = 3
    chi-square         = 1.55118427
    reduced chi-square = 0.03300392
    Akaike info crit   = -167.650216
    Bayesian info crit = -161.914147
[[Variables]]
    amp:  0.15379860 +/- 0.05974957 (38.85%) (init = 5)
    cen:  0.89503041 +/- 0.02227176 (2.49%) (init = 5)
    wid:  0.11113639 +/- 0.04642177 (41.77%) (init = 1)
[[Correlations]] (unreported correlations are < 0.100)
    C(amp, wid) = 0.992
    C(cen, wid) = 0.364
    C(amp, cen) = 0.340
```



In [150…

```
y = gaussian(norml2[:,2], 2.33, 0.21, 1.51) + np.random.normal(0, 0.2, norm

init_vals = [1, 0, 1]
best_vals, covar = curve_fit(gaussian, norml2[:,2], y, p0=init_vals)
gmodel = Model(gaussian)
result = gmodel.fit(y, x=norml2[:,0], amp=5, cen=5, wid=1)
plt.figure(figsize=(12,6))
plt.title('Curve Fitting')
plt.plot(norml2[:,2],"go",label="Data")
plt.plot(result.init_fit, 'r--', label='initial fit')
plt.plot(result.best_fit, 'b-', label='best fit')
plt.legend()
plt.grid()
plt.show()
```


Curve Fitting
Data
initial fit
best fit
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40
50

In [ ]: