# Assignment-1

# Name - Karishma Kansal

```
Uid-18bcs6103
```

In [1]:

```
# Supress Warnings
import warnings
warnings.filterwarnings('ignore')
```

Now we are inserting all libraries which are required in this program

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.model_selection import GridSearchCV
import os
```

Now we are reading or extracting data which is required.

In [3]:

```
car= pd.read_csv('cars_trucks_and_buses_per_1000_persons.csv',encoding="latin-1")
```

In [4]:

```
car.head()
```

Out[4]:

| | geo | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 |
|---|---|---|---|---|---|---|---|
| **0** | Afghanistan | NaN | NaN | NaN | NaN | NaN | 22.8 |
| **1** | Albania | 73.0 | NaN | 85.0 | 87.5 | 97.3 | 102.0 |
| **2** | Algeria | NaN | 88.0 | 89.0 | 91.0 | NaN | NaN |
| **3** | Angola | NaN | NaN | NaN | NaN | NaN | 39.6 |
| **4** | Argentina | NaN | NaN | NaN | NaN | NaN | 314.0 |

In [5]:

```
co2=pd.read_csv('co2_emissions_tonnes_per_person.csv',index_col='geo')
co2.head()
```

Out[5]:

| | 1800 | 1801 | 1802 | 1803 | 1804 | 1805 | 1806 | 1807 | 1808 | 1809 | ... | 2005 | 2006 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **geo** | | | | | | | | | | | | | |
| **Afghanistan** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 0.0529 | 0.0637 |
| **Albania** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 1.3800 | 1.2800 |
| **Algeria** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 3.2200 | 2.9900 |
| **Andorra** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 7.3000 | 6.7500 |
| **Angola** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 0.9800 | 1.1000 |

5 rows × 215 columns

In [6]:

```
co2.rename(columns = {'2014':'co2_emissions_tonnes_per_person'}, inplace = True)
```

In [7]:

```
d1=co2.pop('co2_emissions_tonnes_per_person')
d1.head()
```

Out[7]:

```
geo
Afghanistan    0.299
Albania        1.960
Algeria        3.720
Andorra        5.830
Angola         1.290
Name: co2_emissions_tonnes_per_person, dtype: float64
```

In [8]:

```
df1=pd.DataFrame(d1)
df1.head()
```

Out[8]:

| | co2_emissions_tonnes_per_person |
|---|---|
| **geo** | |
| **Afghanistan** | 0.299 |
| **Albania** | 1.960 |
| **Algeria** | 3.720 |
| **Andorra** | 5.830 |
| **Angola** | 1.290 |

In [9]:

```
coal=pd.read_csv('coal_consumption_per_cap.csv',index_col='geo')
```

In [10]:

```
coal.head()
```

Out[10]:

| | 1965 | 1966 | 1967 | 1968 | 1969 | 1970 | 1971 | 1972 | 1973 | 197 |
|---|---|---|---|---|---|---|---|---|---|---|
| **geo** | | | | | | | | | | |
| **Algeria** | 0.00554 | 0.00524 | 0.00389 | 0.0040 | 0.00495 | 0.0057 | 0.00154 | 0.0013 | 0.00146 | 0.0011 |
| **Argentina** | 0.03570 | 0.03690 | 0.03560 | 0.0282 | 0.03710 | 0.0409 | 0.03310 | 0.0291 | 0.03010 | 0.038C |
| **Australia** | 1.53000 | 1.55000 | 1.55000 | 1.5500 | 1.57000 | 1.5500 | 1.53000 | 1.5700 | 1.61000 | 1.660C |
| **Austria** | 0.69600 | 0.66000 | 0.62100 | 0.6100 | 0.59700 | 0.6390 | 0.58300 | 0.5270 | 0.52200 | 0.549C |
| **Azerbaijan** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | Na |

5 rows × 52 columns

In [11]:

```
coal.rename(columns = {'2014':'coal_consumption_per_cap'}, inplace = True)
```

In [12]:

```
d2=coal.pop('coal_consumption_per_cap')
d2.head()
```

Out[12]:

```
geo
Algeria       0.00458
Argentina     0.03460
Australia     1.82000
Austria       0.34700
Azerbaijan    0.00017
Name: coal_consumption_per_cap, dtype: float64
```

In [13]:

```
df2=pd.DataFrame(d2)
```

In [14]:

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 65 entries, Algeria to Vietnam
Data columns (total 1 columns):
coal_consumption_per_cap    65 non-null float64
dtypes: float64(1)
memory usage: 1.0+ KB
```

In [15]:

```
df2.head()
```

Out[15]:

| | coal_consumption_per_cap |
|---|---|
| **geo** | |
| **Algeria** | 0.00458 |
| **Argentina** | 0.03460 |
| **Australia** | 1.82000 |
| **Austria** | 0.34700 |
| **Azerbaijan** | 0.00017 |

In [16]:

```
electricity_gen=pd.read_csv('electricity_generation_per_person.csv',index_col='geo' )
electricity_gen.head()
```

Out[16]:

| | 1985 | 1986 | 1987 | 1988 | 1989 | 1990 | 1991 | 1992 | 1993 | 1994 | ... | 200 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **geo** | | | | | | | | | | | | |
| **Algeria** | 544.0 | 559.0 | 532.0 | 568.0 | 607.0 | 621.0 | 653.0 | 673 | 699 | 701.0 | ... | 109 |
| **Argentina** | 1490.0 | 1590.0 | 1660.0 | 1670.0 | 1580.0 | 1560.0 | 1620.0 | 1680 | 1810 | 1890.0 | ... | 288 |
| **Australia** | 7860.0 | 8100.0 | 8360.0 | 8670.0 | 9020.0 | 9130.0 | 9150.0 | 9230 | 9350 | 9510.0 | ... | 1160 |
| **Austria** | 5850.0 | 5860.0 | 6610.0 | 6400.0 | 6530.0 | 6530.0 | 6620.0 | 6540 | 6670 | 6710.0 | ... | 780 |
| **Azerbaijan** | 3110.0 | 3180.0 | 3320.0 | 3360.0 | 3270.0 | 3200.0 | 3170.0 | 2630 | 2520 | 2290.0 | ... | 250 |

5 rows × 32 columns

In [17]:

```
electricity_gen.rename(columns = {'2014':'electricity_generation_per_person'}, inplace = Tr
```

In [18]:

```
d3=electricity_gen.pop('electricity_generation_per_person')
```

In [19]:

```
df3=pd.DataFrame(d3)
df3.head()
```

Out[19]:

| geo | electricity_generation_per_person |
|---|---|
| **Algeria** | 1640 |
| **Argentina** | 3290 |
| **Australia** | 10500 |
| **Austria** | 7540 |
| **Azerbaijan** | 2600 |

In [20]:

```
electircity_use=pd.read_csv('electricity_use_per_person.csv',index_col='geo')
electircity_use.head()
```

Out[20]:

| geo | 1960 | 1961 | 1962 | 1963 | 1964 | 1965 | 1966 | 1967 | 1968 | 1969 | ... | 2005 | 2006 | : |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Albania** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 1720.0 | 1220.0 | 12 |
| **Algeria** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 887.0 | 859.0 | 8 |
| **Angola** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 109.0 | 144.0 | 1 |
| **Argentina** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 2390.0 | 2360.0 | 24 |
| **Armenia** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 1520.0 | 1640.0 | 17 |

5 rows × 55 columns

In [21]:

```
electircity_use.rename(columns = {'2014':'electricity_use_per_person'}, inplace = True)
```

In [22]:

```
d4=electircity_use.pop('electricity_use_per_person')
```

In [23]:

```
df4=pd.DataFrame(d4)
df4.head()
```

Out[23]:

| | electricity_use_per_person |
|---|---|
| **geo** | |
| **Albania** | 2310.0 |
| **Algeria** | 1360.0 |
| **Angola** | 312.0 |
| **Argentina** | 3050.0 |
| **Armenia** | 1970.0 |

In [24]:

```
forest=pd.read_csv('forest_coverage_percent.csv',index_col='geo')
forest.head()
```

Out[24]:

| | 1990 | 1991 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | ... | 2006 | 2( |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **geo** | | | | | | | | | | | | | |
| **Afghanistan** | 2.07 | 2.07 | 2.07 | 2.07 | 2.07 | 2.07 | 2.07 | 2.07 | 2.07 | 2.07 | ... | 2.07 | 2 |
| **Albania** | 28.80 | 28.70 | 28.60 | 28.60 | 28.50 | 28.40 | 28.40 | 28.30 | 28.20 | 28.10 | ... | 28.50 | 28 |
| **Algeria** | 0.70 | 0.70 | 0.69 | 0.69 | 0.69 | 0.68 | 0.68 | 0.67 | 0.67 | 0.67 | ... | 0.68 | 0 |
| **Andorra** | 34.00 | 34.00 | 34.00 | 34.00 | 34.00 | 34.00 | 34.00 | 34.00 | 34.00 | 34.00 | ... | 34.00 | 34 |
| **Angola** | 48.90 | 48.80 | 48.70 | 48.60 | 48.50 | 48.40 | 48.30 | 48.20 | 48.10 | 48.00 | ... | 47.30 | 47 |

5 rows × 26 columns

In [25]:

```
forest.rename(columns = {'2014':'forest_coverage_percent'}, inplace = True)
```

In [26]:

```
d5=forest.pop('forest_coverage_percent')
```

In [27]:

```
df5=pd.DataFrame(d5)
df5.head()
```

Out[27]:

| | forest_coverage_percent |
|---|---|
| **geo** | |
| **Afghanistan** | 2.07 |
| **Albania** | 28.20 |
| **Algeria** | 0.82 |
| **Andorra** | 34.00 |
| **Angola** | 46.50 |

In [28]:

```
hydro=pd.read_csv('hydro_power_generation_per_person.csv',index_col='geo')
hydro.head()
```

Out[28]:

| | 1960 | 1961 | 1962 | 1963 | 1964 | 1965 | 1966 | 1967 | 1968 | 1969 | ... | 2002 | 2003 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **geo** | | | | | | | | | | | | | |
| **Albania** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 0.09770 | 0.13500 |
| **Algeria** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 0.00016 | 0.00071 |
| **Angola** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 0.00660 | 0.00692 |
| **Argentina** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 0.08190 | 0.07650 |
| **Armenia** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 0.04660 | 0.05570 |

5 rows × 52 columns

now as in this dataframe 2014 column is not presentand we have to work only for column 2014 so we dont required this dataframe

In [29]:

```
income=pd.read_csv('income_per_person_gdppercapita_ppp_inflation_adjusted.csv',index_col='g
income.head()
```

Out[29]:

| | 1800 | 1801 | 1802 | 1803 | 1804 | 1805 | 1806 | 1807 | 1808 | 1809 | ... | 2009 | 2010 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **geo** | | | | | | | | | | | | | | |
| **Afghanistan** | 603 | 603 | 603 | 603 | 603 | 603 | 603 | 603 | 603 | 603 | ... | 1530 | 1610 | |
| **Albania** | 667 | 667 | 667 | 667 | 667 | 668 | 668 | 668 | 668 | 668 | ... | 9530 | 9930 | 1 |
| **Algeria** | 715 | 716 | 717 | 718 | 719 | 720 | 721 | 722 | 723 | 724 | ... | 12600 | 12900 | 1 |
| **Andorra** | 1200 | 1200 | 1200 | 1200 | 1210 | 1210 | 1210 | 1210 | 1220 | 1220 | ... | 41700 | 39000 | 4 |
| **Angola** | 618 | 620 | 623 | 626 | 628 | 631 | 634 | 637 | 640 | 642 | ... | 5910 | 5900 | |

5 rows × 219 columns

In [30]:

```
income.rename(columns = {'2014':'income_per_person_gdppercapita_ppp_inflation_adjusted'}, i
```

In [31]:

```
d6=income.pop('income_per_person_gdppercapita_ppp_inflation_adjusted')
```

In [32]:

```
df6=pd.DataFrame(d6);
df6.head();
```

In [33]:

```
indusrty=pd.read_csv('industry_percent_of_gdp.csv',index_col='geo')
```

In [34]:

```
indusrty.head()
```

Out[34]:

| | 1960 | 1961 | 1962 | 1963 | 1964 | 1965 | 1966 | 1967 | 1968 | 1969 | ... | 2008 | 2009 | 201 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **geo** | | | | | | | | | | | | | | |
| **Afghanistan** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 26.7 | 21.8 | 21. |
| **Albania** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 25.2 | 24.4 | 24. |
| **Algeria** | 29.4 | 29.5 | 31.7 | 40.8 | 42.7 | 42.2 | 46.5 | 47.8 | 47.7 | 47.8 | ... | 58.6 | 47.9 | 50. |
| **Andorra** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 15.5 | 14.4 | 13. |
| **Angola** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | Na |

5 rows × 58 columns

In [35]:

```
indusrty.rename(columns = {'2014':'industry_percent_of_gdp'}, inplace = True)
```

In [36]:

```
d7=indusrty.pop('industry_percent_of_gdp')
```

In [37]:

```
df7=pd.DataFrame(d7)
df7.head()
```

Out[37]:

| | industry_percent_of_gdp |
|---|---|
| **geo** | |
| **Afghanistan** | 21.10 |
| **Albania** | 21.50 |
| **Algeria** | 42.30 |
| **Andorra** | 9.91 |
| **Angola** | NaN |

In [38]:

```
naturalgas=pd.read_csv('natural_gas_production_per_person.csv',index_col='geo')
```

In [39]:

```
naturalgas.head()
```

Out[39]:

| | 1970 | 1971 | 1972 | 1973 | 1974 | 1975 | 1976 | 1977 | 1978 | 1979 | ... | 2007 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **geo** | | | | | | | | | | | | | |
| **Algeria** | 0.157 | 0.161 | 0.198 | 0.256 | 0.288 | 0.344 | 0.422 | 0.392 | 0.591 | 0.940 | ... | 2.23 | 2.2 |
| **Argentina** | 0.226 | 0.232 | 0.224 | 0.228 | 0.221 | 0.247 | 0.251 | 0.250 | 0.233 | 0.271 | ... | 1.01 | 0.9 |
| **Australia** | 0.122 | 0.179 | 0.252 | 0.316 | 0.356 | 0.377 | 0.440 | 0.497 | 0.530 | 0.603 | ... | 1.77 | 1.7 |
| **Azerbaijan** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 1.01 | 1.5 |
| **Bahrain** | 2.630 | 3.700 | 4.440 | 6.020 | 7.060 | 7.020 | 6.910 | 7.020 | 7.220 | 7.560 | ... | 10.20 | 10.2 |

5 rows × 47 columns

In [40]:

```
naturalgas.rename(columns = {'2014':'natural_gas_production_per_person'}, inplace = True)
```

In [41]:

```
d8=naturalgas.pop('natural_gas_production_per_person')
```

In [42]:

```
df8=pd.DataFrame(d8)
df8.head()
```

Out[42]:

| | natural_gas_production_per_person |
|---|---|
| **geo** | |
| **Algeria** | 1.920 |
| **Argentina** | 0.742 |
| **Australia** | 2.440 |
| **Azerbaijan** | 1.660 |
| **Bahrain** | 10.400 |

In [43]:

```
oilconsumption=pd.read_csv('oil_consumption_per_cap.csv',index_col='geo')
oilconsumption.head()
```

Out[43]:

| | 1965 | 1966 | 1967 | 1968 | 1969 | 1970 | 1971 | 1972 | 1973 | 1974 | ... | 2007 | 200 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **geo** | | | | | | | | | | | | | |
| **Algeria** | 0.102 | 0.130 | 0.118 | 0.122 | 0.125 | 0.140 | 0.153 | 0.163 | 0.173 | 0.187 | ... | 0.376 | 0.40 |
| **Argentina** | 0.990 | 1.010 | 1.020 | 1.020 | 1.050 | 0.923 | 0.969 | 0.954 | 0.939 | 0.930 | ... | 0.605 | 0.61 |
| **Australia** | 1.330 | 1.550 | 1.650 | 1.750 | 1.760 | 1.900 | 1.960 | 1.970 | 2.070 | 2.160 | ... | 2.030 | 2.02 |
| **Austria** | 0.761 | 0.832 | 0.880 | 1.010 | 1.110 | 1.210 | 1.350 | 1.450 | 1.560 | 1.390 | ... | 1.620 | 1.60 |
| **Azerbaijan** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 0.519 | 0.40 |

5 rows × 52 columns

In [44]:

```
oilconsumption.rename(columns = {'2014':'oil_consumption_per_cap'}, inplace = True)
```

In [45]:

```
d9=oilconsumption.pop('oil_consumption_per_cap')
```

In [46]:

```
df9=pd.DataFrame(d9)
df9.head()
```

Out[46]:

| | oil_consumption_per_cap |
|---|---|
| **geo** | |
| **Algeria** | 0.452 |
| **Argentina** | 0.729 |
| **Australia** | 2.050 |
| **Austria** | 1.440 |
| **Azerbaijan** | 0.468 |

In [47]:

```
oilproduction=pd.read_csv('oil_production_per_person.csv',index_col='geo')
oilproduction.head()
```

Out[47]:

| geo | 1965 | 1966 | 1967 | 1968 | 1969 | 1970 | 1971 | 1972 | 1973 | 1974 | ... | 2007 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Algeria** | 2.1000 | 2.6100 | 2.9300 | 3.120 | 3.170 | 3.310 | 2.480 | 3.260 | 3.220 | 2.910 | ... | 2.520 |
| **Angola** | 0.1060 | 0.1000 | 0.0837 | 0.115 | 0.370 | 0.747 | 0.826 | 0.995 | 1.120 | 1.140 | ... | 3.930 |
| **Argentina** | 0.6180 | 0.6480 | 0.6960 | 0.752 | 0.767 | 0.834 | 0.885 | 0.896 | 0.857 | 0.825 | ... | 0.957 |
| **Australia** | 0.0305 | 0.0382 | 0.0869 | 0.158 | 0.177 | 0.678 | 1.170 | 1.220 | 1.510 | 1.480 | ... | 1.170 |
| **Azerbaijan** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 4.880 |

5 rows × 52 columns

In [48]:

```
oilproduction.rename(columns = {'2014':'oil_production_per_person'}, inplace = True)
```

In [49]:

```
d10=oilproduction.pop('oil_production_per_person')
```

In [50]:

```
df10=pd.DataFrame(d10)
df10.head()
```

Out[50]:

| geo | oil_production_per_person |
|---|---|
| **Algeria** | 1.760 |
| **Angola** | 3.080 |
| **Argentina** | 0.695 |
| **Australia** | 0.813 |
| **Azerbaijan** | 4.430 |

In [51]:

```
yearlyCo2=pd.read_csv('yearly_co2_emissions_1000_tonnes.csv',index_col='geo')
yearlyCo2.head()
```

Out[51]:

| | 1751 | 1752 | 1753 | 1754 | 1755 | 1756 | 1757 | 1758 | 1759 | 1760 | ... | 2005 | 2( |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **geo** | | | | | | | | | | | | | |
| **Afghanistan** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 1330.0 | 165 |
| **Albania** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 4250.0 | 390 |
| **Algeria** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 107000.0 | 10100 |
| **Andorra** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 576.0 | 54 |
| **Angola** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 19200.0 | 2230 |

5 rows × 264 columns

In [52]:

```
yearlyCo2.rename(columns = {'2014':'yearly_co2_emissions_1000_tonnes'}, inplace = True)
```

In [53]:

```
d11=yearlyCo2.pop('yearly_co2_emissions_1000_tonnes')
```

In [54]:

```
df11=pd.DataFrame(d11)
df11.head()
```

Out[54]:

| | yearly_co2_emissions_1000_tonnes |
|---|---|
| **geo** | |
| **Afghanistan** | 9810.0 |
| **Albania** | 5720.0 |
| **Algeria** | 145000.0 |
| **Andorra** | 462.0 |
| **Angola** | 34800.0 |

Now we are merging dataframes by taking two at a time.

In [55]:

```
dff1=pd.merge(df1, df2, left_index=True, right_index=True)
```

In [56]:

```
dff1.head()
```

Out[56]:

| geo | co2_emissions_tonnes_per_person | coal_consumption_per_cap |
|---|---|---|
| Algeria | 3.72 | 0.00458 |
| Argentina | 4.75 | 0.03460 |
| Australia | 15.40 | 1.82000 |
| Austria | 6.80 | 0.34700 |
| Azerbaijan | 3.94 | 0.00017 |

In [57]:

```
dff2=pd.merge(df3,df4,left_index=True,right_index=True)
dff2.head()
```

Out[57]:

| geo | electricity_generation_per_person | electricity_use_per_person |
|---|---|---|
| Algeria | 1640 | 1360.0 |
| Argentina | 3290 | 3050.0 |
| Australia | 10500 | 10100.0 |
| Austria | 7540 | 8360.0 |
| Azerbaijan | 2600 | 2200.0 |

In [58]:

```
dff3=pd.merge(df5, df6, left_index=True, right_index=True)
dff3.head()
```

Out[58]:

| geo | forest_coverage_percent | income_per_person_gdppercapita_ppp_inflation_adjusted |
|---|---|---|
| Afghanistan | 2.07 | 1780 |
| Albania | 28.20 | 10700 |
| Algeria | 0.82 | 13500 |
| Andorra | 34.00 | 44900 |
| Angola | 46.50 | 6260 |

In [59]:

```
dff4=pd.merge(df7, df8, left_index=True, right_index=True)
```

In [60]:

```
dff4.head()
```

Out[60]:

| geo | industry_percent_of_gdp | natural_gas_production_per_person |
|---|---|---|
| Algeria | 42.3 | 1.920 |
| Argentina | 24.3 | 0.742 |
| Australia | 25.4 | 2.440 |
| Azerbaijan | 53.6 | 1.660 |
| Bahrain | 46.5 | 10.400 |

In [61]:

```
dff5=pd.merge(df9, df10, left_index=True, right_index=True)
dff5.head()
```

Out[61]:

| geo | oil_consumption_per_cap | oil_production_per_person |
|---|---|---|
| Algeria | 0.452 | 1.760 |
| Argentina | 0.729 | 0.695 |
| Australia | 2.050 | 0.813 |
| Azerbaijan | 0.468 | 4.430 |
| Brazil | 0.737 | 0.600 |

In [62]:

```
dff6=pd.merge(dff5, df11, left_index=True, right_index=True)
```

In [63]:

```
dff6.head()
```

Out[63]:

| geo | oil_consumption_per_cap | oil_production_per_person | yearly_co2_emissions_1000_tonı |
|---|---|---|---|
| Algeria | 0.452 | 1.760 | 14500 |
| Argentina | 0.729 | 0.695 | 20400 |
| Australia | 2.050 | 0.813 | 36100 |
| Azerbaijan | 0.468 | 4.430 | 3750 |
| Brazil | 0.737 | 0.600 | 53000 |

In [64]:

```
dff7=pd.merge(dff4, dff6, left_index=True, right_index=True)
```

In [65]:

```
dff7.head()
```

Out[65]:

| geo | industry_percent_of_gdp | natural_gas_production_per_person | oil_consumption_per_ca |
|---|---|---|---|
| Algeria | 42.3 | 1.920 | 0.45 |
| Argentina | 24.3 | 0.742 | 0.72 |
| Australia | 25.4 | 2.440 | 2.05 |
| Azerbaijan | 53.6 | 1.660 | 0.46 |
| Brazil | 20.5 | 0.100 | 0.73 |

In [66]:

```
dff8=pd.merge(dff7, dff3, left_index=True, right_index=True)
```

In [67]:

```
dff8.head()
```

Out[67]:

| geo | industry_percent_of_gdp | natural_gas_production_per_person | oil_consumption_per_ca |
|---|---|---|---|
| Algeria | 42.3 | 1.920 | 0.45 |
| Argentina | 24.3 | 0.742 | 0.72 |
| Australia | 25.4 | 2.440 | 2.05 |
| Azerbaijan | 53.6 | 1.660 | 0.46 |
| Brazil | 20.5 | 0.100 | 0.73 |

In [68]:

```
dff9=pd.merge(dff8, dff2, left_index=True, right_index=True)
```

In [69]:

```
dff9.head()
```

Out[69]:

| geo | industry_percent_of_gdp | natural_gas_production_per_person | oil_consumption_per_ca |
|---|---|---|---|
| Algeria | 42.3 | 1.920 | 0.45 |
| Argentina | 24.3 | 0.742 | 0.72 |
| Australia | 25.4 | 2.440 | 2.05 |
| Azerbaijan | 53.6 | 1.660 | 0.46 |
| Brazil | 20.5 | 0.100 | 0.73 |

In [70]:

```
df=pd.merge(dff1, dff9, left_index=True, right_index=True)
```

Finally we get dataframe by merging different dataframes and this dataframe is our required dataframe.

In [71]:

```
df.head()
```

Out[71]:

| | co2_emissions_tonnes_per_person | coal_consumption_per_cap | industry_percent_of_g |
|---|---|---|---|
| **geo** | | | |
| **Algeria** | 3.72 | 0.00458 | 4. |
| **Argentina** | 4.75 | 0.03460 | 2. |
| **Australia** | 15.40 | 1.82000 | 2. |
| **Azerbaijan** | 3.94 | 0.00017 | 5. |
| **Brazil** | 2.59 | 0.08580 | 2. |

In [72]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 33 entries, Algeria to Vietnam
Data columns (total 11 columns):
co2_emissions_tonnes_per_person                       33 non-null float64
coal_consumption_per_cap                              33 non-null float64
industry_percent_of_gdp                               33 non-null float64
natural_gas_production_per_person                     33 non-null float64
oil_consumption_per_cap                               33 non-null float64
oil_production_per_person                             33 non-null float64
yearly_co2_emissions_1000_tonnes                      33 non-null float64
forest_coverage_percent                               33 non-null float64
income_per_person_gdppercapita_ppp_inflation_adjusted 33 non-null int64
electricity_generation_per_person                     33 non-null int64
electricity_use_per_person                            33 non-null float64
dtypes: float64(9), int64(2)
memory usage: 3.1+ KB
```

# Checking for outliers

In [73]:

```
df.describe()
```

Out[73]:

| | co2_emissions_tonnes_per_person | coal_consumption_per_cap | industry_percent_of_gdp |
|---|---|---|---|
| count | 33.000000 | 33.000000 | 33.000000 |
| mean | 10.051515 | 0.357886 | 37.066667 |
| std | 10.074812 | 0.570206 | 13.844509 |
| min | 1.730000 | 0.000000 | 17.800000 |
| 25% | 3.500000 | 0.006410 | 27.700000 |
| 50% | 6.030000 | 0.161000 | 33.200000 |
| 75% | 14.200000 | 0.457000 | 42.300000 |
| max | 45.400000 | 2.340000 | 70.500000 |

In [74]:

```
# data visualization
from sklearn.model_selection import train_test_split
```

In [75]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [76]:

```
sns.pairplot(df)
plt.show()
```

This is quite hard to read, and we can rather plot correlations between variables. Also, a heatmap is pretty useful to visualise multiple correlations in one plot.

In [77]:

```
# correlation matrix
cor = df.corr()
cor
```

Out[77]:

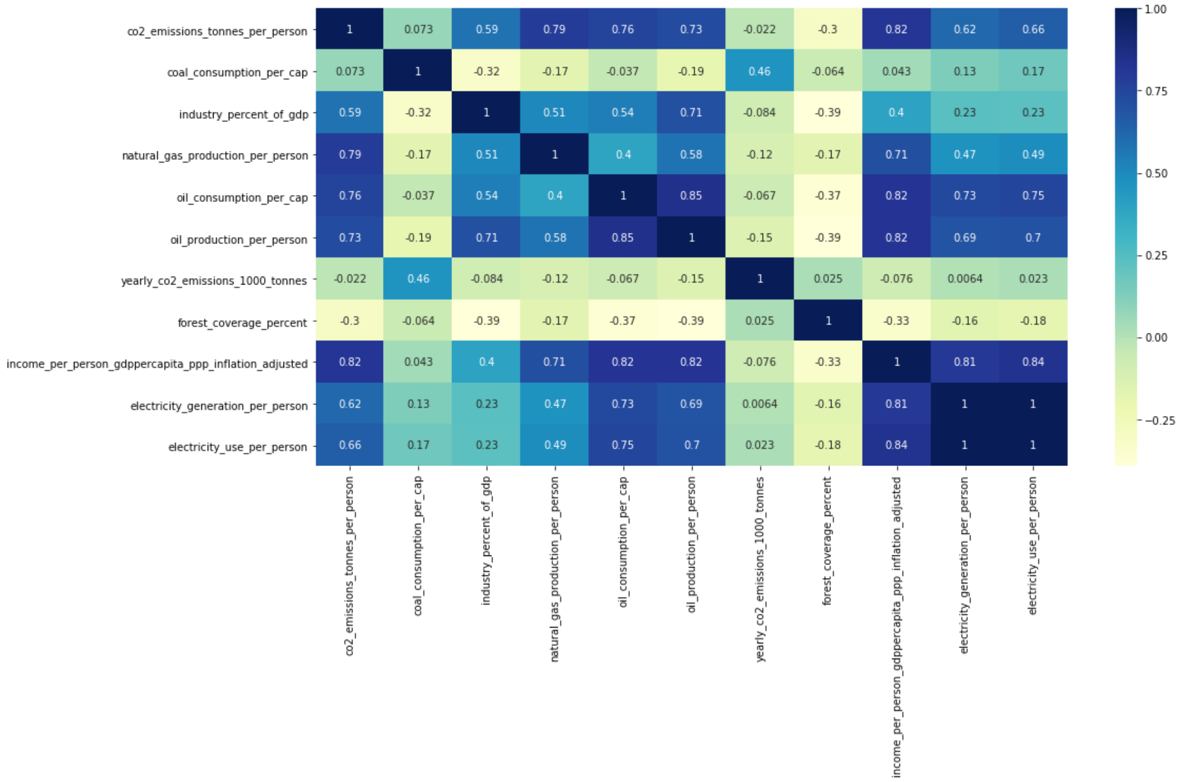
| | co2_emissions_tonnes_per_person | coal_ |
|---|---|---|
| **co2_emissions_tonnes_per_person** | 1.000000 | |
| **coal_consumption_per_cap** | 0.072586 | |
| **industry_percent_of_gdp** | 0.585031 | |
| **natural_gas_production_per_person** | 0.789657 | |
| **oil_consumption_per_cap** | 0.758837 | |
| **oil_production_per_person** | 0.730895 | |
| **yearly_co2_emissions_1000_tonnes** | -0.021843 | |
| **forest_coverage_percent** | -0.303634 | |
| **income_per_person_gdppercapita_ppp_inflation_adjusted** | 0.819585 | |
| **electricity_generation_per_person** | 0.616939 | |
| **electricity_use_per_person** | 0.655008 | |

In [78]:

```
# plotting correlations on a heatmap

# figure size
plt.figure(figsize=(16,8))

# heatmap
sns.heatmap(cor, cmap="YlGnBu", annot=True)
plt.show()
```




In [79]:

```
# now from this it is clear that forest_coverage_percent is negatively correlated so we sho
df.drop('forest_coverage_percent',axis=1,inplace=True)
```

In [80]:

```
#now again we see on correlation matrix
cor = df.corr()
cor
```

Out[80]:

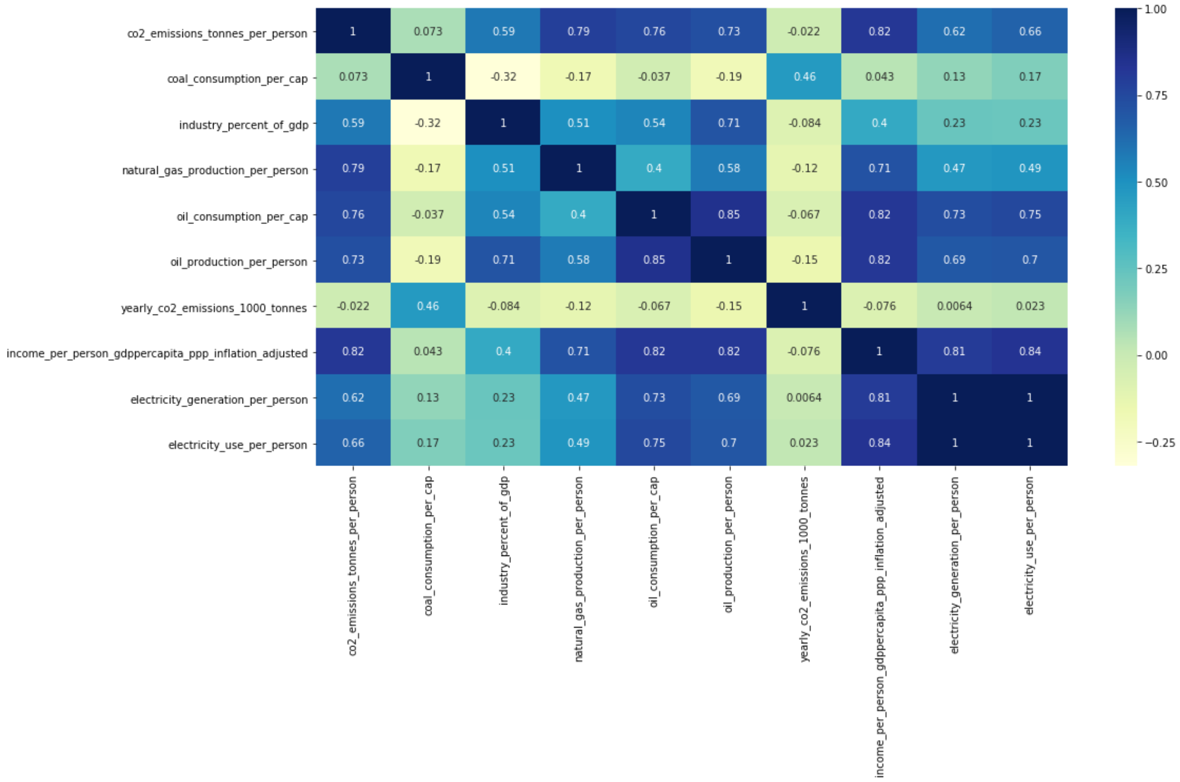
| | co2_emissions_tonnes_per_person | coal_ |
|---|---|---|
| **co2_emissions_tonnes_per_person** | 1.000000 | |
| **coal_consumption_per_cap** | 0.072586 | |
| **industry_percent_of_gdp** | 0.585031 | |
| **natural_gas_production_per_person** | 0.789657 | |
| **oil_consumption_per_cap** | 0.758837 | |
| **oil_production_per_person** | 0.730895 | |
| **yearly_co2_emissions_1000_tonnes** | -0.021843 | |
| **income_per_person_gdppercapita_ppp_inflation_adjusted** | 0.819585 | |
| **electricity_generation_per_person** | 0.616939 | |
| **electricity_use_per_person** | 0.655008 | |

In [81]:

```
# plotting correlations on a heatmap

# figure size
plt.figure(figsize=(16,8))

# heatmap
sns.heatmap(cor, cmap="YlGnBu", annot=True)
plt.show()
```




In [82]:

```
#Applying IQR method
q1 = df.quantile(0.25)
q3 = df.quantile(0.75)
IQR = q3 - q1
df = df[~((df < (q1 - 1.5 * IQR)) |(df > (q3 + 1.5 * IQR))).any(axis=1)]
```

In [83]:

```
#Checking Shape
df.shape
```

Out[83]:

```
(19, 10)
```

In [84]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 19 entries, Algeria to Vietnam
Data columns (total 10 columns):
co2_emissions_tonnes_per_person                          19 non-null float64
coal_consumption_per_cap                                 19 non-null float64
industry_percent_of_gdp                                  19 non-null float64
natural_gas_production_per_person                        19 non-null float64
oil_consumption_per_cap                                  19 non-null float64
oil_production_per_person                                19 non-null float64
yearly_co2_emissions_1000_tonnes                         19 non-null float64
income_per_person_gdppercapita_ppp_inflation_adjusted    19 non-null int64
electricity_generation_per_person                        19 non-null int64
electricity_use_per_person                               19 non-null float64
dtypes: float64(8), int64(2)
memory usage: 1.6+ KB
```

# Data Preparation

Data Preparation Let's now prepare the data and build the model.

In [93]:

```
oil_production_per_person','yearly_co2_emissions_1000_tonnes','income_per_person_gdppercapit
```

In [94]:

```
X.head()
```

Out[94]:

| | coal_consumption_per_cap | industry_percent_of_gdp | natural_gas_production_per_pers |
|---|---|---|---|
| **geo** | | | |
| **Algeria** | 0.00458 | 42.3 | 1.! |
| **Argentina** | 0.03460 | 24.3 | 0.' |
| **Azerbaijan** | 0.00017 | 53.6 | 1.( |
| **Brazil** | 0.08580 | 20.5 | 0.' |
| **Colombia** | 0.11000 | 32.7 | 0.: |

In [95]:

```
y = df['co2_emissions_tonnes_per_person']
```

In [96]:

```
y.head()
```

Out[96]:

```
geo
Algeria       3.72
Argentina     4.75
Azerbaijan    3.94
Brazil        2.59
Colombia      1.76
Name: co2_emissions_tonnes_per_person, dtype: float64
```

# Model Building and Evaluation

In [97]:

```
# scaling the features
from sklearn.preprocessing import scale

# storing column names in cols, since column names are (annoyingly) lost after
# scaling (the df is converted to a numpy array)
cols = X.columns
X = pd.DataFrame(scale(X))
X.columns = cols
X.columns
```

```
C:\Users\dell\Anaconda3\lib\site-packages\ipykernel_launcher.py:7: DataConve
rsionWarning: Data with input dtype int64, float64 were all converted to flo
at64 by the scale function.
  import sys
```

Out[97]:

```
Index(['coal_consumption_per_cap', 'industry_percent_of_gdp',
       'natural_gas_production_per_person', 'oil_consumption_per_cap',
       'oil_production_per_person', 'yearly_co2_emissions_1000_tonnes',
       'income_per_person_gdppercapita_ppp_inflation_adjusted',
       'electricity_generation_per_person', 'electricity_use_per_person'],
      dtype='object')
```

In [98]:

```
from sklearn.model_selection import train_test_split
```

In [99]:

```
X_train, X_test, y_train, y_test = train_test_split(X,y,train_size=0.7,test_size = 0.3, rar
```

In [164]:

```
# list of alphas to tune
params = {'alpha': [ 0.01, 0.05, 0.1,
 0.2, 0.8, 0.9, 1.0, 2.0]}
folds = 5
lasso = Lasso()
# cross validation
model_cv = GridSearchCV(estimator = lasso,
                        param_grid = params,
                        scoring= 'neg_mean_absolute_error',
                        cv = 7,
                        return_train_score=True,
                        verbose = 1)

model_cv.fit(X_train, y_train)
```

```
Fitting 7 folds for each of 8 candidates, totalling 56 fits

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent work
ers.
[Parallel(n_jobs=1)]: Done  56 out of  56 | elapsed:    0.2s finished
C:\Users\dell\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:841: DeprecationWarning: The default of the `iid` parameter will change fr
om True to False in version 0.22 and will be removed in 0.24. This will chan
ge numeric results when test-set sizes are unequal.
  DeprecationWarning)
```

Out[164]:

```
GridSearchCV(cv=7, error_score='raise-deprecating',
       estimator=Lasso(alpha=1.0, copy_X=True, fit_intercept=True, max_iter=
1000,
   normalize=False, positive=False, precompute=False, random_state=None,
   selection='cyclic', tol=0.0001, warm_start=False),
       fit_params=None, iid='warn', n_jobs=None,
       param_grid={'alpha': [0.01, 0.05, 0.1, 0.2, 0.8, 0.9, 1.0, 2.0]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score=True,
       scoring='neg_mean_absolute_error', verbose=1)
```

In [165]:

```
cv_results = pd.DataFrame(model_cv.cv_results_)
cv_results.head()
```

Out[165]:

| | mean_fit_time | std_fit_time | mean_score_time | std_score_time | param_alpha | params | split0_t |
|---|---|---|---|---|---|---|---|
| **0** | 0.002447 | 0.000496 | 0.001106 | 0.000334 | 0.01 | {'alpha': 0.01} | |
| **1** | 0.002275 | 0.000453 | 0.000717 | 0.000453 | 0.05 | {'alpha': 0.05} | |
| **2** | 0.002442 | 0.001184 | 0.001117 | 0.000299 | 0.1 | {'alpha': 0.1} | |
| **3** | 0.002337 | 0.000436 | 0.001419 | 0.000499 | 0.2 | {'alpha': 0.2} | |
| **4** | 0.002435 | 0.000483 | 0.000984 | 0.000033 | 0.8 | {'alpha': 0.8} | |

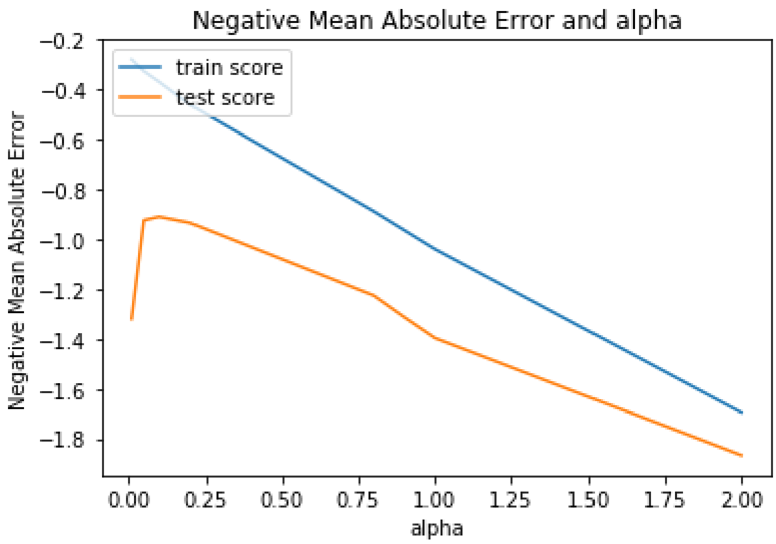
5 rows × 25 columns

In [166]:

```
# plotting mean test and train scoes with alpha
cv_results['param_alpha'] = cv_results['param_alpha'].astype('float32')

# plotting
plt.plot(cv_results['param_alpha'], cv_results['mean_train_score'])
plt.plot(cv_results['param_alpha'], cv_results['mean_test_score'])
plt.xlabel('alpha')
plt.ylabel('Negative Mean Absolute Error')

plt.title("Negative Mean Absolute Error and alpha")
plt.legend(['train score', 'test score'], loc='upper left')
plt.show()
```



In [180]:

```
alpha =0.01

lasso = Lasso(alpha=alpha)

lasso.fit(X_train, y_train)
```

Out[180]:

```
Lasso(alpha=0.01, copy_X=True, fit_intercept=True, max_iter=1000,
   normalize=False, positive=False, precompute=False, random_state=None,
   selection='cyclic', tol=0.0001, warm_start=False)
```

In [181]:

```
lasso.coef_
```

Out[181]:

```
array([ 0.06687721,  0.25555553,  0.93478149,  0.17609756, -0.20150787,
        0.49535097, -0.        ,  1.34951223,  0.        ])
```

In [182]:

```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, model_cv.predict(X_test))
```

Out[182]:

```
0.48751017821885156
```

In [183]:

```
#Importing Linear regression Libraries
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
Linear_model= LinearRegression()
```

In [184]:

```
#now we creating Linear Model
Linear_model.fit(X_train,y_train)
pred = Linear_model.predict(X_test)
pred[:10]
```

Out[184]:

```
array([6.38249686, 3.62003115, 4.29230985, 1.32588428, 3.78845719,
       3.99720296])
```

In [185]:

```
#Importing mean_squared_error and calculating meansquare for linear
from sklearn.metrics import mean_squared_error
print(mean_squared_error(y_test,pred)**(0.5))
```

```
0.8887442223615221
```

In [186]:

```
#Importing r2 score
from sklearn.metrics import r2_score
```

In [187]:

```
#Checking r2 score for linear
r2_score(y_test,prediction)
```

Out[187]:

```
0.6499616968145859
```

In [188]:

```
#Creating Lasso model for prediction
lasso_model = Lasso(alpha=0.00001)
lasso_model.fit(X_train,y_train)
pred2 = lasso_model.predict(X_test)
```

```
C:\Users\dell\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_de
scent.py:492: ConvergenceWarning: Objective did not converge. You might want
to increase the number of iterations. Fitting data with very small alpha may
cause precision problems.
  ConvergenceWarning)
```

In [189]:

```
#Checking Mean squared error for Lasso
print(mean_squared_error(y_test,pred2)**(0.5))
```

```
0.8877982278062921
```

In [190]:

```
#Checking r2 score for Lasso
r2_score(y_test,pred2)
```

Out[190]:

```
0.6507064737438435
```

In [ ]:

```
We can see that our lasso model's r2 and mean squared value are almost same as linear model
```

In [ ]:

```

```