Machine Learning Assignment No.1

Que.1 Create a scatter plot between cylinder vs Co2Emission (green color)

Import the libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("C:/Users/Lenovo/Documents/Data Set/FuelConsumption.csv")

df

      MODELYEAR   MAKE       MODEL    VEHICLECLASS  ENGINESIZE
CYLINDERS  \
0          2014  ACURA         ILX         COMPACT         2.0
4
1          2014  ACURA         ILX         COMPACT         2.4
4
2          2014  ACURA  ILX HYBRID         COMPACT         1.5
4
3          2014  ACURA     MDX 4WD     SUV - SMALL         3.5
6
4          2014  ACURA     RDX AWD     SUV - SMALL         3.5
6
...         ...    ...         ...             ...         ...
...
1062       2014  VOLVO    XC60 AWD     SUV - SMALL         3.0
6
1063       2014  VOLVO    XC60 AWD     SUV - SMALL         3.2
6
1064       2014  VOLVO    XC70 AWD     SUV - SMALL         3.0
6
1065       2014  VOLVO    XC70 AWD     SUV - SMALL         3.2
6
1066       2014  VOLVO    XC90 AWD  SUV - STANDARD         3.2
6

     TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY
\
0             AS5        Z                   9.9                  6.7

1              M6        Z                  11.2                  7.7
```

```
2              AV7        Z                     6.0                   5.8

3              AS6        Z                    12.7                   9.1

4              AS6        Z                    12.1                   8.7

...            ...      ...                     ...                   ...

1062           AS6        X                    13.4                   9.8

1063           AS6        X                    13.2                   9.5

1064           AS6        X                    13.4                   9.8

1065           AS6        X                    12.9                   9.3

1066           AS6        X                    14.9                  10.2


      FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                      8.5                        33           196
1                      9.6                        29           221
2                      5.9                        48           136
3                     11.1                        25           255
4                     10.6                        27           244
...                    ...                       ...           ...
1062                  11.8                        24           271
1063                  11.5                        25           264
1064                  11.8                        24           271
1065                  11.3                        25           260
1066                  12.8                        22           294

[1067 rows x 13 columns]
```

```
df.head()
```

```
   MODELYEAR   MAKE       MODEL VEHICLECLASS  ENGINESIZE  CYLINDERS  \
0       2014  ACURA         ILX      COMPACT         2.0          4
1       2014  ACURA         ILX      COMPACT         2.4          4
2       2014  ACURA  ILX HYBRID      COMPACT         1.5          4
3       2014  ACURA     MDX 4WD  SUV - SMALL         3.5          6
4       2014  ACURA     RDX AWD  SUV - SMALL         3.5          6

  TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY  \
0          AS5        Z                   9.9                  6.7
1           M6        Z                  11.2                  7.7
2          AV7        Z                   6.0                  5.8
3          AS6        Z                  12.7                  9.1
```

```
4          AS6        Z                  12.1                 8.7

   FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                   8.5                        33           196
1                   9.6                        29           221
2                   5.9                        48           136
3                  11.1                        25           255
4                  10.6                        27           244
```

```
df.tail()
```

```
      MODELYEAR   MAKE     MODEL    VEHICLECLASS  ENGINESIZE
CYLINDERS  \
1062       2014  VOLVO  XC60 AWD     SUV - SMALL         3.0
6
1063       2014  VOLVO  XC60 AWD     SUV - SMALL         3.2
6
1064       2014  VOLVO  XC70 AWD     SUV - SMALL         3.0
6
1065       2014  VOLVO  XC70 AWD     SUV - SMALL         3.2
6
1066       2014  VOLVO  XC90 AWD  SUV - STANDARD         3.2
6

     TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY
\
1062          AS6        X                  13.4                  9.8

1063          AS6        X                  13.2                  9.5

1064          AS6        X                  13.4                  9.8

1065          AS6        X                  12.9                  9.3

1066          AS6        X                  14.9                 10.2


      FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
1062                  11.8                        24           271
1063                  11.5                        25           264
1064                  11.8                        24           271
1065                  11.3                        25           260
1066                  12.8                        22           294
```

```
df.info
```

```
<bound method DataFrame.info of       MODELYEAR   MAKE       MODEL
VEHICLECLASS  ENGINESIZE  CYLINDERS  \
0          2014  ACURA         ILX         COMPACT         2.0
4
```

```
1          2014  ACURA         ILX         COMPACT         2.4
4
2          2014  ACURA  ILX HYBRID         COMPACT         1.5
4
3          2014  ACURA     MDX 4WD     SUV - SMALL         3.5
6
4          2014  ACURA     RDX AWD     SUV - SMALL         3.5
6
...         ...    ...         ...             ...         ...
...
1062       2014  VOLVO    XC60 AWD     SUV - SMALL         3.0
6
1063       2014  VOLVO    XC60 AWD     SUV - SMALL         3.2
6
1064       2014  VOLVO    XC70 AWD     SUV - SMALL         3.0
6
1065       2014  VOLVO    XC70 AWD     SUV - SMALL         3.2
6
1066       2014  VOLVO    XC90 AWD  SUV - STANDARD         3.2
6

     TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY
\
0             AS5        Z                   9.9                  6.7

1              M6        Z                  11.2                  7.7

2             AV7        Z                   6.0                  5.8

3             AS6        Z                  12.7                  9.1

4             AS6        Z                  12.1                  8.7

...           ...      ...                   ...                  ...

1062          AS6        X                  13.4                  9.8

1063          AS6        X                  13.2                  9.5

1064          AS6        X                  13.4                  9.8

1065          AS6        X                  12.9                  9.3

1066          AS6        X                  14.9                 10.2


      FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                      8.5                        33           196
1                      9.6                        29           221
```

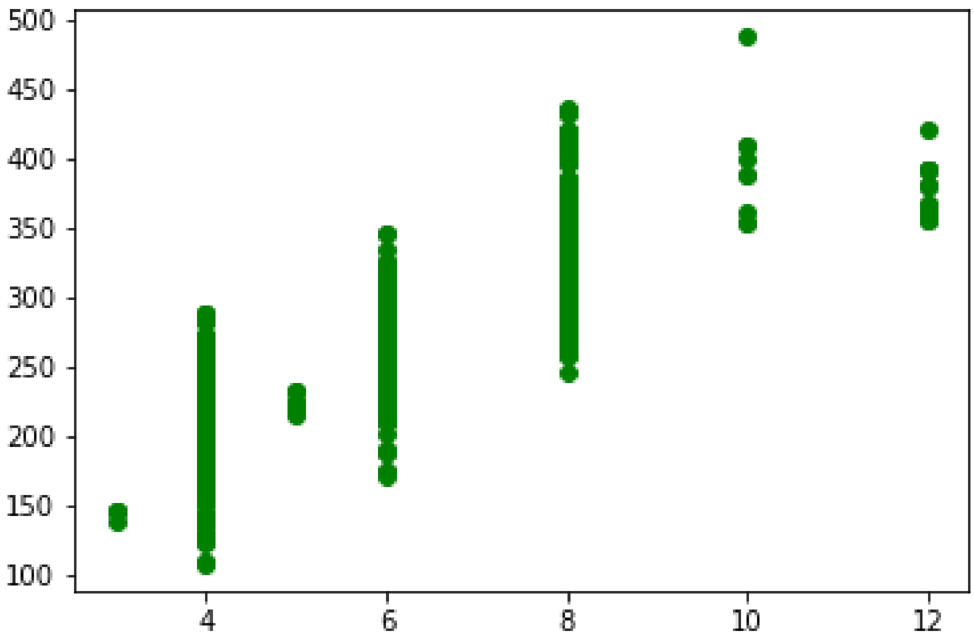
```
2                      5.9                        48            136
3                     11.1                        25            255
4                     10.6                        27            244
...                    ...                       ...            ...
1062                  11.8                        24            271
1063                  11.5                        25            264
1064                  11.8                        24            271
1065                  11.3                        25            260
1066                  12.8                        22            294

[1067 rows x 13 columns]>
```

```
df.isnull().sum()
```

```
MODELYEAR                   0
MAKE                        0
MODEL                       0
VEHICLECLASS                0
ENGINESIZE                  0
CYLINDERS                   0
TRANSMISSION                0
FUELTYPE                    0
FUELCONSUMPTION_CITY        0
FUELCONSUMPTION_HWY         0
FUELCONSUMPTION_COMB        0
FUELCONSUMPTION_COMB_MPG    0
CO2EMISSIONS                0
dtype: int64
```

```
plt.scatter(df['CYLINDERS'],df['CO2EMISSIONS'] , color = "green")
```

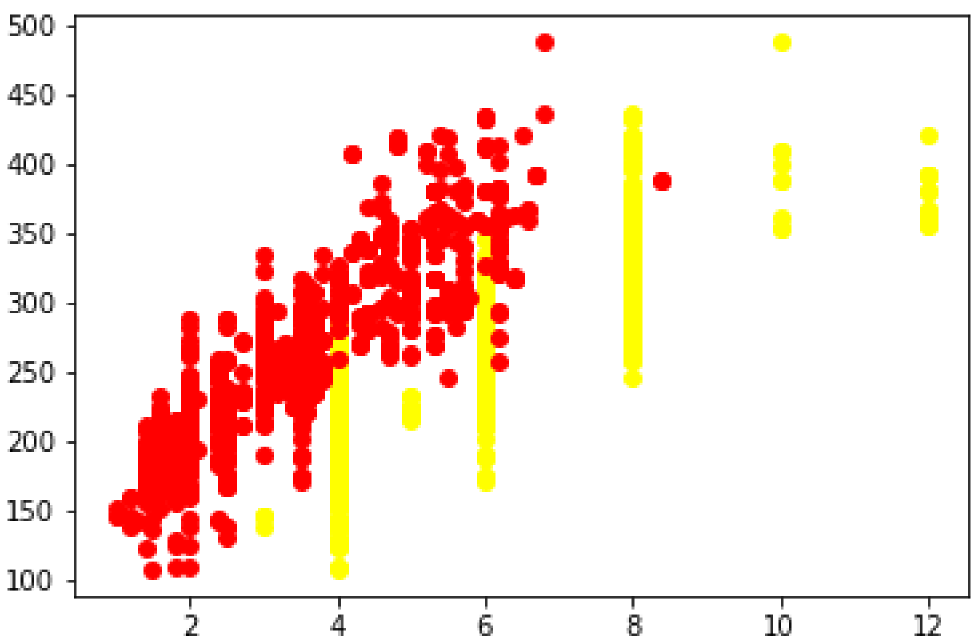
```
<matplotlib.collections.PathCollection at 0x21189b46b50>
```



Que.2 Using scatter plot compare data cylinder vs Co2Emission and Enginesize Vs Co2Emission using different colors

```
plt.scatter(df["CYLINDERS"],df["CO2EMISSIONS"], color='yellow')
plt.scatter(df["ENGINESIZE"],df["CO2EMISSIONS"],color='red')
```

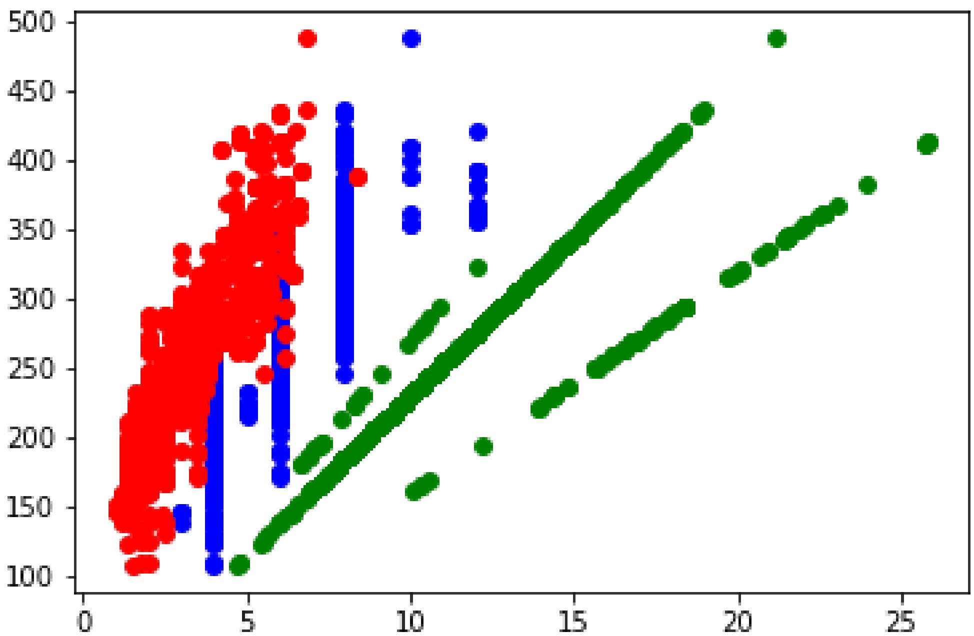
```
<matplotlib.collections.PathCollection at 0x21189c4ed60>
```



Que.3 Using scatter plot compare data cylinder vs Co2Emission and Enginesize Vs Co2Emission and FuelConsumption_comb Co2Emission using different colors

```
plt.scatter(df["CYLINDERS"],df["CO2EMISSIONS"], color="blue")
plt.scatter(df["ENGINESIZE"],df["CO2EMISSIONS"],color='red')
plt.scatter(df["FUELCONSUMPTION_COMB"],df["CO2EMISSIONS"],
color='green')
```

```
<matplotlib.collections.PathCollection at 0x21189cd6520>
```



Que.4 Train your model with indepedent variable as cylinder and dependent variable as Co2Emission

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("C:/Users/Lenovo/Documents/Data
Set/FuelConsumption.csv")

df
```

```
      MODELYEAR   MAKE       MODEL    VEHICLECLASS  ENGINESIZE
CYLINDERS  \
0          2014  ACURA         ILX         COMPACT         2.0
4
1          2014  ACURA         ILX         COMPACT         2.4
4
```

```
2          2014  ACURA  ILX HYBRID         COMPACT         1.5
4
3          2014  ACURA     MDX 4WD     SUV - SMALL         3.5
6
4          2014  ACURA     RDX AWD     SUV - SMALL         3.5
6
...         ...    ...         ...             ...         ...
...
1062       2014  VOLVO    XC60 AWD     SUV - SMALL         3.0
6
1063       2014  VOLVO    XC60 AWD     SUV - SMALL         3.2
6
1064       2014  VOLVO    XC70 AWD     SUV - SMALL         3.0
6
1065       2014  VOLVO    XC70 AWD     SUV - SMALL         3.2
6
1066       2014  VOLVO    XC90 AWD  SUV - STANDARD         3.2
6

     TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY
\
0             AS5        Z                   9.9                  6.7

1              M6        Z                  11.2                  7.7

2             AV7        Z                   6.0                  5.8

3             AS6        Z                  12.7                  9.1

4             AS6        Z                  12.1                  8.7

...           ...      ...                   ...                  ...

1062          AS6        X                  13.4                  9.8

1063          AS6        X                  13.2                  9.5

1064          AS6        X                  13.4                  9.8

1065          AS6        X                  12.9                  9.3

1066          AS6        X                  14.9                 10.2


      FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                      8.5                        33           196
1                      9.6                        29           221
2                      5.9                        48           136
3                     11.1                        25           255
```

```
4                    10.6                           27          244
...                   ...                          ...          ...
1062                 11.8                           24          271
1063                 11.5                           25          264
1064                 11.8                           24          271
1065                 11.3                           25          260
1066                 12.8                           22          294

[1067 rows x 13 columns]
```

df.head()

```
   MODELYEAR   MAKE       MODEL VEHICLECLASS  ENGINESIZE  CYLINDERS  \
0       2014  ACURA         ILX      COMPACT         2.0          4
1       2014  ACURA         ILX      COMPACT         2.4          4
2       2014  ACURA  ILX HYBRID      COMPACT         1.5          4
3       2014  ACURA     MDX 4WD  SUV - SMALL         3.5          6
4       2014  ACURA     RDX AWD  SUV - SMALL         3.5          6

  TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY  \
0          AS5        Z                   9.9                  6.7
1           M6        Z                  11.2                  7.7
2          AV7        Z                   6.0                  5.8
3          AS6        Z                  12.7                  9.1
4          AS6        Z                  12.1                  8.7

   FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                   8.5                        33           196
1                   9.6                        29           221
2                   5.9                        48           136
3                  11.1                        25           255
4                  10.6                        27           244
```

df.tail()

```
      MODELYEAR   MAKE     MODEL    VEHICLECLASS  ENGINESIZE
CYLINDERS  \
1062       2014  VOLVO  XC60 AWD     SUV - SMALL         3.0
6
1063       2014  VOLVO  XC60 AWD     SUV - SMALL         3.2
6
1064       2014  VOLVO  XC70 AWD     SUV - SMALL         3.0
6
1065       2014  VOLVO  XC70 AWD     SUV - SMALL         3.2
6
1066       2014  VOLVO  XC90 AWD  SUV - STANDARD         3.2
6

     TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY
\
1062          AS6        X                  13.4                  9.8
```

```
1063          AS6        X                  13.2                   9.5

1064          AS6        X                  13.4                   9.8

1065          AS6        X                  12.9                   9.3

1066          AS6        X                  14.9                  10.2


      FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
1062                  11.8                        24           271
1063                  11.5                        25           264
1064                  11.8                        24           271
1065                  11.3                        25           260
1066                  12.8                        22           294

df.info

<bound method DataFrame.info of       MODELYEAR   MAKE       MODEL
VEHICLECLASS  ENGINESIZE  CYLINDERS  \
0          2014  ACURA         ILX         COMPACT         2.0
4
1          2014  ACURA         ILX         COMPACT         2.4
4
2          2014  ACURA  ILX HYBRID         COMPACT         1.5
4
3          2014  ACURA     MDX 4WD     SUV - SMALL         3.5
6
4          2014  ACURA     RDX AWD     SUV - SMALL         3.5
6
...         ...    ...         ...             ...         ...
...
1062       2014  VOLVO    XC60 AWD     SUV - SMALL         3.0
6
1063       2014  VOLVO    XC60 AWD     SUV - SMALL         3.2
6
1064       2014  VOLVO    XC70 AWD     SUV - SMALL         3.0
6
1065       2014  VOLVO    XC70 AWD     SUV - SMALL         3.2
6
1066       2014  VOLVO    XC90 AWD  SUV - STANDARD         3.2
6

     TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY
\
0             AS5        Z                   9.9                   6.7

1              M6        Z                  11.2                   7.7
```

```
2             AV7        Z                     6.0                    5.8

3             AS6        Z                    12.7                    9.1

4             AS6        Z                    12.1                    8.7

...           ...      ...                     ...                    ...

1062          AS6        X                    13.4                    9.8

1063          AS6        X                    13.2                    9.5

1064          AS6        X                    13.4                    9.8

1065          AS6        X                    12.9                    9.3

1066          AS6        X                    14.9                   10.2

      FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                      8.5                        33           196
1                      9.6                        29           221
2                      5.9                        48           136
3                     11.1                        25           255
4                     10.6                        27           244
...                    ...                       ...           ...
1062                  11.8                        24           271
1063                  11.5                        25           264
1064                  11.8                        24           271
1065                  11.3                        25           260
1066                  12.8                        22           294

[1067 rows x 13 columns]>
```

Data Exploration

```
# Summarize the Data
df.describe()
```

```
       MODELYEAR   ENGINESIZE    CYLINDERS  FUELCONSUMPTION_CITY  \
count     1067.0  1067.000000  1067.000000           1067.000000
mean      2014.0     3.346298     5.794752             13.296532
std          0.0     1.415895     1.797447              4.101253
min       2014.0     1.000000     3.000000              4.600000
25%       2014.0     2.000000     4.000000             10.250000
50%       2014.0     3.400000     6.000000             12.600000
75%       2014.0     4.300000     8.000000             15.550000
max       2014.0     8.400000    12.000000             30.200000
```

```
       FUELCONSUMPTION_HWY  FUELCONSUMPTION_COMB
FUELCONSUMPTION_COMB_MPG  \
count          1067.000000           1067.000000
1067.000000
mean              9.474602             11.580881
26.441425
std               2.794510              3.485595
7.468702
min               4.900000              4.700000
11.000000
25%               7.500000              9.000000
21.000000
50%               8.800000             10.900000
26.000000
75%              10.850000             13.350000
31.000000
max              20.500000             25.800000
60.000000

       CO2EMISSIONS
count   1067.000000
mean     256.228679
std       63.372304
min      108.000000
25%      207.000000
50%      251.000000
75%      294.000000
max      488.000000
```

```
df[['MODELYEAR','ENGINESIZE','CYLINDERS']]
```

```
      MODELYEAR  ENGINESIZE  CYLINDERS
0          2014         2.0          4
1          2014         2.4          4
2          2014         1.5          4
3          2014         3.5          6
4          2014         3.5          6
...         ...         ...        ...
1062       2014         3.0          6
1063       2014         3.2          6
1064       2014         3.0          6
1065       2014         3.2          6
1066       2014         3.2          6

[1067 rows x 3 columns]
```

```
df[['CYLINDERS','CO2EMISSIONS']]
```

```
      CYLINDERS  CO2EMISSIONS
0             4           196
1             4           221
```

```
2             4          136
3             6          255
4             6          244
...         ...          ...
1062          6          271
1063          6          264
1064          6          271
1065          6          260
1066          6          294

[1067 rows x 2 columns]
```

df

```
      MODELYEAR   MAKE       MODEL    VEHICLECLASS  ENGINESIZE
CYLINDERS  \
0          2014  ACURA         ILX         COMPACT         2.0
4
1          2014  ACURA         ILX         COMPACT         2.4
4
2          2014  ACURA  ILX HYBRID         COMPACT         1.5
4
3          2014  ACURA     MDX 4WD     SUV - SMALL         3.5
6
4          2014  ACURA     RDX AWD     SUV - SMALL         3.5
6
...         ...    ...         ...             ...         ...
...
1062       2014  VOLVO    XC60 AWD     SUV - SMALL         3.0
6
1063       2014  VOLVO    XC60 AWD     SUV - SMALL         3.2
6
1064       2014  VOLVO    XC70 AWD     SUV - SMALL         3.0
6
1065       2014  VOLVO    XC70 AWD     SUV - SMALL         3.2
6
1066       2014  VOLVO    XC90 AWD  SUV - STANDARD         3.2
6

     TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY
\
0             AS5        Z                   9.9                  6.7

1              M6        Z                  11.2                  7.7

2             AV7        Z                   6.0                  5.8

3             AS6        Z                  12.7                  9.1

4             AS6        Z                  12.1                  8.7
```

```
...          ...      ...                     ...                     ...
1062         AS6        X                    13.4                     9.8
1063         AS6        X                    13.2                     9.5
1064         AS6        X                    13.4                     9.8
1065         AS6        X                    12.9                     9.3
1066         AS6        X                    14.9                    10.2

      FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                      8.5                        33           196
1                      9.6                        29           221
2                      5.9                        48           136
3                     11.1                        25           255
4                     10.6                        27           244
...                    ...                       ...           ...
1062                  11.8                        24           271
1063                  11.5                        25           264
1064                  11.8                        24           271
1065                  11.3                        25           260
1066                  12.8                        22           294

[1067 rows x 13 columns]
```

```
cdf =
df[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB','CO2EMISSIONS']]
cdf.head()
```

```
   ENGINESIZE  CYLINDERS  FUELCONSUMPTION_COMB  CO2EMISSIONS
0         2.0          4                   8.5           196
1         2.4          4                   9.6           221
2         1.5          4                   5.9           136
3         3.5          6                  11.1           255
4         3.5          6                  10.6           244
```

```
cdf.tail()
```

```
      ENGINESIZE  CYLINDERS  FUELCONSUMPTION_COMB  CO2EMISSIONS
1062         3.0          6                  11.8           271
1063         3.2          6                  11.5           264
1064         3.0          6                  11.8           271
1065         3.2          6                  11.3           260
1066         3.2          6                  12.8           294
```

Creating train and test dataset

```
cdf
```

```
      ENGINESIZE  CYLINDERS  FUELCONSUMPTION_COMB  CO2EMISSIONS
0            2.0          4                   8.5           196
1            2.4          4                   9.6           221
2            1.5          4                   5.9           136
3            3.5          6                  11.1           255
4            3.5          6                  10.6           244
...          ...        ...                   ...           ...
1062         3.0          6                  11.8           271
1063         3.2          6                  11.5           264
1064         3.0          6                  11.8           271
1065         3.2          6                  11.3           260
1066         3.2          6                  12.8           294

[1067 rows x 4 columns]
```

```
len(cdf)
```

```
1067
```

```
msk = np.random.rand(len(cdf)) <= 0.80 #80% #len 1067 #853
```

```
msk
```

```
array([ True,  True,  True, ...,  True,  True,  True])
```

```
~msk
```

```
array([False, False, False, ..., False, False, False])
```

```
len(msk)
```

```
1067
```

```
len(~msk)
```

```
1067
```

```
train = cdf[msk] #80% of rows are your trainig example
test = cdf[~msk] #rest of the data out of the msk are for testing data
#~not
```

```
train.shape
```

```
(845, 4)
```

```
test.shape
```

```
(222, 4)
```

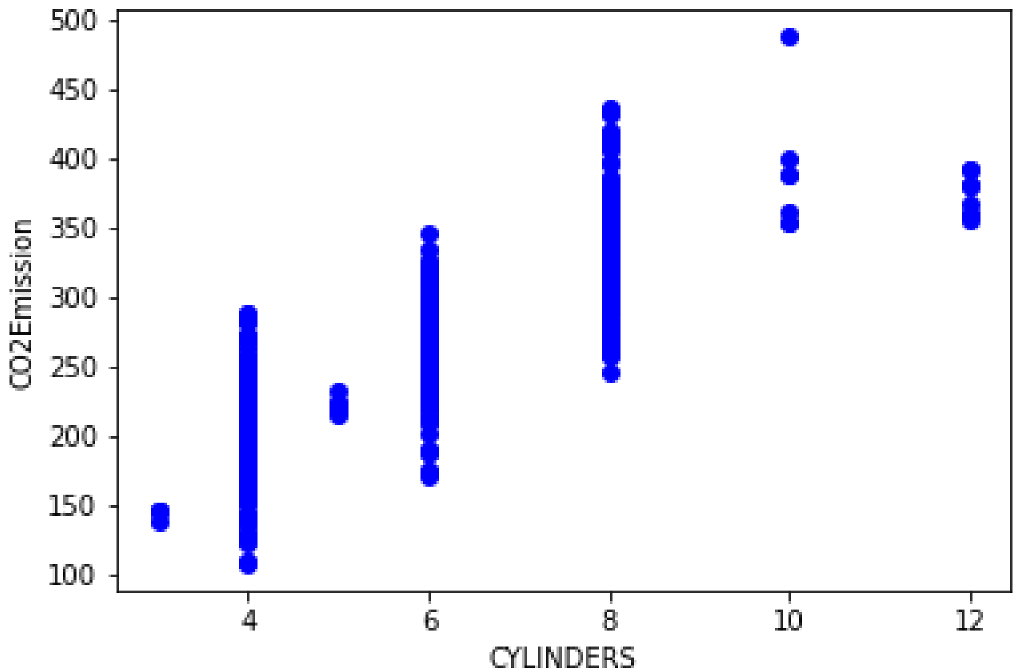
```
train
```

```
      ENGINESIZE  CYLINDERS  FUELCONSUMPTION_COMB  CO2EMISSIONS
0            2.0          4                   8.5           196
1            2.4          4                   9.6           221
2            1.5          4                   5.9           136
3            3.5          6                  11.1           255
4            3.5          6                  10.6           244
...          ...        ...                   ...           ...
1062         3.0          6                  11.8           271
1063         3.2          6                  11.5           264
1064         3.0          6                  11.8           271
1065         3.2          6                  11.3           260
1066         3.2          6                  12.8           294

[845 rows x 4 columns]

test

      ENGINESIZE  CYLINDERS  FUELCONSUMPTION_COMB  CO2EMISSIONS
5            3.5          6                  10.0           230
7            3.7          6                  11.1           255
9            2.4          4                   9.2           212
13           5.9         12                  15.6           359
19           2.0          4                   8.8           202
...          ...        ...                   ...           ...
1041         2.0          4                   6.9           186
1045         2.5          5                   9.7           223
1046         2.5          5                   9.8           225
1047         3.6          6                  10.8           248
1052         2.0          4                  11.6           267

[222 rows x 4 columns]
```

Simple Regression Model

Linear Regression fits a linear model with coefficients B = (B1, ..., Bn) to minimize the 'residual sum of squares' between the actual value y in the dataset, and the predicted value yhat using linear approximation.

```
# Train data distribution

plt.scatter(train.CYLINDERS, train.CO2EMISSIONS,  color='blue')
plt.xlabel("CYLINDERS")
plt.ylabel("CO2Emission")
plt.show()
```



Modeling

Using sklearn package to model data.

```
train[['CYLINDERS']]
```

```
      CYLINDERS
0             4
1             4
2             4
3             6
4             6
...         ...
1062          6
1063          6
1064          6
1065          6
1066          6

[845 rows x 1 columns]
```

```
train[['CO2EMISSIONS']]
```

```
      CO2EMISSIONS
0              196
1              221
2              136
3              255
```

```
4               244
...             ...
1062            271
1063            264
1064            271
1065            260
1066            294

[845 rows x 1 columns]
```

```
np.asanyarray(train[['CYLINDERS']])
```

```
array([[ 4],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 4],
       [ 6],
       [12],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [12],
       [ 4],
       [ 4],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 8],
       [ 6],
       [ 6],
       [ 8],
       [ 6],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 8],
       [ 8],
       [10],
       [ 8],
       [ 8],
       [10],
```

```
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 8],
[12],
[ 8],
[12],
[ 8],
[12],
[12],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 8],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 8],
```

```
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 6],
[ 4],
[ 6],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 6],
[ 6],
[ 4],
[ 6],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
```

```
       [ 8],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 4],
       [ 4],
       [ 4],
       [ 4],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 6],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 6],
       [ 4],
       [ 4],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 8],
       [ 8],
       [ 8],
       [ 6],
       [ 8],
       [ 8],
```

```
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
```

```
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[10],
[ 4],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
```

```
[ 6],
[ 6],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 8],
[ 6],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
```

```
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 8],
[ 4],
[ 6],
[ 4],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 8],
[ 6],
[ 6],
[ 6],
```

```
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
```

```
[ 4],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 6],
[ 6],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[10],
[10],
[ 4],
[ 6],
```

```
[ 4],
[ 4],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 4],
```

```
[ 4],
[ 4],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 8],
[ 8],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
```

```
[ 8],
[ 8],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
```

```
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 3],
[ 3],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 6],
[ 8],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 6],
[ 4],
[ 4],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
```

```
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[12],
[12],
[12],
[12],
[ 4],
[ 4],
[ 4],
[ 4],
[ 3],
[ 3],
[10],
```

```
[10],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 6],
[ 6],
[ 4],
[ 4],
[ 6],
[ 4],
```

```
[ 4],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 5],
[ 5],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 5],
[ 5],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 5],
[ 5],
[ 6],
[ 6],
```

```
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 6]], dtype=int64)
```

```
np.asanyarray(train[['CO2EMISSIONS']])
```

```
array([[196],
       [221],
       [136],
       [255],
       [244],
       [232],
       [267],
       [225],
       [239],
       [359],
       [338],
       [354],
       [338],
       [354],
       [359],
       [230],
       [214],
       [230],
       [214],
       [251],
       [224],
       [258],
       [224],
       [260],
       [227],
       [258],
       [288],
       [230],
       [242],
       [239],
       [258],
       [294],
       [336],
       [407],
       [354],
       [336],
       [407],
       [354],
       [306],
       [308],
       [262],
```

[285],
[262],
[267],
[281],
[281],
[292],
[209],
[209],
[297],
[356],
[320],
[380],
[322],
[380],
[380],
[437],
[193],
[200],
[181],
[181],
[193],
[200],
[209],
[209],
[209],
[230],
[228],
[237],
[232],
[193],
[200],
[209],
[221],
[230],
[237],
[202],
[209],
[213],
[232],
[292],
[232],
[292],
[281],
[292],
[292],
[191],
[292],
[338],
[338],
[317],

```
[338],
[317],
[209],
[260],
[230],
[253],
[320],
[370],
[274],
[209],
[200],
[278],
[253],
[278],
[283],
[290],
[196],
[212],
[264],
[250],
[264],
[264],
[225],
[246],
[189],
[225],
[218],
[221],
[235],
[207],
[251],
[232],
[258],
[262],
[232],
[248],
[248],
[260],
[253],
[260],
[255],
[382],
[340],
[382],
[340],
[338],
[322],
[377],
[283],
[294],
```

[271],
[290],
[258],
[269],
[294],
[382],
[340],
[276],
[258],
[179],
[189],
[192],
[177],
[163],
[209],
[259],
[290],
[272],
[366],
[334],
[380],
[344],
[380],
[344],
[362],
[380],
[344],
[380],
[362],
[414],
[432],
[411],
[419],
[435],
[229],
[200],
[228],
[184],
[244],
[271],
[269],
[292],
[270],
[324],
[290],
[299],
[329],
[184],
[173],
[184],

[172],
[182],
[191],
[182],
[191],
[182],
[159],
[317],
[294],
[317],
[294],
[317],
[294],
[317],
[294],
[290],
[207],
[232],
[246],
[230],
[230],
[230],
[294],
[251],
[251],
[237],
[222],
[276],
[269],
[225],
[230],
[301],
[294],
[320],
[317],
[258],
[237],
[294],
[251],
[308],
[251],
[237],
[235],
[222],
[186],
[198],
[202],
[195],
[202],
[168],

[168],
[278],
[250],
[271],
[264],
[255],
[285],
[276],
[179],
[179],
[182],
[179],
[156],
[200],
[193],
[373],
[350],
[396],
[360],
[386],
[368],
[488],
[225],
[264],
[281],
[209],
[216],
[214],
[230],
[368],
[346],
[274],
[254],
[288],
[267],
[294],
[366],
[317],
[308],
[283],
[283],
[264],
[336],
[336],
[315],
[308],
[288],
[308],
[288],
[320],

```
[318],
[152],
[184],
[264],
[278],
[294],
[205],
[177],
[165],
[179],
[165],
[184],
[170],
[166],
[162],
[196],
[186],
[205],
[186],
[216],
[126],
[244],
[264],
[244],
[209],
[271],
[237],
[229],
[258],
[251],
[218],
[228],
[232],
[239],
[283],
[297],
[347],
[366],
[334],
[344],
[380],
[344],
[380],
[380],
[344],
[414],
[432],
[411],
[419],
[435],
```

[271],
[269],
[292],
[270],
[324],
[290],
[285],
[277],
[329],
[209],
[276],
[230],
[290],
[272],
[317],
[294],
[338],
[322],
[354],
[317],
[294],
[317],
[294],
[182],
[186],
[200],
[209],
[221],
[251],
[110],
[161],
[166],
[246],
[209],
[159],
[172],
[244],
[265],
[276],
[308],
[177],
[177],
[196],
[304],
[255],
[258],
[267],
[237],
[264],
[255],

[221],
[143],
[145],
[214],
[232],
[230],
[244],
[177],
[184],
[202],
[196],
[235],
[172],
[189],
[265],
[269],
[283],
[255],
[267],
[253],
[283],
[267],
[294],
[175],
[271],
[244],
[294],
[334],
[345],
[237],
[244],
[294],
[306],
[290],
[306],
[285],
[280],
[306],
[306],
[301],
[315],
[310],
[315],
[310],
[218],
[230],
[248],
[232],
[251],
[267],

```
[225],
[209],
[225],
[216],
[235],
[216],
[255],
[340],
[267],
[278],
[363],
[225],
[216],
[216],
[255],
[294],
[308],
[251],
[184],
[193],
[193],
[223],
[225],
[193],
[196],
[216],
[221],
[193],
[198],
[200],
[175],
[177],
[175],
[212],
[214],
[276],
[237],
[262],
[253],
[271],
[207],
[209],
[230],
[361],
[400],
[274],
[334],
[225],
[225],
[354],
```

```
[342],
[285],
[278],
[345],
[331],
[283],
[283],
[277],
[354],
[347],
[129],
[138],
[221],
[239],
[253],
[175],
[320],
[223],
[239],
[244],
[253],
[244],
[290],
[283],
[297],
[262],
[380],
[255],
[189],
[271],
[260],
[294],
[235],
[281],
[207],
[262],
[145],
[368],
[346],
[292],
[292],
[347],
[334],
[301],
[189],
[184],
[198],
[196],
[205],
[283],
```

```
[290],
[177],
[170],
[159],
[161],
[170],
[163],
[166],
[177],
[168],
[166],
[223],
[225],
[175],
[168],
[235],
[221],
[228],
[184],
[216],
[216],
[242],
[224],
[239],
[244],
[242],
[347],
[352],
[304],
[308],
[179],
[212],
[264],
[301],
[194],
[235],
[230],
[239],
[242],
[235],
[202],
[267],
[269],
[262],
[297],
[310],
[310],
[419],
[407],
[324],
```

```
[359],
[260],
[256],
[278],
[327],
[356],
[278],
[276],
[327],
[205],
[207],
[225],
[246],
[380],
[361],
[361],
[179],
[179],
[179],
[179],
[200],
[179],
[175],
[200],
[179],
[179],
[179],
[184],
[191],
[207],
[196],
[191],
[184],
[207],
[196],
[191],
[184],
[191],
[184],
[191],
[184],
[207],
[196],
[191],
[207],
[196],
[191],
[184],
[189],
[191],
```

```
[288],
[283],
[267],
[198],
[198],
[138],
[147],
[198],
[235],
[200],
[202],
[207],
[253],
[264],
[262],
[175],
[214],
[380],
[283],
[292],
[315],
[301],
[186],
[198],
[251],
[244],
[253],
[202],
[255],
[161],
[179],
[359],
[398],
[152],
[177],
[327],
[315],
[232],
[237],
[246],
[242],
[253],
[248],
[253],
[253],
[258],
[232],
[246],
[244],
[248],
```

[246],
[251],
[322],
[274],
[274],
[274],
[274],
[212],
[228],
[225],
[313],
[281],
[320],
[299],
[251],
[315],
[212],
[225],
[239],
[251],
[255],
[260],
[264],
[292],
[260],
[294],
[340],
[313],
[354],
[290],
[285],
[275],
[274],
[260],
[262],
[254],
[368],
[393],
[393],
[359],
[193],
[207],
[184],
[182],
[147],
[147],
[389],
[389],
[221],
[223],

```
[179],
[196],
[258],
[179],
[196],
[258],
[288],
[198],
[228],
[267],
[207],
[301],
[196],
[212],
[177],
[297],
[297],
[221],
[131],
[138],
[177],
[168],
[163],
[170],
[156],
[159],
[196],
[189],
[297],
[320],
[237],
[258],
[264],
[191],
[189],
[110],
[108],
[126],
[216],
[218],
[384],
[260],
[285],
[251],
[235],
[301],
[274],
[271],
[322],
[310],
```

[329],
[356],
[347],
[373],
[230],
[253],
[235],
[260],
[163],
[196],
[200],
[209],
[216],
[225],
[221],
[200],
[214],
[216],
[197],
[197],
[225],
[283],
[221],
[218],
[216],
[194],
[189],
[189],
[216],
[207],
[207],
[212],
[186],
[124],
[198],
[198],
[192],
[184],
[278],
[246],
[246],
[281],
[281],
[223],
[232],
[264],
[235],
[264],
[258],
[271],

```
       [264],
       [271],
       [260],
       [294]], dtype=int64)
```

```python
X = np.asanyarray(train[['CYLINDERS']])
```

```python
X
```

```
array([[ 4],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 4],
       [ 6],
       [12],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [12],
       [ 4],
       [ 4],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 8],
       [ 6],
       [ 6],
       [ 8],
       [ 6],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 8],
       [ 8],
       [10],
       [ 8],
       [ 8],
       [10],
       [ 8],
       [ 8],
       [ 6],
       [ 6],
```

```
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 8],
[12],
[ 8],
[12],
[ 8],
[12],
[12],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 8],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
```

```
[ 8],
[ 4],
[ 6],
[ 4],
[ 6],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 6],
[ 6],
[ 4],
[ 6],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
```

```
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 6],
[ 8],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
```

```
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
```

```
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[10],
[ 4],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
```

```
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 8],
[ 6],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
```

```
[ 6],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 8],
[ 4],
[ 6],
[ 4],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 8],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
```

```
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 4],
[ 4],
[ 6],
[ 4],
```

```
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 6],
[ 6],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[10],
[10],
[ 4],
[ 6],
[ 4],
[ 4],
[ 8],
[ 8],
```

```
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
```

```
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 8],
[ 8],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 8],
```

```
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
```

```
[ 4],
[ 4],
[ 4],
[ 4],
[ 3],
[ 3],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 6],
[ 8],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 6],
[ 4],
[ 4],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
```

```
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[12],
[12],
[12],
[12],
[ 4],
[ 4],
[ 4],
[ 4],
[ 3],
[ 3],
[10],
[10],
[ 4],
[ 4],
[ 4],
```

[4],
[4],
[4],
[4],
[4],
[4],
[4],
[4],
[6],
[4],
[6],
[4],
[4],
[4],
[6],
[6],
[6],
[4],
[4],
[4],
[4],
[4],
[4],
[4],
[4],
[4],
[4],
[6],
[6],
[4],
[6],
[6],
[6],
[6],
[4],
[4],
[4],
[4],
[4],
[8],
[6],
[6],
[4],
[4],
[6],
[4],
[4],
[6],
[6],
[8],

```
[ 8],
[ 8],
[ 8],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 5],
[ 5],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 5],
[ 5],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 5],
[ 5],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
```

```
       [ 6],
       [ 6],
       [ 6]], dtype=int64)
```

```python
y = np.asanyarray(train[['CO2EMISSIONS']])
```

```python
y
```

```
array([[196],
       [221],
       [136],
       [255],
       [244],
       [232],
       [267],
       [225],
       [239],
       [359],
       [338],
       [354],
       [338],
       [354],
       [359],
       [230],
       [214],
       [230],
       [214],
       [251],
       [224],
       [258],
       [224],
       [260],
       [227],
       [258],
       [288],
       [230],
       [242],
       [239],
       [258],
       [294],
       [336],
       [407],
       [354],
       [336],
       [407],
       [354],
       [306],
       [308],
       [262],
       [285],
       [262],
```

[267],
[281],
[281],
[292],
[209],
[209],
[297],
[356],
[320],
[380],
[322],
[380],
[380],
[437],
[193],
[200],
[181],
[181],
[193],
[200],
[209],
[209],
[209],
[230],
[228],
[237],
[232],
[193],
[200],
[209],
[221],
[230],
[237],
[202],
[209],
[213],
[232],
[292],
[232],
[292],
[281],
[292],
[292],
[191],
[292],
[338],
[338],
[317],
[338],
[317],

```
[209],
[260],
[230],
[253],
[320],
[370],
[274],
[209],
[200],
[278],
[253],
[278],
[283],
[290],
[196],
[212],
[264],
[250],
[264],
[264],
[225],
[246],
[189],
[225],
[218],
[221],
[235],
[207],
[251],
[232],
[258],
[262],
[232],
[248],
[248],
[260],
[253],
[260],
[255],
[382],
[340],
[382],
[340],
[338],
[322],
[377],
[283],
[294],
[271],
[290],
```

```
[258],
[269],
[294],
[382],
[340],
[276],
[258],
[179],
[189],
[192],
[177],
[163],
[209],
[259],
[290],
[272],
[366],
[334],
[380],
[344],
[380],
[344],
[362],
[380],
[344],
[380],
[362],
[414],
[432],
[411],
[419],
[435],
[229],
[200],
[228],
[184],
[244],
[271],
[269],
[292],
[270],
[324],
[290],
[299],
[329],
[184],
[173],
[184],
[172],
[182],
```

[191],
[182],
[191],
[182],
[159],
[317],
[294],
[317],
[294],
[317],
[294],
[317],
[294],
[290],
[207],
[232],
[246],
[230],
[230],
[230],
[294],
[251],
[251],
[237],
[222],
[276],
[269],
[225],
[230],
[301],
[294],
[320],
[317],
[258],
[237],
[294],
[251],
[308],
[251],
[237],
[235],
[222],
[186],
[198],
[202],
[195],
[202],
[168],
[168],
[278],

[250],
[271],
[264],
[255],
[285],
[276],
[179],
[179],
[182],
[179],
[156],
[200],
[193],
[373],
[350],
[396],
[360],
[386],
[368],
[488],
[225],
[264],
[281],
[209],
[216],
[214],
[230],
[368],
[346],
[274],
[254],
[288],
[267],
[294],
[366],
[317],
[308],
[283],
[283],
[264],
[336],
[336],
[315],
[308],
[288],
[308],
[288],
[320],
[318],
[152],

```
[184],
[264],
[278],
[294],
[205],
[177],
[165],
[179],
[165],
[184],
[170],
[166],
[162],
[196],
[186],
[205],
[186],
[216],
[126],
[244],
[264],
[244],
[209],
[271],
[237],
[229],
[258],
[251],
[218],
[228],
[232],
[239],
[283],
[297],
[347],
[366],
[334],
[344],
[380],
[344],
[380],
[380],
[344],
[414],
[432],
[411],
[419],
[435],
[271],
[269],
```

[292],
[270],
[324],
[290],
[285],
[277],
[329],
[209],
[276],
[230],
[290],
[272],
[317],
[294],
[338],
[322],
[354],
[317],
[294],
[317],
[294],
[182],
[186],
[200],
[209],
[221],
[251],
[110],
[161],
[166],
[246],
[209],
[159],
[172],
[244],
[265],
[276],
[308],
[177],
[177],
[196],
[304],
[255],
[258],
[267],
[237],
[264],
[255],
[221],
[143],

[145],
[214],
[232],
[230],
[244],
[177],
[184],
[202],
[196],
[235],
[172],
[189],
[265],
[269],
[283],
[255],
[267],
[253],
[283],
[267],
[294],
[175],
[271],
[244],
[294],
[334],
[345],
[237],
[244],
[294],
[306],
[290],
[306],
[285],
[280],
[306],
[306],
[301],
[315],
[310],
[315],
[310],
[218],
[230],
[248],
[232],
[251],
[267],
[225],
[209],

```
[225],
[216],
[235],
[216],
[255],
[340],
[267],
[278],
[363],
[225],
[216],
[216],
[255],
[294],
[308],
[251],
[184],
[193],
[193],
[223],
[225],
[193],
[196],
[216],
[221],
[193],
[198],
[200],
[175],
[177],
[175],
[212],
[214],
[276],
[237],
[262],
[253],
[271],
[207],
[209],
[230],
[361],
[400],
[274],
[334],
[225],
[225],
[354],
[342],
[285],
```

[278],
[345],
[331],
[283],
[283],
[277],
[354],
[347],
[129],
[138],
[221],
[239],
[253],
[175],
[320],
[223],
[239],
[244],
[253],
[244],
[290],
[283],
[297],
[262],
[380],
[255],
[189],
[271],
[260],
[294],
[235],
[281],
[207],
[262],
[145],
[368],
[346],
[292],
[292],
[347],
[334],
[301],
[189],
[184],
[198],
[196],
[205],
[283],
[290],
[177],

```
[170],
[159],
[161],
[170],
[163],
[166],
[177],
[168],
[166],
[223],
[225],
[175],
[168],
[235],
[221],
[228],
[184],
[216],
[216],
[242],
[224],
[239],
[244],
[242],
[347],
[352],
[304],
[308],
[179],
[212],
[264],
[301],
[194],
[235],
[230],
[239],
[242],
[235],
[202],
[267],
[269],
[262],
[297],
[310],
[310],
[419],
[407],
[324],
[359],
[260],
```

```
[256],
[278],
[327],
[356],
[278],
[276],
[327],
[205],
[207],
[225],
[246],
[380],
[361],
[361],
[179],
[179],
[179],
[179],
[200],
[179],
[175],
[200],
[179],
[179],
[179],
[184],
[191],
[207],
[196],
[191],
[184],
[207],
[196],
[191],
[184],
[191],
[184],
[191],
[184],
[207],
[196],
[191],
[207],
[196],
[191],
[184],
[189],
[191],
[288],
[283],
```

```
[267],
[198],
[198],
[138],
[147],
[198],
[235],
[200],
[202],
[207],
[253],
[264],
[262],
[175],
[214],
[380],
[283],
[292],
[315],
[301],
[186],
[198],
[251],
[244],
[253],
[202],
[255],
[161],
[179],
[359],
[398],
[152],
[177],
[327],
[315],
[232],
[237],
[246],
[242],
[253],
[248],
[253],
[253],
[258],
[232],
[246],
[244],
[248],
[246],
[251],
```

[322],
[274],
[274],
[274],
[274],
[212],
[228],
[225],
[313],
[281],
[320],
[299],
[251],
[315],
[212],
[225],
[239],
[251],
[255],
[260],
[264],
[292],
[260],
[294],
[340],
[313],
[354],
[290],
[285],
[275],
[274],
[260],
[262],
[254],
[368],
[393],
[393],
[359],
[193],
[207],
[184],
[182],
[147],
[147],
[389],
[389],
[221],
[223],
[179],
[196],

[258],
[179],
[196],
[258],
[288],
[198],
[228],
[267],
[207],
[301],
[196],
[212],
[177],
[297],
[297],
[221],
[131],
[138],
[177],
[168],
[163],
[170],
[156],
[159],
[196],
[189],
[297],
[320],
[237],
[258],
[264],
[191],
[189],
[110],
[108],
[126],
[216],
[218],
[384],
[260],
[285],
[251],
[235],
[301],
[274],
[271],
[322],
[310],
[329],
[356],

```
[347],
[373],
[230],
[253],
[235],
[260],
[163],
[196],
[200],
[209],
[216],
[225],
[221],
[200],
[214],
[216],
[197],
[197],
[225],
[283],
[221],
[218],
[216],
[194],
[189],
[189],
[216],
[207],
[207],
[212],
[186],
[124],
[198],
[198],
[192],
[184],
[278],
[246],
[246],
[281],
[281],
[223],
[232],
[264],
[235],
[264],
[258],
[271],
[264],
[271],
```

```
       [260],
       [294]], dtype=int64)
```

```
from sklearn import linear_model
```

```
# y = mx + c == y is dependent variable, x is the data(indep.
variable) m is COEFFICIENT  and c is INTERCEPT
```

```
regr = linear_model.LinearRegression()
```

```
regr
```

```
LinearRegression()
```

```
train.shape
```

```
(845, 4)
```

```
test.shape
```

```
(222, 4)
```

```
train_x = np.asanyarray(train[['CYLINDERS']]) #x represents
independent variables
train_y = np.asanyarray(train[['CO2EMISSIONS']]) #y represent
dependent variable
```

```
train_x # All Values of Cylinders column this will be used for
training
```

```
array([[ 4],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 4],
       [ 6],
       [12],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [12],
       [ 4],
       [ 4],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 6],
```

```
[ 6],
[ 8],
[ 6],
[ 6],
[ 8],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 8],
[ 8],
[10],
[ 8],
[ 8],
[10],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 8],
[12],
[ 8],
[12],
[ 8],
[12],
[12],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
```

```
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 8],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 6],
[ 4],
[ 6],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
```

```
[ 4],
[ 6],
[ 6],
[ 4],
[ 6],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
```

```
[ 8],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 6],
[ 8],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
```

```
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[10],
[ 4],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 8],
```

```
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 8],
[ 6],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
```

```
       [ 4],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 6],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 6],
       [ 6],
       [ 8],
       [ 8],
       [ 8],
       [ 6],
       [ 6],
       [ 8],
       [ 8],
       [ 4],
       [ 6],
       [ 4],
       [ 6],
       [ 6],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 8],
       [ 4],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 6],
       [ 4],
       [ 4],
```

```
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 8],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 6],
```

```
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 4],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 6],
[ 6],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
```

```
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[10],
[10],
[ 4],
[ 6],
[ 4],
[ 4],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
```

```
[ 6],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 8],
[ 8],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
```

```
[ 4],
[ 8],
[ 8],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
```

```
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 3],
[ 3],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 6],
[ 8],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 6],
[ 4],
[ 4],
```

```
[ 8],
[ 8],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 8],
[ 6],
[ 8],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[ 8],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 6],
[ 6],
```

```
[ 6],
[ 6],
[ 6],
[ 6],
[ 6],
[12],
[12],
[12],
[12],
[ 4],
[ 4],
[ 4],
[ 4],
[ 3],
[ 3],
[10],
[10],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 6],
[ 4],
```

```
[ 6],
[ 6],
[ 6],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 8],
[ 6],
[ 6],
[ 4],
[ 4],
[ 6],
[ 4],
[ 4],
[ 6],
[ 6],
[ 8],
[ 8],
[ 8],
[ 8],
[ 4],
[ 6],
[ 4],
[ 6],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 5],
[ 5],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 6],
[ 4],
[ 5],
[ 5],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
[ 4],
```

```
       [ 4],
       [ 4],
       [ 4],
       [ 4],
       [ 4],
       [ 4],
       [ 4],
       [ 6],
       [ 4],
       [ 4],
       [ 6],
       [ 6],
       [ 5],
       [ 5],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 6],
       [ 6]], dtype=int64)
```

```
train_y # Column Values of column co2emissions  will be used for the
training only as dependent varibale
```

```
array([[196],
       [221],
       [136],
       [255],
       [244],
       [232],
       [267],
       [225],
       [239],
       [359],
       [338],
       [354],
       [338],
       [354],
       [359],
       [230],
       [214],
       [230],
       [214],
       [251],
       [224],
       [258],
       [224],
       [260],
```

```
[227],
[258],
[288],
[230],
[242],
[239],
[258],
[294],
[336],
[407],
[354],
[336],
[407],
[354],
[306],
[308],
[262],
[285],
[262],
[267],
[281],
[281],
[292],
[209],
[209],
[297],
[356],
[320],
[380],
[322],
[380],
[380],
[437],
[193],
[200],
[181],
[181],
[193],
[200],
[209],
[209],
[209],
[230],
[228],
[237],
[232],
[193],
[200],
[209],
[221],
```

[230],
[237],
[202],
[209],
[213],
[232],
[292],
[232],
[292],
[281],
[292],
[292],
[191],
[292],
[338],
[338],
[317],
[338],
[317],
[209],
[260],
[230],
[253],
[320],
[370],
[274],
[209],
[200],
[278],
[253],
[278],
[283],
[290],
[196],
[212],
[264],
[250],
[264],
[264],
[225],
[246],
[189],
[225],
[218],
[221],
[235],
[207],
[251],
[232],
[258],

```
[262],
[232],
[248],
[248],
[260],
[253],
[260],
[255],
[382],
[340],
[382],
[340],
[338],
[322],
[377],
[283],
[294],
[271],
[290],
[258],
[269],
[294],
[382],
[340],
[276],
[258],
[179],
[189],
[192],
[177],
[163],
[209],
[259],
[290],
[272],
[366],
[334],
[380],
[344],
[380],
[344],
[362],
[380],
[344],
[380],
[362],
[414],
[432],
[411],
[419],
```

[435],
[229],
[200],
[228],
[184],
[244],
[271],
[269],
[292],
[270],
[324],
[290],
[299],
[329],
[184],
[173],
[184],
[172],
[182],
[191],
[182],
[191],
[182],
[159],
[317],
[294],
[317],
[294],
[317],
[294],
[317],
[294],
[290],
[207],
[232],
[246],
[230],
[230],
[230],
[294],
[251],
[251],
[237],
[222],
[276],
[269],
[225],
[230],
[301],
[294],

```
[320],
[317],
[258],
[237],
[294],
[251],
[308],
[251],
[237],
[235],
[222],
[186],
[198],
[202],
[195],
[202],
[168],
[168],
[278],
[250],
[271],
[264],
[255],
[285],
[276],
[179],
[179],
[182],
[179],
[156],
[200],
[193],
[373],
[350],
[396],
[360],
[386],
[368],
[488],
[225],
[264],
[281],
[209],
[216],
[214],
[230],
[368],
[346],
[274],
[254],
```

```
[288],
[267],
[294],
[366],
[317],
[308],
[283],
[283],
[264],
[336],
[336],
[315],
[308],
[288],
[308],
[288],
[320],
[318],
[152],
[184],
[264],
[278],
[294],
[205],
[177],
[165],
[179],
[165],
[184],
[170],
[166],
[162],
[196],
[186],
[205],
[186],
[216],
[126],
[244],
[264],
[244],
[209],
[271],
[237],
[229],
[258],
[251],
[218],
[228],
[232],
```

```
[239],
[283],
[297],
[347],
[366],
[334],
[344],
[380],
[344],
[380],
[380],
[344],
[414],
[432],
[411],
[419],
[435],
[271],
[269],
[292],
[270],
[324],
[290],
[285],
[277],
[329],
[209],
[276],
[230],
[290],
[272],
[317],
[294],
[338],
[322],
[354],
[317],
[294],
[317],
[294],
[182],
[186],
[200],
[209],
[221],
[251],
[110],
[161],
[166],
[246],
```

```
[209],
[159],
[172],
[244],
[265],
[276],
[308],
[177],
[177],
[196],
[304],
[255],
[258],
[267],
[237],
[264],
[255],
[221],
[143],
[145],
[214],
[232],
[230],
[244],
[177],
[184],
[202],
[196],
[235],
[172],
[189],
[265],
[269],
[283],
[255],
[267],
[253],
[283],
[267],
[294],
[175],
[271],
[244],
[294],
[334],
[345],
[237],
[244],
[294],
[306],
```

[290],
[306],
[285],
[280],
[306],
[306],
[301],
[315],
[310],
[315],
[310],
[218],
[230],
[248],
[232],
[251],
[267],
[225],
[209],
[225],
[216],
[235],
[216],
[255],
[340],
[267],
[278],
[363],
[225],
[216],
[216],
[255],
[294],
[308],
[251],
[184],
[193],
[193],
[223],
[225],
[193],
[196],
[216],
[221],
[193],
[198],
[200],
[175],
[177],
[175],

```
[212],
[214],
[276],
[237],
[262],
[253],
[271],
[207],
[209],
[230],
[361],
[400],
[274],
[334],
[225],
[225],
[354],
[342],
[285],
[278],
[345],
[331],
[283],
[283],
[277],
[354],
[347],
[129],
[138],
[221],
[239],
[253],
[175],
[320],
[223],
[239],
[244],
[253],
[244],
[290],
[283],
[297],
[262],
[380],
[255],
[189],
[271],
[260],
[294],
[235],
```

[281],
[207],
[262],
[145],
[368],
[346],
[292],
[292],
[347],
[334],
[301],
[189],
[184],
[198],
[196],
[205],
[283],
[290],
[177],
[170],
[159],
[161],
[170],
[163],
[166],
[177],
[168],
[166],
[223],
[225],
[175],
[168],
[235],
[221],
[228],
[184],
[216],
[216],
[242],
[224],
[239],
[244],
[242],
[347],
[352],
[304],
[308],
[179],
[212],
[264],

```
[301],
[194],
[235],
[230],
[239],
[242],
[235],
[202],
[267],
[269],
[262],
[297],
[310],
[310],
[419],
[407],
[324],
[359],
[260],
[256],
[278],
[327],
[356],
[278],
[276],
[327],
[205],
[207],
[225],
[246],
[380],
[361],
[361],
[179],
[179],
[179],
[179],
[200],
[179],
[175],
[200],
[179],
[179],
[179],
[184],
[191],
[207],
[196],
[191],
[184],
```

[207],
[196],
[191],
[184],
[191],
[184],
[191],
[184],
[207],
[196],
[191],
[207],
[196],
[191],
[184],
[189],
[191],
[288],
[283],
[267],
[198],
[198],
[138],
[147],
[198],
[235],
[200],
[202],
[207],
[253],
[264],
[262],
[175],
[214],
[380],
[283],
[292],
[315],
[301],
[186],
[198],
[251],
[244],
[253],
[202],
[255],
[161],
[179],
[359],
[398],

```
[152],
[177],
[327],
[315],
[232],
[237],
[246],
[242],
[253],
[248],
[253],
[253],
[258],
[232],
[246],
[244],
[248],
[246],
[251],
[322],
[274],
[274],
[274],
[274],
[212],
[228],
[225],
[313],
[281],
[320],
[299],
[251],
[315],
[212],
[225],
[239],
[251],
[255],
[260],
[264],
[292],
[260],
[294],
[340],
[313],
[354],
[290],
[285],
[275],
[274],
```

[260],
[262],
[254],
[368],
[393],
[393],
[359],
[193],
[207],
[184],
[182],
[147],
[147],
[389],
[389],
[221],
[223],
[179],
[196],
[258],
[179],
[196],
[258],
[288],
[198],
[228],
[267],
[207],
[301],
[196],
[212],
[177],
[297],
[297],
[221],
[131],
[138],
[177],
[168],
[163],
[170],
[156],
[159],
[196],
[189],
[297],
[320],
[237],
[258],
[264],

[191],
[189],
[110],
[108],
[126],
[216],
[218],
[384],
[260],
[285],
[251],
[235],
[301],
[274],
[271],
[322],
[310],
[329],
[356],
[347],
[373],
[230],
[253],
[235],
[260],
[163],
[196],
[200],
[209],
[216],
[225],
[221],
[200],
[214],
[216],
[197],
[197],
[225],
[283],
[221],
[218],
[216],
[194],
[189],
[189],
[216],
[207],
[207],
[212],
[186],

```
       [124],
       [198],
       [198],
       [192],
       [184],
       [278],
       [246],
       [246],
       [281],
       [281],
       [223],
       [232],
       [264],
       [235],
       [264],
       [258],
       [271],
       [264],
       [271],
       [260],
       [294]], dtype=int64)
```

regr.fit(independentvariable,dependentvariable)

```
regr.fit(train_x, train_y)
```

```
LinearRegression()
```

```
# The coefficients
print ('Coefficients: ', regr.coef_)
print ('Intercept: ',regr.intercept_)
```

```
Coefficients:  [[30.70676327]]
Intercept:  [78.19444198]
```

```
regr.coef_
```

```
array([[30.70676327]])
```

```
regr.intercept_
```

```
array([78.19444198])
```

```
regr.coef_[0][0]
```

```
30.706763266880237
```

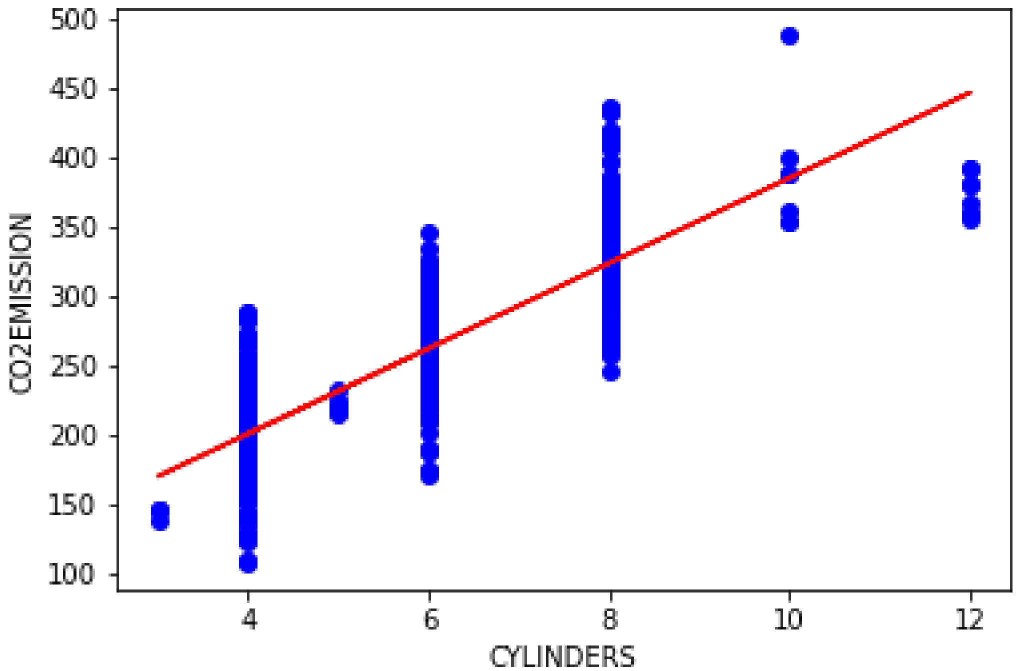
```
regr.intercept_[0]
```

```
78.19444197620655
```

Plot Outputs

```
plt.scatter(train.CYLINDERS, train.CO2EMISSIONS,  color='blue')
plt.plot(train_x, regr.coef_[0][0]*train_x + regr.intercept_[0], 'r')
#y = mx+c
plt.xlabel("CYLINDERS")
plt.ylabel("CO2EMISSION")
```

```
Text(0, 0.5, 'CO2EMISSION')
```



Evaluation

```
test_x = np.asanyarray(test[['CYLINDERS']]) #question in exam
Questions
test_y = np.asanyarray(test[['CO2EMISSIONS']])#(ACTUAL answers)

test_x.shape #independent variable data for testing
```

```
(222, 1)
```

```
test_y.shape #dependent variable for testing
```

```
(222, 1)
```

```
predicted_y = regr.predict(test_x) #whatever answerr will be generated
will be stored in predicted_y predicted

predicted_y
```

```
array([[262.43502158],
       [262.43502158],
```

```
[201.02149504],
[446.67560118],
[201.02149504],
[201.02149504],
[201.02149504],
[262.43502158],
[446.67560118],
[201.02149504],
[262.43502158],
[262.43502158],
[385.26207465],
[385.26207465],
[323.84854811],
[262.43502158],
[323.84854811],
[201.02149504],
[201.02149504],
[201.02149504],
[262.43502158],
[262.43502158],
[262.43502158],
[323.84854811],
[262.43502158],
[323.84854811],
[446.67560118],
[262.43502158],
[262.43502158],
[323.84854811],
[323.84854811],
[323.84854811],
[323.84854811],
[262.43502158],
[323.84854811],
[201.02149504],
[201.02149504],
[201.02149504],
[262.43502158],
[262.43502158],
[262.43502158],
[323.84854811],
[323.84854811],
[323.84854811],
[262.43502158],
[262.43502158],
[323.84854811],
[201.02149504],
[201.02149504],
[262.43502158],
[201.02149504],
[262.43502158],
```

```
[323.84854811],
[323.84854811],
[201.02149504],
[262.43502158],
[201.02149504],
[262.43502158],
[323.84854811],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[262.43502158],
[201.02149504],
[201.02149504],
[262.43502158],
[262.43502158],
[201.02149504],
[262.43502158],
[323.84854811],
[262.43502158],
[323.84854811],
[262.43502158],
[262.43502158],
[262.43502158],
[262.43502158],
[262.43502158],
[323.84854811],
[201.02149504],
[201.02149504],
[323.84854811],
[262.43502158],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[323.84854811],
[323.84854811],
[262.43502158],
[201.02149504],
[201.02149504],
[262.43502158],
[262.43502158],
[323.84854811],
[262.43502158],
[323.84854811],
[323.84854811],
[323.84854811],
[323.84854811],
[201.02149504],
[201.02149504],
```

```
[201.02149504],
[201.02149504],
[262.43502158],
[323.84854811],
[323.84854811],
[323.84854811],
[323.84854811],
[323.84854811],
[323.84854811],
[323.84854811],
[323.84854811],
[262.43502158],
[323.84854811],
[323.84854811],
[323.84854811],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[262.43502158],
[262.43502158],
[201.02149504],
[201.02149504],
[262.43502158],
[323.84854811],
[262.43502158],
[323.84854811],
[323.84854811],
[323.84854811],
[262.43502158],
[262.43502158],
[262.43502158],
[201.02149504],
[201.02149504],
[201.02149504],
[262.43502158],
[262.43502158],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[446.67560118],
[323.84854811],
[262.43502158],
```

```
[323.84854811],
[323.84854811],
[262.43502158],
[323.84854811],
[262.43502158],
[262.43502158],
[323.84854811],
[201.02149504],
[201.02149504],
[201.02149504],
[262.43502158],
[323.84854811],
[262.43502158],
[323.84854811],
[323.84854811],
[323.84854811],
[201.02149504],
[262.43502158],
[323.84854811],
[323.84854811],
[323.84854811],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
[262.43502158],
[201.02149504],
[262.43502158],
[201.02149504],
[262.43502158],
[201.02149504],
[201.02149504],
[201.02149504],
[262.43502158],
[262.43502158],
[262.43502158],
[262.43502158],
[323.84854811],
[323.84854811],
[262.43502158],
[262.43502158],
[446.67560118],
[446.67560118],
[446.67560118],
[201.02149504],
[201.02149504],
[201.02149504],
[201.02149504],
```

```
       [201.02149504],
       [201.02149504],
       [201.02149504],
       [201.02149504],
       [201.02149504],
       [201.02149504],
       [262.43502158],
       [201.02149504],
       [262.43502158],
       [201.02149504],
       [262.43502158],
       [201.02149504],
       [231.72825831],
       [201.02149504],
       [201.02149504],
       [201.02149504],
       [231.72825831],
       [231.72825831],
       [262.43502158],
       [201.02149504]])
```

```
test_y.shape
```

```
(222, 1)
```

```
predicted_y.shape
```

```
(222, 1)
```

```
from sklearn.metrics import r2_score
print(f"Mean absolute error: {np.mean(np.absolute(predicted_y - test_y))} ")# pred - actual
```

```
Mean absolute error: 26.218296344468087
```

```
print("Residual sum of squares (MSE): %.2f" % np.mean((predicted_y - test_y) **2))
```

```
Residual sum of squares (MSE): 1109.97
```

```
from sklearn.metrics import r2_score
print(f"R2-score:{r2_score(test_y , predicted_y)*100} %") # Most accurate thing or evaluation metrics
```

```
R2-score:72.38654274529334 %
```

Apply Lasso Regression to cover the overfitting and underfitting problem

```
from sklearn.linear_model import Lasso
```

```
L = Lasso(alpha = 1)
```

```
L.fit(train_x,train_y)
```

```
Lasso(alpha=1)
```

```
y_pred1 = L.predict(test_x)
```

```
from sklearn.metrics import r2_score
print("R2-score",r2_score(test_y,y_pred1))
print(f"Mean absolute error: {np.mean(np.absolute(y_pred1 - test_y))}
")# pred - actual
```

```
R2-score 0.7256514539177052
Mean absolute error: 68.97569534677652
```

Que.5 Train another model with independent variable as FuelConsumption_comb and dependent variable as Co2Emission

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
df = pd.read_csv("C:/Users/Lenovo/Documents/Data
Set/FuelConsumption.csv")
```

```
df
```

```
      MODELYEAR   MAKE       MODEL    VEHICLECLASS  ENGINESIZE
CYLINDERS  \
0          2014  ACURA         ILX         COMPACT         2.0
4
1          2014  ACURA         ILX         COMPACT         2.4
4
2          2014  ACURA  ILX HYBRID         COMPACT         1.5
4
3          2014  ACURA     MDX 4WD     SUV - SMALL         3.5
6
4          2014  ACURA     RDX AWD     SUV - SMALL         3.5
6
...         ...    ...         ...             ...         ...
...
1062       2014  VOLVO    XC60 AWD     SUV - SMALL         3.0
6
1063       2014  VOLVO    XC60 AWD     SUV - SMALL         3.2
6
1064       2014  VOLVO    XC70 AWD     SUV - SMALL         3.0
6
1065       2014  VOLVO    XC70 AWD     SUV - SMALL         3.2
6
1066       2014  VOLVO    XC90 AWD  SUV - STANDARD         3.2
```

```
6

    TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY
\
0             AS5        Z                   9.9                  6.7

1              M6        Z                  11.2                  7.7

2             AV7        Z                   6.0                  5.8

3             AS6        Z                  12.7                  9.1

4             AS6        Z                  12.1                  8.7

...           ...      ...                   ...                  ...

1062          AS6        X                  13.4                  9.8

1063          AS6        X                  13.2                  9.5

1064          AS6        X                  13.4                  9.8

1065          AS6        X                  12.9                  9.3

1066          AS6        X                  14.9                 10.2


      FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                      8.5                        33           196
1                      9.6                        29           221
2                      5.9                        48           136
3                     11.1                        25           255
4                     10.6                        27           244
...                    ...                       ...           ...
1062                  11.8                        24           271
1063                  11.5                        25           264
1064                  11.8                        24           271
1065                  11.3                        25           260
1066                  12.8                        22           294

[1067 rows x 13 columns]
```

```
df.head()
```

```
   MODELYEAR   MAKE       MODEL VEHICLECLASS  ENGINESIZE  CYLINDERS  \
0       2014  ACURA         ILX      COMPACT         2.0          4
1       2014  ACURA         ILX      COMPACT         2.4          4
2       2014  ACURA  ILX HYBRID      COMPACT         1.5          4
3       2014  ACURA     MDX 4WD  SUV - SMALL         3.5          6
```

```
4       2014  ACURA     RDX AWD  SUV - SMALL         3.5          6

  TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY  \
0          AS5        Z                   9.9                  6.7
1           M6        Z                  11.2                  7.7
2          AV7        Z                   6.0                  5.8
3          AS6        Z                  12.7                  9.1
4          AS6        Z                  12.1                  8.7

   FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                   8.5                        33           196
1                   9.6                        29           221
2                   5.9                        48           136
3                  11.1                        25           255
4                  10.6                        27           244
```

```
df.tail()
```

```
      MODELYEAR   MAKE     MODEL    VEHICLECLASS  ENGINESIZE
CYLINDERS  \
1062       2014  VOLVO  XC60 AWD     SUV - SMALL         3.0
6
1063       2014  VOLVO  XC60 AWD     SUV - SMALL         3.2
6
1064       2014  VOLVO  XC70 AWD     SUV - SMALL         3.0
6
1065       2014  VOLVO  XC70 AWD     SUV - SMALL         3.2
6
1066       2014  VOLVO  XC90 AWD  SUV - STANDARD         3.2
6

     TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY
\
1062          AS6        X                  13.4                  9.8

1063          AS6        X                  13.2                  9.5

1064          AS6        X                  13.4                  9.8

1065          AS6        X                  12.9                  9.3

1066          AS6        X                  14.9                 10.2


      FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
1062                  11.8                        24           271
1063                  11.5                        25           264
1064                  11.8                        24           271
1065                  11.3                        25           260
1066                  12.8                        22           294
```

```
df.describe()
```

```
       MODELYEAR   ENGINESIZE    CYLINDERS  FUELCONSUMPTION_CITY  \
count     1067.0  1067.000000  1067.000000           1067.000000
mean      2014.0     3.346298     5.794752             13.296532
std          0.0     1.415895     1.797447              4.101253
min       2014.0     1.000000     3.000000              4.600000
25%       2014.0     2.000000     4.000000             10.250000
50%       2014.0     3.400000     6.000000             12.600000
75%       2014.0     4.300000     8.000000             15.550000
max       2014.0     8.400000    12.000000             30.200000

       FUELCONSUMPTION_HWY  FUELCONSUMPTION_COMB
FUELCONSUMPTION_COMB_MPG  \
count          1067.000000           1067.000000
1067.000000
mean              9.474602             11.580881
26.441425
std               2.794510              3.485595
7.468702
min               4.900000              4.700000
11.000000
25%               7.500000              9.000000
21.000000
50%               8.800000             10.900000
26.000000
75%              10.850000             13.350000
31.000000
max              20.500000             25.800000
60.000000

       CO2EMISSIONS
count   1067.000000
mean     256.228679
std       63.372304
min      108.000000
25%      207.000000
50%      251.000000
75%      294.000000
max      488.000000
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1067 entries, 0 to 1066
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   MODELYEAR                 1067 non-null   int64
 1   MAKE                      1067 non-null   object
 2   MODEL                     1067 non-null   object
```

```
 3   VEHICLECLASS              1067 non-null   object
 4   ENGINESIZE                1067 non-null   float64
 5   CYLINDERS                 1067 non-null   int64
 6   TRANSMISSION              1067 non-null   object
 7   FUELTYPE                  1067 non-null   object
 8   FUELCONSUMPTION_CITY      1067 non-null   float64
 9   FUELCONSUMPTION_HWY       1067 non-null   float64
 10  FUELCONSUMPTION_COMB      1067 non-null   float64
 11  FUELCONSUMPTION_COMB_MPG  1067 non-null   int64
 12  CO2EMISSIONS              1067 non-null   int64
dtypes: float64(4), int64(4), object(5)
memory usage: 108.5+ KB
```

```
df.isnull().sum()
```

```
MODELYEAR                   0
MAKE                        0
MODEL                       0
VEHICLECLASS                0
ENGINESIZE                  0
CYLINDERS                   0
TRANSMISSION                0
FUELTYPE                    0
FUELCONSUMPTION_CITY        0
FUELCONSUMPTION_HWY         0
FUELCONSUMPTION_COMB        0
FUELCONSUMPTION_COMB_MPG    0
CO2EMISSIONS                0
dtype: int64
```

```
df[['MODELYEAR','ENGINESIZE','CYLINDERS']]
```

```
      MODELYEAR  ENGINESIZE  CYLINDERS
0          2014         2.0          4
1          2014         2.4          4
2          2014         1.5          4
3          2014         3.5          6
4          2014         3.5          6
...         ...         ...        ...
1062       2014         3.0          6
1063       2014         3.2          6
1064       2014         3.0          6
1065       2014         3.2          6
1066       2014         3.2          6

[1067 rows x 3 columns]
```

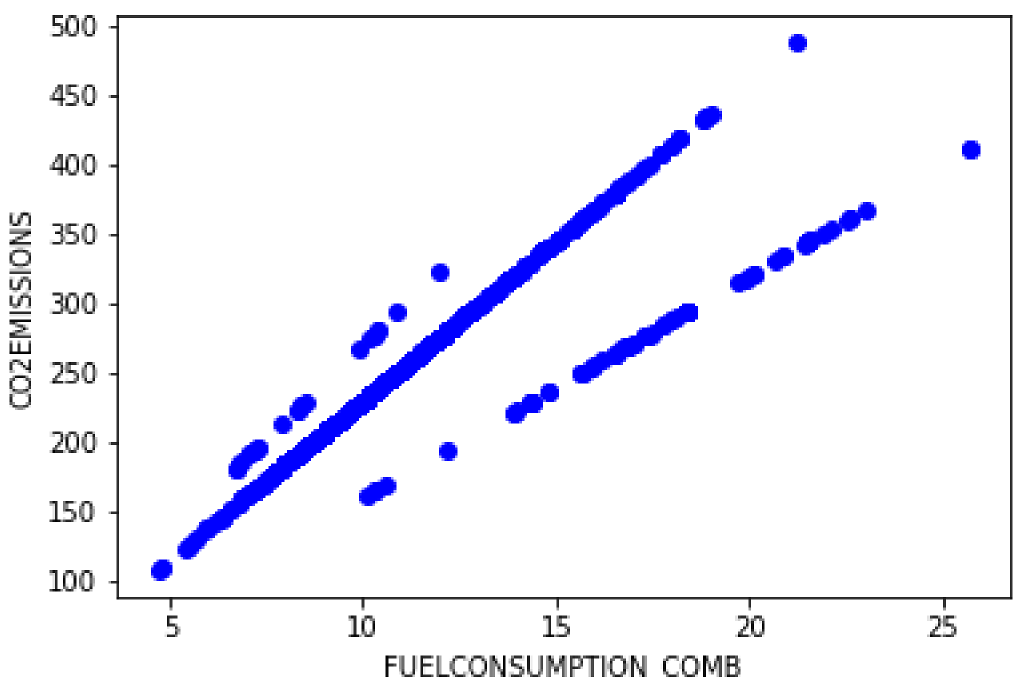
```
df[["FUELCONSUMPTION_COMB","CO2EMISSIONS"]]
```

```
      FUELCONSUMPTION_COMB  CO2EMISSIONS
0                      8.5           196
```

```
1                         9.6           221
2                         5.9           136
3                        11.1           255
4                        10.6           244
...                       ...           ...
1062                     11.8           271
1063                     11.5           264
1064                     11.8           271
1065                     11.3           260
1066                     12.8           294

[1067 rows x 2 columns]
```

Simple Regression Model

```
# Train Data Distribution

plt.scatter(train.FUELCONSUMPTION_COMB, train.CO2EMISSIONS,
color='blue')
plt.xlabel("FUELCONSUMPTION_COMB")
plt.ylabel("CO2EMISSIONS")
plt.show()
```



Modelling

```
train[['FUELCONSUMPTION_COMB']]
```

```
      FUELCONSUMPTION_COMB
0                      8.5
1                      9.6
2                      5.9
3                     11.1
4                     10.6
...                    ...
1062                  11.8
1063                  11.5
1064                  11.8
1065                  11.3
1066                  12.8

[845 rows x 1 columns]
```

```
train[['CO2EMISSIONS']]
```

```
      CO2EMISSIONS
0              196
1              221
2              136
3              255
4              244
...            ...
1062           271
1063           264
1064           271
1065           260
1066           294

[845 rows x 1 columns]
```

```
np.asanyarray(train[['FUELCONSUMPTION_COMB']])
```

```
array([[ 8.5],
       [ 9.6],
       [ 5.9],
       [11.1],
       [10.6],
       [10.1],
       [11.6],
       [ 9.8],
       [10.4],
       [15.6],
       [14.7],
       [15.4],
       [14.7],
       [15.4],
       [15.6],
       [10. ],
       [ 9.3],
```

```
[10. ],
[ 9.3],
[10.9],
[ 8.3],
[11.2],
[ 8.3],
[11.3],
[ 8.4],
[11.2],
[12.5],
[ 8.5],
[10.5],
[10.4],
[11.2],
[10.9],
[14.6],
[17.7],
[15.4],
[14.6],
[17.7],
[15.4],
[13.3],
[13.4],
[11.4],
[12.4],
[11.4],
[11.6],
[12.2],
[12.2],
[12.7],
[ 9.1],
[ 9.1],
[12.9],
[15.5],
[13.9],
[16.5],
[14. ],
[16.5],
[16.5],
[19. ],
[ 8.4],
[ 8.7],
[ 6.7],
[ 6.7],
[ 8.4],
[ 8.7],
[ 9.1],
[ 9.1],
[ 9.1],
[10. ],
```

```
[ 9.9],
[10.3],
[10.1],
[ 8.4],
[ 8.7],
[ 9.1],
[ 9.6],
[10. ],
[10.3],
[ 8.8],
[ 9.1],
[ 7.9],
[10.1],
[12.7],
[10.1],
[12.7],
[12.2],
[12.7],
[12.7],
[ 8.3],
[12.7],
[14.7],
[14.7],
[13.8],
[14.7],
[13.8],
[ 9.1],
[11.3],
[10. ],
[11. ],
[13.9],
[16.1],
[11.9],
[ 9.1],
[ 8.7],
[12.1],
[11. ],
[12.1],
[12.3],
[12.6],
[ 8.5],
[ 9.2],
[11.5],
[15.6],
[11.5],
[16.5],
[ 9.8],
[10.7],
[ 8.2],
[ 9.8],
```

```
[ 9.5],
[ 9.6],
[10.2],
[ 9. ],
[10.9],
[10.1],
[11.2],
[11.4],
[10.1],
[10.8],
[10.8],
[11.3],
[11. ],
[11.3],
[11.1],
[16.6],
[14.8],
[16.6],
[14.8],
[14.7],
[20.1],
[16.4],
[12.3],
[12.8],
[11.8],
[12.6],
[11.2],
[11.7],
[12.8],
[16.6],
[14.8],
[12. ],
[11.2],
[ 7.8],
[ 8.2],
[ 7.1],
[ 7.7],
[ 7.1],
[ 9.1],
[16.2],
[12.6],
[17. ],
[15.9],
[20.9],
[16.5],
[21.5],
[16.5],
[21.5],
[22.6],
[16.5],
```

```
[21.5],
[16.5],
[22.6],
[18. ],
[18.8],
[25.7],
[18.2],
[18.9],
[14.3],
[ 8.7],
[ 9.9],
[ 8. ],
[10.6],
[11.8],
[16.8],
[12.7],
[16.9],
[14.1],
[12.6],
[13. ],
[14.3],
[ 8. ],
[ 7.5],
[ 8. ],
[ 7.5],
[ 7.9],
[ 8.3],
[ 7.9],
[ 8.3],
[ 7.9],
[ 6.9],
[13.8],
[18.4],
[13.8],
[18.4],
[13.8],
[18.4],
[13.8],
[18.4],
[12.6],
[ 9. ],
[10.1],
[10.7],
[14.4],
[10. ],
[14.4],
[12.8],
[10.9],
[10.9],
[14.8],
```

```
[13.9],
[12. ],
[16.8],
[ 9.8],
[10. ],
[13.1],
[12.8],
[13.9],
[13.8],
[11.2],
[10.3],
[12.8],
[10.9],
[13.4],
[10.9],
[14.8],
[10.2],
[13.9],
[ 8.1],
[ 8.6],
[ 8.8],
[12.2],
[ 8.8],
[ 7.3],
[ 7.3],
[12.1],
[15.6],
[11.8],
[16.5],
[11.1],
[12.4],
[12. ],
[ 7.8],
[ 7.8],
[ 7.9],
[ 7.8],
[ 6.8],
[ 8.7],
[ 8.4],
[16.2],
[21.9],
[17.2],
[22.5],
[16.8],
[23. ],
[21.2],
[ 9.8],
[11.5],
[12.2],
[ 9.1],
```

```
[ 9.4],
[ 9.3],
[10. ],
[16. ],
[21.6],
[11.9],
[15.9],
[12.5],
[16.7],
[12.8],
[15.9],
[13.8],
[13.4],
[12.3],
[12.3],
[16.5],
[14.6],
[14.6],
[19.7],
[13.4],
[18. ],
[13.4],
[18. ],
[20. ],
[19.9],
[ 6.6],
[ 8. ],
[11.5],
[12.1],
[12.8],
[ 8.9],
[ 7.7],
[10.3],
[ 7.8],
[10.3],
[ 8. ],
[10.6],
[ 7.2],
[10.1],
[ 8.5],
[ 8.1],
[ 8.9],
[ 8.1],
[ 9.4],
[ 5.5],
[10.6],
[11.5],
[10.6],
[ 9.1],
[11.8],
```

```
[10.3],
[14.3],
[11.2],
[15.7],
[ 9.5],
[ 9.9],
[10.1],
[10.4],
[12.3],
[12.9],
[15.1],
[15.9],
[20.9],
[21.5],
[16.5],
[21.5],
[16.5],
[16.5],
[21.5],
[18. ],
[18.8],
[25.7],
[18.2],
[18.9],
[11.8],
[16.8],
[12.7],
[16.9],
[14.1],
[12.6],
[17.8],
[17.3],
[14.3],
[ 9.1],
[12. ],
[10. ],
[12.6],
[17. ],
[13.8],
[18.4],
[14.7],
[20.1],
[22.1],
[13.8],
[18.4],
[13.8],
[18.4],
[ 7.9],
[ 8.1],
[ 8.7],
```

```
[ 9.1],
[ 9.6],
[10.9],
[ 4.8],
[ 7. ],
[ 7.2],
[10.7],
[ 9.1],
[ 6.9],
[ 7.5],
[10.6],
[11.5],
[12. ],
[13.4],
[ 7.7],
[ 7.7],
[ 8.5],
[13.2],
[11.1],
[11.2],
[11.6],
[10.3],
[11.5],
[11.1],
[ 9.6],
[ 6.2],
[ 6.3],
[ 9.3],
[10.1],
[10. ],
[10.6],
[ 7.7],
[ 8. ],
[ 8.8],
[ 8.5],
[10.2],
[ 7.5],
[ 8.2],
[11.5],
[11.7],
[12.3],
[11.1],
[11.6],
[11. ],
[12.3],
[11.6],
[12.8],
[ 7.6],
[11.8],
[10.6],
```

```
[12.8],
[14.5],
[15. ],
[10.3],
[10.6],
[12.8],
[13.3],
[18.1],
[13.3],
[12.4],
[17.5],
[13.3],
[13.3],
[13.1],
[13.7],
[13.5],
[13.7],
[13.5],
[ 9.5],
[10. ],
[10.8],
[10.1],
[10.9],
[11.6],
[ 9.8],
[ 9.1],
[ 9.8],
[ 9.4],
[10.2],
[ 9.4],
[11.1],
[14.8],
[ 9.9],
[12.1],
[15.8],
[ 9.8],
[ 9.4],
[ 9.4],
[11.1],
[12.8],
[13.4],
[10.9],
[ 8. ],
[ 8.4],
[ 8.4],
[ 9.7],
[ 9.8],
[ 8.4],
[ 8.5],
[ 9.4],
```

```
[ 9.6],
[ 8.4],
[ 8.6],
[ 8.7],
[ 7.6],
[ 7.7],
[ 7.6],
[ 9.2],
[ 9.3],
[12. ],
[10.3],
[11.4],
[11. ],
[11.8],
[ 9. ],
[ 9.1],
[10. ],
[15.7],
[17.4],
[11.9],
[14.5],
[ 9.8],
[ 9.8],
[15.4],
[21.4],
[12.4],
[17.4],
[15. ],
[20.7],
[12.3],
[12.3],
[17.3],
[15.4],
[15.1],
[ 5.6],
[ 6. ],
[ 9.6],
[10.4],
[11. ],
[ 7.6],
[13.9],
[ 9.7],
[10.4],
[10.6],
[11. ],
[10.6],
[12.6],
[12.3],
[12.9],
[11.4],
```

```
[16.5],
[11.1],
[ 8.2],
[11.8],
[11.3],
[12.8],
[10.2],
[12.2],
[ 9. ],
[11.4],
[ 6.3],
[16. ],
[21.6],
[12.7],
[12.7],
[15.1],
[14.5],
[13.1],
[ 8.2],
[ 8. ],
[ 8.6],
[ 8.5],
[ 8.9],
[12.3],
[12.6],
[ 7.7],
[ 7.4],
[ 6.9],
[ 7. ],
[ 7.4],
[ 7.1],
[ 7.2],
[ 7.7],
[ 7.3],
[ 7.2],
[ 9.7],
[ 9.8],
[ 7.6],
[ 7.3],
[10.2],
[ 9.6],
[ 9.9],
[ 8. ],
[ 9.4],
[ 9.4],
[10.5],
[14. ],
[10.4],
[10.6],
[10.5],
```

```
[15.1],
[15.3],
[13.2],
[13.4],
[ 7.8],
[ 9.2],
[11.5],
[13.1],
[ 7.2],
[10.2],
[10. ],
[10.4],
[10.5],
[10.2],
[ 8.8],
[11.6],
[11.7],
[11.4],
[12.9],
[13.5],
[13.5],
[18.2],
[17.7],
[12. ],
[15.6],
[11.3],
[16. ],
[10.3],
[14.2],
[15.5],
[12.1],
[12. ],
[14.2],
[ 8.9],
[ 9. ],
[ 9.8],
[10.7],
[16.5],
[15.7],
[15.7],
[ 7.8],
[ 7.8],
[ 7.8],
[ 7.8],
[ 8.7],
[ 7.8],
[ 7.6],
[ 8.7],
[ 7.8],
[ 7.8],
```

```
[ 7.8],
[ 8. ],
[ 8.3],
[ 9. ],
[ 8.5],
[ 8.3],
[ 8. ],
[ 9. ],
[ 8.5],
[ 8.3],
[ 8. ],
[ 8.3],
[ 8. ],
[ 8.3],
[ 8. ],
[ 9. ],
[ 8.5],
[ 8.3],
[ 9. ],
[ 8.5],
[ 8.3],
[ 8. ],
[ 8.2],
[ 8.3],
[12.5],
[12.3],
[11.6],
[ 8.6],
[ 8.6],
[ 6. ],
[ 6.4],
[ 8.6],
[10.2],
[ 8.7],
[ 8.8],
[ 9. ],
[11. ],
[11.5],
[11.4],
[ 7.6],
[ 9.3],
[16.5],
[12.3],
[12.7],
[13.7],
[13.1],
[ 8.1],
[ 8.6],
[10.9],
[10.6],
```

```
[11. ],
[ 8.8],
[11.1],
[ 7. ],
[ 7.8],
[15.6],
[17.3],
[ 6.6],
[ 7.7],
[14.2],
[13.7],
[10.1],
[10.3],
[10.7],
[10.5],
[11. ],
[10.8],
[11. ],
[11. ],
[11.2],
[10.1],
[10.7],
[10.6],
[10.8],
[10.7],
[10.9],
[14. ],
[11.9],
[11.9],
[11.9],
[11.9],
[ 9.2],
[ 9.9],
[ 9.8],
[13.6],
[10.4],
[13.9],
[13. ],
[10.9],
[13.7],
[ 9.2],
[ 9.8],
[10.4],
[10.9],
[11.1],
[11.3],
[11.5],
[12.7],
[11.3],
[12.8],
```

```
[14.8],
[13.6],
[15.4],
[12.6],
[17.8],
[10.2],
[11.9],
[11.3],
[11.4],
[15.9],
[16. ],
[17.1],
[17.1],
[15.6],
[ 8.4],
[ 9. ],
[ 8. ],
[ 7.9],
[ 6.4],
[ 6.4],
[16.9],
[16.9],
[ 9.6],
[ 9.7],
[ 7.8],
[ 8.5],
[11.2],
[ 7.8],
[ 8.5],
[11.2],
[12.5],
[ 8.6],
[ 9.9],
[11.6],
[ 9. ],
[13.1],
[ 8.5],
[ 9.2],
[ 7.7],
[12.9],
[12.9],
[ 9.6],
[ 5.7],
[ 6. ],
[ 7.7],
[ 7.3],
[ 7.1],
[ 7.4],
[ 6.8],
[ 6.9],
```

```
[ 8.5],
[ 8.2],
[12.9],
[13.9],
[10.3],
[11.2],
[11.5],
[ 8.3],
[ 8.2],
[ 4.8],
[ 4.7],
[ 5.5],
[ 9.4],
[ 9.5],
[16.7],
[11.3],
[12.4],
[10.9],
[10.2],
[13.1],
[11.9],
[11.8],
[14. ],
[13.5],
[14.3],
[15.5],
[15.1],
[16.2],
[10. ],
[11. ],
[10.2],
[11.3],
[ 7.1],
[ 8.5],
[ 8.7],
[ 9.1],
[ 9.4],
[ 9.8],
[ 9.6],
[ 8.7],
[ 9.3],
[ 9.4],
[ 7.3],
[ 7.3],
[ 9.8],
[12.3],
[ 9.6],
[ 9.5],
[ 9.4],
[ 7.2],
```

```
       [ 8.2],
       [ 8.2],
       [ 9.4],
       [ 9. ],
       [ 9. ],
       [ 9.2],
       [ 6.9],
       [ 5.4],
       [ 8.6],
       [ 8.6],
       [ 7.1],
       [ 6.8],
       [12.1],
       [10.7],
       [10.7],
       [12.2],
       [10.4],
       [ 9.7],
       [10.1],
       [11.5],
       [10.2],
       [11.5],
       [11.2],
       [11.8],
       [11.5],
       [11.8],
       [11.3],
       [12.8]])
```

```
np.asanyarray(train[['CO2EMISSIONS']])
```

```
array([[196],
       [221],
       [136],
       [255],
       [244],
       [232],
       [267],
       [225],
       [239],
       [359],
       [338],
       [354],
       [338],
       [354],
       [359],
       [230],
       [214],
       [230],
       [214],
       [251],
```

```
[224],
[258],
[224],
[260],
[227],
[258],
[288],
[230],
[242],
[239],
[258],
[294],
[336],
[407],
[354],
[336],
[407],
[354],
[306],
[308],
[262],
[285],
[262],
[267],
[281],
[281],
[292],
[209],
[209],
[297],
[356],
[320],
[380],
[322],
[380],
[380],
[437],
[193],
[200],
[181],
[181],
[193],
[200],
[209],
[209],
[209],
[230],
[228],
[237],
[232],
```

```
[193],
[200],
[209],
[221],
[230],
[237],
[202],
[209],
[213],
[232],
[292],
[232],
[292],
[281],
[292],
[292],
[191],
[292],
[338],
[338],
[317],
[338],
[317],
[209],
[260],
[230],
[253],
[320],
[370],
[274],
[209],
[200],
[278],
[253],
[278],
[283],
[290],
[196],
[212],
[264],
[250],
[264],
[264],
[225],
[246],
[189],
[225],
[218],
[221],
[235],
```

```
[207],
[251],
[232],
[258],
[262],
[232],
[248],
[248],
[260],
[253],
[260],
[255],
[382],
[340],
[382],
[340],
[338],
[322],
[377],
[283],
[294],
[271],
[290],
[258],
[269],
[294],
[382],
[340],
[276],
[258],
[179],
[189],
[192],
[177],
[163],
[209],
[259],
[290],
[272],
[366],
[334],
[380],
[344],
[380],
[344],
[362],
[380],
[344],
[380],
[362],
```

```
[414],
[432],
[411],
[419],
[435],
[229],
[200],
[228],
[184],
[244],
[271],
[269],
[292],
[270],
[324],
[290],
[299],
[329],
[184],
[173],
[184],
[172],
[182],
[191],
[182],
[191],
[182],
[159],
[317],
[294],
[317],
[294],
[317],
[294],
[317],
[294],
[290],
[207],
[232],
[246],
[230],
[230],
[230],
[294],
[251],
[251],
[237],
[222],
[276],
[269],
```

[225],
[230],
[301],
[294],
[320],
[317],
[258],
[237],
[294],
[251],
[308],
[251],
[237],
[235],
[222],
[186],
[198],
[202],
[195],
[202],
[168],
[168],
[278],
[250],
[271],
[264],
[255],
[285],
[276],
[179],
[179],
[182],
[179],
[156],
[200],
[193],
[373],
[350],
[396],
[360],
[386],
[368],
[488],
[225],
[264],
[281],
[209],
[216],
[214],
[230],

```
[368],
[346],
[274],
[254],
[288],
[267],
[294],
[366],
[317],
[308],
[283],
[283],
[264],
[336],
[336],
[315],
[308],
[288],
[308],
[288],
[320],
[318],
[152],
[184],
[264],
[278],
[294],
[205],
[177],
[165],
[179],
[165],
[184],
[170],
[166],
[162],
[196],
[186],
[205],
[186],
[216],
[126],
[244],
[264],
[244],
[209],
[271],
[237],
[229],
[258],
```

[251],
[218],
[228],
[232],
[239],
[283],
[297],
[347],
[366],
[334],
[344],
[380],
[344],
[380],
[380],
[344],
[414],
[432],
[411],
[419],
[435],
[271],
[269],
[292],
[270],
[324],
[290],
[285],
[277],
[329],
[209],
[276],
[230],
[290],
[272],
[317],
[294],
[338],
[322],
[354],
[317],
[294],
[317],
[294],
[182],
[186],
[200],
[209],
[221],
[251],

```
[110],
[161],
[166],
[246],
[209],
[159],
[172],
[244],
[265],
[276],
[308],
[177],
[177],
[196],
[304],
[255],
[258],
[267],
[237],
[264],
[255],
[221],
[143],
[145],
[214],
[232],
[230],
[244],
[177],
[184],
[202],
[196],
[235],
[172],
[189],
[265],
[269],
[283],
[255],
[267],
[253],
[283],
[267],
[294],
[175],
[271],
[244],
[294],
[334],
[345],
```

[237],
[244],
[294],
[306],
[290],
[306],
[285],
[280],
[306],
[306],
[301],
[315],
[310],
[315],
[310],
[218],
[230],
[248],
[232],
[251],
[267],
[225],
[209],
[225],
[216],
[235],
[216],
[255],
[340],
[267],
[278],
[363],
[225],
[216],
[216],
[255],
[294],
[308],
[251],
[184],
[193],
[193],
[223],
[225],
[193],
[196],
[216],
[221],
[193],
[198],

```
[200],
[175],
[177],
[175],
[212],
[214],
[276],
[237],
[262],
[253],
[271],
[207],
[209],
[230],
[361],
[400],
[274],
[334],
[225],
[225],
[354],
[342],
[285],
[278],
[345],
[331],
[283],
[283],
[277],
[354],
[347],
[129],
[138],
[221],
[239],
[253],
[175],
[320],
[223],
[239],
[244],
[253],
[244],
[290],
[283],
[297],
[262],
[380],
[255],
[189],
```

[271],
[260],
[294],
[235],
[281],
[207],
[262],
[145],
[368],
[346],
[292],
[292],
[347],
[334],
[301],
[189],
[184],
[198],
[196],
[205],
[283],
[290],
[177],
[170],
[159],
[161],
[170],
[163],
[166],
[177],
[168],
[166],
[223],
[225],
[175],
[168],
[235],
[221],
[228],
[184],
[216],
[216],
[242],
[224],
[239],
[244],
[242],
[347],
[352],
[304],

[308],
[179],
[212],
[264],
[301],
[194],
[235],
[230],
[239],
[242],
[235],
[202],
[267],
[269],
[262],
[297],
[310],
[310],
[419],
[407],
[324],
[359],
[260],
[256],
[278],
[327],
[356],
[278],
[276],
[327],
[205],
[207],
[225],
[246],
[380],
[361],
[361],
[179],
[179],
[179],
[179],
[200],
[179],
[175],
[200],
[179],
[179],
[179],
[184],
[191],

```
[207],
[196],
[191],
[184],
[207],
[196],
[191],
[184],
[191],
[184],
[191],
[184],
[207],
[196],
[191],
[207],
[196],
[191],
[184],
[189],
[191],
[288],
[283],
[267],
[198],
[198],
[138],
[147],
[198],
[235],
[200],
[202],
[207],
[253],
[264],
[262],
[175],
[214],
[380],
[283],
[292],
[315],
[301],
[186],
[198],
[251],
[244],
[253],
[202],
[255],
```

```
[161],
[179],
[359],
[398],
[152],
[177],
[327],
[315],
[232],
[237],
[246],
[242],
[253],
[248],
[253],
[253],
[258],
[232],
[246],
[244],
[248],
[246],
[251],
[322],
[274],
[274],
[274],
[274],
[212],
[228],
[225],
[313],
[281],
[320],
[299],
[251],
[315],
[212],
[225],
[239],
[251],
[255],
[260],
[264],
[292],
[260],
[294],
[340],
[313],
[354],
```

[290],
[285],
[275],
[274],
[260],
[262],
[254],
[368],
[393],
[393],
[359],
[193],
[207],
[184],
[182],
[147],
[147],
[389],
[389],
[221],
[223],
[179],
[196],
[258],
[179],
[196],
[258],
[288],
[198],
[228],
[267],
[207],
[301],
[196],
[212],
[177],
[297],
[297],
[221],
[131],
[138],
[177],
[168],
[163],
[170],
[156],
[159],
[196],
[189],
[297],

```
[320],
[237],
[258],
[264],
[191],
[189],
[110],
[108],
[126],
[216],
[218],
[384],
[260],
[285],
[251],
[235],
[301],
[274],
[271],
[322],
[310],
[329],
[356],
[347],
[373],
[230],
[253],
[235],
[260],
[163],
[196],
[200],
[209],
[216],
[225],
[221],
[200],
[214],
[216],
[197],
[197],
[225],
[283],
[221],
[218],
[216],
[194],
[189],
[189],
[216],
```

```
       [207],
       [207],
       [212],
       [186],
       [124],
       [198],
       [198],
       [192],
       [184],
       [278],
       [246],
       [246],
       [281],
       [281],
       [223],
       [232],
       [264],
       [235],
       [264],
       [258],
       [271],
       [264],
       [271],
       [260],
       [294]], dtype=int64)
```

```python
X = np.asanyarray(train[['FUELCONSUMPTION_COMB']]) # Independant
Variable

X
```

```
array([[ 8.5],
       [ 9.6],
       [ 5.9],
       [11.1],
       [10.6],
       [10.1],
       [11.6],
       [ 9.8],
       [10.4],
       [15.6],
       [14.7],
       [15.4],
       [14.7],
       [15.4],
       [15.6],
       [10. ],
       [ 9.3],
       [10. ],
       [ 9.3],
       [10.9],
```

```
[ 8.3],
[11.2],
[ 8.3],
[11.3],
[ 8.4],
[11.2],
[12.5],
[ 8.5],
[10.5],
[10.4],
[11.2],
[10.9],
[14.6],
[17.7],
[15.4],
[14.6],
[17.7],
[15.4],
[13.3],
[13.4],
[11.4],
[12.4],
[11.4],
[11.6],
[12.2],
[12.2],
[12.7],
[ 9.1],
[ 9.1],
[12.9],
[15.5],
[13.9],
[16.5],
[14. ],
[16.5],
[16.5],
[19. ],
[ 8.4],
[ 8.7],
[ 6.7],
[ 6.7],
[ 8.4],
[ 8.7],
[ 9.1],
[ 9.1],
[ 9.1],
[10. ],
[ 9.9],
[10.3],
[10.1],
```

```
[ 8.4],
[ 8.7],
[ 9.1],
[ 9.6],
[10. ],
[10.3],
[ 8.8],
[ 9.1],
[ 7.9],
[10.1],
[12.7],
[10.1],
[12.7],
[12.2],
[12.7],
[12.7],
[ 8.3],
[12.7],
[14.7],
[14.7],
[13.8],
[14.7],
[13.8],
[ 9.1],
[11.3],
[10. ],
[11. ],
[13.9],
[16.1],
[11.9],
[ 9.1],
[ 8.7],
[12.1],
[11. ],
[12.1],
[12.3],
[12.6],
[ 8.5],
[ 9.2],
[11.5],
[15.6],
[11.5],
[16.5],
[ 9.8],
[10.7],
[ 8.2],
[ 9.8],
[ 9.5],
[ 9.6],
[10.2],
```

```
[ 9. ],
[10.9],
[10.1],
[11.2],
[11.4],
[10.1],
[10.8],
[10.8],
[11.3],
[11. ],
[11.3],
[11.1],
[16.6],
[14.8],
[16.6],
[14.8],
[14.7],
[20.1],
[16.4],
[12.3],
[12.8],
[11.8],
[12.6],
[11.2],
[11.7],
[12.8],
[16.6],
[14.8],
[12. ],
[11.2],
[ 7.8],
[ 8.2],
[ 7.1],
[ 7.7],
[ 7.1],
[ 9.1],
[16.2],
[12.6],
[17. ],
[15.9],
[20.9],
[16.5],
[21.5],
[16.5],
[21.5],
[22.6],
[16.5],
[21.5],
[16.5],
[22.6],
```

```
[18. ],
[18.8],
[25.7],
[18.2],
[18.9],
[14.3],
[ 8.7],
[ 9.9],
[ 8. ],
[10.6],
[11.8],
[16.8],
[12.7],
[16.9],
[14.1],
[12.6],
[13. ],
[14.3],
[ 8. ],
[ 7.5],
[ 8. ],
[ 7.5],
[ 7.9],
[ 8.3],
[ 7.9],
[ 8.3],
[ 7.9],
[ 6.9],
[13.8],
[18.4],
[13.8],
[18.4],
[13.8],
[18.4],
[13.8],
[18.4],
[12.6],
[ 9. ],
[10.1],
[10.7],
[14.4],
[10. ],
[14.4],
[12.8],
[10.9],
[10.9],
[14.8],
[13.9],
[12. ],
[16.8],
```

```
[ 9.8],
[10. ],
[13.1],
[12.8],
[13.9],
[13.8],
[11.2],
[10.3],
[12.8],
[10.9],
[13.4],
[10.9],
[14.8],
[10.2],
[13.9],
[ 8.1],
[ 8.6],
[ 8.8],
[12.2],
[ 8.8],
[ 7.3],
[ 7.3],
[12.1],
[15.6],
[11.8],
[16.5],
[11.1],
[12.4],
[12. ],
[ 7.8],
[ 7.8],
[ 7.9],
[ 7.8],
[ 6.8],
[ 8.7],
[ 8.4],
[16.2],
[21.9],
[17.2],
[22.5],
[16.8],
[23. ],
[21.2],
[ 9.8],
[11.5],
[12.2],
[ 9.1],
[ 9.4],
[ 9.3],
[10. ],
```

```
[16. ],
[21.6],
[11.9],
[15.9],
[12.5],
[16.7],
[12.8],
[15.9],
[13.8],
[13.4],
[12.3],
[12.3],
[16.5],
[14.6],
[14.6],
[19.7],
[13.4],
[18. ],
[13.4],
[18. ],
[20. ],
[19.9],
[ 6.6],
[ 8. ],
[11.5],
[12.1],
[12.8],
[ 8.9],
[ 7.7],
[10.3],
[ 7.8],
[10.3],
[ 8. ],
[10.6],
[ 7.2],
[10.1],
[ 8.5],
[ 8.1],
[ 8.9],
[ 8.1],
[ 9.4],
[ 5.5],
[10.6],
[11.5],
[10.6],
[ 9.1],
[11.8],
[10.3],
[14.3],
[11.2],
```

```
[15.7],
[ 9.5],
[ 9.9],
[10.1],
[10.4],
[12.3],
[12.9],
[15.1],
[15.9],
[20.9],
[21.5],
[16.5],
[21.5],
[16.5],
[16.5],
[21.5],
[18. ],
[18.8],
[25.7],
[18.2],
[18.9],
[11.8],
[16.8],
[12.7],
[16.9],
[14.1],
[12.6],
[17.8],
[17.3],
[14.3],
[ 9.1],
[12. ],
[10. ],
[12.6],
[17. ],
[13.8],
[18.4],
[14.7],
[20.1],
[22.1],
[13.8],
[18.4],
[13.8],
[18.4],
[ 7.9],
[ 8.1],
[ 8.7],
[ 9.1],
[ 9.6],
[10.9],
```

```
[ 4.8],
[ 7. ],
[ 7.2],
[10.7],
[ 9.1],
[ 6.9],
[ 7.5],
[10.6],
[11.5],
[12. ],
[13.4],
[ 7.7],
[ 7.7],
[ 8.5],
[13.2],
[11.1],
[11.2],
[11.6],
[10.3],
[11.5],
[11.1],
[ 9.6],
[ 6.2],
[ 6.3],
[ 9.3],
[10.1],
[10. ],
[10.6],
[ 7.7],
[ 8. ],
[ 8.8],
[ 8.5],
[10.2],
[ 7.5],
[ 8.2],
[11.5],
[11.7],
[12.3],
[11.1],
[11.6],
[11. ],
[12.3],
[11.6],
[12.8],
[ 7.6],
[11.8],
[10.6],
[12.8],
[14.5],
[15. ],
```

```
[10.3],
[10.6],
[12.8],
[13.3],
[18.1],
[13.3],
[12.4],
[17.5],
[13.3],
[13.3],
[13.1],
[13.7],
[13.5],
[13.7],
[13.5],
[ 9.5],
[10. ],
[10.8],
[10.1],
[10.9],
[11.6],
[ 9.8],
[ 9.1],
[ 9.8],
[ 9.4],
[10.2],
[ 9.4],
[11.1],
[14.8],
[ 9.9],
[12.1],
[15.8],
[ 9.8],
[ 9.4],
[ 9.4],
[11.1],
[12.8],
[13.4],
[10.9],
[ 8. ],
[ 8.4],
[ 8.4],
[ 9.7],
[ 9.8],
[ 8.4],
[ 8.5],
[ 9.4],
[ 9.6],
[ 8.4],
[ 8.6],
```

```
[ 8.7],
[ 7.6],
[ 7.7],
[ 7.6],
[ 9.2],
[ 9.3],
[12. ],
[10.3],
[11.4],
[11. ],
[11.8],
[ 9. ],
[ 9.1],
[10. ],
[15.7],
[17.4],
[11.9],
[14.5],
[ 9.8],
[ 9.8],
[15.4],
[21.4],
[12.4],
[17.4],
[15. ],
[20.7],
[12.3],
[12.3],
[17.3],
[15.4],
[15.1],
[ 5.6],
[ 6. ],
[ 9.6],
[10.4],
[11. ],
[ 7.6],
[13.9],
[ 9.7],
[10.4],
[10.6],
[11. ],
[10.6],
[12.6],
[12.3],
[12.9],
[11.4],
[16.5],
[11.1],
[ 8.2],
```

```
[11.8],
[11.3],
[12.8],
[10.2],
[12.2],
[ 9. ],
[11.4],
[ 6.3],
[16. ],
[21.6],
[12.7],
[12.7],
[15.1],
[14.5],
[13.1],
[ 8.2],
[ 8. ],
[ 8.6],
[ 8.5],
[ 8.9],
[12.3],
[12.6],
[ 7.7],
[ 7.4],
[ 6.9],
[ 7. ],
[ 7.4],
[ 7.1],
[ 7.2],
[ 7.7],
[ 7.3],
[ 7.2],
[ 9.7],
[ 9.8],
[ 7.6],
[ 7.3],
[10.2],
[ 9.6],
[ 9.9],
[ 8. ],
[ 9.4],
[ 9.4],
[10.5],
[14. ],
[10.4],
[10.6],
[10.5],
[15.1],
[15.3],
[13.2],
```

```
[13.4],
[ 7.8],
[ 9.2],
[11.5],
[13.1],
[ 7.2],
[10.2],
[10. ],
[10.4],
[10.5],
[10.2],
[ 8.8],
[11.6],
[11.7],
[11.4],
[12.9],
[13.5],
[13.5],
[18.2],
[17.7],
[12. ],
[15.6],
[11.3],
[16. ],
[10.3],
[14.2],
[15.5],
[12.1],
[12. ],
[14.2],
[ 8.9],
[ 9. ],
[ 9.8],
[10.7],
[16.5],
[15.7],
[15.7],
[ 7.8],
[ 7.8],
[ 7.8],
[ 7.8],
[ 8.7],
[ 7.8],
[ 7.6],
[ 8.7],
[ 7.8],
[ 7.8],
[ 7.8],
[ 8. ],
[ 8.3],
```

```
[ 9. ],
[ 8.5],
[ 8.3],
[ 8. ],
[ 9. ],
[ 8.5],
[ 8.3],
[ 8. ],
[ 8.3],
[ 8. ],
[ 8.3],
[ 8. ],
[ 9. ],
[ 8.5],
[ 8.3],
[ 9. ],
[ 8.5],
[ 8.3],
[ 8. ],
[ 8.2],
[ 8.3],
[12.5],
[12.3],
[11.6],
[ 8.6],
[ 8.6],
[ 6. ],
[ 6.4],
[ 8.6],
[10.2],
[ 8.7],
[ 8.8],
[ 9. ],
[11. ],
[11.5],
[11.4],
[ 7.6],
[ 9.3],
[16.5],
[12.3],
[12.7],
[13.7],
[13.1],
[ 8.1],
[ 8.6],
[10.9],
[10.6],
[11. ],
[ 8.8],
[11.1],
```

```
[ 7. ],
[ 7.8],
[15.6],
[17.3],
[ 6.6],
[ 7.7],
[14.2],
[13.7],
[10.1],
[10.3],
[10.7],
[10.5],
[11. ],
[10.8],
[11. ],
[11. ],
[11.2],
[10.1],
[10.7],
[10.6],
[10.8],
[10.7],
[10.9],
[14. ],
[11.9],
[11.9],
[11.9],
[11.9],
[ 9.2],
[ 9.9],
[ 9.8],
[13.6],
[10.4],
[13.9],
[13. ],
[10.9],
[13.7],
[ 9.2],
[ 9.8],
[10.4],
[10.9],
[11.1],
[11.3],
[11.5],
[12.7],
[11.3],
[12.8],
[14.8],
[13.6],
[15.4],
```

```
[12.6],
[17.8],
[10.2],
[11.9],
[11.3],
[11.4],
[15.9],
[16. ],
[17.1],
[17.1],
[15.6],
[ 8.4],
[ 9. ],
[ 8. ],
[ 7.9],
[ 6.4],
[ 6.4],
[16.9],
[16.9],
[ 9.6],
[ 9.7],
[ 7.8],
[ 8.5],
[11.2],
[ 7.8],
[ 8.5],
[11.2],
[12.5],
[ 8.6],
[ 9.9],
[11.6],
[ 9. ],
[13.1],
[ 8.5],
[ 9.2],
[ 7.7],
[12.9],
[12.9],
[ 9.6],
[ 5.7],
[ 6. ],
[ 7.7],
[ 7.3],
[ 7.1],
[ 7.4],
[ 6.8],
[ 6.9],
[ 8.5],
[ 8.2],
[12.9],
```

```
[13.9],
[10.3],
[11.2],
[11.5],
[ 8.3],
[ 8.2],
[ 4.8],
[ 4.7],
[ 5.5],
[ 9.4],
[ 9.5],
[16.7],
[11.3],
[12.4],
[10.9],
[10.2],
[13.1],
[11.9],
[11.8],
[14. ],
[13.5],
[14.3],
[15.5],
[15.1],
[16.2],
[10. ],
[11. ],
[10.2],
[11.3],
[ 7.1],
[ 8.5],
[ 8.7],
[ 9.1],
[ 9.4],
[ 9.8],
[ 9.6],
[ 8.7],
[ 9.3],
[ 9.4],
[ 7.3],
[ 7.3],
[ 9.8],
[12.3],
[ 9.6],
[ 9.5],
[ 9.4],
[ 7.2],
[ 8.2],
[ 8.2],
[ 9.4],
```

```
       [ 9. ],
       [ 9. ],
       [ 9.2],
       [ 6.9],
       [ 5.4],
       [ 8.6],
       [ 8.6],
       [ 7.1],
       [ 6.8],
       [12.1],
       [10.7],
       [10.7],
       [12.2],
       [10.4],
       [ 9.7],
       [10.1],
       [11.5],
       [10.2],
       [11.5],
       [11.2],
       [11.8],
       [11.5],
       [11.8],
       [11.3],
       [12.8]])
```

```
y = np.asanyarray(train[['CO2EMISSIONS']]) # Dependent Variable
```

```
y
```

```
array([[196],
       [221],
       [136],
       [255],
       [244],
       [232],
       [267],
       [225],
       [239],
       [359],
       [338],
       [354],
       [338],
       [354],
       [359],
       [230],
       [214],
       [230],
       [214],
       [251],
       [224],
```

[258],
[224],
[260],
[227],
[258],
[288],
[230],
[242],
[239],
[258],
[294],
[336],
[407],
[354],
[336],
[407],
[354],
[306],
[308],
[262],
[285],
[262],
[267],
[281],
[281],
[292],
[209],
[209],
[297],
[356],
[320],
[380],
[322],
[380],
[380],
[437],
[193],
[200],
[181],
[181],
[193],
[200],
[209],
[209],
[209],
[230],
[228],
[237],
[232],
[193],

```
[200],
[209],
[221],
[230],
[237],
[202],
[209],
[213],
[232],
[292],
[232],
[292],
[281],
[292],
[292],
[191],
[292],
[338],
[338],
[317],
[338],
[317],
[209],
[260],
[230],
[253],
[320],
[370],
[274],
[209],
[200],
[278],
[253],
[278],
[283],
[290],
[196],
[212],
[264],
[250],
[264],
[264],
[225],
[246],
[189],
[225],
[218],
[221],
[235],
[207],
```

[251],
[232],
[258],
[262],
[232],
[248],
[248],
[260],
[253],
[260],
[255],
[382],
[340],
[382],
[340],
[338],
[322],
[377],
[283],
[294],
[271],
[290],
[258],
[269],
[294],
[382],
[340],
[276],
[258],
[179],
[189],
[192],
[177],
[163],
[209],
[259],
[290],
[272],
[366],
[334],
[380],
[344],
[380],
[344],
[362],
[380],
[344],
[380],
[362],
[414],

```
[432],
[411],
[419],
[435],
[229],
[200],
[228],
[184],
[244],
[271],
[269],
[292],
[270],
[324],
[290],
[299],
[329],
[184],
[173],
[184],
[172],
[182],
[191],
[182],
[191],
[182],
[159],
[317],
[294],
[317],
[294],
[317],
[294],
[317],
[294],
[290],
[207],
[232],
[246],
[230],
[230],
[230],
[294],
[251],
[251],
[237],
[222],
[276],
[269],
[225],
```

```
[230],
[301],
[294],
[320],
[317],
[258],
[237],
[294],
[251],
[308],
[251],
[237],
[235],
[222],
[186],
[198],
[202],
[195],
[202],
[168],
[168],
[278],
[250],
[271],
[264],
[255],
[285],
[276],
[179],
[179],
[182],
[179],
[156],
[200],
[193],
[373],
[350],
[396],
[360],
[386],
[368],
[488],
[225],
[264],
[281],
[209],
[216],
[214],
[230],
[368],
```

[346],
[274],
[254],
[288],
[267],
[294],
[366],
[317],
[308],
[283],
[283],
[264],
[336],
[336],
[315],
[308],
[288],
[308],
[288],
[320],
[318],
[152],
[184],
[264],
[278],
[294],
[205],
[177],
[165],
[179],
[165],
[184],
[170],
[166],
[162],
[196],
[186],
[205],
[186],
[216],
[126],
[244],
[264],
[244],
[209],
[271],
[237],
[229],
[258],
[251],

[218],
[228],
[232],
[239],
[283],
[297],
[347],
[366],
[334],
[344],
[380],
[344],
[380],
[380],
[344],
[414],
[432],
[411],
[419],
[435],
[271],
[269],
[292],
[270],
[324],
[290],
[285],
[277],
[329],
[209],
[276],
[230],
[290],
[272],
[317],
[294],
[338],
[322],
[354],
[317],
[294],
[317],
[294],
[182],
[186],
[200],
[209],
[221],
[251],
[110],

```
[161],
[166],
[246],
[209],
[159],
[172],
[244],
[265],
[276],
[308],
[177],
[177],
[196],
[304],
[255],
[258],
[267],
[237],
[264],
[255],
[221],
[143],
[145],
[214],
[232],
[230],
[244],
[177],
[184],
[202],
[196],
[235],
[172],
[189],
[265],
[269],
[283],
[255],
[267],
[253],
[283],
[267],
[294],
[175],
[271],
[244],
[294],
[334],
[345],
[237],
```

```
[244],
[294],
[306],
[290],
[306],
[285],
[280],
[306],
[306],
[301],
[315],
[310],
[315],
[310],
[218],
[230],
[248],
[232],
[251],
[267],
[225],
[209],
[225],
[216],
[235],
[216],
[255],
[340],
[267],
[278],
[363],
[225],
[216],
[216],
[255],
[294],
[308],
[251],
[184],
[193],
[193],
[223],
[225],
[193],
[196],
[216],
[221],
[193],
[198],
[200],
```

```
[175],
[177],
[175],
[212],
[214],
[276],
[237],
[262],
[253],
[271],
[207],
[209],
[230],
[361],
[400],
[274],
[334],
[225],
[225],
[354],
[342],
[285],
[278],
[345],
[331],
[283],
[283],
[277],
[354],
[347],
[129],
[138],
[221],
[239],
[253],
[175],
[320],
[223],
[239],
[244],
[253],
[244],
[290],
[283],
[297],
[262],
[380],
[255],
[189],
[271],
```

```
[260],
[294],
[235],
[281],
[207],
[262],
[145],
[368],
[346],
[292],
[292],
[347],
[334],
[301],
[189],
[184],
[198],
[196],
[205],
[283],
[290],
[177],
[170],
[159],
[161],
[170],
[163],
[166],
[177],
[168],
[166],
[223],
[225],
[175],
[168],
[235],
[221],
[228],
[184],
[216],
[216],
[242],
[224],
[239],
[244],
[242],
[347],
[352],
[304],
[308],
```

```
[179],
[212],
[264],
[301],
[194],
[235],
[230],
[239],
[242],
[235],
[202],
[267],
[269],
[262],
[297],
[310],
[310],
[419],
[407],
[324],
[359],
[260],
[256],
[278],
[327],
[356],
[278],
[276],
[327],
[205],
[207],
[225],
[246],
[380],
[361],
[361],
[179],
[179],
[179],
[179],
[200],
[179],
[175],
[200],
[179],
[179],
[179],
[184],
[191],
[207],
```

```
[196],
[191],
[184],
[207],
[196],
[191],
[184],
[191],
[184],
[191],
[184],
[207],
[196],
[191],
[207],
[196],
[191],
[184],
[189],
[191],
[288],
[283],
[267],
[198],
[198],
[138],
[147],
[198],
[235],
[200],
[202],
[207],
[253],
[264],
[262],
[175],
[214],
[380],
[283],
[292],
[315],
[301],
[186],
[198],
[251],
[244],
[253],
[202],
[255],
[161],
```

```
[179],
[359],
[398],
[152],
[177],
[327],
[315],
[232],
[237],
[246],
[242],
[253],
[248],
[253],
[253],
[258],
[232],
[246],
[244],
[248],
[246],
[251],
[322],
[274],
[274],
[274],
[274],
[212],
[228],
[225],
[313],
[281],
[320],
[299],
[251],
[315],
[212],
[225],
[239],
[251],
[255],
[260],
[264],
[292],
[260],
[294],
[340],
[313],
[354],
[290],
```

[285],
[275],
[274],
[260],
[262],
[254],
[368],
[393],
[393],
[359],
[193],
[207],
[184],
[182],
[147],
[147],
[389],
[389],
[221],
[223],
[179],
[196],
[258],
[179],
[196],
[258],
[288],
[198],
[228],
[267],
[207],
[301],
[196],
[212],
[177],
[297],
[297],
[221],
[131],
[138],
[177],
[168],
[163],
[170],
[156],
[159],
[196],
[189],
[297],
[320],

[237],
[258],
[264],
[191],
[189],
[110],
[108],
[126],
[216],
[218],
[384],
[260],
[285],
[251],
[235],
[301],
[274],
[271],
[322],
[310],
[329],
[356],
[347],
[373],
[230],
[253],
[235],
[260],
[163],
[196],
[200],
[209],
[216],
[225],
[221],
[200],
[214],
[216],
[197],
[197],
[225],
[283],
[221],
[218],
[216],
[194],
[189],
[189],
[216],
[207],

```
       [207],
       [212],
       [186],
       [124],
       [198],
       [198],
       [192],
       [184],
       [278],
       [246],
       [246],
       [281],
       [281],
       [223],
       [232],
       [264],
       [235],
       [264],
       [258],
       [271],
       [264],
       [271],
       [260],
       [294]], dtype=int64)
```

```
from sklearn import linear_model

# y = mx + c == y is dependent variable, x is the data(indep.
variable) m is COEFFICIENT  and c is INTERCEPT

regr = linear_model.LinearRegression()

regr
```

```
LinearRegression()
```

```
train.shape
```

```
(845, 4)
```

```
test.shape
```

```
(222, 4)
```

```
train_x = np.asanyarray(train[['FUELCONSUMPTION_COMB']]) #x represents
independent variables
train_y = np.asanyarray(train[['CO2EMISSIONS']]) #y represent
dependent variable

train_x # All Values of FUELCONSUMPTION_COMB column this will be used
for training
```

```
array([[ 8.5],
       [ 9.6],
       [ 5.9],
       [11.1],
       [10.6],
       [10.1],
       [11.6],
       [ 9.8],
       [10.4],
       [15.6],
       [14.7],
       [15.4],
       [14.7],
       [15.4],
       [15.6],
       [10. ],
       [ 9.3],
       [10. ],
       [ 9.3],
       [10.9],
       [ 8.3],
       [11.2],
       [ 8.3],
       [11.3],
       [ 8.4],
       [11.2],
       [12.5],
       [ 8.5],
       [10.5],
       [10.4],
       [11.2],
       [10.9],
       [14.6],
       [17.7],
       [15.4],
       [14.6],
       [17.7],
       [15.4],
       [13.3],
       [13.4],
       [11.4],
       [12.4],
       [11.4],
       [11.6],
       [12.2],
       [12.2],
       [12.7],
       [ 9.1],
       [ 9.1],
       [12.9],
```

```
[15.5],
[13.9],
[16.5],
[14. ],
[16.5],
[16.5],
[19. ],
[ 8.4],
[ 8.7],
[ 6.7],
[ 6.7],
[ 8.4],
[ 8.7],
[ 9.1],
[ 9.1],
[ 9.1],
[10. ],
[ 9.9],
[10.3],
[10.1],
[ 8.4],
[ 8.7],
[ 9.1],
[ 9.6],
[10. ],
[10.3],
[ 8.8],
[ 9.1],
[ 7.9],
[10.1],
[12.7],
[10.1],
[12.7],
[12.2],
[12.7],
[12.7],
[ 8.3],
[12.7],
[14.7],
[14.7],
[13.8],
[14.7],
[13.8],
[ 9.1],
[11.3],
[10. ],
[11. ],
[13.9],
[16.1],
[11.9],
```

```
[ 9.1],
[ 8.7],
[12.1],
[11. ],
[12.1],
[12.3],
[12.6],
[ 8.5],
[ 9.2],
[11.5],
[15.6],
[11.5],
[16.5],
[ 9.8],
[10.7],
[ 8.2],
[ 9.8],
[ 9.5],
[ 9.6],
[10.2],
[ 9. ],
[10.9],
[10.1],
[11.2],
[11.4],
[10.1],
[10.8],
[10.8],
[11.3],
[11. ],
[11.3],
[11.1],
[16.6],
[14.8],
[16.6],
[14.8],
[14.7],
[20.1],
[16.4],
[12.3],
[12.8],
[11.8],
[12.6],
[11.2],
[11.7],
[12.8],
[16.6],
[14.8],
[12. ],
[11.2],
```

```
[ 7.8],
[ 8.2],
[ 7.1],
[ 7.7],
[ 7.1],
[ 9.1],
[16.2],
[12.6],
[17. ],
[15.9],
[20.9],
[16.5],
[21.5],
[16.5],
[21.5],
[22.6],
[16.5],
[21.5],
[16.5],
[22.6],
[18. ],
[18.8],
[25.7],
[18.2],
[18.9],
[14.3],
[ 8.7],
[ 9.9],
[ 8. ],
[10.6],
[11.8],
[16.8],
[12.7],
[16.9],
[14.1],
[12.6],
[13. ],
[14.3],
[ 8. ],
[ 7.5],
[ 8. ],
[ 7.5],
[ 7.9],
[ 8.3],
[ 7.9],
[ 8.3],
[ 7.9],
[ 6.9],
[13.8],
[18.4],
```

```
[13.8],
[18.4],
[13.8],
[18.4],
[13.8],
[18.4],
[12.6],
[ 9. ],
[10.1],
[10.7],
[14.4],
[10. ],
[14.4],
[12.8],
[10.9],
[10.9],
[14.8],
[13.9],
[12. ],
[16.8],
[ 9.8],
[10. ],
[13.1],
[12.8],
[13.9],
[13.8],
[11.2],
[10.3],
[12.8],
[10.9],
[13.4],
[10.9],
[14.8],
[10.2],
[13.9],
[ 8.1],
[ 8.6],
[ 8.8],
[12.2],
[ 8.8],
[ 7.3],
[ 7.3],
[12.1],
[15.6],
[11.8],
[16.5],
[11.1],
[12.4],
[12. ],
[ 7.8],
```

```
[ 7.8],
[ 7.9],
[ 7.8],
[ 6.8],
[ 8.7],
[ 8.4],
[16.2],
[21.9],
[17.2],
[22.5],
[16.8],
[23. ],
[21.2],
[ 9.8],
[11.5],
[12.2],
[ 9.1],
[ 9.4],
[ 9.3],
[10. ],
[16. ],
[21.6],
[11.9],
[15.9],
[12.5],
[16.7],
[12.8],
[15.9],
[13.8],
[13.4],
[12.3],
[12.3],
[16.5],
[14.6],
[14.6],
[19.7],
[13.4],
[18. ],
[13.4],
[18. ],
[20. ],
[19.9],
[ 6.6],
[ 8. ],
[11.5],
[12.1],
[12.8],
[ 8.9],
[ 7.7],
[10.3],
```

```
[ 7.8],
[10.3],
[ 8. ],
[10.6],
[ 7.2],
[10.1],
[ 8.5],
[ 8.1],
[ 8.9],
[ 8.1],
[ 9.4],
[ 5.5],
[10.6],
[11.5],
[10.6],
[ 9.1],
[11.8],
[10.3],
[14.3],
[11.2],
[15.7],
[ 9.5],
[ 9.9],
[10.1],
[10.4],
[12.3],
[12.9],
[15.1],
[15.9],
[20.9],
[21.5],
[16.5],
[21.5],
[16.5],
[16.5],
[21.5],
[18. ],
[18.8],
[25.7],
[18.2],
[18.9],
[11.8],
[16.8],
[12.7],
[16.9],
[14.1],
[12.6],
[17.8],
[17.3],
[14.3],
```

```
[ 9.1],
[12. ],
[10. ],
[12.6],
[17. ],
[13.8],
[18.4],
[14.7],
[20.1],
[22.1],
[13.8],
[18.4],
[13.8],
[18.4],
[ 7.9],
[ 8.1],
[ 8.7],
[ 9.1],
[ 9.6],
[10.9],
[ 4.8],
[ 7. ],
[ 7.2],
[10.7],
[ 9.1],
[ 6.9],
[ 7.5],
[10.6],
[11.5],
[12. ],
[13.4],
[ 7.7],
[ 7.7],
[ 8.5],
[13.2],
[11.1],
[11.2],
[11.6],
[10.3],
[11.5],
[11.1],
[ 9.6],
[ 6.2],
[ 6.3],
[ 9.3],
[10.1],
[10. ],
[10.6],
[ 7.7],
[ 8. ],
```

```
[ 8.8],
[ 8.5],
[10.2],
[ 7.5],
[ 8.2],
[11.5],
[11.7],
[12.3],
[11.1],
[11.6],
[11. ],
[12.3],
[11.6],
[12.8],
[ 7.6],
[11.8],
[10.6],
[12.8],
[14.5],
[15. ],
[10.3],
[10.6],
[12.8],
[13.3],
[18.1],
[13.3],
[12.4],
[17.5],
[13.3],
[13.3],
[13.1],
[13.7],
[13.5],
[13.7],
[13.5],
[ 9.5],
[10. ],
[10.8],
[10.1],
[10.9],
[11.6],
[ 9.8],
[ 9.1],
[ 9.8],
[ 9.4],
[10.2],
[ 9.4],
[11.1],
[14.8],
[ 9.9],
```

```
[12.1],
[15.8],
[ 9.8],
[ 9.4],
[ 9.4],
[11.1],
[12.8],
[13.4],
[10.9],
[ 8. ],
[ 8.4],
[ 8.4],
[ 9.7],
[ 9.8],
[ 8.4],
[ 8.5],
[ 9.4],
[ 9.6],
[ 8.4],
[ 8.6],
[ 8.7],
[ 7.6],
[ 7.7],
[ 7.6],
[ 9.2],
[ 9.3],
[12. ],
[10.3],
[11.4],
[11. ],
[11.8],
[ 9. ],
[ 9.1],
[10. ],
[15.7],
[17.4],
[11.9],
[14.5],
[ 9.8],
[ 9.8],
[15.4],
[21.4],
[12.4],
[17.4],
[15. ],
[20.7],
[12.3],
[12.3],
[17.3],
[15.4],
```

```
[15.1],
[ 5.6],
[ 6. ],
[ 9.6],
[10.4],
[11. ],
[ 7.6],
[13.9],
[ 9.7],
[10.4],
[10.6],
[11. ],
[10.6],
[12.6],
[12.3],
[12.9],
[11.4],
[16.5],
[11.1],
[ 8.2],
[11.8],
[11.3],
[12.8],
[10.2],
[12.2],
[ 9. ],
[11.4],
[ 6.3],
[16. ],
[21.6],
[12.7],
[12.7],
[15.1],
[14.5],
[13.1],
[ 8.2],
[ 8. ],
[ 8.6],
[ 8.5],
[ 8.9],
[12.3],
[12.6],
[ 7.7],
[ 7.4],
[ 6.9],
[ 7. ],
[ 7.4],
[ 7.1],
[ 7.2],
[ 7.7],
```

```
[ 7.3],
[ 7.2],
[ 9.7],
[ 9.8],
[ 7.6],
[ 7.3],
[10.2],
[ 9.6],
[ 9.9],
[ 8. ],
[ 9.4],
[ 9.4],
[10.5],
[14. ],
[10.4],
[10.6],
[10.5],
[15.1],
[15.3],
[13.2],
[13.4],
[ 7.8],
[ 9.2],
[11.5],
[13.1],
[ 7.2],
[10.2],
[10. ],
[10.4],
[10.5],
[10.2],
[ 8.8],
[11.6],
[11.7],
[11.4],
[12.9],
[13.5],
[13.5],
[18.2],
[17.7],
[12. ],
[15.6],
[11.3],
[16. ],
[10.3],
[14.2],
[15.5],
[12.1],
[12. ],
[14.2],
```

```
[ 8.9],
[ 9. ],
[ 9.8],
[10.7],
[16.5],
[15.7],
[15.7],
[ 7.8],
[ 7.8],
[ 7.8],
[ 7.8],
[ 8.7],
[ 7.8],
[ 7.6],
[ 8.7],
[ 7.8],
[ 7.8],
[ 7.8],
[ 8. ],
[ 8.3],
[ 9. ],
[ 8.5],
[ 8.3],
[ 8. ],
[ 9. ],
[ 8.5],
[ 8.3],
[ 8. ],
[ 8.3],
[ 8. ],
[ 8.3],
[ 8. ],
[ 9. ],
[ 8.5],
[ 8.3],
[ 9. ],
[ 8.5],
[ 8.3],
[ 8. ],
[ 8.2],
[ 8.3],
[12.5],
[12.3],
[11.6],
[ 8.6],
[ 8.6],
[ 6. ],
[ 6.4],
[ 8.6],
[10.2],
```

```
[ 8.7],
[ 8.8],
[ 9. ],
[11. ],
[11.5],
[11.4],
[ 7.6],
[ 9.3],
[16.5],
[12.3],
[12.7],
[13.7],
[13.1],
[ 8.1],
[ 8.6],
[10.9],
[10.6],
[11. ],
[ 8.8],
[11.1],
[ 7. ],
[ 7.8],
[15.6],
[17.3],
[ 6.6],
[ 7.7],
[14.2],
[13.7],
[10.1],
[10.3],
[10.7],
[10.5],
[11. ],
[10.8],
[11. ],
[11. ],
[11.2],
[10.1],
[10.7],
[10.6],
[10.8],
[10.7],
[10.9],
[14. ],
[11.9],
[11.9],
[11.9],
[11.9],
[ 9.2],
[ 9.9],
```

```
[ 9.8],
[13.6],
[10.4],
[13.9],
[13. ],
[10.9],
[13.7],
[ 9.2],
[ 9.8],
[10.4],
[10.9],
[11.1],
[11.3],
[11.5],
[12.7],
[11.3],
[12.8],
[14.8],
[13.6],
[15.4],
[12.6],
[17.8],
[10.2],
[11.9],
[11.3],
[11.4],
[15.9],
[16. ],
[17.1],
[17.1],
[15.6],
[ 8.4],
[ 9. ],
[ 8. ],
[ 7.9],
[ 6.4],
[ 6.4],
[16.9],
[16.9],
[ 9.6],
[ 9.7],
[ 7.8],
[ 8.5],
[11.2],
[ 7.8],
[ 8.5],
[11.2],
[12.5],
[ 8.6],
[ 9.9],
```

```
[11.6],
[ 9. ],
[13.1],
[ 8.5],
[ 9.2],
[ 7.7],
[12.9],
[12.9],
[ 9.6],
[ 5.7],
[ 6. ],
[ 7.7],
[ 7.3],
[ 7.1],
[ 7.4],
[ 6.8],
[ 6.9],
[ 8.5],
[ 8.2],
[12.9],
[13.9],
[10.3],
[11.2],
[11.5],
[ 8.3],
[ 8.2],
[ 4.8],
[ 4.7],
[ 5.5],
[ 9.4],
[ 9.5],
[16.7],
[11.3],
[12.4],
[10.9],
[10.2],
[13.1],
[11.9],
[11.8],
[14. ],
[13.5],
[14.3],
[15.5],
[15.1],
[16.2],
[10. ],
[11. ],
[10.2],
[11.3],
[ 7.1],
```

```
       [ 8.5],
       [ 8.7],
       [ 9.1],
       [ 9.4],
       [ 9.8],
       [ 9.6],
       [ 8.7],
       [ 9.3],
       [ 9.4],
       [ 7.3],
       [ 7.3],
       [ 9.8],
       [12.3],
       [ 9.6],
       [ 9.5],
       [ 9.4],
       [ 7.2],
       [ 8.2],
       [ 8.2],
       [ 9.4],
       [ 9. ],
       [ 9. ],
       [ 9.2],
       [ 6.9],
       [ 5.4],
       [ 8.6],
       [ 8.6],
       [ 7.1],
       [ 6.8],
       [12.1],
       [10.7],
       [10.7],
       [12.2],
       [10.4],
       [ 9.7],
       [10.1],
       [11.5],
       [10.2],
       [11.5],
       [11.2],
       [11.8],
       [11.5],
       [11.8],
       [11.3],
       [12.8]])
```

```
train_y # Column Values of column CO2EMISSIONS  will be used for the
training only as dependent varibale
```

```
array([[196],
       [221],
```

[136],
[255],
[244],
[232],
[267],
[225],
[239],
[359],
[338],
[354],
[338],
[354],
[359],
[230],
[214],
[230],
[214],
[251],
[224],
[258],
[224],
[260],
[227],
[258],
[288],
[230],
[242],
[239],
[258],
[294],
[336],
[407],
[354],
[336],
[407],
[354],
[306],
[308],
[262],
[285],
[262],
[267],
[281],
[281],
[292],
[209],
[209],
[297],
[356],
[320],

```
[380],
[322],
[380],
[380],
[437],
[193],
[200],
[181],
[181],
[193],
[200],
[209],
[209],
[209],
[230],
[228],
[237],
[232],
[193],
[200],
[209],
[221],
[230],
[237],
[202],
[209],
[213],
[232],
[292],
[232],
[292],
[281],
[292],
[292],
[191],
[292],
[338],
[338],
[317],
[338],
[317],
[209],
[260],
[230],
[253],
[320],
[370],
[274],
[209],
[200],
```

```
[278],
[253],
[278],
[283],
[290],
[196],
[212],
[264],
[250],
[264],
[264],
[225],
[246],
[189],
[225],
[218],
[221],
[235],
[207],
[251],
[232],
[258],
[262],
[232],
[248],
[248],
[260],
[253],
[260],
[255],
[382],
[340],
[382],
[340],
[338],
[322],
[377],
[283],
[294],
[271],
[290],
[258],
[269],
[294],
[382],
[340],
[276],
[258],
[179],
[189],
```

[192],
[177],
[163],
[209],
[259],
[290],
[272],
[366],
[334],
[380],
[344],
[380],
[344],
[362],
[380],
[344],
[380],
[362],
[414],
[432],
[411],
[419],
[435],
[229],
[200],
[228],
[184],
[244],
[271],
[269],
[292],
[270],
[324],
[290],
[299],
[329],
[184],
[173],
[184],
[172],
[182],
[191],
[182],
[191],
[182],
[159],
[317],
[294],
[317],
[294],

[317],
[294],
[317],
[294],
[290],
[207],
[232],
[246],
[230],
[230],
[230],
[294],
[251],
[251],
[237],
[222],
[276],
[269],
[225],
[230],
[301],
[294],
[320],
[317],
[258],
[237],
[294],
[251],
[308],
[251],
[237],
[235],
[222],
[186],
[198],
[202],
[195],
[202],
[168],
[168],
[278],
[250],
[271],
[264],
[255],
[285],
[276],
[179],
[179],
[182],

```
[179],
[156],
[200],
[193],
[373],
[350],
[396],
[360],
[386],
[368],
[488],
[225],
[264],
[281],
[209],
[216],
[214],
[230],
[368],
[346],
[274],
[254],
[288],
[267],
[294],
[366],
[317],
[308],
[283],
[283],
[264],
[336],
[336],
[315],
[308],
[288],
[308],
[288],
[320],
[318],
[152],
[184],
[264],
[278],
[294],
[205],
[177],
[165],
[179],
[165],
```

[184],
[170],
[166],
[162],
[196],
[186],
[205],
[186],
[216],
[126],
[244],
[264],
[244],
[209],
[271],
[237],
[229],
[258],
[251],
[218],
[228],
[232],
[239],
[283],
[297],
[347],
[366],
[334],
[344],
[380],
[344],
[380],
[380],
[344],
[414],
[432],
[411],
[419],
[435],
[271],
[269],
[292],
[270],
[324],
[290],
[285],
[277],
[329],
[209],
[276],

```
[230],
[290],
[272],
[317],
[294],
[338],
[322],
[354],
[317],
[294],
[317],
[294],
[182],
[186],
[200],
[209],
[221],
[251],
[110],
[161],
[166],
[246],
[209],
[159],
[172],
[244],
[265],
[276],
[308],
[177],
[177],
[196],
[304],
[255],
[258],
[267],
[237],
[264],
[255],
[221],
[143],
[145],
[214],
[232],
[230],
[244],
[177],
[184],
[202],
[196],
```

[235],
[172],
[189],
[265],
[269],
[283],
[255],
[267],
[253],
[283],
[267],
[294],
[175],
[271],
[244],
[294],
[334],
[345],
[237],
[244],
[294],
[306],
[290],
[306],
[285],
[280],
[306],
[306],
[301],
[315],
[310],
[315],
[310],
[218],
[230],
[248],
[232],
[251],
[267],
[225],
[209],
[225],
[216],
[235],
[216],
[255],
[340],
[267],
[278],
[363],

[225],
[216],
[216],
[255],
[294],
[308],
[251],
[184],
[193],
[193],
[223],
[225],
[193],
[196],
[216],
[221],
[193],
[198],
[200],
[175],
[177],
[175],
[212],
[214],
[276],
[237],
[262],
[253],
[271],
[207],
[209],
[230],
[361],
[400],
[274],
[334],
[225],
[225],
[354],
[342],
[285],
[278],
[345],
[331],
[283],
[283],
[277],
[354],
[347],
[129],

```
[138],
[221],
[239],
[253],
[175],
[320],
[223],
[239],
[244],
[253],
[244],
[290],
[283],
[297],
[262],
[380],
[255],
[189],
[271],
[260],
[294],
[235],
[281],
[207],
[262],
[145],
[368],
[346],
[292],
[292],
[347],
[334],
[301],
[189],
[184],
[198],
[196],
[205],
[283],
[290],
[177],
[170],
[159],
[161],
[170],
[163],
[166],
[177],
[168],
[166],
```

```
[223],
[225],
[175],
[168],
[235],
[221],
[228],
[184],
[216],
[216],
[242],
[224],
[239],
[244],
[242],
[347],
[352],
[304],
[308],
[179],
[212],
[264],
[301],
[194],
[235],
[230],
[239],
[242],
[235],
[202],
[267],
[269],
[262],
[297],
[310],
[310],
[419],
[407],
[324],
[359],
[260],
[256],
[278],
[327],
[356],
[278],
[276],
[327],
[205],
[207],
```

```
[225],
[246],
[380],
[361],
[361],
[179],
[179],
[179],
[179],
[200],
[179],
[175],
[200],
[179],
[179],
[179],
[184],
[191],
[207],
[196],
[191],
[184],
[207],
[196],
[191],
[184],
[191],
[184],
[191],
[184],
[207],
[196],
[191],
[207],
[196],
[191],
[184],
[189],
[191],
[288],
[283],
[267],
[198],
[198],
[138],
[147],
[198],
[235],
[200],
[202],
```

[207],
[253],
[264],
[262],
[175],
[214],
[380],
[283],
[292],
[315],
[301],
[186],
[198],
[251],
[244],
[253],
[202],
[255],
[161],
[179],
[359],
[398],
[152],
[177],
[327],
[315],
[232],
[237],
[246],
[242],
[253],
[248],
[253],
[253],
[258],
[232],
[246],
[244],
[248],
[246],
[251],
[322],
[274],
[274],
[274],
[274],
[212],
[228],
[225],
[313],

[281],
[320],
[299],
[251],
[315],
[212],
[225],
[239],
[251],
[255],
[260],
[264],
[292],
[260],
[294],
[340],
[313],
[354],
[290],
[285],
[275],
[274],
[260],
[262],
[254],
[368],
[393],
[393],
[359],
[193],
[207],
[184],
[182],
[147],
[147],
[389],
[389],
[221],
[223],
[179],
[196],
[258],
[179],
[196],
[258],
[288],
[198],
[228],
[267],
[207],

[301],
[196],
[212],
[177],
[297],
[297],
[221],
[131],
[138],
[177],
[168],
[163],
[170],
[156],
[159],
[196],
[189],
[297],
[320],
[237],
[258],
[264],
[191],
[189],
[110],
[108],
[126],
[216],
[218],
[384],
[260],
[285],
[251],
[235],
[301],
[274],
[271],
[322],
[310],
[329],
[356],
[347],
[373],
[230],
[253],
[235],
[260],
[163],
[196],
[200],

```
       [209],
       [216],
       [225],
       [221],
       [200],
       [214],
       [216],
       [197],
       [197],
       [225],
       [283],
       [221],
       [218],
       [216],
       [194],
       [189],
       [189],
       [216],
       [207],
       [207],
       [212],
       [186],
       [124],
       [198],
       [198],
       [192],
       [184],
       [278],
       [246],
       [246],
       [281],
       [281],
       [223],
       [232],
       [264],
       [235],
       [264],
       [258],
       [271],
       [264],
       [271],
       [260],
       [294]], dtype=int64)
```

regr.fit(independentvariable,dependentvariable)

```
regr.fit(train_x,train_y)
```

```
LinearRegression()
```

```
# The coefficients
print ('Coefficients: ', regr.coef_)
print ('Intercept: ',regr.intercept_)
```

```
Coefficients:  [[16.37691523]]
Intercept:  [66.31999663]
```

```
regr.coef_
```

```
array([[16.37691523]])
```

```
regr.intercept_
```

```
array([66.31999663])
```

```
regr.coef_[0][0]
```

```
16.37691522903722
```

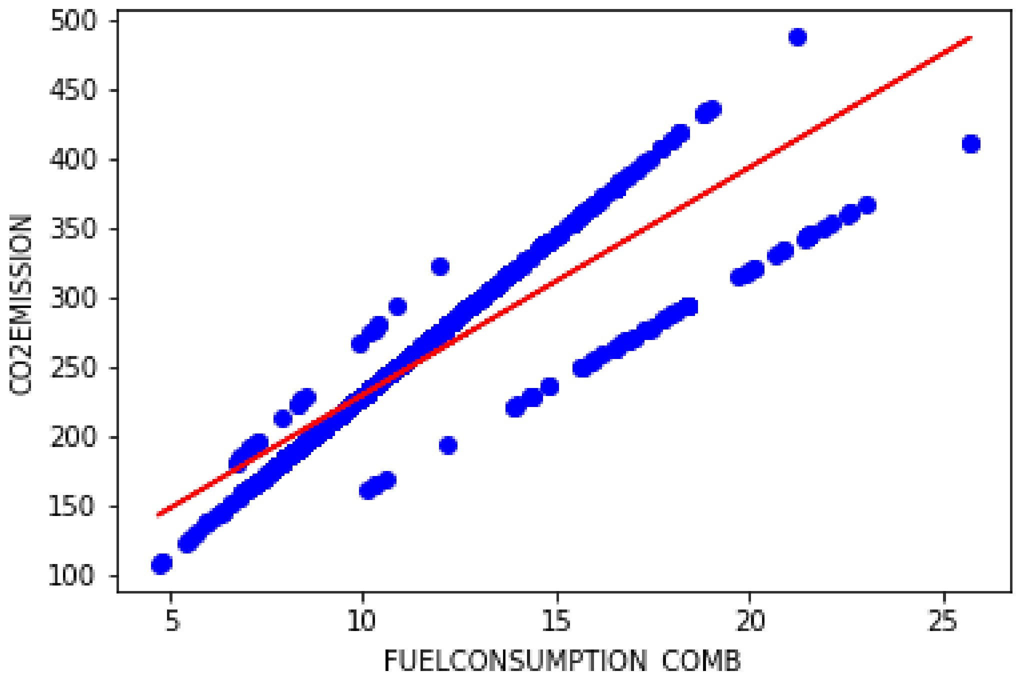
```
regr.intercept_[0]
```

```
66.31999662763249
```

Plot Outputs

```
plt.scatter(train.FUELCONSUMPTION_COMB, train.CO2EMISSIONS,
color='blue')
plt.plot(train_x, regr.coef_[0][0]*train_x + regr.intercept_[0], 'r')
# y = mx+c
plt.xlabel("FUELCONSUMPTION_COMB")
plt.ylabel("CO2EMISSION")
```

```
Text(0, 0.5, 'CO2EMISSION')
```



Evaluation

```
test_x = np.asanyarray(test[['FUELCONSUMPTION_COMB']]) #question in exam Questions
test_y = np.asanyarray(test[['CO2EMISSIONS']])#(ACTUAL answers)
```

```
test_x.shape # Independent variable data for testing
```

```
(222, 1)
```

```
test_y.shape # Dependent variable data for testing
```

```
(222, 1)
```

```
predicted_y = regr.predict(test_x) #whatever answerr will be generated will be stored in predicted_y predicted
```

```
predicted_y
```

```
array([[230.08914892],
       [248.10375567],
       [216.98761673],
       [321.7998742 ],
       [210.43685064],
       [230.08914892],
       [233.36453196],
       [249.74144719],
       [323.43756572],
       [216.98761673],
       [215.34992521],
```

```
[282.49527765],
[357.8290877 ],
[357.8290877 ],
[272.66912851],
[269.39374547],
[277.58220308],
[235.00222349],
[235.00222349],
[210.43685064],
[223.53838283],
[228.4514574 ],
[241.55298958],
[266.11836242],
[241.55298958],
[274.30682004],
[320.16218268],
[218.62530826],
[216.98761673],
[274.30682004],
[292.32142679],
[307.06065049],
[292.32142679],
[248.10375567],
[297.23450136],
[200.61070151],
[228.4514574 ],
[228.4514574 ],
[251.37913872],
[275.94451156],
[253.01683024],
[338.17678943],
[308.69834202],
[428.24982319],
[253.01683024],
[238.27760653],
[274.30682004],
[194.05993541],
[210.43685064],
[262.84297938],
[230.08914892],
[313.61141659],
[336.53909791],
[488.84440954],
[223.53838283],
[239.91529806],
[233.36453196],
[357.8290877 ],
[349.64063009],
[202.24839303],
[195.69762694],
```

```
[202.24839303],
[179.32071171],
[267.75605394],
[203.88608455],
[198.97300998],
[235.00222349],
[238.27760653],
[226.81376587],
[238.27760653],
[285.7706607 ],
[235.00222349],
[292.32142679],
[238.27760653],
[302.14757593],
[249.74144719],
[249.74144719],
[323.43756572],
[292.32142679],
[207.1614676 ],
[216.98761673],
[302.14757593],
[331.62602334],
[184.23378628],
[195.69762694],
[194.05993541],
[164.581488  ],
[366.01754532],
[457.7282706 ],
[243.1906811 ],
[223.53838283],
[233.36453196],
[279.21989461],
[274.30682004],
[352.91601314],
[336.53909791],
[388.94522664],
[310.33603354],
[308.69834202],
[361.10447075],
[190.78455237],
[190.78455237],
[180.95840323],
[213.71223369],
[233.36453196],
[274.30682004],
[282.49527765],
[336.53909791],
[436.4382808 ],
[336.53909791],
[436.4382808 ],
```

```
[488.84440954],
[279.21989461],
[331.62602334],
[292.32142679],
[367.65523684],
[334.90140638],
[190.78455237],
[220.26299978],
[169.49456257],
[192.42224389],
[207.1614676 ],
[205.52377607],
[238.27760653],
[243.1906811 ],
[246.46606415],
[205.52377607],
[238.27760653],
[266.11836242],
[284.13296917],
[269.39374547],
[352.91601314],
[284.13296917],
[272.66912851],
[238.27760653],
[246.46606415],
[321.7998742 ],
[215.34992521],
[226.81376587],
[233.36453196],
[275.94451156],
[285.7706607 ],
[202.24839303],
[228.4514574 ],
[212.07454217],
[210.43685064],
[251.37913872],
[253.01683024],
[241.55298958],
[366.01754532],
[313.61141659],
[269.39374547],
[307.06065049],
[416.78598253],
[225.17607435],
[277.58220308],
[257.92990481],
[266.11836242],
[316.88679963],
[185.8714778 ],
[180.95840323],
```

```
[198.97300998],
[238.27760653],
[280.85758613],
[236.63991501],
[277.58220308],
[315.24910811],
[325.07525725],
[205.52377607],
[264.4806709 ],
[282.49527765],
[257.92990481],
[274.30682004],
[185.8714778 ],
[202.24839303],
[197.33531846],
[197.33531846],
[221.9006913 ],
[215.34992521],
[257.92990481],
[251.37913872],
[277.58220308],
[208.79915912],
[259.56759633],
[218.62530826],
[202.24839303],
[207.1614676 ],
[241.55298958],
[236.63991501],
[262.84297938],
[228.4514574 ],
[277.58220308],
[297.23450136],
[239.91529806],
[341.45217248],
[328.35064029],
[346.36524704],
[346.36524704],
[221.9006913 ],
[167.85687105],
[213.71223369],
[228.4514574 ],
[228.4514574 ],
[205.52377607],
[221.9006913 ],
[207.1614676 ],
[271.03143699],
[226.81376587],
[257.92990481],
[200.61070151],
[223.53838283],
```

```
       [212.07454217],
       [284.13296917],
       [185.8714778 ],
       [231.72684044],
       [225.17607435],
       [179.32071171],
       [179.32071171],
       [225.17607435],
       [226.81376587],
       [243.1906811 ],
       [256.29221328]])
```

```
test_y.shape
```

```
(222, 1)
```

```
predicted_y.shape
```

```
(222, 1)
```

```
from sklearn.metrics import r2_score
print(f"Mean absolute error: {np.mean(np.absolute(predicted_y - test_y))} ")# pred - actual
```

```
Mean absolute error: 20.0264532711564
```

```
print("Residual sum of squares (MSE): %.2f" % np.mean((predicted_y - test_y) **2))
```

```
Residual sum of squares (MSE): 822.80
```

```
from sklearn.metrics import r2_score
print(f"R2-score:{r2_score(test_y ,predicted_y)*100} %") # Most accurate thing or evaluation metrics
```

```
R2-score:79.53048969001148 %
```

Apply Lasso Regression to cover the overfitting and underfitting problem

```
from sklearn.linear_model import Lasso
```

```
L = Lasso(alpha=1)
```

```
L.fit(train_x,train_y)
```

```
Lasso(alpha=1)
```

```
ypred2 = L.predict(test_x)
```

```
from sklearn.metrics import r2_score
print("R2-score",r2_score(test_y,ypred2))
```

```
print(f"Mean absolute error: {np.mean(np.absolute(ypred2 - test_y))}
")# pred - actual
```

```
R2-score 0.7956622176158492
Mean absolute error: 67.56303603234201
```

Que.6 Train your model on different train test ratio and train the models and note down there accuracies

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier

Bill_Authentication = pd.read_csv("C:/Users/Lenovo/Documents/Data
Set/bill_authentication.csv")
Bill_Authentication
```

```
      Variance  Skewness  Curtosis  Entropy  Class
0      3.62160   8.66610   -2.8073 -0.44699      0
1      4.54590   8.16740   -2.4586 -1.46210      0
2      3.86600  -2.63830    1.9242  0.10645      0
3      3.45660   9.52280   -4.0112 -3.59440      0
4      0.32924  -4.45520    4.5718 -0.98880      0
...        ...       ...       ...      ...    ...
1367   0.40614   1.34920   -1.4501 -0.55949      1
1368  -1.38870  -4.87730    6.4774  0.34179      1
1369  -3.75030 -13.45860   17.5932 -2.77710      1
1370  -3.56370  -8.38270   12.3930 -1.28230      1
1371  -2.54190  -0.65804    2.6842  1.19520      1

[1372 rows x 5 columns]
```

```
Bill_Authentication.head()
```

```
   Variance  Skewness  Curtosis  Entropy  Class
0   3.62160    8.6661   -2.8073 -0.44699      0
1   4.54590    8.1674   -2.4586 -1.46210      0
2   3.86600   -2.6383    1.9242  0.10645      0
3   3.45660    9.5228   -4.0112 -3.59440      0
4   0.32924   -4.4552    4.5718 -0.98880      0
```

```
Bill_Authentication.tail()
```

```
      Variance  Skewness  Curtosis  Entropy  Class
1367   0.40614   1.34920   -1.4501 -0.55949      1
1368  -1.38870  -4.87730    6.4774  0.34179      1
1369  -3.75030 -13.45860   17.5932 -2.77710      1
1370  -3.56370  -8.38270   12.3930 -1.28230      1
1371  -2.54190  -0.65804    2.6842  1.19520      1
```

```
X = Bill_Authentication[["Variance","Skewness","Curtosis","Entropy"]]
# Independant Variable

X

      Variance  Skewness  Curtosis  Entropy
0      3.62160   8.66610   -2.8073 -0.44699
1      4.54590   8.16740   -2.4586 -1.46210
2      3.86600  -2.63830    1.9242  0.10645
3      3.45660   9.52280   -4.0112 -3.59440
4      0.32924  -4.45520    4.5718 -0.98880
...        ...       ...       ...      ...
1367   0.40614   1.34920   -1.4501 -0.55949
1368  -1.38870  -4.87730    6.4774  0.34179
1369  -3.75030 -13.45860   17.5932 -2.77710
1370  -3.56370  -8.38270   12.3930 -1.28230
1371  -2.54190  -0.65804    2.6842  1.19520

[1372 rows x 4 columns]

Y = Bill_Authentication[["Class"]]

Y

      Class
0         0
1         0
2         0
3         0
4         0
...     ...
1367      1
1368      1
1369      1
1370      1
1371      1

[1372 rows x 1 columns]
```

Setting up Decision Tree

```
from sklearn.model_selection import train_test_split
X_trainset, X_testset, Y_trainset, Y_testset = train_test_split(X, Y,
test_size=0.2, random_state=500)
```

Modelling

```
ClassTree = DecisionTreeClassifier(criterion="entropy", max_depth = 4)
ClassTree # it shows the default parameters
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=4)
```

```
ClassTree.fit(X_trainset,Y_trainset)
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=4)
```

Prediction

Let's make some predictions on the testing dataset and store it into a variable called predTree

```
Y_pred = ClassTree.predict(X_testset)
```

```
Y_pred
```

```
array([1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1,
1,
       1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0,
0,
       0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1,
0,
       0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
1,
       1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1,
0,
       0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0,
1,
       1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1,
0,
       1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0,
1,
       0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0,
1,
       1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0,
0,
       0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1,
1,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0,
0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0], dtype=int64)
```

```
# You can print out predTree and y_testset if you want to visually
compare the prediction to the actual values.
```

```
print (Y_pred [0:5])# Predicted by the ml model
print (Y_testset [0:5])# Actual values we have
```

```
[1 0 1 1 0]
      Class
```

```
816     1
316     0
762     1
1128    1
232     0
```

```
print (Y_testset [0:5])
```

```
      Class
816       1
316       0
762       1
1128      1
232       0
```

Evaluation

Next, let's import metrics from sklearn and check the accuracy of our model.

```
from sklearn import metrics
print("DecisionTrees's Accuracy: ", metrics.accuracy_score(Y_testset, 
Y_pred))
```

```
DecisionTrees's Accuracy:  0.9490909090909091
```

```
Bill_Authentication.head()
```

```
   Variance  Skewness  Curtosis  Entropy  Class
0   3.62160    8.6661   -2.8073 -0.44699      0
1   4.54590    8.1674   -2.4586 -1.46210      0
2   3.86600   -2.6383    1.9242  0.10645      0
3   3.45660    9.5228   -4.0112 -3.59440      0
4   0.32924   -4.4552    4.5718 -0.98880      0
```

```
Bill_Authentication.tail()
```

```
      Variance  Skewness  Curtosis  Entropy  Class
1367   0.40614   1.34920   -1.4501 -0.55949      1
1368  -1.38870  -4.87730    6.4774  0.34179      1
1369  -3.75030 -13.45860   17.5932 -2.77710      1
1370  -3.56370  -8.38270   12.3930 -1.28230      1
1371  -2.54190  -0.65804    2.6842  1.19520      1
```

```
ClassTree.predict([[3,4,-4,3]])
```

```
C:\Users\Lenovo\anaconda3\lib\site-packages\sklearn\base.py:450: 
UserWarning: X does not have valid feature names, but 
DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

```
array([1], dtype=int64)
```

```
max(Bill_Authentication['Variance'])
```

```
6.8248
```

```
min(Bill_Authentication['Variance'])
```

```
-7.0421
```

```
X_testset[0:5]
```

```
      Variance  Skewness  Curtosis   Entropy
816    -4.8554 -5.903700   10.9818 -0.821990
316     5.7353  5.280800   -2.2598  0.075416
762    -1.3971  3.319100   -1.3927 -1.994800
1128   -2.1802  3.379100   -1.2256 -2.662100
232     2.2596 -0.033118    4.7355 -0.277600
```

```
Y_testset = np.array(Y_testset)
Y_testset
```

```
array([[1],
       [0],
       [1],
       [1],
       [0],
       [1],
       [1],
       [1],
       [0],
       [1],
       [0],
       [1],
       [1],
       [0],
       [1],
       [1],
       [1],
       [0],
       [1],
       [1],
       [1],
       [1],
       [0],
       [0],
       [1],
       [1],
       [0],
       [1],
       [0],
       [0],
```

```
[1],
[0],
[0],
[0],
[1],
[0],
[1],
[0],
[0],
[1],
[1],
[1],
[0],
[0],
[0],
[0],
[0],
[0],
[1],
[0],
[1],
[1],
[0],
[1],
[0],
[0],
[0],
[0],
[1],
[0],
[0],
[1],
[1],
[0],
[1],
[0],
[0],
[0],
[1],
[0],
[0],
[1],
[0],
[1],
[1],
[0],
[1],
[0],
[1],
[0],
```

```
[0],
[1],
[1],
[0],
[1],
[1],
[0],
[1],
[1],
[1],
[0],
[1],
[1],
[0],
[1],
[1],
[0],
[1],
[0],
[0],
[0],
[1],
[1],
[0],
[1],
[0],
[0],
[0],
[0],
[0],
[0],
[0],
[1],
[1],
[0],
[0],
[0],
[1],
[1],
[1],
[0],
[0],
[0],
[0],
[1],
[0],
[1],
[0],
[0],
[1],
```

```
[0],
[1],
[1],
[0],
[1],
[1],
[0],
[1],
[0],
[1],
[1],
[1],
[0],
[0],
[0],
[0],
[1],
[0],
[1],
[1],
[0],
[0],
[1],
[0],
[0],
[1],
[1],
[1],
[0],
[1],
[0],
[1],
[0],
[0],
[0],
[1],
[0],
[1],
[0],
[1],
[1],
[1],
[0],
[1],
[0],
[1],
[0],
[1],
[0],
[1],
```

```
[1],
[0],
[1],
[0],
[0],
[0],
[1],
[1],
[1],
[0],
[1],
[0],
[0],
[0],
[0],
[0],
[0],
[1],
[1],
[0],
[0],
[0],
[0],
[1],
[1],
[0],
[1],
[1],
[0],
[1],
[1],
[1],
[1],
[0],
[0],
[0],
[1],
[1],
[0],
[0],
[0],
[1],
[0],
[0],
[0],
[1],
[0],
[1],
[1],
[0],
```

```
       [0],
       [1],
       [1],
       [0],
       [0],
       [1],
       [0],
       [0],
       [0],
       [1],
       [1],
       [1],
       [1],
       [0],
       [0],
       [0],
       [0],
       [0],
       [1],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [1],
       [0],
       [1],
       [1],
       [0],
       [1],
       [0],
       [0],
       [1],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [1],
       [0],
       [0],
       [0],
       [0]], dtype=int64)
```

```
X_testset
```

```
      Variance  Skewness  Curtosis   Entropy
816    -4.8554 -5.903700  10.98180 -0.821990
316     5.7353  5.280800  -2.25980  0.075416
```

```
762     -1.3971  3.319100  -1.39270 -1.994800
1128    -2.1802  3.379100  -1.22560 -2.662100
232      2.2596 -0.033118   4.73550 -0.277600
...         ...       ...       ...       ...
1122    -2.0285  3.846800  -0.63435 -1.175000
141      1.7317 -0.347650   4.19050 -0.991380
633      4.4295 -2.350700   1.70480  0.909460
428      3.4246 -0.146930   0.80342  0.291360
277      1.3638 -4.775900   8.41820 -1.883600

[275 rows x 4 columns]
```

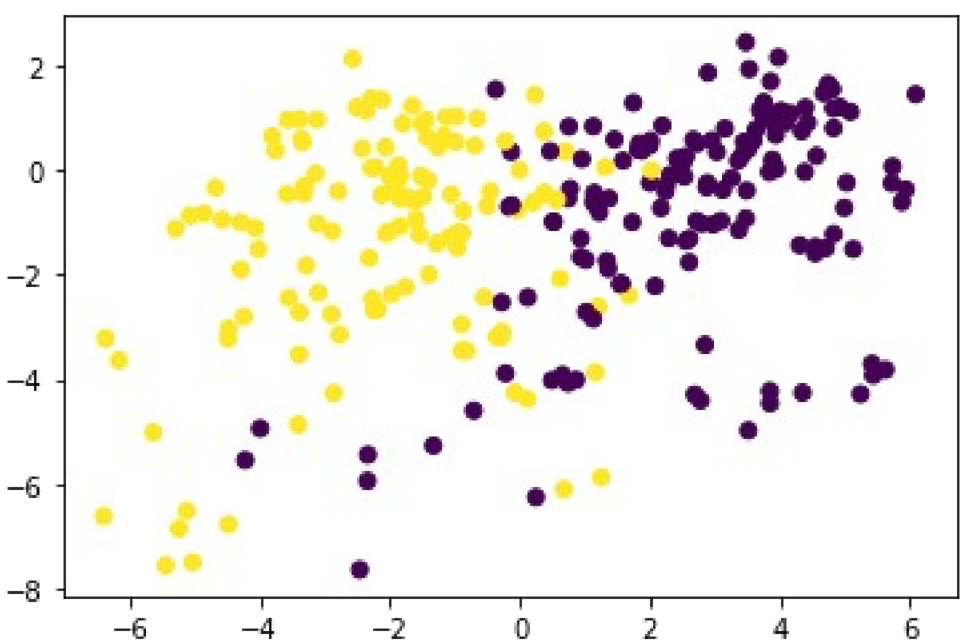
```
X_testset = X_testset.values
```

```
X_testset
```

```
array([[-4.8554  , -5.9037  , 10.9818  , -0.82199 ],
       [ 5.7353  ,  5.2808  , -2.2598  ,  0.075416],
       [-1.3971  ,  3.3191  , -1.3927  , -1.9948  ],
       ...,
       [ 4.4295  , -2.3507  ,  1.7048  ,  0.90946 ],
       [ 3.4246  , -0.14693 ,  0.80342 ,  0.29136 ],
       [ 1.3638  , -4.7759  ,  8.4182  , -1.8836  ]])
```

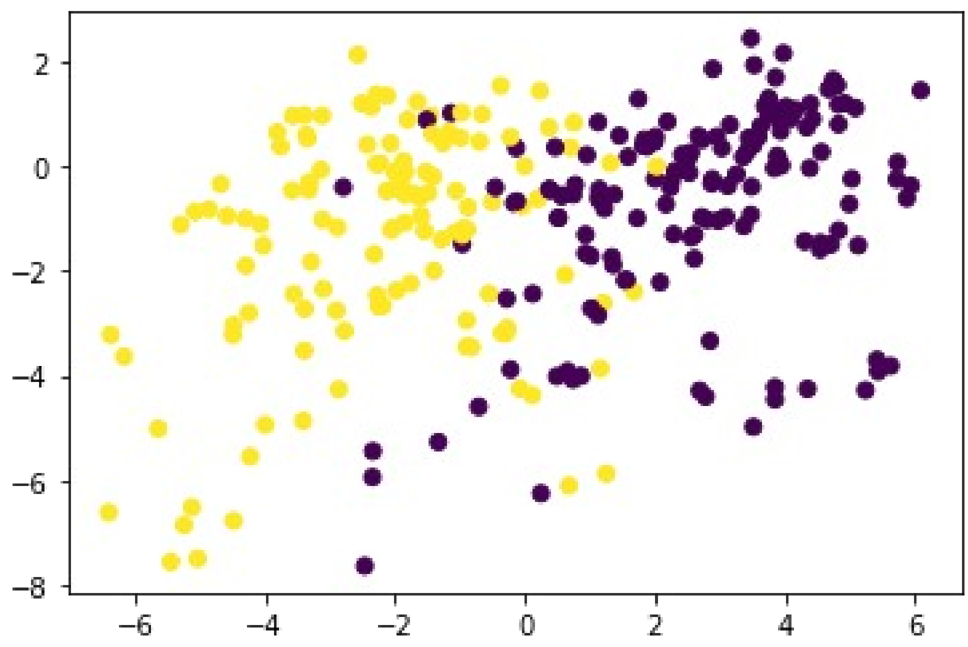
```
plt.scatter(X_testset[:,0],X_testset[:,-1], c = Y_pred)
```

```
<matplotlib.collections.PathCollection at 0x2118c333d00>
```




```
plt.scatter(X_testset[:,0],X_testset[:,-1], c = Y_testset)
```

```
<matplotlib.collections.PathCollection at 0x2118c3aa250>
```


2
0
−2
−4
−6
−8
−6
−4
−2
0
2
4
6

For the accuracy of training and testing are

1) When take tranning data 80% and testing data 20% then accuracy is 94.90%

2) When take tranning data 70% and testing data the 30% then accuracy is 96.35%

3) When take tranning data is 60% and testing data is 40% then accuracy is 96.90%

4) When take tranning data 50% and testing data 50% then accuracy is 96.50%

Que.7 we are providing you another dataset regarding housing prediction to need to apply Linear Regression on atleast 5 pairs of independent and dependent variable and store their accuracy and thenmake a plot of those accuracyNote : Dependent Variable is SalesPrice

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("C:/Users/Lenovo/Documents/Data Set/housing.csv")

df

        Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley
LotShape  \
0        1          60       RL         65.0     8450   Pave   NaN
Reg
1        2          20       RL         80.0     9600   Pave   NaN
Reg
2        3          60       RL         68.0    11250   Pave   NaN
IR1
3        4          70       RL         60.0     9550   Pave   NaN
IR1
4        5          60       RL         84.0    14260   Pave   NaN
IR1
```

```
...    ...          ...       ...          ...       ...    ...   ...
...
1455  1456           60        RL         62.0      7917   Pave   NaN
Reg
1456  1457           20        RL         85.0     13175   Pave   NaN
Reg
1457  1458           70        RL         66.0      9042   Pave   NaN
Reg
1458  1459           20        RL         68.0      9717   Pave   NaN
Reg
1459  1460           20        RL         75.0      9937   Pave   NaN
Reg

     LandContour Utilities  ... PoolArea PoolQC  Fence MiscFeature
MiscVal  \
0            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
2            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
3            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
4            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
...          ...       ...  ...      ...    ...    ...         ...
...
1455         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1456         Lvl    AllPub  ...        0    NaN  MnPrv         NaN
0
1457         Lvl    AllPub  ...        0    NaN  GdPrv        Shed
2500
1458         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1459         Lvl    AllPub  ...        0    NaN    NaN         NaN
0

     MoSold YrSold  SaleType  SaleCondition  SalePrice
0         2   2008        WD         Normal     208500
1         5   2007        WD         Normal     181500
2         9   2008        WD         Normal     223500
3         2   2006        WD        Abnorml     140000
4        12   2008        WD         Normal     250000
...     ...    ...       ...            ...        ...
1455      8   2007        WD         Normal     175000
1456      2   2010        WD         Normal     210000
1457      5   2010        WD         Normal     266500
1458      4   2010        WD         Normal     142125
1459      6   2008        WD         Normal     147500
```

```
[1460 rows x 81 columns]
```

df.head()

```
   Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape
\
0   1          60       RL         65.0     8450   Pave   NaN      Reg

1   2          20       RL         80.0     9600   Pave   NaN      Reg

2   3          60       RL         68.0    11250   Pave   NaN      IR1

3   4          70       RL         60.0     9550   Pave   NaN      IR1

4   5          60       RL         84.0    14260   Pave   NaN      IR1


  LandContour Utilities  ... PoolArea PoolQC Fence MiscFeature MiscVal
MoSold  \
0         Lvl    AllPub  ...        0    NaN   NaN         NaN       0
2
1         Lvl    AllPub  ...        0    NaN   NaN         NaN       0
5
2         Lvl    AllPub  ...        0    NaN   NaN         NaN       0
9
3         Lvl    AllPub  ...        0    NaN   NaN         NaN       0
2
4         Lvl    AllPub  ...        0    NaN   NaN         NaN       0
12

  YrSold  SaleType  SaleCondition  SalePrice
0   2008        WD         Normal     208500
1   2007        WD         Normal     181500
2   2008        WD         Normal     223500
3   2006        WD        Abnorml     140000
4   2008        WD         Normal     250000

[5 rows x 81 columns]
```

df.tail()

```
        Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley
LotShape  \
1455  1456          60       RL         62.0     7917   Pave   NaN
Reg
1456  1457          20       RL         85.0    13175   Pave   NaN
Reg
1457  1458          70       RL         66.0     9042   Pave   NaN
Reg
```

```
1458  1459          20       RL         68.0     9717   Pave   NaN
Reg
1459  1460          20       RL         75.0     9937   Pave   NaN
Reg

     LandContour Utilities  ... PoolArea PoolQC  Fence MiscFeature
MiscVal  \
1455         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1456         Lvl    AllPub  ...        0    NaN  MnPrv         NaN
0
1457         Lvl    AllPub  ...        0    NaN  GdPrv        Shed
2500
1458         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1459         Lvl    AllPub  ...        0    NaN    NaN         NaN
0

     MoSold YrSold  SaleType  SaleCondition  SalePrice
1455      8   2007        WD         Normal     175000
1456      2   2010        WD         Normal     210000
1457      5   2010        WD         Normal     266500
1458      4   2010        WD         Normal     142125
1459      6   2008        WD         Normal     147500

[5 rows x 81 columns]
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1460 non-null   int64
 1   MSSubClass     1460 non-null   int64
 2   MSZoning       1460 non-null   object
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64
 5   Street         1460 non-null   object
 6   Alley          91 non-null     object
 7   LotShape       1460 non-null   object
 8   LandContour    1460 non-null   object
 9   Utilities      1460 non-null   object
 10  LotConfig      1460 non-null   object
 11  LandSlope      1460 non-null   object
 12  Neighborhood   1460 non-null   object
 13  Condition1     1460 non-null   object
 14  Condition2     1460 non-null   object
 15  BldgType       1460 non-null   object
```

```
 16  HouseStyle     1460 non-null   object
 17  OverallQual    1460 non-null   int64
 18  OverallCond    1460 non-null   int64
 19  YearBuilt      1460 non-null   int64
 20  YearRemodAdd   1460 non-null   int64
 21  RoofStyle      1460 non-null   object
 22  RoofMatl       1460 non-null   object
 23  Exterior1st    1460 non-null   object
 24  Exterior2nd    1460 non-null   object
 25  MasVnrType     1452 non-null   object
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object
 28  ExterCond      1460 non-null   object
 29  Foundation     1460 non-null   object
 30  BsmtQual       1423 non-null   object
 31  BsmtCond       1423 non-null   object
 32  BsmtExposure   1422 non-null   object
 33  BsmtFinType1   1423 non-null   object
 34  BsmtFinSF1     1460 non-null   int64
 35  BsmtFinType2   1422 non-null   object
 36  BsmtFinSF2     1460 non-null   int64
 37  BsmtUnfSF      1460 non-null   int64
 38  TotalBsmtSF    1460 non-null   int64
 39  Heating        1460 non-null   object
 40  HeatingQC      1460 non-null   object
 41  CentralAir     1460 non-null   object
 42  Electrical     1459 non-null   object
 43  1stFlrSF       1460 non-null   int64
 44  2ndFlrSF       1460 non-null   int64
 45  LowQualFinSF   1460 non-null   int64
 46  GrLivArea      1460 non-null   int64
 47  BsmtFullBath   1460 non-null   int64
 48  BsmtHalfBath   1460 non-null   int64
 49  FullBath       1460 non-null   int64
 50  HalfBath       1460 non-null   int64
 51  BedroomAbvGr   1460 non-null   int64
 52  KitchenAbvGr   1460 non-null   int64
 53  KitchenQual    1460 non-null   object
 54  TotRmsAbvGrd   1460 non-null   int64
 55  Functional     1460 non-null   object
 56  Fireplaces     1460 non-null   int64
 57  FireplaceQu    770 non-null    object
 58  GarageType     1379 non-null   object
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1379 non-null   object
 64  GarageCond     1379 non-null   object
 65  PavedDrive     1460 non-null   object
```

```
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          281 non-null    object
 74  MiscFeature    54 non-null     object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   int64
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

```
df.describe()
```

```
                Id   MSSubClass  LotFrontage        LotArea  
OverallQual  \
count  1460.000000  1460.000000  1201.000000    1460.000000  
1460.000000   
mean    730.500000    56.897260    70.049958   10516.828082  
6.099315   
std     421.610009    42.300571    24.284752    9981.264932  
1.382997   
min       1.000000    20.000000    21.000000    1300.000000  
1.000000   
25%     365.750000    20.000000    59.000000    7553.500000  
5.000000   
50%     730.500000    50.000000    69.000000    9478.500000  
6.000000   
75%    1095.250000    70.000000    80.000000   11601.500000  
7.000000   
max    1460.000000   190.000000   313.000000  215245.000000  
10.000000   

       OverallCond    YearBuilt  YearRemodAdd   MasVnrArea  
BsmtFinSF1  ...  \
count  1460.000000  1460.000000   1460.000000  1452.000000  
1460.000000  ...   
mean      5.575342  1971.267808   1984.865753   103.685262  
443.639726  ...   
std       1.112799    30.202904     20.645407   181.066207  
456.098091  ...   
min       1.000000  1872.000000   1950.000000     0.000000  
0.000000  ...   
25%       5.000000  1954.000000   1967.000000     0.000000
```

```
0.000000  ...
50%       5.000000  1973.000000   1994.000000     0.000000
383.500000  ...
75%       6.000000  2000.000000   2004.000000   166.000000
712.250000  ...
max       9.000000  2010.000000   2010.000000  1600.000000
5644.000000  ...

        WoodDeckSF  OpenPorchSF  EnclosedPorch    3SsnPorch
ScreenPorch  \
count  1460.000000  1460.000000    1460.000000  1460.000000
1460.000000
mean     94.244521    46.660274      21.954110     3.409589
15.060959
std     125.338794    66.256028      61.119149    29.317331
55.757415
min       0.000000     0.000000       0.000000     0.000000
0.000000
25%       0.000000     0.000000       0.000000     0.000000
0.000000
50%       0.000000    25.000000       0.000000     0.000000
0.000000
75%     168.000000    68.000000       0.000000     0.000000
0.000000
max     857.000000   547.000000     552.000000   508.000000
480.000000

          PoolArea       MiscVal       MoSold       YrSold
SalePrice
count  1460.000000   1460.000000  1460.000000  1460.000000
1460.000000
mean      2.758904     43.489041     6.321918  2007.815753
180921.195890
std      40.177307    496.123024     2.703626     1.328095
79442.502883
min       0.000000      0.000000     1.000000  2006.000000
34900.000000
25%       0.000000      0.000000     5.000000  2007.000000
129975.000000
50%       0.000000      0.000000     6.000000  2008.000000
163000.000000
75%       0.000000      0.000000     8.000000  2009.000000
214000.000000
max     738.000000  15500.000000    12.000000  2010.000000
755000.000000

[8 rows x 38 columns]
```

```
df.isnull().sum()
```

```
Id                 0
MSSubClass         0
MSZoning           0
LotFrontage      259
LotArea            0
                ... 
MoSold             0
YrSold             0
SaleType           0
SaleCondition      0
SalePrice          0
Length: 81, dtype: int64
```

```
df[["1stFlrSF"]]
```

```
      1stFlrSF
0          856
1         1262
2          920
3          961
4         1145
...        ...
1455       953
1456      2073
1457      1188
1458      1078
1459      1256

[1460 rows x 1 columns]
```

```
df[["2ndFlrSF"]]
```

```
      2ndFlrSF
0          854
1            0
2          866
3          756
4         1053
...        ...
1455       694
1456         0
1457      1152
1458         0
1459         0

[1460 rows x 1 columns]
```

```
df[["OpenPorchSF"]]
```

```
      OpenPorchSF
0              61
```

```
1               0
2              42
3              35
4              84
...           ...
1455           40
1456            0
1457           60
1458            0
1459           68

[1460 rows x 1 columns]
```

```
df[["BsmtFinSF1"]]
```

```
      BsmtFinSF1
0            706
1            978
2            486
3            216
4            655
...          ...
1455           0
1456         790
1457         275
1458          49
1459         830

[1460 rows x 1 columns]
```

```
set(df['1stFlrSF'])
```

```
{334,
 372,
 438,
 480,
 483,
 495,
 520,
 525,
 526,
 536,
 546,
 551,
 561,
 572,
 575,
 576,
 581,
 596,
 600,
```

605,
612,
616,
624,
625,
626,
630,
649,
658,
660,
661,
663,
664,
672,
673,
676,
679,
680,
682,
684,
686,
689,
691,
693,
694,
696,
697,
698,
702,
703,
707,
708,
713,
716,
720,
725,
728,
729,
734,
735,
736,
738,
741,
742,
747,
750,
751,
752,
753,
754,

755,
756,
757,
760,
764,
765,
767,
768,
769,
770,
772,
773,
774,
778,
779,
780,
783,
784,
786,
788,
789,
790,
792,
793,
794,
796,
798,
799,
800,
802,
803,
804,
806,
807,
808,
810,
811,
812,
813,
814,
815,
816,
818,
820,
822,
824,
825,
827,
829,
831,

832,
833,
835,
838,
840,
841,
842,
845,
846,
847,
848,
849,
851,
854,
855,
856,
858,
859,
860,
861,
864,
865,
866,
869,
872,
874,
875,
876,
877,
879,
880,
882,
884,
885,
886,
887,
888,
889,
892,
893,
894,
896,
897,
899,
900,
901,
902,
904,
905,
907,

908,
909,
910,
912,
913,
914,
915,
916,
918,
920,
923,
924,
925,
926,
927,
928,
929,
930,
932,
933,
935,
936,
938,
939,
941,
943,
944,
948,
949,
950,
951,
952,
953,
954,
955,
956,
958,
959,
960,
961,
962,
963,
964,
965,
966,
968,
969,
970,
971,
972,

974,
975,
976,
977,
979,
980,
981,
983,
984,
985,
986,
988,
989,
990,
991,
992,
993,
996,
997,
998,
999,
1001,
1002,
1003,
1004,
1005,
1006,
1007,
1008,
1010,
1012,
1013,
1014,
1015,
1020,
1021,
1022,
1024,
1026,
1028,
1029,
1032,
1034,
1035,
1036,
1038,
1039,
1040,
1041,
1042,

1044,
1047,
1048,
1050,
1051,
1052,
1053,
1054,
1055,
1056,
1057,
1058,
1060,
1061,
1062,
1063,
1064,
1065,
1067,
1068,
1069,
1071,
1072,
1073,
1074,
1075,
1077,
1078,
1079,
1080,
1082,
1085,
1086,
1088,
1089,
1090,
1091,
1092,
1094,
1095,
1096,
1097,
1098,
1099,
1100,
1103,
1104,
1105,
1107,
1108,

1110,
1112,
1113,
1114,
1116,
1117,
1118,
1120,
1121,
1122,
1124,
1125,
1126,
1127,
1128,
1130,
1131,
1132,
1133,
1134,
1136,
1137,
1138,
1140,
1141,
1142,
1143,
1144,
1145,
1146,
1148,
1149,
1150,
1152,
1153,
1154,
1155,
1156,
1158,
1159,
1160,
1161,
1163,
1164,
1165,
1166,
1167,
1168,
1170,
1172,

1173,
1175,
1178,
1180,
1181,
1182,
1184,
1186,
1187,
1188,
1190,
1192,
1194,
1195,
1196,
1199,
1200,
1203,
1204,
1207,
1208,
1210,
1211,
1212,
1214,
1215,
1216,
1217,
1218,
1220,
1221,
1222,
1223,
1224,
1225,
1226,
1228,
1229,
1232,
1234,
1235,
1236,
1238,
1240,
1241,
1242,
1244,
1246,
1247,
1248,

1249,
1251,
1252,
1253,
1256,
1258,
1260,
1261,
1262,
1264,
1265,
1266,
1268,
1269,
1272,
1274,
1276,
1277,
1279,
1281,
1282,
1283,
1284,
1285,
1287,
1288,
1291,
1294,
1296,
1297,
1298,
1299,
1301,
1302,
1304,
1306,
1307,
1309,
1310,
1314,
1316,
1318,
1319,
1320,
1324,
1325,
1327,
1328,
1332,
1334,

1336,
1337,
1338,
1339,
1340,
1342,
1344,
1349,
1350,
1352,
1357,
1358,
1360,
1361,
1362,
1363,
1364,
1368,
1370,
1372,
1375,
1377,
1378,
1381,
1382,
1383,
1389,
1390,
1391,
1392,
1394,
1402,
1405,
1407,
1411,
1412,
1414,
1416,
1419,
1422,
1423,
1425,
1426,
1428,
1429,
1430,
1431,
1432,
1434,
1436,

1437,
1440,
1442,
1444,
1445,
1453,
1454,
1456,
1459,
1462,
1464,
1465,
1466,
1468,
1470,
1472,
1473,
1476,
1478,
1479,
1482,
1484,
1486,
1489,
1490,
1493,
1494,
1496,
1498,
1500,
1501,
1502,
1504,
1505,
1506,
1507,
1509,
1512,
1516,
1518,
1520,
1521,
1523,
1525,
1526,
1530,
1532,
1533,
1535,
1536,

1537,
1541,
1542,
1547,
1548,
1549,
1552,
1553,
1554,
1555,
1557,
1559,
1560,
1561,
1563,
1566,
1567,
1569,
1571,
1572,
1573,
1574,
1575,
1576,
1578,
1580,
1582,
1584,
1586,
1588,
1593,
1600,
1601,
1602,
1604,
1610,
1614,
1616,
1617,
1619,
1620,
1621,
1622,
1624,
1625,
1629,
1630,
1632,
1634,
1636,

1640,
1644,
1646,
1647,
1651,
1652,
1654,
1656,
1657,
1659,
1660,
1661,
1663,
1664,
1668,
1670,
1675,
1679,
1680,
1682,
1684,
1686,
1687,
1689,
1690,
1694,
1696,
1698,
1699,
1700,
1701,
1702,
1704,
1707,
1709,
1710,
1712,
1713,
1717,
1718,
1719,
1720,
1721,
1724,
1726,
1728,
1733,
1734,
1742,
1746,

1752,
1766,
1768,
1771,
1776,
1779,
1787,
1788,
1792,
1795,
1800,
1801,
1803,
1811,
1824,
1826,
1828,
1831,
1836,
1838,
1839,
1842,
1844,
1848,
1850,
1856,
1867,
1869,
1872,
1888,
1898,
1902,
1905,
1922,
1932,
1940,
1944,
1959,
1966,
1968,
1973,
1976,
1980,
1987,
1992,
2000,
2018,
2020,
2028,
2036,

```
 2042,
 2046,
 2053,
 2069,
 2073,
 2076,
 2084,
 2097,
 2110,
 2113,
 2117,
 2121,
 2129,
 2136,
 2156,
 2158,
 2196,
 2207,
 2217,
 2223,
 2234,
 2259,
 2364,
 2392,
 2402,
 2411,
 2444,
 2515,
 2524,
 2633,
 2898,
 3138,
 3228,
 4692}
```

```
X =
df[["1stFlrSF","2ndFlrSF","OpenPorchSF","PoolArea","BsmtFinSF1"]].valu
es # Independent variable
```

```
X
```

```
array([[ 856,  854,   61,    0,  706],
       [1262,    0,    0,    0,  978],
       [ 920,  866,   42,    0,  486],
       ...,
       [1188, 1152,   60,    0,  275],
       [1078,    0,    0,    0,   49],
       [1256,    0,   68,    0,  830]], dtype=int64)
```

```
Y = df[["SalePrice"]].values # Dependent Variable
```

```
Y
```

```
array([[208500],
       [181500],
       [223500],
       ...,
       [266500],
       [142125],
       [147500]], dtype=int64)
```

Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size =
0.2, random_state = 500)
```

```
X_train.shape
```

```
(1168, 5)
```

```
X_test.shape
```

```
(292, 5)
```

Feature Scaling

```
# Feature Scaling

from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
sc_Y = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
Y_train = sc_Y.fit_transform(Y_train)
Y_test = sc_Y.transform(Y_test)
```

```
X_train
```

```
array([[ 0.10235513, -0.78522929,  0.94757715, -0.07684505,
0.53193612],
       [ 0.60747084, -0.78522929,  0.85855721, -0.07684505, -
0.90231314],
       [ 1.40050251,  0.52565775,  3.36595228, -0.07684505,
1.90633593],
       ...,
       [ 1.15804697, -0.78522929, -0.69929179, -0.07684505,  0.3716251
],
       [-0.52651394, -0.78522929, -0.69929179, -0.07684505,
0.41651219],
       [ 0.49634538, -0.78522929, -0.69929179, -0.07684505, -
0.95361266]])
```

X_test

```
array([[-1.06951333,  0.9456955 , -0.18000879, -0.07684505, -
0.95361266],
       [-0.19061199,  1.21341187,  0.32443755, -0.07684505, -
0.09220812],
       [-0.72098349,  0.9595429 , -0.69929179, -0.07684505, -
0.12854528],
       ...,
       [-0.4128629 ,  2.11580066, -0.01680556, -0.07684505, -
0.95361266],
       [ 1.12016329, -0.78522929,  0.32443755, -0.07684505, -
0.60092842],
       [-0.73613696,  1.31034366,  0.60633404, -0.07684505,
0.39941234]])
```

Y_train

```
array([[-0.47654967],
       [ 0.04412298],
       [ 3.30843421],
       ...,
       [-0.52555415],
       [-0.75955057],
       [-0.36628958]])
```

Y_test

```
array([[-9.79900262e-02],
       [ 2.83019842e-01],
       [-4.33670746e-01],
       [-5.56181958e-01],
       [-9.67649141e-02],
       [ 6.24996610e-02],
       [ 6.07674553e-01],
       [-7.52199896e-01],
       [ 4.91288902e-01],
       [-2.49903929e-01],
       [-2.75214745e-01],
       [ 5.15791144e-01],
       [-5.90485097e-01],
       [-5.13303034e-01],
       [-7.22626718e-02],
       [ 4.07981278e-01],
       [ 1.23755267e-01],
       [-6.66442048e-01],
       [-3.81603481e-01],
       [-9.54343396e-01],
       [-3.41787337e-01],
       [ 4.90234277e-02],
       [ 5.28042265e-01],
```

```
[ 5.26817153e-01],
[ 2.52392039e-01],
[-7.38723663e-01],
[-2.74406171e-01],
[-1.49339273e+00],
[-5.99060882e-01],
[-4.15294064e-01],
[ 1.48257509e-01],
[-3.67344204e-02],
[ 2.70874183e+00],
[ 8.94350788e-01],
[-5.13303034e-01],
[-1.00334788e+00],
[-5.38859900e-02],
[ 4.60661099e-01],
[-4.82675231e-01],
[-9.42092275e-01],
[-2.65830386e-01],
[-2.92782853e-01],
[ 2.54413474e-01],
[-3.47912898e-01],
[-4.77604295e-02],
[-1.10070660e-02],
[ 5.28042265e-01],
[-7.83882324e-02],
[-1.10070660e-02],
[ 4.36264319e+00],
[ 3.13647645e-01],
[-7.95078821e-01],
[ 1.10384496e+00],
[-9.42092275e-01],
[-2.74406171e-01],
[-5.56181958e-01],
[ 1.89430863e-05],
[ 1.48257509e-01],
[-1.09016035e-01],
[-4.39796307e-01],
[-3.49909831e-01],
[-9.42092275e-01],
[-6.11312003e-01],
[ 2.57462975e-02],
[-5.01051913e-01],
[ 5.63741004e-02],
[-6.35814245e-01],
[ 5.09665584e-01],
[ 1.80093375e+00],
[-7.59550569e-01],
[-4.27545186e-01],
[-4.52047428e-01],
[-4.58172988e-01],
```

```
[-1.15036133e+00],
[-3.43012450e-01],
[-4.53272540e-01],
[-4.03042943e-01],
[ 2.21764236e-01],
[-5.37805276e-01],
[ 1.69189878e+00],
[-5.74558640e-01],
[-1.33412815e+00],
[ 1.20185393e+00],
[-1.70271641e-01],
[-2.68280610e-01],
[-1.21774250e+00],
[ 8.34320294e-01],
[ 5.63741004e-02],
[-1.09016035e-01],
[-4.64298549e-01],
[-6.48065367e-01],
[-9.05338911e-01],
[-4.64298549e-01],
[-1.04010124e+00],
[ 3.44275448e-01],
[-4.76549670e-01],
[-3.84666262e-01],
[ 6.75055719e-01],
[-1.64146081e-01],
[-4.88150538e-03],
[-6.35814245e-01],
[-2.49903929e-01],
[-3.29536216e-01],
[-3.29536216e-01],
[-8.45137930e-02],
[ 7.10583971e-01],
[-2.43778368e-01],
[-7.14221421e-01],
[-8.57389051e-02],
[-2.68280610e-01],
[ 4.60661099e-01],
[-1.13233994e+00],
[ 4.17782175e-01],
[ 1.53263420e+00],
[-7.64451018e-01],
[ 2.39633824e+00],
[ 1.16510057e+00],
[ 4.66786659e-01],
[ 6.86252216e-02],
[ 7.11809083e-01],
[-2.32581871e-02],
[-7.33823215e-01],
[-5.14528146e-01],
```

```
[-1.22386806e+00],
[-3.41787337e-01],
[-4.76549670e-01],
[-5.13915590e-01],
[ 8.09818052e-01],
[ 1.69189878e+00],
[ 9.92530245e-02],
[-2.43778368e-01],
[-3.29536216e-01],
[-3.29536216e-01],
[-8.25657619e-01],
[-7.95078821e-01],
[ 4.91288902e-01],
[ 2.79449968e+00],
[ 1.62835221e+00],
[-6.60316488e-01],
[-2.68280610e-01],
[ 1.96036882e-01],
[-1.33518278e-01],
[-2.19276126e-01],
[-5.13303034e-01],
[-4.70424110e-01],
[-9.35966714e-01],
[-8.45137930e-02],
[-2.68280610e-01],
[-7.70576578e-01],
[-5.31679715e-01],
[-1.49339273e+00],
[-1.08298017e+00],
[ 9.92530245e-02],
[-4.52047428e-01],
[ 1.03033823e+00],
[-7.27697654e-01],
[ 2.50353555e+00],
[ 1.32436514e+00],
[-1.00334788e+00],
[ 1.63064317e+00],
[-1.00334788e+00],
[ 1.75315438e+00],
[-5.75783752e-01],
[ 6.38302356e-01],
[-3.29536216e-01],
[-1.71326265e-02],
[-5.68433079e-01],
[ 9.06601909e-01],
[-5.74558640e-01],
[ 3.30799215e-01],
[-6.05186442e-01],
[-1.08910573e+00],
[-3.05033974e-01],
```

```
[ 6.50553477e-01],
[ 3.13647645e-01],
[-7.70576578e-01],
[-6.37039358e-01],
[ 2.34015357e-01],
[-1.45891910e-01],
[-6.66442048e-01],
[ 1.60508630e-01],
[-1.71326265e-02],
[-8.93590086e-01],
[ 4.23907735e-01],
[-5.77008864e-01],
[-6.35814245e-01],
[-5.13303034e-01],
[-1.45769399e-01],
[-3.05033974e-01],
[-9.78845638e-01],
[-2.69566978e-01],
[-6.11312003e-01],
[-8.45137930e-02],
[-2.31527247e-01],
[ 2.57462975e-02],
[-1.09016035e-01],
[-7.09320972e-01],
[-5.01051913e-01],
[ 2.58517600e-01],
[-4.64298549e-01],
[-1.34025371e+00],
[ 1.65024496e+00],
[ 9.44580385e-01],
[-2.68893166e-01],
[ 6.26051235e-01],
[-9.78845638e-01],
[ 5.89297871e-01],
[ 9.07827021e-01],
[-6.29688685e-01],
[-1.18711470e+00],
[ 8.70019034e-02],
[-9.64144293e-01],
[ 1.01270887e+00],
[-1.23611918e+00],
[ 2.57292488e-01],
[ 1.42131949e-01],
[-8.56334426e-01],
[-4.52047428e-01],
[ 8.33265670e-02],
[-2.93837477e-02],
[ 4.97414462e-01],
[ 1.07321716e+00],
[ 6.99557962e-01],
```

```
[-1.24959542e+00],
[-1.33518278e-01],
[-3.78540701e-01],
[-6.48065367e-01],
[ 1.02911312e+00],
[-5.50056397e-01],
[ 7.07962231e-01],
[-9.11464472e-01],
[-2.32581871e-02],
[-8.01204381e-01],
[ 3.48668803e+00],
[-5.25554155e-01],
[-8.80836669e-01],
[-2.32581871e-02],
[ 1.85010873e-01],
[ 1.25085841e+00],
[ 3.79974187e-02],
[-4.29995410e-01],
[ 4.05531054e-01],
[-4.76549670e-01],
[-1.17486358e+00],
[-2.25401686e-01],
[ 1.12834720e+00],
[-1.71496753e-01],
[ 2.17091761e+00],
[ 5.89297871e-01],
[ 6.87306840e-01],
[ 1.15284944e+00],
[-3.78540701e-01],
[-2.92782853e-01],
[ 3.91960590e+00],
[-3.72415140e-01],
[-8.19581063e-01],
[ 3.44275448e-01],
[ 3.67723066e-02],
[ 2.58517600e-01],
[-4.22644737e-01],
[-5.25554155e-01],
[-5.13303034e-01],
[-6.11312003e-01],
[ 1.23755267e-01],
[ 2.21764236e-01],
[-5.37805276e-01],
[ 2.42696605e+00],
[ 6.21150786e-01],
[-3.94467158e-01],
[-1.15036133e+00],
[-1.55464833e+00],
[-7.39948775e-01],
[-7.76702139e-01],
```

```
       [ 1.26310954e+00],
       [-1.45769399e-01],
       [-2.07025005e-01],
       [-4.21419625e-01],
       [-5.13303034e-01],
       [-4.15294064e-01],
       [-3.29536216e-01],
       [-6.05186442e-01],
       [-7.76702139e-01],
       [-7.70576578e-01],
       [-9.67649141e-02],
       [ 6.24996610e-02],
       [ 1.13202254e+00],
       [-4.76549670e-01],
       [-3.99367607e-01],
       [-1.00947344e+00],
       [ 2.21764236e-01],
       [ 1.36006388e-01],
       [ 1.36006388e-01]])
```

Multiple Regression Model

```
from sklearn import linear_model
from sklearn.linear_model import Lasso
regr = linear_model.LinearRegression()
regr.fit(X_train, Y_train)#training func question + answers
# The coefficients
print ('Intercept: ',regr.intercept_)
print ('Coefficient : ',regr.coef_)
```

```
Intercept:  [-8.10072062e-17]
Coefficient :  [[ 0.60519595  0.44512289  0.06762205 -0.05752323
0.16796952]]
```

```
regr.intercept_
```

```
array([-8.10072062e-17])
```

```
regr.coef_
```

```
array([[ 0.60519595,  0.44512289,  0.06762205, -0.05752323,
0.16796952]])
```

```
regr.coef_[0][0]
```

```
0.6051959545443171
```

```
regr.coef_[0][1]
```

```
0.44512289318862164
```

```
regr.coef_[0][3]
```

```
-0.05752322506300601
```

```python
y_pred = regr.predict(X_test)
```

```python
y_pred
```

```
array([[-0.39424447],
       [ 0.43563115],
       [-0.07368063],
       [-0.09273092],
       [-0.36267396],
       [ 0.05064675],
       [ 0.20328702],
       [-0.8909513 ],
       [ 0.44851715],
       [ 0.04391717],
       [-0.29435498],
       [ 0.02776111],
       [-0.62409479],
       [-0.21444144],
       [-0.30049126],
       [-0.32029542],
       [ 0.07722696],
       [-0.30865709],
       [-0.17249878],
       [ 0.22677364],
       [-0.06008962],
       [-0.11402423],
       [ 0.89433856],
       [ 0.22671411],
       [-0.2201243 ],
       [-0.49964672],
       [ 1.11553858],
       [-0.39585851],
       [-0.46073544],
       [-0.25803403],
       [ 0.26731555],
       [-0.35702107],
       [ 1.6646843 ],
       [ 0.36677717],
       [-0.45637249],
       [-0.97169607],
       [-0.23762948],
       [ 0.06964113],
       [-0.706152  ],
       [-0.93373737],
       [-0.57429101],
       [-0.18820504],
       [-0.2201243 ],
       [ 0.27443221],
       [-0.49368059],
```

```
[-0.19819601],
[ 0.44379244],
[-0.26581468],
[-0.04162286],
[ 2.41381306],
[ 0.04899152],
[-0.16488018],
[ 1.98164081],
[-0.75123254],
[-0.2860274 ],
[ 0.24255629],
[ 0.16162348],
[-0.35757792],
[-0.33531867],
[-0.21049214],
[-0.08543071],
[-0.31536566],
[-0.61637194],
[ 0.67924379],
[-0.51443501],
[-0.32675775],
[ 0.37725886],
[ 0.59308954],
[ 0.77566353],
[-0.61820833],
[-0.46732462],
[-0.33917436],
[-0.81236477],
[-0.9612609 ],
[-0.34263428],
[-0.57137771],
[-0.60408211],
[-0.02313707],
[-0.80598496],
[ 2.343037  ],
[-0.51389148],
[-1.17843512],
[ 0.59308929],
[ 0.14954728],
[ 1.25717844],
[ 0.10560884],
[ 1.32354575],
[-0.0231399 ],
[ 0.17137702],
[-0.09975317],
[-0.82649847],
[-0.41658061],
[-0.20159653],
[-0.95292784],
[ 0.44542798],
```

```
[-0.74045428],
[-0.56557265],
[ 0.80593344],
[-0.05871606],
[-0.06612426],
[-0.17198534],
[-0.20486258],
[-0.49452428],
[ 0.19415472],
[-0.25643087],
[ 0.07191543],
[-0.33613923],
[-0.62189038],
[-0.32407706],
[ 0.10955637],
[ 0.10709579],
[-0.48493283],
[ 0.06914469],
[ 0.79174333],
[-0.60996888],
[ 0.83557209],
[ 0.92860065],
[ 0.51944553],
[ 0.37254083],
[ 0.77134671],
[-0.14344445],
[-0.60803221],
[-0.1727019 ],
[-1.06074294],
[-0.25845798],
[ 0.14827678],
[-0.39225051],
[ 0.55704345],
[ 0.75717653],
[ 0.56203734],
[ 0.05282437],
[ 0.09608483],
[-0.3233979 ],
[-0.43952115],
[-0.51835199],
[ 0.55588536],
[ 2.38836419],
[ 0.5734452 ],
[-0.63315472],
[-0.28944078],
[ 0.47846466],
[ 0.0454706 ],
[-0.04010479],
[-0.28958818],
[-0.33771609],
```

```
[-1.07449917],
[-0.34975182],
[ 0.18119894],
[-0.93419024],
[-0.63671654],
[-1.1234102 ],
[-0.75657742],
[ 0.44216756],
[-0.58670529],
[ 0.31004663],
[-0.35530514],
[ 1.43536464],
[ 1.19561933],
[-1.04624946],
[ 1.5125534 ],
[ 0.19718782],
[ 1.50177389],
[-0.74416217],
[ 0.43008988],
[-0.76636746],
[-0.10802081],
[-0.47813759],
[ 0.58217396],
[-0.90227556],
[ 0.33182492],
[-0.47570051],
[-0.51487281],
[-0.27914205],
[ 0.5234929 ],
[ 0.09877503],
[ 0.30937116],
[-0.31406017],
[-0.08164559],
[-0.32466683],
[-0.80586083],
[ 0.12764177],
[ 0.00780896],
[ 0.0067488 ],
[ 0.37087928],
[-0.85444471],
[-0.7276939 ],
[-0.80410705],
[-0.1151538 ],
[-0.03591672],
[-0.18130247],
[-0.39509065],
[-0.9285049 ],
[ 0.10398306],
[ 0.25502138],
[ 0.30977458],
```

```
[-0.06224381],
[-0.00263775],
[-0.79979867],
[ 0.03151065],
[-0.17740457],
[-1.2456878 ],
[ 1.15126287],
[ 0.4952321 ],
[-0.50100753],
[ 0.55449689],
[-0.7106943 ],
[ 0.4057285 ],
[ 0.7337886 ],
[-0.22795365],
[-0.66531054],
[ 0.82197778],
[-0.99807567],
[ 0.46324862],
[-1.20552949],
[ 0.11156308],
[ 0.67083072],
[ 0.26827984],
[ 0.31410351],
[-0.09833141],
[ 0.16121226],
[ 0.23556011],
[ 0.13869097],
[ 0.09115838],
[-1.20441911],
[-0.24428647],
[-0.64015975],
[-0.33229225],
[ 0.10883714],
[-0.63099913],
[ 1.15044974],
[-1.08214152],
[-0.31630063],
[-0.784873  ],
[ 2.27901352],
[-0.73078841],
[-0.7408098 ],
[-0.03945174],
[ 0.01712556],
[ 0.63297054],
[-0.04344139],
[-0.52943309],
[ 0.13894206],
[-0.08600372],
[-1.26994773],
[-0.61675165],
```

```
       [ 0.55179153],
       [-0.04485907],
       [ 2.06094397],
       [ 0.52814623],
       [ 0.50581097],
       [ 1.189303  ],
       [-0.76845706],
       [-0.25917978],
       [ 1.81220882],
       [-0.70311425],
       [-0.81007387],
       [ 1.10596693],
       [-0.17181428],
       [ 0.2665812 ],
       [-0.74482918],
       [-0.7929351 ],
       [-0.57948478],
       [-0.79241117],
       [-0.0770204 ],
       [ 1.53639455],
       [ 0.243747  ],
       [ 1.49016372],
       [ 0.0701655 ],
       [-0.08305935],
       [-0.85272071],
       [-1.12952408],
       [ 0.24890305],
       [-0.77387983],
       [ 0.62866392],
       [-0.60896776],
       [ 0.14668456],
       [-0.06020836],
       [-0.24001964],
       [-0.51477893],
       [-0.28171621],
       [-0.48217394],
       [-0.52132441],
       [-0.44227057],
       [-0.43681607],
       [ 0.20806301],
       [ 0.88397146],
       [-0.32906096],
       [-0.79979867],
       [-0.94305075],
       [ 0.53503444],
       [ 0.25381661],
       [ 0.25026787]])
```

```
from sklearn.metrics import r2_score
```

```
print(f"R2 Score : {r2_score(Y_test,y_pred)*100} % ")
```

```
R2 Score : 69.52190340964152 %
```

```
print(f"Mean absolute error: {np.mean(np.absolute(y_pred - Y_test))}
")#pred - actual
```

```
Mean absolute error: 0.3362259554459409
```

```
print("Residual sum of squares (MSE): %.2f" % np.mean((y_pred -
Y_test) **2))
```

```
Residual sum of squares (MSE): 0.22
```

```
accuracy = (f"accuracy : {r2_score(Y_test,y_pred)*100} % ")
```

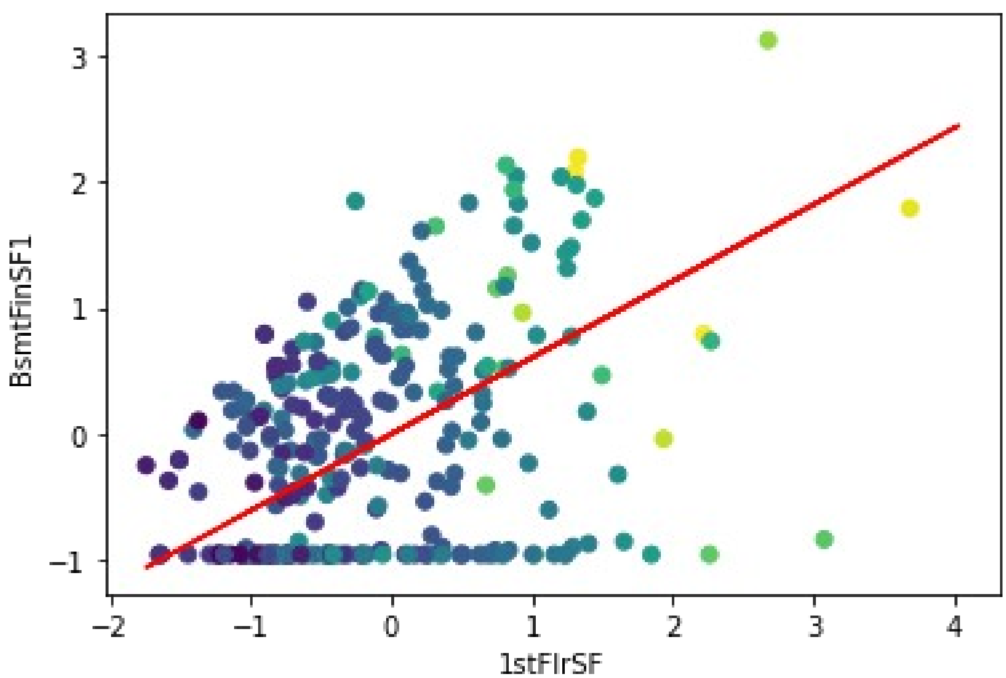
```
accuracy
```

```
'accuracy : 69.52190340964152 % '
```

Plots

```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,0],X_test[:,-1], c = y_pred)
plt.plot(X_test, regr.coef_[0][0]*X_test + regr.intercept_[0], 'r') #
y = mx+c
plt.xlabel("1stFlrSF")
plt.ylabel("BsmtFinSF1")
```

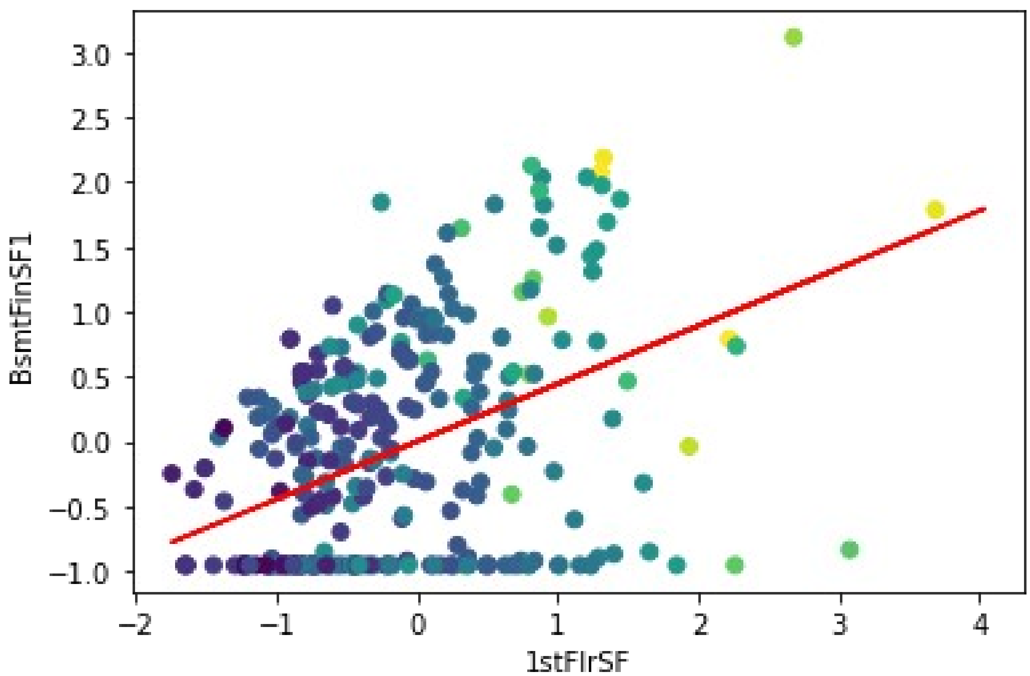
```
Text(0, 0.5, 'BsmtFinSF1')
```




```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,0],X_test[:,-1], c = y_pred)
```

```
plt.plot(X_test, regr.coef_[0][1]*X_test + regr.intercept_[0], 'r') #y
= mx+c
plt.xlabel("1stFlrSF")
plt.ylabel("BsmtFinSF1")
```

```
Text(0, 0.5, 'BsmtFinSF1')
```



Apply Lasso Regression to cover overfitting and undefitting Problem

```
from sklearn.linear_model import Lasso
```

```
L = Lasso(alpha = 1)
```

```
L.fit(X_train,Y_train)
```

```
Lasso(alpha=1)
```

```
ypred3 = L.predict(X_test)
```

```
from sklearn.metrics import r2_score
print("R2-score",r2_score(Y_test,ypred3))
print(f"Mean absolute error: {np.mean(np.absolute(ypred3 - Y_test))}
")# pred - actual
```

```
R2-score -0.004908518792945626
Mean absolute error: 0.6243935831081134
```