Dataset Preparation

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
import warnings
warnings.filterwarnings("ignore")
```

Importing Target variable

In [3]:

```
price=pd.read_csv('C:/Users/artia/OneDrive/Documents/CSUSHPISA.csv')
price
```

Out[3]:

	DATE	CSUSHPISA
0	1987-01-01	63.965
1	1987-02-01	64.424
2	1987-03-01	64.736
3	1987-04-01	65.132
4	1987-05-01	65.563
...	...	...
436	2023-05-01	302.566
437	2023-06-01	304.593
438	2023-07-01	306.767
439	2023-08-01	309.155
440	2023-09-01	311.175

441 rows × 2 columns

Importing other variables

In [4]:

```
unemp_rate=pd.read_csv("C:/Users/artia/Downloads/UNRATE.csv")
unemp_rate
```

Out[4]:

	DATE	UNRATE
0	1948-01-01	3.4
1	1948-02-01	3.8
2	1948-03-01	4.0
3	1948-04-01	3.9
4	1948-05-01	3.5

	DATE	UNRATE
...	...	...
906	2023-07-01	3.5
907	2023-08-01	3.8
908	2023-09-01	3.8
909	2023-10-01	3.9
910	2023-11-01	3.7

911 rows × 2 columns

In [155…

```
Interest_rate=pd.read_csv("C:/Users/artia/Downloads/INTDSRUSM193N.csv")   # Interest_Rate
Interest_rate
```

Out[155…

	DATE	INTDSRUSM193N
0	1950-01-01	1.50
1	1950-02-01	1.50
2	1950-03-01	1.50
3	1950-04-01	1.50
4	1950-05-01	1.50
...	...	...
855	2021-04-01	0.25
856	2021-05-01	0.25
857	2021-06-01	0.25
858	2021-07-01	0.25
859	2021-08-01	0.25

860 rows × 2 columns

In [156…

```
Active_pop=pd.read_csv("C:/Users/artia/Downloads/LFACTTTTUSM657S.csv")   # Active_Populati
Active_pop
```

Out[156…

	DATE	LFACTTTTUSM657S
0	1960-01-01	-0.046381
1	1960-02-01	-0.018851
2	1960-03-01	-0.797691
3	1960-04-01	1.725171
4	1960-05-01	0.067549
...	...	...
761	2023-06-01	0.079728
762	2023-07-01	0.091045

	DATE	LFACTTTTUSM657S
763	2023-08-01	0.440447
764	2023-09-01	0.053623
765	2023-10-01	-0.119693

766 rows × 2 columns

In [157…

```
under_constr=pd.read_csv("C:/Users/artia/Downloads/UNDCONTSA.csv") # under_construction
under_constr
```

Out[157…

	DATE	UNDCONTSA
0	1970-01-01	889.0
1	1970-02-01	888.0
2	1970-03-01	890.0
3	1970-04-01	891.0
4	1970-05-01	883.0
...	...	...
641	2023-06-01	1692.0
642	2023-07-01	1697.0
643	2023-08-01	1694.0
644	2023-09-01	1676.0
645	2023-10-01	1674.0

646 rows × 2 columns

In [6]:

```
complete_constr=pd.read_csv("C:/Users/artia/Downloads/COMPUTSA.csv") # Comlete_Constructi
complete_constr
```

Out[6]:

	DATE	COMPUTSA
0	1968-01-01	1257.0
1	1968-02-01	1174.0
2	1968-03-01	1323.0
3	1968-04-01	1328.0
4	1968-05-01	1367.0
...	...	...
665	2023-06-01	1492.0
666	2023-07-01	1334.0
667	2023-08-01	1370.0
668	2023-09-01	1478.0
669	2023-10-01	1410.0

670 rows × 2 columns

In [158…

```
Total_constr=pd.read_csv("C:/Users/artia/Downloads/TTLCONS (1).csv") # Total_construction
Total_constr
```

Out[158…

	DATE	TTLCONS
0	1993-01-01	458080.0
1	1993-02-01	462967.0
2	1993-03-01	458399.0
3	1993-04-01	469425.0
4	1993-05-01	468998.0
...	...	...
365	2023-06-01	1956226.0
366	2023-07-01	1969005.0
367	2023-08-01	2010143.0
368	2023-09-01	2014718.0
369	2023-10-01	2027072.0

370 rows × 2 columns

In [159…

```
Privtly_owned_house=pd.read_csv("C:/Users/artia/Downloads/PERMIT.csv")
Privtly_owned_house
```

Out[159…

	DATE	PERMIT
0	1960-01-01	1092.0
1	1960-02-01	1088.0
2	1960-03-01	955.0
3	1960-04-01	1016.0
4	1960-05-01	1052.0
...	...	...
761	2023-06-01	1441.0
762	2023-07-01	1443.0
763	2023-08-01	1541.0
764	2023-09-01	1471.0
765	2023-10-01	1498.0

766 rows × 2 columns

In [19]:

```
total_constn=pd.read_csv("C:/Users/artia/Downloads/TTLCONS.csv")
total_constn
```

Out[19]:

	DATE	TTLCONS
0	1993-01-01	458080.0
1	1993-02-01	462967.0
2	1993-03-01	458399.0
3	1993-04-01	469425.0
4	1993-05-01	468998.0
...	...	...
365	2023-06-01	1956226.0
366	2023-07-01	1969005.0
367	2023-08-01	2010143.0
368	2023-09-01	2014718.0
369	2023-10-01	2027072.0

370 rows × 2 columns

In [160…

```
def date_col(arr):
    arr = arr.str.replace(" ","-")
    for i in range(len(arr)):
        arr[i]=dt.datetime.strptime(arr[i], '%d-%m-%y').strftime('%Y-%m-%d')
    return arr
```

In [161…

```
file=pd.merge(price,unemp_rate,on='DATE',how='inner')
file
```

Out[161…

	DATE	CSUSHPISA	UNRATE
0	1987-01-01	63.965	6.6
1	1987-02-01	64.424	6.6
2	1987-03-01	64.736	6.6
3	1987-04-01	65.132	6.3
4	1987-05-01	65.563	6.3
...	...	...	...
436	2023-05-01	302.566	3.7
437	2023-06-01	304.593	3.6
438	2023-07-01	306.767	3.5
439	2023-08-01	309.155	3.8
440	2023-09-01	311.175	3.8

441 rows × 3 columns

In [162…

```
file02=pd.merge(file,Interest_rate,on='DATE',how='inner')
file02
```

Out[162...

	DATE	CSUSHPISA	UNRATE	INTDSRUSM193N
0	1987-01-01	63.965	6.6	5.50
1	1987-02-01	64.424	6.6	5.50
2	1987-03-01	64.736	6.6	5.50
3	1987-04-01	65.132	6.3	5.50
4	1987-05-01	65.563	6.3	5.50
...	...	...	...	...
411	2021-04-01	249.070	6.1	0.25
412	2021-05-01	253.407	5.8	0.25
413	2021-06-01	258.358	5.9	0.25
414	2021-07-01	262.820	5.4	0.25
415	2021-08-01	266.845	5.2	0.25

416 rows × 4 columns

In [163...

```
file03=pd.merge(file02,Active_pop,on='DATE',how='inner')
file03
```

Out[163...

	DATE	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S
0	1987-01-01	63.965	6.6	5.50	0.197284
1	1987-02-01	64.424	6.6	5.50	0.233077
2	1987-03-01	64.736	6.6	5.50	0.124242
3	1987-04-01	65.132	6.3	5.50	0.055337
4	1987-05-01	65.563	6.3	5.50	0.563116
...	...	...	...	...	...
411	2021-04-01	249.070	6.1	0.25	0.277322
412	2021-05-01	253.407	5.8	0.25	-0.104407
413	2021-06-01	258.358	5.9	0.25	0.281199
414	2021-07-01	262.820	5.4	0.25	0.182391
415	2021-08-01	266.845	5.2	0.25	0.030343

416 rows × 5 columns

In [164...

```
file04=pd.merge(file03,under_constr,on='DATE',how='inner')
file04
```

Out[164...

	DATE	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA
0	1987-01-01	63.965	6.6	5.50	0.197284	1090.0
1	1987-02-01	64.424	6.6	5.50	0.233077	1096.0
2	1987-03-01	64.736	6.6	5.50	0.124242	1084.0

	DATE	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA
3	1987-04-01	65.132	6.3	5.50	0.055337	1079.0
4	1987-05-01	65.563	6.3	5.50	0.563116	1070.0
...	...	...	...	...	...	...
411	2021-04-01	249.070	6.1	0.25	0.277322	1320.0
412	2021-05-01	253.407	5.8	0.25	-0.104407	1338.0
413	2021-06-01	258.358	5.9	0.25	0.281199	1372.0
414	2021-07-01	262.820	5.4	0.25	0.182391	1387.0
415	2021-08-01	266.845	5.2	0.25	0.030343	1412.0

416 rows × 6 columns

In [165…

```
files=pd.merge(file04,complete_constr,on='DATE',how='inner')
files
```

Out[165…

	DATE	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA	COMPUTSA
0	1987-01-01	63.965	6.6	5.50	0.197284	1090.0	1862.0
1	1987-02-01	64.424	6.6	5.50	0.233077	1096.0	1771.0
2	1987-03-01	64.736	6.6	5.50	0.124242	1084.0	1694.0
3	1987-04-01	65.132	6.3	5.50	0.055337	1079.0	1735.0
4	1987-05-01	65.563	6.3	5.50	0.563116	1070.0	1713.0
...	...	...	...	...	...	...	...
411	2021-04-01	249.070	6.1	0.25	0.277322	1320.0	1438.0
412	2021-05-01	253.407	5.8	0.25	-0.104407	1338.0	1337.0
413	2021-06-01	258.358	5.9	0.25	0.281199	1372.0	1298.0
414	2021-07-01	262.820	5.4	0.25	0.182391	1387.0	1361.0
415	2021-08-01	266.845	5.2	0.25	0.030343	1412.0	1312.0

416 rows × 7 columns

In [166…

```
df_files=pd.merge(files,Total_constr,on='DATE',how='inner')
df_files
```

Out[166…

	DATE	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA	COMPUTSA	TTLCONS
0	1993-01-01	76.784	7.3	3.00	-0.119794	636.0	1135.0	458080.0
1	1993-02-01	76.837	7.1	3.00	0.045171	640.0	1236.0	462967.0
2	1993-03-01	76.867	7.0	3.00	0.108985	633.0	1105.0	458399.0
3	1993-04-01	76.936	7.1	3.00	-0.010887	636.0	1216.0	469425.0

	DATE	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA	COMPUTSA	TTLCONS
4	1993-05-01	77.037	7.1	3.00	0.528837	648.0	1111.0	468998.0
...	...	...	...	...	...	...	...	...
339	2021-04-01	249.070	6.1	0.25	0.277322	1320.0	1438.0	1615638.0
340	2021-05-01	253.407	5.8	0.25	-0.104407	1338.0	1337.0	1628281.0
341	2021-06-01	258.358	5.9	0.25	0.281199	1372.0	1298.0	1639779.0
342	2021-07-01	262.820	5.4	0.25	0.182391	1387.0	1361.0	1658346.0
343	2021-08-01	266.845	5.2	0.25	0.030343	1412.0	1312.0	1666756.0

344 rows × 8 columns

In [167…

```
df_files.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 344 entries, 0 to 343
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   DATE             344 non-null    object 
 1   CSUSHPISA        344 non-null    float64
 2   UNRATE           344 non-null    float64
 3   INTDSRUSM193N    344 non-null    float64
 4   LFACTTTTUSM657S  344 non-null    float64
 5   UNDCONTSA        344 non-null    float64
 6   COMPUTSA         344 non-null    float64
 7   TTLCONS          344 non-null    float64
dtypes: float64(7), object(1)
memory usage: 24.2+ KB
```

In [168…

```
df_files.DATE = pd.to_datetime(df_files.DATE)
```

In [169…

```
df_files.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 344 entries, 0 to 343
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype         
---  ------           --------------  -----         
 0   DATE             344 non-null    datetime64[ns]
 1   CSUSHPISA        344 non-null    float64       
 2   UNRATE           344 non-null    float64       
 3   INTDSRUSM193N    344 non-null    float64       
 4   LFACTTTTUSM657S  344 non-null    float64       
 5   UNDCONTSA        344 non-null    float64       
 6   COMPUTSA         344 non-null    float64       
 7   TTLCONS          344 non-null    float64       
dtypes: datetime64[ns](1), float64(7)
memory usage: 24.2 KB
```

In dataset, date was in object datatype so converted it into datetime Dtype. Other key factors and target is in float dtype.

Checking for null values

In [170…

```
df_files.isnull()
```

Out[170…

	DATE	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA	COMPUTSA	TTLCONS
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
339	False	False	False	False	False	False	False	False
340	False	False	False	False	False	False	False	False
341	False	False	False	False	False	False	False	False
342	False	False	False	False	False	False	False	False
343	False	False	False	False	False	False	False	False

344 rows × 8 columns

so we have data with 344 rows and 8 columns.

In [171…

```
df_files.isnull().sum()
```

Out[171…

```
DATE                0
CSUSHPISA           0
UNRATE              0
INTDSRUSM193N       0
LFACTTTTUSM657S     0
UNDCONTSA           0
COMPUTSA            0
TTLCONS             0
dtype: int64
```

There is no null value in the dataset, so we can procceed.

In [172…

```
df_files['year']=pd.DatetimeIndex(df_files['DATE']).year
df_files
```

Out[172…

	DATE	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA	COMPUTSA	TTLCONS	yea
0	1993-01-01	76.784	7.3	3.00	-0.119794	636.0	1135.0	458080.0	199
1	1993-02-01	76.837	7.1	3.00	0.045171	640.0	1236.0	462967.0	199
2	1993-03-01	76.867	7.0	3.00	0.108985	633.0	1105.0	458399.0	199

	DATE	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA	COMPUTSA	TTLCONS	yea
3	1993-04-01	76.936	7.1	3.00	-0.010887	636.0	1216.0	469425.0	199
4	1993-05-01	77.037	7.1	3.00	0.528837	648.0	1111.0	468998.0	199
...	...	...	...	...	...	...	...	...	.
339	2021-04-01	249.070	6.1	0.25	0.277322	1320.0	1438.0	1615638.0	202
340	2021-05-01	253.407	5.8	0.25	-0.104407	1338.0	1337.0	1628281.0	202
341	2021-06-01	258.358	5.9	0.25	0.281199	1372.0	1298.0	1639779.0	202
342	2021-07-01	262.820	5.4	0.25	0.182391	1387.0	1361.0	1658346.0	202
343	2021-08-01	266.845	5.2	0.25	0.030343	1412.0	1312.0	1666756.0	202

344 rows × 9 columns

For the better understanding i have splitted date column in year column.

Calculating Statistical Values

In [173...

```
df_files.describe()
```

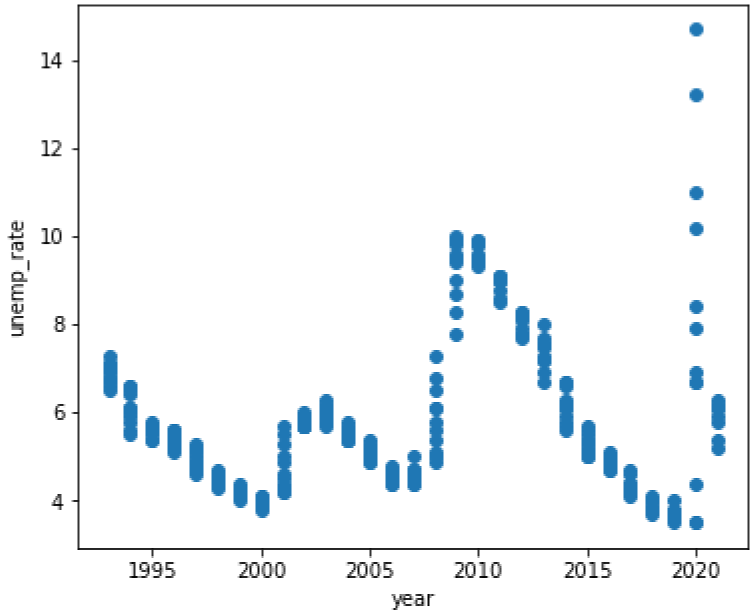
Out[173...

	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA	COMPUTSA	TTLCONS	
count	344.000000	344.000000	344.000000	344.000000	344.000000	344.000000	3.440000e+02	34
mean	144.957683	5.836337	2.721192	0.066925	930.572674	1281.750000	9.532798e+05	20
std	45.429680	1.767949	1.935252	0.319807	258.445105	383.856814	2.842743e+05	
min	76.784000	3.500000	0.250000	-3.890307	414.000000	520.000000	4.580800e+05	19
25%	101.930000	4.600000	0.750000	-0.041459	763.500000	1020.000000	7.737975e+05	20
50%	147.111500	5.400000	2.250000	0.091572	982.000000	1313.500000	8.952795e+05	20
75%	178.314750	6.600000	4.822500	0.198490	1123.250000	1560.000000	1.155674e+06	20
max	266.845000	14.700000	6.250000	1.490972	1424.000000	2245.000000	1.666756e+06	20

Data Visualization

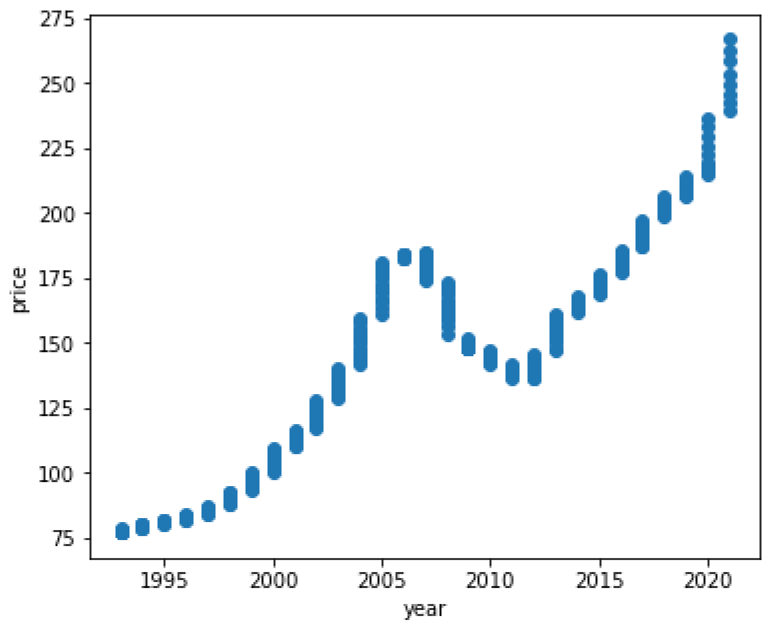
In [176...

```
plt.figure(figsize=(6,5))
plt.scatter(df_files.year,df_files.UNRATE)
plt.xlabel('year')
plt.ylabel('unemp_rate')
plt.show()
```



In [177…

```
plt.figure(figsize=(6,5))
plt.scatter(df_files.year,df_files.CSUSHPISA)
plt.xlabel('year')
plt.ylabel('price')
plt.show()
```

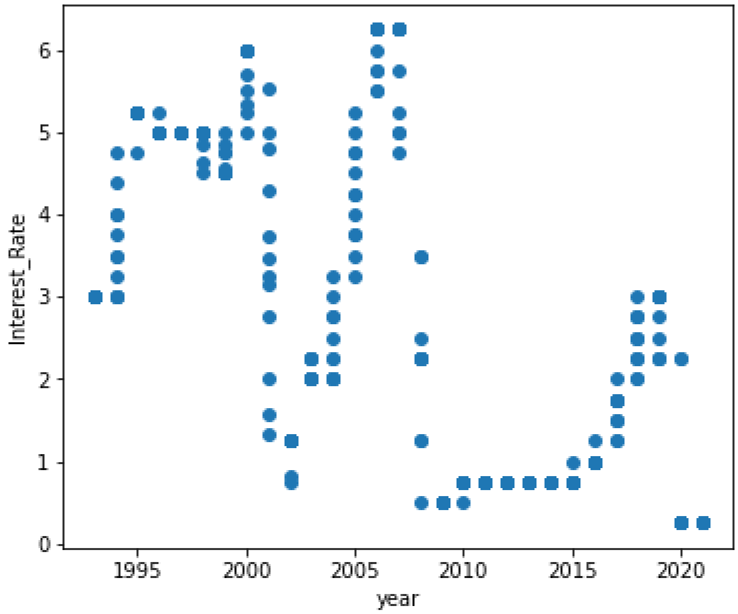


As we can see from both the graphs,

1. The unemployment rate for year 2007-2012 was high, same year the house prices were started decreasing
2. Again the unemployment rate for year 2020 was too high and then started to decerase in year 2023 it became lowest. and house prices are gradually increasing from 2012 in 2023 it is high.

In [178…

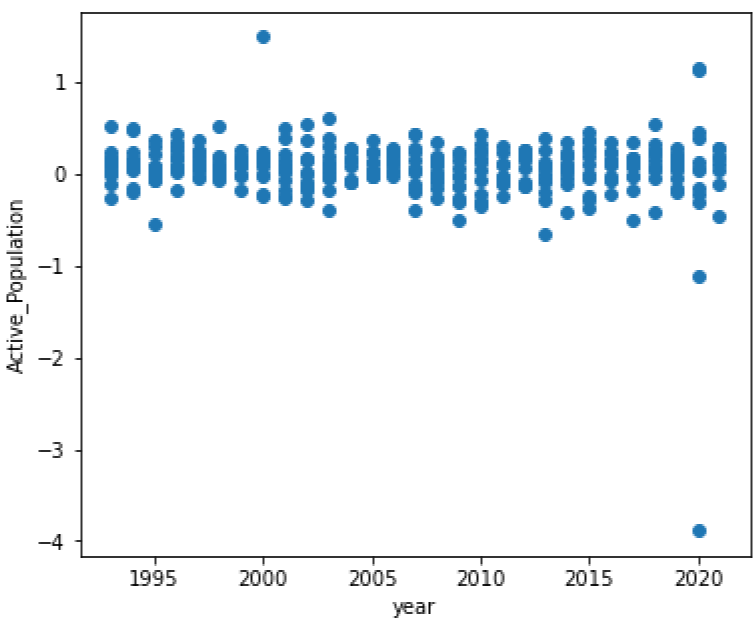
```
plt.figure(figsize=(6,5))
plt.scatter(df_files.year,df_files.INTDSRUSM193N)
plt.xlabel('year')
plt.ylabel('Interest_Rate')
plt.show()
```



As we can see, the interest rate was high in the year 2006 and 2007 followed by 2005. In 2020 and 2021 it was low.

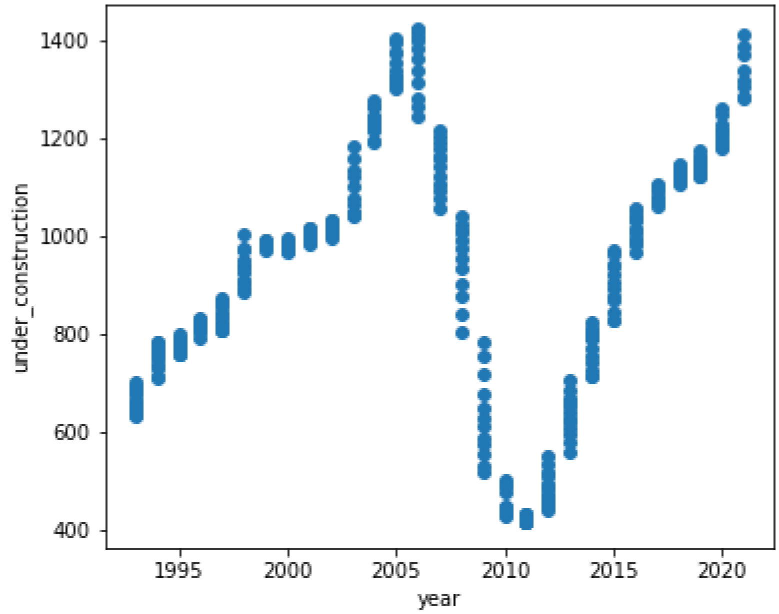
In [179…

```
plt.figure(figsize=(6,5))
plt.scatter(df_files.year,df_files.LFACTTTTUSM657S)
plt.xlabel('year')
plt.ylabel('Active_Population')
plt.show()
```



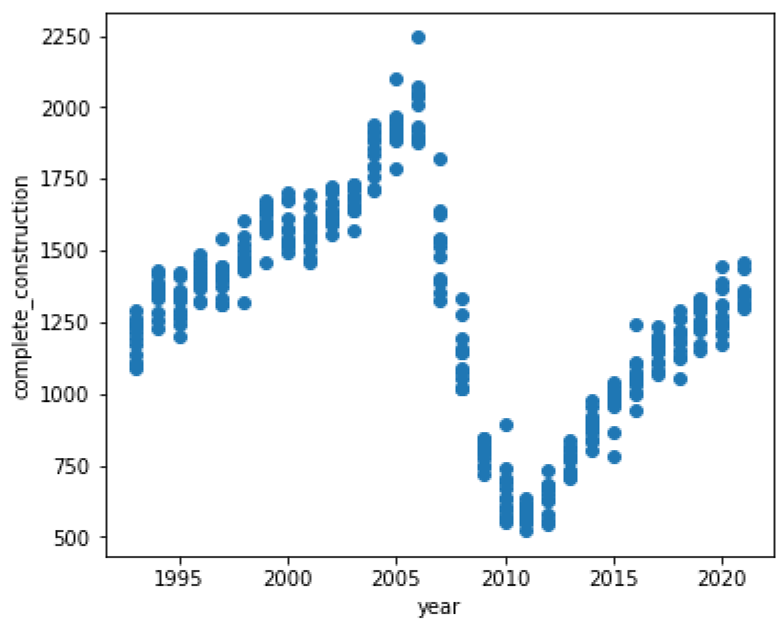
In [181…

```
plt.figure(figsize=(6,5))
plt.scatter(df_files.year,df_files.UNDCONTSA)
plt.xlabel('year')
plt.ylabel('under_construction')
plt.show()
```



In [182…

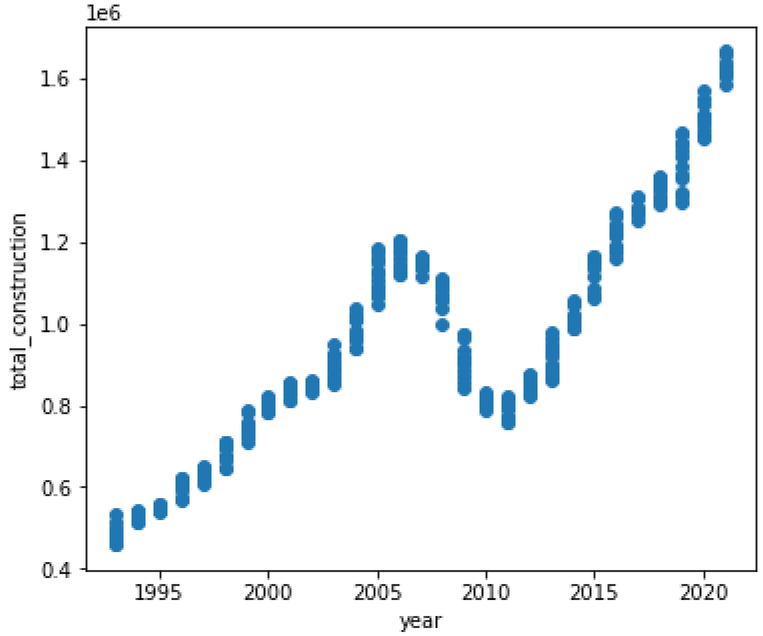
```
plt.figure(figsize=(6,5))
plt.scatter(df_files.year,df_files.COMPUTSA)
plt.xlabel('year')
plt.ylabel('complete_construction')
plt.show()
```



As we can see the construction was almost complete in the same year when it has been statred.

In [183…

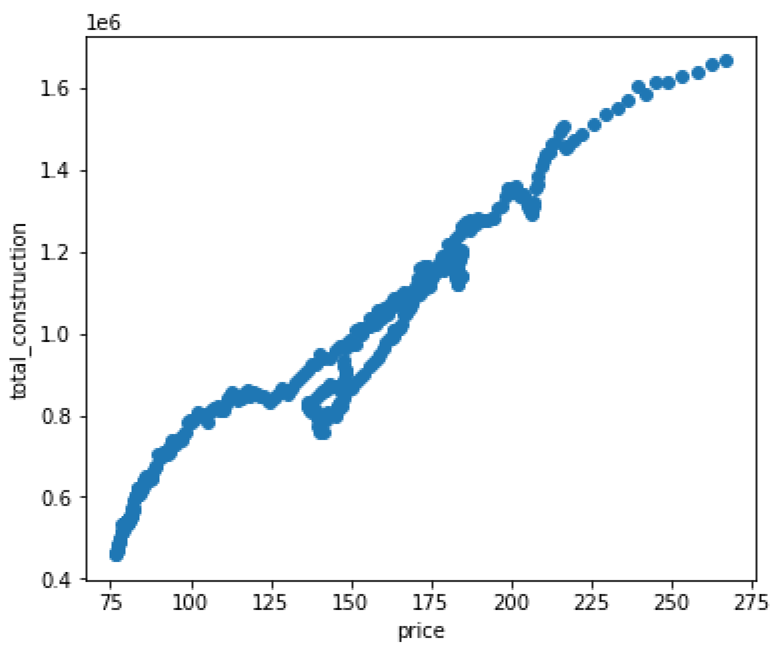
```
plt.figure(figsize=(6,5))
plt.scatter(df_files.year,df_files.TTLCONS)
plt.xlabel('year')
plt.ylabel('total_construction')
plt.show()
```



we can see the graph for total constriction was gradually increasing and it was high in the year 2023.

In [184…

```
plt.figure(figsize=(6,5))
plt.scatter(df_files.CSUSHPISA,df_files.TTLCONS)
plt.xlabel('price')
plt.ylabel('total_construction')
plt.show()
```



we can say that the as the construction rate is increasing the price was also increasing.

Checking Correlation

In [185…

```
cor=df_files.corr()
cor
```

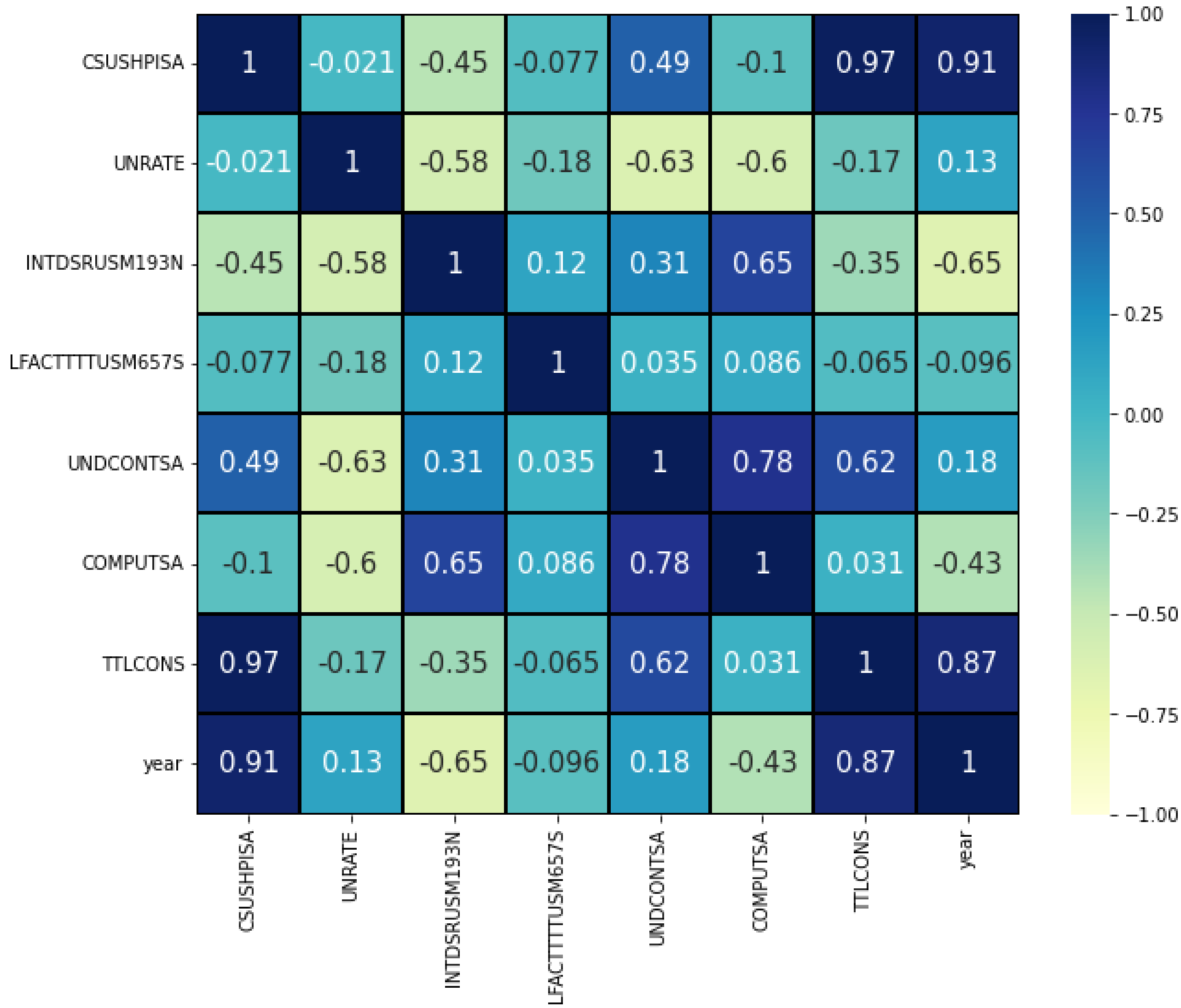
Out[185…

	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA	COMPUTSA	TTLCO
CSUSHPISA	1.000000	-0.020833	-0.446602	-0.076625	0.488785	-0.103728	0.973

	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA	COMPUTSA	TTLCO
UNRATE	-0.020833	1.000000	-0.582199	-0.184693	-0.627291	-0.597563	-0.174
INTDSRUSM193N	-0.446602	-0.582199	1.000000	0.121992	0.312216	0.649074	-0.352
LFACTTTTUSM657S	-0.076625	-0.184693	0.121992	1.000000	0.035034	0.086184	-0.065
UNDCONTSA	0.488785	-0.627291	0.312216	0.035034	1.000000	0.779675	0.619
COMPUTSA	-0.103728	-0.597563	0.649074	0.086184	0.779675	1.000000	0.031
TTLCONS	0.973775	-0.174769	-0.352253	-0.065320	0.619543	0.031316	1.000
year	0.914631	0.128267	-0.650182	-0.095796	0.175737	-0.428203	0.868

In [186…

```
plt.figure(figsize=(10,8))
sns.heatmap(df_files.corr(),linewidths=.1,vmin=-1, vmax=1, fmt='.2g', annot = True, linec
plt.yticks(rotation=0);
```

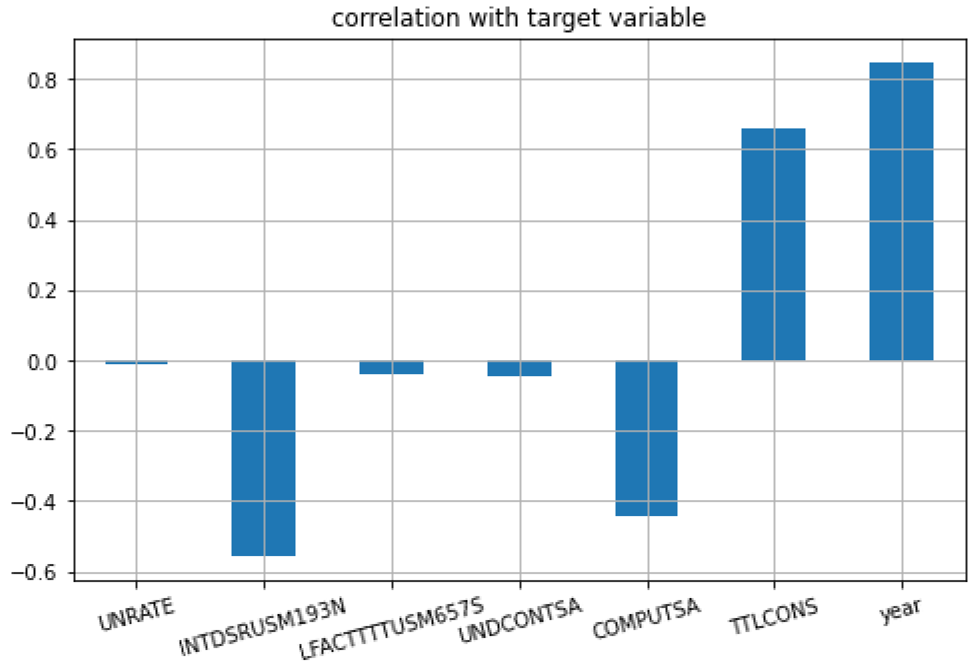


In [187…

```
plt.figure(figsize=(8,5))
df_files.drop('CSUSHPISA',axis=1).corrwith(df01['CSUSHPISA']).plot(kind='bar',grid=True)
plt.xticks(rotation=15)
plt.title('correlation with target variable')
```

Out[187…

```
Text(0.5, 1.0, 'correlation with target variable')
```



1. From the above bar graph we can say that unemployment rate, interest rate, active population, under construction, complete construction these key features are negatively corelated with the target variable i.e. price.

2. Total construction and year both are positively correlate with target variable.

Checking skewness

In [188…

```
df_files.skew()
```

Out[188…

```
CSUSHPISA          0.145970
UNRATE             1.361248
INTDSRUSM193N      0.354687
LFACTTTTUSM657S   -5.306864
UNDCONTSA         -0.258626
COMPUTSA          -0.107500
TTLCONS            0.352128
year               0.002754
dtype: float64
```

In [189…

```
df_files
```

Out[189…

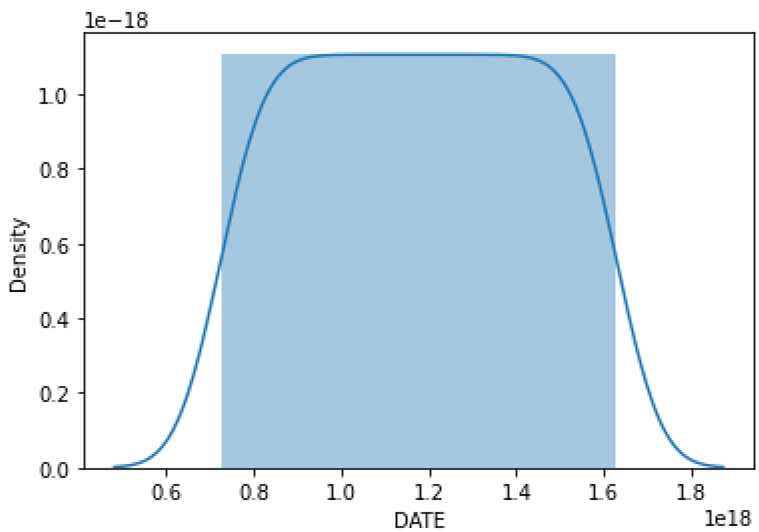
	DATE	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA	COMPUTSA	TTLCONS	yea
0	1993-01-01	76.784	7.3	3.00	-0.119794	636.0	1135.0	458080.0	199
1	1993-02-01	76.837	7.1	3.00	0.045171	640.0	1236.0	462967.0	199
2	1993-03-01	76.867	7.0	3.00	0.108985	633.0	1105.0	458399.0	199
3	1993-04-01	76.936	7.1	3.00	-0.010887	636.0	1216.0	469425.0	199
4	1993-05-01	77.037	7.1	3.00	0.528837	648.0	1111.0	468998.0	199

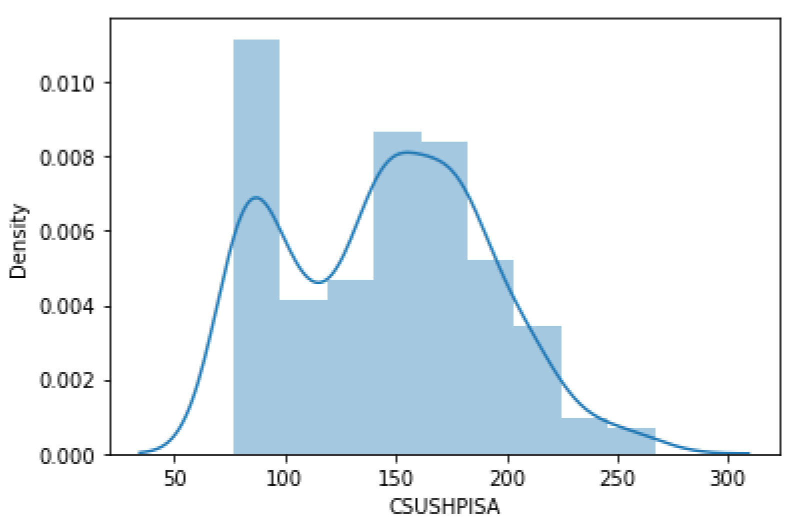
	DATE	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA	COMPUTSA	TTLCONS	yea
...	...	...	...	...	...	...	...	...	.
339	2021-04-01	249.070	6.1	0.25	0.277322	1320.0	1438.0	1615638.0	202
340	2021-05-01	253.407	5.8	0.25	-0.104407	1338.0	1337.0	1628281.0	202
341	2021-06-01	258.358	5.9	0.25	0.281199	1372.0	1298.0	1639779.0	202
342	2021-07-01	262.820	5.4	0.25	0.182391	1387.0	1361.0	1658346.0	202
343	2021-08-01	266.845	5.2	0.25	0.030343	1412.0	1312.0	1666756.0	202

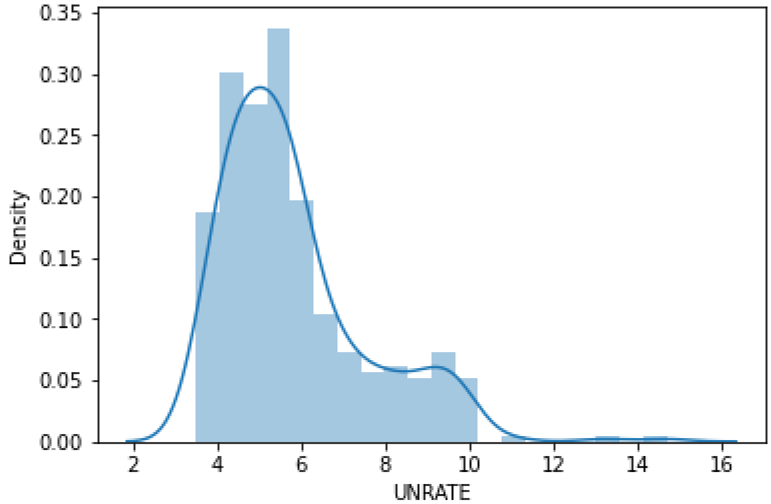
344 rows × 9 columns

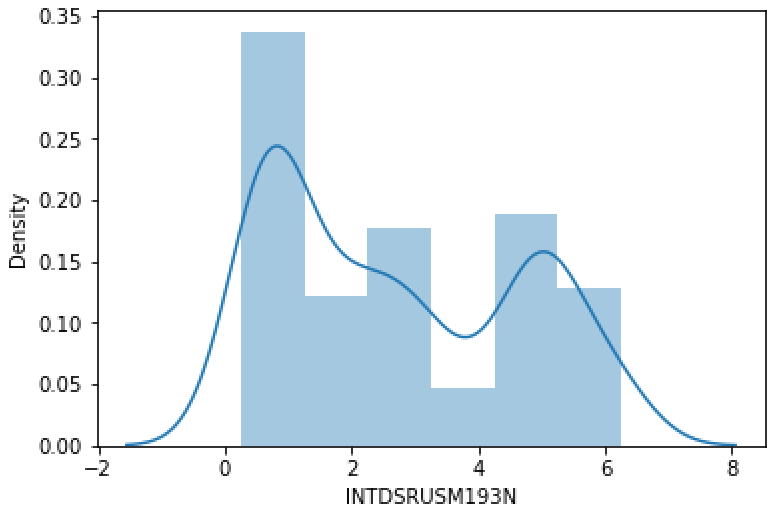
In [190…

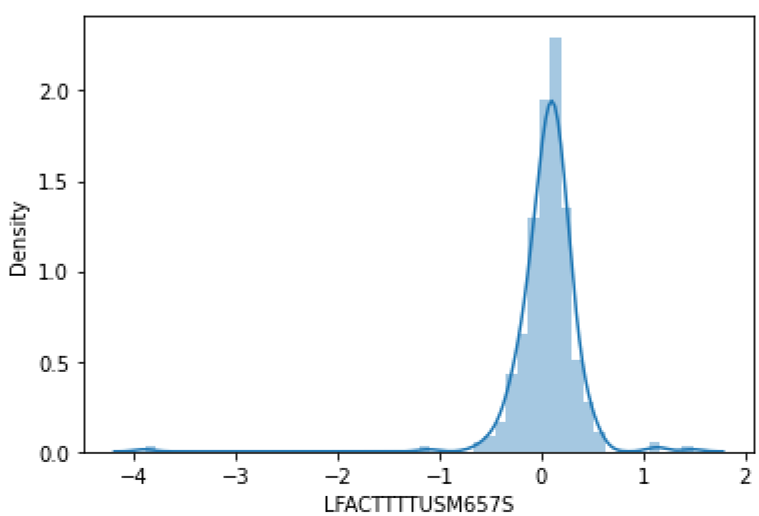
```
#KDE plot to check the distribution.
for i in df_files.columns:
    sns.distplot(df_files[i],kde=True)
    plt.show()
```

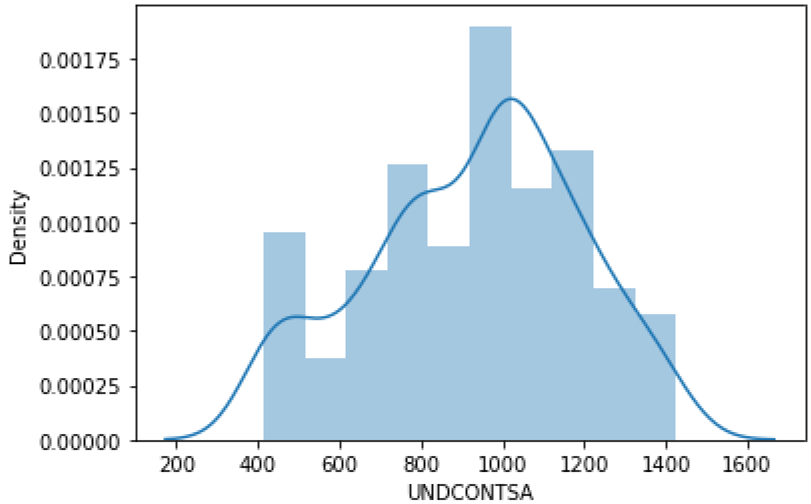


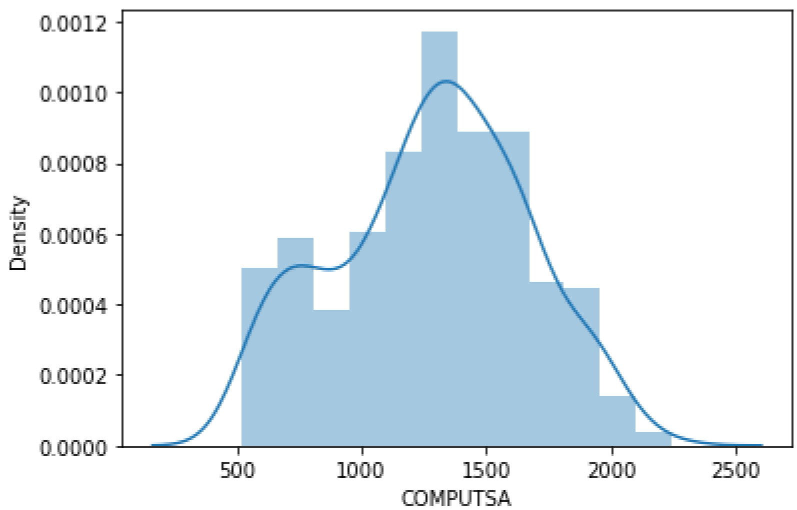


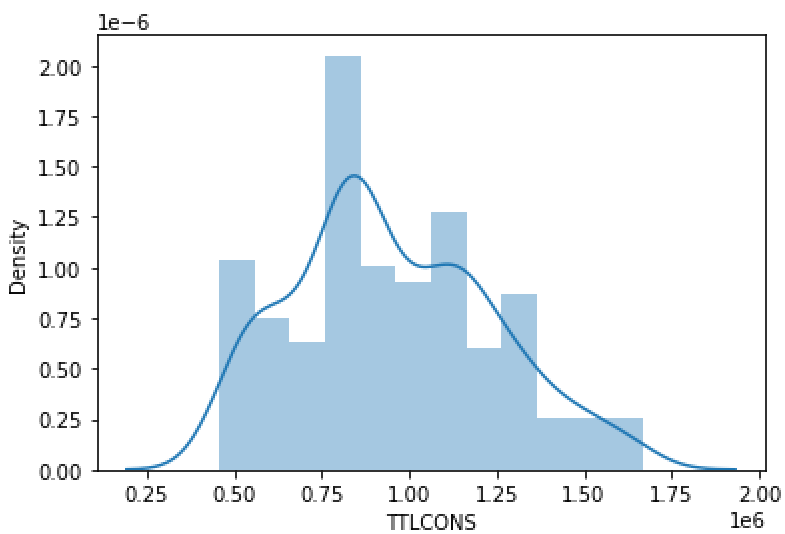

Density
UNRATE

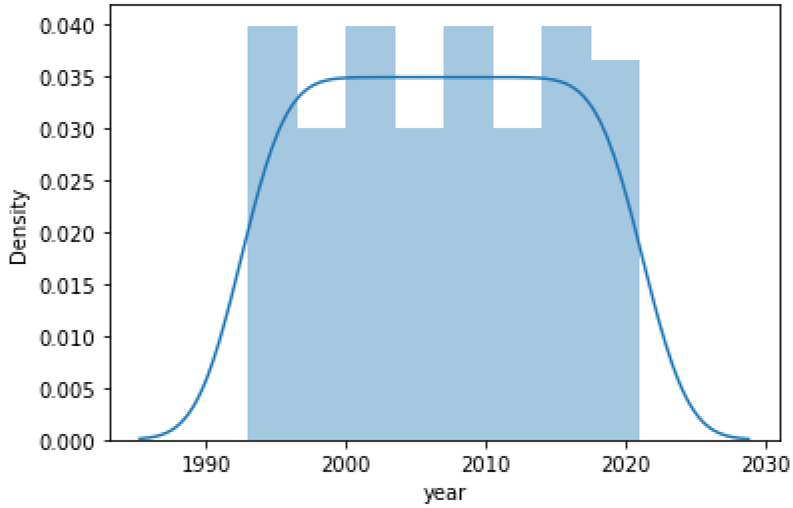

Density
INTDSRUSM193N


Density
LFACTTTTUSM657S


Density
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
200
400
600
800
1000
1200
1400
1600
UNDCONTSA


Density
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
0.0012
500
1000
1500
2000
2500
COMPUTSA


1e−6
Density
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
1e6
TTLCONS



I have plotted graph to see whether the data is normally distributed or not.

In [197…

```
df_files= df_files.drop('DATE',axis=1)
df_files
```

Out[197…

	CSUSHPISA	UNRATE	INTDSRUSM193N	LFACTTTTUSM657S	UNDCONTSA	COMPUTSA	TTLCONS	year
0	76.784	7.3	3.00	-0.119794	636.0	1135.0	458080.0	1993
1	76.837	7.1	3.00	0.045171	640.0	1236.0	462967.0	1993
2	76.867	7.0	3.00	0.108985	633.0	1105.0	458399.0	1993
3	76.936	7.1	3.00	-0.010887	636.0	1216.0	469425.0	1993
4	77.037	7.1	3.00	0.528837	648.0	1111.0	468998.0	1993
...	...	...	...	...	...	...	...	...
339	249.070	6.1	0.25	0.277322	1320.0	1438.0	1615638.0	2021
340	253.407	5.8	0.25	-0.104407	1338.0	1337.0	1628281.0	2021
341	258.358	5.9	0.25	0.281199	1372.0	1298.0	1639779.0	2021
342	262.820	5.4	0.25	0.182391	1387.0	1361.0	1658346.0	2021
343	266.845	5.2	0.25	0.030343	1412.0	1312.0	1666756.0	2021

344 rows × 8 columns

Z score

In [198…

```
from scipy.stats import zscore

z= np.abs(zscore(df_files))
print(np.where(z>3))
```

```
(array([ 84, 326, 327, 327, 328, 328, 329], dtype=int64), array([3, 3, 1, 3, 1, 3, 3], dty
pe=int64))
```

In [199…

```
df= df_files[(z<3).all(axis=1)]
print("with outliers::",df01.shape)
print("After removing outliers::",df02.shape)
```

```
with outliers:: (297, 8)
After removing outliers:: (294, 8)
```

Splitting data into X and Y:

In [200…

```
X=df.drop(columns=['CSUSHPISA'],axis=1)
Y=df['CSUSHPISA']
```

In [201…

```
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)
```

In [202…

```
from sklearn.metrics import mean_squared_error,mean_absolute_error
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression
```

splitting into x_train and y_train:

In [203…

```
LR=LinearRegression()
```

In [204…

```
LR.fit(x_train,y_train)
```

Out[204…

```
▾ LinearRegression
LinearRegression()
```

In [205…

```
for i in range (0,100):
    x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.20,random_state=i)
    LR.fit(x_train,y_train)
    LR_predict_train=LR.predict(x_train)
    LR_predict_test=LR.predict(x_test)
    print(f'At random state {i}, The training accuracy is :-{r2_score(y_train,LR_predict_t
    print(f'At random state {i}, The test accuracy is :-{r2_score(y_test,LR_predict_test)}
    print('\n')
```

```
At random state 0, The training accuracy is :-0.9837019628959501
At random state 0, The test accuracy is :-0.9839083930443696


At random state 1, The training accuracy is :-0.9825868728705563
At random state 1, The test accuracy is :-0.9879207129664563


At random state 2, The training accuracy is :-0.9844122562180208
At random state 2, The test accuracy is :-0.9820168632167986


At random state 3, The training accuracy is :-0.9838156682317519
At random state 3, The test accuracy is :-0.9837633519800233


At random state 4, The training accuracy is :-0.9833607618982051
At random state 4, The test accuracy is :-0.9850412415412659
```

```
At random state 5, The training accuracy is :-0.9828706333129351
At random state 5, The test accuracy is :-0.9861158642693874


At random state 6, The training accuracy is :-0.984864200861637
At random state 6, The test accuracy is :-0.978638087195254


At random state 7, The training accuracy is :-0.9865544905196336
At random state 7, The test accuracy is :-0.9733969145368929


At random state 8, The training accuracy is :-0.9856546061943385
At random state 8, The test accuracy is :-0.9756600970599137


At random state 9, The training accuracy is :-0.9829808751588801
At random state 9, The test accuracy is :-0.987032183037723


At random state 10, The training accuracy is :-0.9836351163152737
At random state 10, The test accuracy is :-0.9835824044341308


At random state 11, The training accuracy is :-0.9839189458202237
At random state 11, The test accuracy is :-0.9836311078463547


At random state 12, The training accuracy is :-0.9829818667145096
At random state 12, The test accuracy is :-0.9865527510785602


At random state 13, The training accuracy is :-0.9839057899782
At random state 13, The test accuracy is :-0.9823096624273824


At random state 14, The training accuracy is :-0.9852399636447073
At random state 14, The test accuracy is :-0.9770148982184224


At random state 15, The training accuracy is :-0.9836105243656936
At random state 15, The test accuracy is :-0.983892420078707


At random state 16, The training accuracy is :-0.9861199401114389
At random state 16, The test accuracy is :-0.9740796654431823


At random state 17, The training accuracy is :-0.9850109961823488
At random state 17, The test accuracy is :-0.9778495822404023


At random state 18, The training accuracy is :-0.983217427776964
At random state 18, The test accuracy is :-0.9854005665552338


At random state 19, The training accuracy is :-0.9852290521173475
At random state 19, The test accuracy is :-0.9781603259655821


At random state 20, The training accuracy is :-0.9835812913951585
At random state 20, The test accuracy is :-0.9845568339206625
```

```
At random state 21, The training accuracy is :-0.9851624457433048
At random state 21, The test accuracy is :-0.9784956818553667


At random state 22, The training accuracy is :-0.9839204676850448
At random state 22, The test accuracy is :-0.98269950308371


At random state 23, The training accuracy is :-0.9835654391891769
At random state 23, The test accuracy is :-0.9845547996010883


At random state 24, The training accuracy is :-0.983402307985781
At random state 24, The test accuracy is :-0.9818108696328139


At random state 25, The training accuracy is :-0.9833886184035506
At random state 25, The test accuracy is :-0.9857506285183733


At random state 26, The training accuracy is :-0.9879609892861339
At random state 26, The test accuracy is :-0.9611117352094634


At random state 27, The training accuracy is :-0.98371918647753
At random state 27, The test accuracy is :-0.9842407283922336


At random state 28, The training accuracy is :-0.9828641024087532
At random state 28, The test accuracy is :-0.9876271119486171


At random state 29, The training accuracy is :-0.9859956093587197
At random state 29, The test accuracy is :-0.97486813922064


At random state 30, The training accuracy is :-0.9844642824530581
At random state 30, The test accuracy is :-0.9799252314742654


At random state 31, The training accuracy is :-0.982842295069321
At random state 31, The test accuracy is :-0.9862989697615607


At random state 32, The training accuracy is :-0.9844648888752827
At random state 32, The test accuracy is :-0.9808170503610831


At random state 33, The training accuracy is :-0.9865635112102779
At random state 33, The test accuracy is :-0.9674168593131254


At random state 34, The training accuracy is :-0.9841914855120729
At random state 34, The test accuracy is :-0.982347475712988


At random state 35, The training accuracy is :-0.9825215569849215
At random state 35, The test accuracy is :-0.9877918904062705


At random state 36, The training accuracy is :-0.9825144087180836
At random state 36, The test accuracy is :-0.9876438765223168


At random state 37, The training accuracy is :-0.9836874862701472
At random state 37, The test accuracy is :-0.9836429562493584
```

```
At random state 38, The training accuracy is :-0.9826791057567701
At random state 38, The test accuracy is :-0.987534279112648


At random state 39, The training accuracy is :-0.9838459846590419
At random state 39, The test accuracy is :-0.9828539581709865


At random state 40, The training accuracy is :-0.9834630531734598
At random state 40, The test accuracy is :-0.9848305778066808


At random state 41, The training accuracy is :-0.984585688022516
At random state 41, The test accuracy is :-0.9802407128492662


At random state 42, The training accuracy is :-0.9850531453166945
At random state 42, The test accuracy is :-0.9792814532851448


At random state 43, The training accuracy is :-0.9836493506666706
At random state 43, The test accuracy is :-0.9846476488636436


At random state 44, The training accuracy is :-0.9843638896296465
At random state 44, The test accuracy is :-0.9817888523426204


At random state 45, The training accuracy is :-0.985379612017482
At random state 45, The test accuracy is :-0.9780195199388358


At random state 46, The training accuracy is :-0.9839347469549192
At random state 46, The test accuracy is :-0.9825308553627116


At random state 47, The training accuracy is :-0.983345601529918
At random state 47, The test accuracy is :-0.9860921709217184


At random state 48, The training accuracy is :-0.9839518363328247
At random state 48, The test accuracy is :-0.9829548252910753


At random state 49, The training accuracy is :-0.9839588417843488
At random state 49, The test accuracy is :-0.982832826282472


At random state 50, The training accuracy is :-0.984327669589227
At random state 50, The test accuracy is :-0.9806424848867563


At random state 51, The training accuracy is :-0.9865918033464692
At random state 51, The test accuracy is :-0.9679036079520638


At random state 52, The training accuracy is :-0.9861003362072761
At random state 52, The test accuracy is :-0.9738557257164632


At random state 53, The training accuracy is :-0.986776510296126
At random state 53, The test accuracy is :-0.9679455204068051
```

```
At random state 54, The training accuracy is :-0.9853914810155507
At random state 54, The test accuracy is :-0.9743421557426369


At random state 55, The training accuracy is :-0.9858293057220334
At random state 55, The test accuracy is :-0.9749644014698786


At random state 56, The training accuracy is :-0.9840543147203029
At random state 56, The test accuracy is :-0.9829316676934995


At random state 57, The training accuracy is :-0.9847059459788977
At random state 57, The test accuracy is :-0.9785268551866974


At random state 58, The training accuracy is :-0.9852677240750924
At random state 58, The test accuracy is :-0.9782624921950968


At random state 59, The training accuracy is :-0.9846032605698412
At random state 59, The test accuracy is :-0.9808579646140885


At random state 60, The training accuracy is :-0.9838915456349199
At random state 60, The test accuracy is :-0.9835794687440246


At random state 61, The training accuracy is :-0.9863681453822553
At random state 61, The test accuracy is :-0.9712452182361818


At random state 62, The training accuracy is :-0.9834657326414411
At random state 62, The test accuracy is :-0.9844843044659137


At random state 63, The training accuracy is :-0.9837069080321805
At random state 63, The test accuracy is :-0.9842052394393077


At random state 64, The training accuracy is :-0.9836749929593297
At random state 64, The test accuracy is :-0.9845969113981919


At random state 65, The training accuracy is :-0.9840029055142765
At random state 65, The test accuracy is :-0.9831788464092482


At random state 66, The training accuracy is :-0.9827104900082605
At random state 66, The test accuracy is :-0.9870516278125271


At random state 67, The training accuracy is :-0.9839941625396151
At random state 67, The test accuracy is :-0.9831434862660958


At random state 68, The training accuracy is :-0.9867607077160683
At random state 68, The test accuracy is :-0.9718978378255102


At random state 69, The training accuracy is :-0.9829434979053501
At random state 69, The test accuracy is :-0.9852587454688384


At random state 70, The training accuracy is :-0.9837317634908899
At random state 70, The test accuracy is :-0.9842159299556288
```

```
At random state 71, The training accuracy is :-0.9839203689883746
At random state 71, The test accuracy is :-0.9828184172018943


At random state 72, The training accuracy is :-0.9841910205473411
At random state 72, The test accuracy is :-0.9824483283975004


At random state 73, The training accuracy is :-0.9832750890996828
At random state 73, The test accuracy is :-0.9857593202954021


At random state 74, The training accuracy is :-0.9843575897642036
At random state 74, The test accuracy is :-0.9816283052103406


At random state 75, The training accuracy is :-0.9839714995629195
At random state 75, The test accuracy is :-0.9833438932124882


At random state 76, The training accuracy is :-0.9838923925358661
At random state 76, The test accuracy is :-0.9835628093928773


At random state 77, The training accuracy is :-0.9835864509296034
At random state 77, The test accuracy is :-0.9835583891408786


At random state 78, The training accuracy is :-0.9842356873793024
At random state 78, The test accuracy is :-0.9823787825863949


At random state 79, The training accuracy is :-0.9828097504711939
At random state 79, The test accuracy is :-0.9875485427927325


At random state 80, The training accuracy is :-0.9830345960497276
At random state 80, The test accuracy is :-0.9858061865344844


At random state 81, The training accuracy is :-0.9834340119999189
At random state 81, The test accuracy is :-0.9857646314775704


At random state 82, The training accuracy is :-0.9845232162189141
At random state 82, The test accuracy is :-0.9801155137105537


At random state 83, The training accuracy is :-0.9870865211247719
At random state 83, The test accuracy is :-0.9686585203541174


At random state 84, The training accuracy is :-0.9843347716325812
At random state 84, The test accuracy is :-0.9808823006085495


At random state 85, The training accuracy is :-0.981984607128656
At random state 85, The test accuracy is :-0.9887561460304538


At random state 86, The training accuracy is :-0.9840911454721342
At random state 86, The test accuracy is :-0.9820685051979058
```

```
At random state 87, The training accuracy is :-0.9823605005195944
At random state 87, The test accuracy is :-0.9869816573334919


At random state 88, The training accuracy is :-0.9836602499980691
At random state 88, The test accuracy is :-0.9839765241417545


At random state 89, The training accuracy is :-0.984271009847888
At random state 89, The test accuracy is :-0.9819716906780734


At random state 90, The training accuracy is :-0.9839507089810294
At random state 90, The test accuracy is :-0.9822031946991499


At random state 91, The training accuracy is :-0.9823124253676819
At random state 91, The test accuracy is :-0.9892741185926806


At random state 92, The training accuracy is :-0.9845765734564226
At random state 92, The test accuracy is :-0.9799942487628255


At random state 93, The training accuracy is :-0.9849677236544128
At random state 93, The test accuracy is :-0.9789324711074341


At random state 94, The training accuracy is :-0.983071060825889
At random state 94, The test accuracy is :-0.9840166009499801


At random state 95, The training accuracy is :-0.9826241037663195
At random state 95, The test accuracy is :-0.9877470512214794


At random state 96, The training accuracy is :-0.98721503743191
At random state 96, The test accuracy is :-0.9684966635106386


At random state 97, The training accuracy is :-0.9831680344383714
At random state 97, The test accuracy is :-0.9855554039148415


At random state 98, The training accuracy is :-0.9829058420255673
At random state 98, The test accuracy is :-0.9873627360935519


At random state 99, The training accuracy is :-0.9822562094352101
At random state 99, The test accuracy is :-0.9893847010963364
```

At Random State 48

In [206…

```
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.20,random_state=48)
```

In [207…

```
x_train.shape
```

Out[207…

```
(271, 7)
```

In [208…

```
y_train.shape
```

Out[208…
```
(271,)
```

In [209…
```
x_test.shape
```

Out[209…
```
(68, 7)
```

In [210…
```
y_test.shape
```

Out[210…
```
(68,)
```

Model Building

In [211…
```
from sklearn.linear_model import LinearRegression

LR = LinearRegression()
LR.fit(x_train,y_train)
print(LR.score(x_train,y_train))
LR_predict=LR.predict(x_test)
```

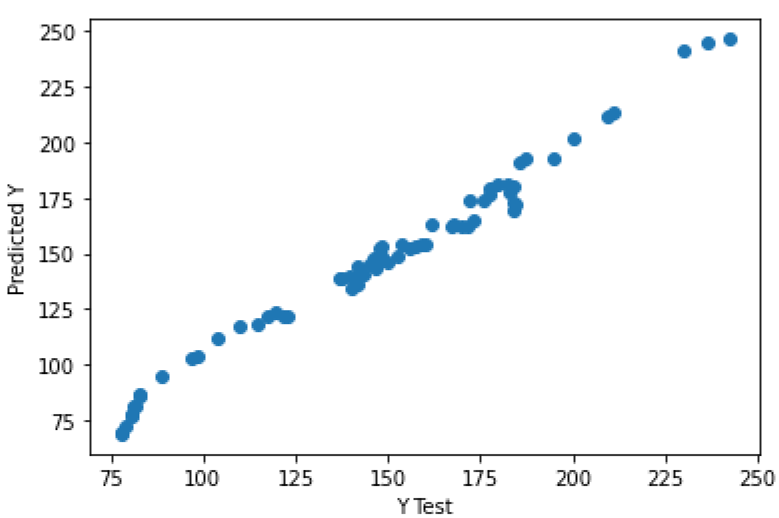
```
0.9839518363328247
```

In [212…
```
print('MSE:',mean_squared_error(LR_predict,y_test))
print('MAE:',mean_absolute_error(LR_predict,y_test))
print('r2_score:',r2_score(LR_predict,y_test))
print('RMSE:', np.sqrt(mean_squared_error(LR_predict,y_test)))
```

```
MSE: 28.003779940627027
MAE: 4.205731620567567
r2_score: 0.983194974987619
RMSE: 5.291859780892444
```

In [213…
```
plt.scatter(x=y_test,y=LR_predict)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[213…
```
Text(0, 0.5, 'Predicted Y')
```



We have got the straight line, but the RMSE is high so we are not sure about the model. so trying another model

In [214…
```
from sklearn.tree import DecisionTreeRegressor
```

```
DTR = DecisionTreeRegressor()
DTR.fit(x_train,y_train)
print(DTR.score(x_train,y_train))
DTR_Predict=DTR.predict(x_test)
```

```
1.0
```

In [215…

```
print('MSE:',mean_squared_error(DTR_Predict,y_test))
print('MAE:',mean_absolute_error(DTR_Predict,y_test))
print('r2_score:',r2_score(DTR_Predict,y_test))
print('RMSE:', np.sqrt(mean_squared_error(DTR_Predict,y_test)))
```

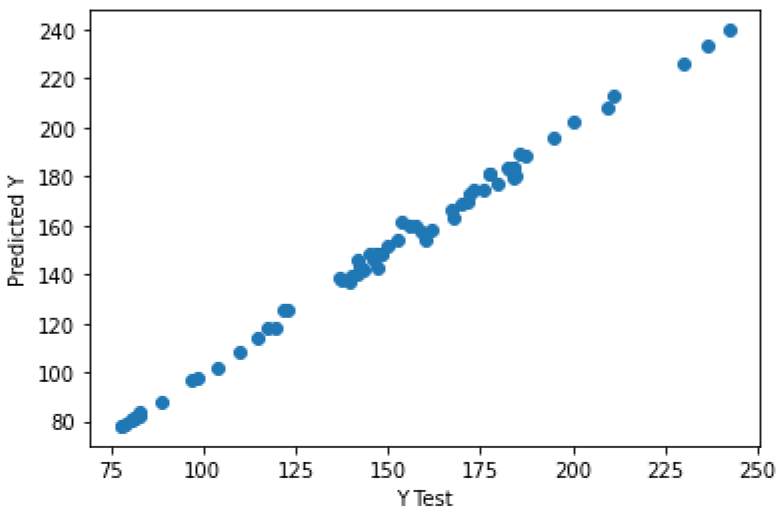
```
MSE: 5.788144102941183
MAE: 1.8091029411764716
r2_score: 0.9964252100179446
RMSE: 2.405856209947133
```

In [216…

```
plt.scatter(x=y_test,y=DTR_Predict)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[216…

```
Text(0, 0.5, 'Predicted Y')
```



Here also RMSE is high. so trying with Random forest model.

In [217…

```
from sklearn.ensemble import RandomForestRegressor
rdr = RandomForestRegressor()
rdr.fit(x_train,y_train)
print(rdr.score(x_train,y_train))
rdr_Predict=rdr.predict(x_test)
```

```
0.9997286310084705
```

In [218…

```
print('MSE:',mean_squared_error(rdr_Predict,y_test))
print('MAE:',mean_absolute_error(rdr_Predict,y_test))
print('r2_score:',r2_score(rdr_Predict,y_test))
print('RMSE:', np.sqrt(mean_squared_error(rdr_Predict,y_test)))
```

```
MSE: 3.1093383494705513
MAE: 1.3522697058823503
r2_score: 0.9981002095238141
RMSE: 1.7633316050790195
```

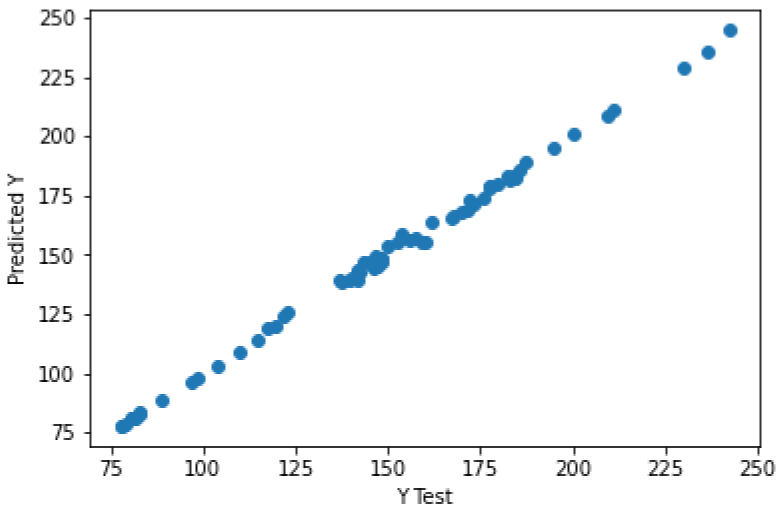
In [219…

```
plt.scatter(x=y_test,y=rdr_Predict)
```

```
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[219…

```
Text(0, 0.5, 'Predicted Y')
```



Here the RMSE is lower than the prevoius two models, so we can say that this is a good model.

In [220…

```
from sklearn.model_selection import cross_val_score
```

In [221…

```
from sklearn.model_selection import GridSearchCV
```

In [222…

```
rdr = RandomForestRegressor()

param ={
        'n_estimators':[100,200],
       'criterion':['mse','mae'],
      'min_samples_split':[2],
     'min_samples_leaf':[1],
}
```

In [223…

```
rdr_grid=GridSearchCV(RandomForestRegressor(),param,cv=4,scoring='accuracy',n_jobs=-1,verb
```

In [224…

```
rdr_grid.fit(x_train,y_train)
rdr_grid_PRED=rdr_grid.best_estimator_.predict(x_test)
```

```
Fitting 4 folds for each of 4 candidates, totalling 16 fits
```

In [225…

```
rdr_grid.best_params_
```

Out[225…

```
{'criterion': 'mse',
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'n_estimators': 100}
```

In [226…

```
print('MSE:',mean_squared_error(rdr_grid_PRED,y_test))
print('MAE:',mean_absolute_error(rdr_grid_PRED,y_test))
print('r2_score:',r2_score(rdr_grid_PRED,y_test))
```

```
MSE: 2.931714742495551
MAE: 1.2620869117647033
```

```
r2_score: 0.9982124364274597
```

In [227…

```
RF = RandomForestRegressor()

param ={
       'n_estimators':[100],
      'criterion':['mse'],
     'min_samples_split':[2],
    'min_samples_leaf':[1],
}
```

In [228…

```
RF_grid=GridSearchCV(RandomForestRegressor(),param,cv=4,scoring='accuracy',n_jobs=-1,verbo
```

In [229…

```
RF_grid.fit(x_train,y_train)
RF_grid_PRED=RF_grid.best_estimator_.predict(x_test)
```

```
Fitting 4 folds for each of 1 candidates, totalling 4 fits
```

In [230…

```
print('MSE:',mean_squared_error(RF_grid_PRED,y_test))
print('MAE:',mean_absolute_error(RF_grid_PRED,y_test))
print('r2_score:',r2_score(RF_grid_PRED,y_test))
```

```
MSE: 3.13481068627351
MAE: 1.3348114705882435
r2_score: 0.9980821872887043
```

In [231…

```
RF_grid_PRED
```

Out[231…

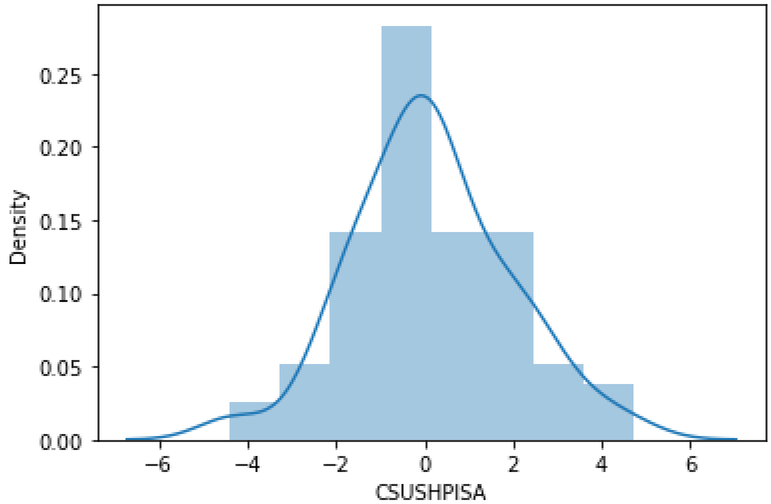
```
array([118.82399,  80.85049, 144.26224, 143.16848, 182.64485, 169.12556,
       209.35452, 138.91058, 182.78286, 245.1112 ,  81.47604, 114.38515,
        96.68331, 124.24765, 228.03087, 153.49959, 155.02455, 179.07568,
       182.75852, 148.26306, 157.36423, 147.46457, 171.87112, 125.59149,
       139.20872, 165.44873, 166.01041, 143.20928, 179.91037, 201.31933,
        80.52649, 139.3327 , 157.68276, 108.94093, 173.10953, 139.01644,
       147.30099, 157.71233, 119.75656, 158.26866, 146.26851,  88.64133,
       183.01024, 195.16235, 185.26611, 145.55483,  77.69633, 147.95621,
       167.86404,  82.7666 , 164.0012 ,  77.88359, 182.41524, 144.98929,
       174.40009, 147.4405 , 182.86663, 235.09693, 139.85685, 211.70979,
        78.58662,  97.94524, 189.47436, 103.64002, 177.4481 , 155.65381,
        81.40197,  83.18641])
```

In [232…

```
sns.distplot(RF_grid_PRED-y_test)
```

Out[232…

<AxesSubplot:xlabel='CSUSHPISA', ylabel='Density'>

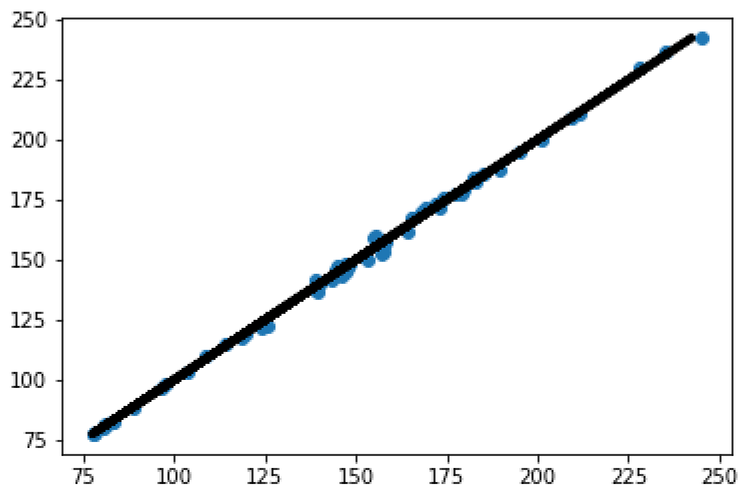


In [233…

```
plt.scatter(RF_grid_PRED,y_test)
plt.plot(y_test,y_test,linewidth=4,color='black')
```

Out[233…

```
[<matplotlib.lines.Line2D at 0x1c15ef52280>]
```



In [236…

```
!pip install -U notebook-as-pdf
!pyppeteer-install
```

```
Collecting notebook-as-pdf
  Downloading notebook_as_pdf-0.5.0-py3-none-any.whl (6.5 kB)
Collecting pyppeteer
  Downloading pyppeteer-1.0.2-py3-none-any.whl (83 kB)
Requirement already satisfied: nbconvert in c:\users\artia\anaconda3\lib\site-packages (fr
om notebook-as-pdf) (6.1.0)
Collecting PyPDF2
  Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
Requirement already satisfied: pygments>=2.4.1 in c:\users\artia\anaconda3\lib\site-packag
es (from nbconvert->notebook-as-pdf) (2.10.0)
Requirement already satisfied: nbformat>=4.4 in c:\users\artia\anaconda3\lib\site-packages
 (from nbconvert->notebook-as-pdf) (5.1.3)
Requirement already satisfied: jupyter-core in c:\users\artia\anaconda3\lib\site-packages
 (from nbconvert->notebook-as-pdf) (4.8.1)
Requirement already satisfied: entrypoints>=0.2.2 in c:\users\artia\anaconda3\lib\site-pac
kages (from nbconvert->notebook-as-pdf) (0.3)
Requirement already satisfied: traitlets>=5.0 in c:\users\artia\anaconda3\lib\site-package
s (from nbconvert->notebook-as-pdf) (5.1.0)
Requirement already satisfied: mistune<2,>=0.8.1 in c:\users\artia\anaconda3\lib\site-pack
ages (from nbconvert->notebook-as-pdf) (0.8.4)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in c:\users\artia\anaconda3\lib\site
-packages (from nbconvert->notebook-as-pdf) (0.5.3)
```

```
Requirement already satisfied: bleach in c:\users\artia\anaconda3\lib\site-packages (from
nbconvert->notebook-as-pdf) (4.0.0)
Requirement already satisfied: testpath in c:\users\artia\anaconda3\lib\site-packages (fro
m nbconvert->notebook-as-pdf) (0.5.0)
Requirement already satisfied: pandocfilters>=1.4.1 in c:\users\artia\anaconda3\lib\site-p
ackages (from nbconvert->notebook-as-pdf) (1.4.3)
Requirement already satisfied: defusedxml in c:\users\artia\anaconda3\lib\site-packages (f
rom nbconvert->notebook-as-pdf) (0.7.1)
Requirement already satisfied: jupyterlab-pygments in c:\users\artia\anaconda3\lib\site-pa
ckages (from nbconvert->notebook-as-pdf) (0.1.2)
Requirement already satisfied: jinja2>=2.4 in c:\users\artia\anaconda3\lib\site-packages
(from nbconvert->notebook-as-pdf) (2.11.3)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\artia\anaconda3\lib\site-packa
ges (from jinja2>=2.4->nbconvert->notebook-as-pdf) (1.1.1)
Requirement already satisfied: jupyter-client>=6.1.5 in c:\users\artia\anaconda3\lib\site-
packages (from nbclient<0.6.0,>=0.5.0->nbconvert->notebook-as-pdf) (6.1.12)
Requirement already satisfied: async-generator in c:\users\artia\anaconda3\lib\site-packag
es (from nbclient<0.6.0,>=0.5.0->nbconvert->notebook-as-pdf) (1.10)
Requirement already satisfied: nest-asyncio in c:\users\artia\anaconda3\lib\site-packages
(from nbclient<0.6.0,>=0.5.0->nbconvert->notebook-as-pdf) (1.5.1)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\artia\anaconda3\lib\site-p
ackages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert->notebook-as-pdf)
(2.8.2)
Requirement already satisfied: pyzmq>=13 in c:\users\artia\anaconda3\lib\site-packages (fr
om jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert->notebook-as-pdf) (22.2.1)
Requirement already satisfied: tornado>=4.1 in c:\users\artia\anaconda3\lib\site-packages
(from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert->notebook-as-pdf) (6.1)
Requirement already satisfied: pywin32>=1.0 in c:\users\artia\anaconda3\lib\site-packages
(from jupyter-core->nbconvert->notebook-as-pdf) (228)
Requirement already satisfied: ipython-genutils in c:\users\artia\anaconda3\lib\site-packa
ges (from nbformat>=4.4->nbconvert->notebook-as-pdf) (0.2.0)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in c:\users\artia\anaconda3\lib\sit
e-packages (from nbformat>=4.4->nbconvert->notebook-as-pdf) (3.2.0)
Requirement already satisfied: six>=1.11.0 in c:\users\artia\anaconda3\lib\site-packages
(from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert->notebook-as-pdf) (1.16.0)
Requirement already satisfied: attrs>=17.4.0 in c:\users\artia\anaconda3\lib\site-packages
(from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert->notebook-as-pdf) (21.2.0)
Requirement already satisfied: pyrsistent>=0.14.0 in c:\users\artia\anaconda3\lib\site-pac
kages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert->notebook-as-pdf) (0.18.0)
Requirement already satisfied: setuptools in c:\users\artia\anaconda3\lib\site-packages (f
rom jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert->notebook-as-pdf) (58.0.4)
Requirement already satisfied: packaging in c:\users\artia\anaconda3\lib\site-packages (fr
om bleach->nbconvert->notebook-as-pdf) (21.0)
Requirement already satisfied: webencodings in c:\users\artia\anaconda3\lib\site-packages
(from bleach->nbconvert->notebook-as-pdf) (0.5.1)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\artia\anaconda3\lib\site-packa
ges (from packaging->bleach->nbconvert->notebook-as-pdf) (3.0.4)
Requirement already satisfied: typing_extensions>=3.10.0.0 in c:\users\artia\anaconda3\lib
\site-packages (from PyPDF2->notebook-as-pdf) (3.10.0.2)
Requirement already satisfied: importlib-metadata>=1.4 in c:\users\artia\anaconda3\lib\sit
e-packages (from pyppeteer->notebook-as-pdf) (4.8.1)
Requirement already satisfied: certifi>=2021 in c:\users\artia\anaconda3\lib\site-packages
(from pyppeteer->notebook-as-pdf) (2021.10.8)
Requirement already satisfied: urllib3<2.0.0,>=1.25.8 in c:\users\artia\anaconda3\lib\site
-packages (from pyppeteer->notebook-as-pdf) (1.26.7)
Collecting websockets<11.0,>=10.0
  Downloading websockets-10.4-cp39-cp39-win_amd64.whl (101 kB)
Collecting pyee<9.0.0,>=8.1.0
  Downloading pyee-8.2.2-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: appdirs<2.0.0,>=1.4.3 in c:\users\artia\anaconda3\lib\site-
packages (from pyppeteer->notebook-as-pdf) (1.4.4)
Requirement already satisfied: tqdm<5.0.0,>=4.42.1 in c:\users\artia\anaconda3\lib\site-pa
ckages (from pyppeteer->notebook-as-pdf) (4.62.3)
Requirement already satisfied: zipp>=0.5 in c:\users\artia\anaconda3\lib\site-packages (fr
om importlib-metadata>=1.4->pyppeteer->notebook-as-pdf) (3.6.0)
Requirement already satisfied: colorama in c:\users\artia\anaconda3\lib\site-packages (fro
```

```
m tqdm<5.0.0,>=4.42.1->pyppeteer->notebook-as-pdf) (0.4.4)
Installing collected packages: websockets, pyee, pyppeteer, PyPDF2, notebook-as-pdf
Successfully installed PyPDF2-3.0.1 notebook-as-pdf-0.5.0 pyee-8.2.2 pyppeteer-1.0.2 webso
ckets-10.4
[INFO] Starting Chromium download.

  0%|          | 0.00/137M [00:00<?, ?b/s]
  0%|          | 20.5k/137M [00:00<21:38, 105kb/s]
  0%|          | 41.0k/137M [00:00<17:05, 134kb/s]
  0%|          | 61.4k/137M [00:00<15:01, 152kb/s]
  0%|          | 92.2k/137M [00:00<12:05, 189kb/s]
  0%|          | 133k/137M [00:00<09:00, 253kb/s]
  0%|          | 195k/137M [00:00<06:34, 346kb/s]
  0%|          | 266k/137M [00:00<05:01, 453kb/s]
  0%|          | 379k/137M [00:00<03:30, 650kb/s]
  0%|          | 532k/137M [00:01<02:31, 898kb/s]
  1%|          | 748k/137M [00:01<01:48, 1.25Mb/s]
  1%|          | 1.05M/137M [00:01<01:16, 1.78Mb/s]
  1%|1         | 1.46M/137M [00:01<00:55, 2.45Mb/s]
  2%|1         | 2.09M/137M [00:01<00:38, 3.51Mb/s]
  2%|2         | 2.91M/137M [00:01<00:27, 4.87Mb/s]
  3%|2         | 3.95M/137M [00:01<00:20, 6.42Mb/s]
  3%|3         | 4.60M/137M [00:02<00:41, 3.21Mb/s]
  6%|5         | 7.74M/137M [00:02<00:18, 6.81Mb/s]
  6%|6         | 8.73M/137M [00:02<00:17, 7.33Mb/s]
  7%|7         | 9.61M/137M [00:02<00:16, 7.58Mb/s]
  8%|7         | 10.5M/137M [00:02<00:16, 7.80Mb/s]
  8%|8         | 11.4M/137M [00:03<00:35, 3.51Mb/s]
  9%|8         | 12.0M/137M [00:03<00:35, 3.48Mb/s]
  9%|9         | 12.9M/137M [00:03<00:29, 4.19Mb/s]
 10%|9         | 13.5M/137M [00:03<00:27, 4.44Mb/s]
 11%|#         | 14.5M/137M [00:03<00:23, 5.26Mb/s]
 11%|#1        | 15.4M/137M [00:03<00:19, 6.08Mb/s]
 12%|#1        | 16.4M/137M [00:04<00:17, 6.91Mb/s]
 13%|#2        | 17.2M/137M [00:04<00:16, 7.30Mb/s]
 13%|#3        | 18.1M/137M [00:04<00:15, 7.64Mb/s]
 14%|#3        | 18.9M/137M [00:04<00:23, 5.08Mb/s]
 16%|#5        | 21.6M/137M [00:04<00:12, 9.23Mb/s]
 17%|#6        | 22.8M/137M [00:04<00:12, 8.98Mb/s]
 17%|#7        | 23.9M/137M [00:04<00:12, 8.90Mb/s]
 18%|#8        | 24.9M/137M [00:05<00:12, 8.70Mb/s]
 19%|#8        | 25.9M/137M [00:05<00:12, 8.77Mb/s]
 20%|#9        | 26.9M/137M [00:05<00:13, 8.41Mb/s]
 20%|##        | 27.8M/137M [00:05<00:12, 8.41Mb/s]
 21%|##        | 28.7M/137M [00:05<00:12, 8.43Mb/s]
 22%|##1       | 29.7M/137M [00:05<00:12, 8.45Mb/s]
 22%|##2       | 30.5M/137M [00:05<00:12, 8.44Mb/s]
 23%|##2       | 31.4M/137M [00:05<00:12, 8.51Mb/s]
 24%|##3       | 32.3M/137M [00:05<00:12, 8.45Mb/s]
 24%|##4       | 33.1M/137M [00:06<00:24, 4.25Mb/s]
 25%|##4       | 33.8M/137M [00:06<00:26, 3.83Mb/s]
 25%|##5       | 34.3M/137M [00:06<00:28, 3.64Mb/s]
 26%|##5       | 35.4M/137M [00:06<00:21, 4.69Mb/s]
 27%|##6       | 36.7M/137M [00:06<00:16, 6.22Mb/s]
 28%|##7       | 37.7M/137M [00:07<00:14, 6.90Mb/s]
 28%|##8       | 38.6M/137M [00:07<00:13, 7.24Mb/s]
 29%|##8       | 39.5M/137M [00:07<00:12, 7.59Mb/s]
 30%|##9       | 40.4M/137M [00:07<00:12, 7.84Mb/s]
 30%|###       | 41.3M/137M [00:07<00:11, 8.02Mb/s]
 31%|###       | 42.2M/137M [00:07<00:11, 8.21Mb/s]
 31%|###1      | 43.1M/137M [00:07<00:11, 8.27Mb/s]
 32%|###2      | 44.0M/137M [00:07<00:11, 8.36Mb/s]
 33%|###2      | 44.8M/137M [00:07<00:10, 8.42Mb/s]
 33%|###3      | 45.7M/137M [00:08<00:10, 8.46Mb/s]
 34%|###4      | 46.6M/137M [00:08<00:10, 8.49Mb/s]
```

```
 35%|###4      | 47.5M/137M [00:08<00:10, 8.48Mb/s]
 35%|###5      | 48.3M/137M [00:08<00:10, 8.51Mb/s]
 36%|###5      | 49.2M/137M [00:08<00:10, 8.52Mb/s]
 37%|###6      | 50.1M/137M [00:08<00:10, 8.48Mb/s]
 37%|###7      | 51.0M/137M [00:08<00:10, 8.46Mb/s]
 38%|###7      | 51.8M/137M [00:08<00:10, 8.45Mb/s]
 39%|###8      | 52.8M/137M [00:08<00:09, 8.48Mb/s]
 39%|###9      | 53.6M/137M [00:08<00:09, 8.50Mb/s]
 40%|###9      | 54.5M/137M [00:09<00:09, 8.47Mb/s]
 40%|####      | 55.4M/137M [00:09<00:09, 8.47Mb/s]
 41%|####1     | 56.2M/137M [00:09<00:09, 8.46Mb/s]
 42%|####1     | 57.1M/137M [00:09<00:09, 8.48Mb/s]
 42%|####2     | 57.9M/137M [00:09<00:09, 8.54Mb/s]
 43%|####2     | 58.8M/137M [00:09<00:09, 8.56Mb/s]
 44%|####3     | 59.7M/137M [00:10<00:18, 4.27Mb/s]
 44%|####4     | 60.3M/137M [00:10<00:20, 3.79Mb/s]
 44%|####4     | 60.9M/137M [00:10<00:20, 3.73Mb/s]
 45%|####5     | 61.8M/137M [00:10<00:16, 4.65Mb/s]
 46%|####6     | 63.3M/137M [00:10<00:11, 6.45Mb/s]
 47%|####6     | 64.2M/137M [00:10<00:10, 6.86Mb/s]
 48%|####7     | 65.0M/137M [00:10<00:10, 7.18Mb/s]
 48%|####8     | 65.9M/137M [00:10<00:09, 7.31Mb/s]
 49%|####8     | 66.7M/137M [00:11<00:09, 7.44Mb/s]
 49%|####9     | 67.5M/137M [00:11<00:09, 7.50Mb/s]
 50%|####9     | 68.3M/137M [00:11<00:09, 7.45Mb/s]
 50%|#####     | 69.0M/137M [00:11<00:09, 7.45Mb/s]
 51%|#####     | 69.8M/137M [00:11<00:09, 7.41Mb/s]
 52%|#####1    | 70.6M/137M [00:11<00:08, 7.38Mb/s]
 52%|#####2    | 71.3M/137M [00:12<00:19, 3.31Mb/s]
 53%|#####2    | 71.9M/137M [00:12<00:18, 3.51Mb/s]
 53%|#####2    | 72.4M/137M [00:12<00:17, 3.77Mb/s]
 53%|#####3    | 73.0M/137M [00:12<00:15, 4.03Mb/s]
 54%|#####3    | 73.5M/137M [00:13<00:32, 1.94Mb/s]
 54%|#####4    | 74.3M/137M [00:13<00:23, 2.72Mb/s]
 55%|#####4    | 74.8M/137M [00:13<00:21, 2.83Mb/s]
 55%|#####5    | 75.5M/137M [00:13<00:18, 3.38Mb/s]
 55%|#####5    | 76.0M/137M [00:13<00:17, 3.58Mb/s]
 56%|#####5    | 76.6M/137M [00:13<00:14, 4.14Mb/s]
 57%|#####6    | 77.4M/137M [00:13<00:11, 5.02Mb/s]
 57%|#####7    | 78.1M/137M [00:13<00:10, 5.35Mb/s]
 57%|#####7    | 78.7M/137M [00:14<00:26, 2.20Mb/s]
 58%|#####7    | 79.2M/137M [00:14<00:26, 2.20Mb/s]
 58%|#####8    | 79.7M/137M [00:14<00:21, 2.66Mb/s]
 59%|#####8    | 80.2M/137M [00:15<00:20, 2.75Mb/s]
 60%|######    | 82.3M/137M [00:15<00:08, 6.11Mb/s]
 61%|######    | 83.3M/137M [00:15<00:09, 5.78Mb/s]
 61%|######1   | 84.0M/137M [00:15<00:09, 5.49Mb/s]
 62%|######1   | 84.8M/137M [00:15<00:09, 5.41Mb/s]
 62%|######2   | 85.4M/137M [00:16<00:16, 3.09Mb/s]
 63%|######2   | 85.9M/137M [00:16<00:18, 2.69Mb/s]
 63%|######3   | 86.3M/137M [00:16<00:18, 2.77Mb/s]
 63%|######3   | 86.7M/137M [00:16<00:17, 2.92Mb/s]
 64%|######3   | 87.1M/137M [00:16<00:16, 3.06Mb/s]
 64%|######3   | 87.4M/137M [00:16<00:15, 3.14Mb/s]
 64%|######4   | 88.0M/137M [00:16<00:13, 3.71Mb/s]
 65%|######4   | 88.4M/137M [00:17<00:13, 3.53Mb/s]
 65%|######4   | 88.8M/137M [00:17<00:14, 3.42Mb/s]
 65%|######5   | 89.2M/137M [00:17<00:20, 2.37Mb/s]
 66%|######5   | 90.3M/137M [00:17<00:11, 4.02Mb/s]
 66%|######6   | 90.8M/137M [00:17<00:11, 3.93Mb/s]
 67%|######6   | 91.3M/137M [00:17<00:12, 3.76Mb/s]
 67%|######7   | 91.7M/137M [00:18<00:12, 3.63Mb/s]
 67%|######7   | 92.1M/137M [00:18<00:12, 3.57Mb/s]
 68%|######7   | 92.5M/137M [00:18<00:12, 3.54Mb/s]
 68%|######7   | 92.9M/137M [00:18<00:12, 3.46Mb/s]
 68%|######8   | 93.3M/137M [00:18<00:12, 3.48Mb/s]
```

```
68%|######8   | 93.6M/137M [00:18<00:12, 3.43Mb/s]
69%|######8   | 94.0M/137M [00:18<00:12, 3.39Mb/s]
69%|######8   | 94.3M/137M [00:18<00:12, 3.33Mb/s]
69%|######9   | 94.7M/137M [00:18<00:13, 3.22Mb/s]
69%|######9   | 95.0M/137M [00:19<00:13, 3.18Mb/s]
70%|######9   | 95.4M/137M [00:19<00:12, 3.39Mb/s]
70%|######9   | 95.8M/137M [00:19<00:11, 3.51Mb/s]
70%|#######   | 96.2M/137M [00:19<00:11, 3.46Mb/s]
71%|#######   | 96.6M/137M [00:19<00:11, 3.44Mb/s]
71%|#######   | 96.9M/137M [00:19<00:24, 1.61Mb/s]
71%|#######1  | 97.3M/137M [00:20<00:20, 1.95Mb/s]
72%|#######1  | 97.9M/137M [00:20<00:14, 2.67Mb/s]
72%|#######1  | 98.3M/137M [00:20<00:13, 2.82Mb/s]
73%|#######2  | 99.5M/137M [00:20<00:07, 4.84Mb/s]
73%|#######3  | 100M/137M [00:20<00:09, 4.00Mb/s]
73%|#######3  | 101M/137M [00:20<00:09, 4.02Mb/s]
74%|#######3  | 101M/137M [00:20<00:08, 4.01Mb/s]
74%|#######4  | 102M/137M [00:21<00:13, 2.60Mb/s]
74%|#######4  | 102M/137M [00:21<00:19, 1.83Mb/s]
75%|#######4  | 102M/137M [00:21<00:16, 2.13Mb/s]
75%|#######4  | 103M/137M [00:21<00:15, 2.22Mb/s]
75%|#######5  | 103M/137M [00:21<00:13, 2.49Mb/s]
76%|#######5  | 103M/137M [00:22<00:12, 2.73Mb/s]
76%|#######5  | 104M/137M [00:22<00:11, 2.93Mb/s]
76%|#######6  | 104M/137M [00:22<00:16, 2.03Mb/s]
77%|#######6  | 105M/137M [00:22<00:14, 2.21Mb/s]
77%|#######6  | 105M/137M [00:22<00:16, 1.91Mb/s]
77%|#######6  | 105M/137M [00:23<00:16, 1.96Mb/s]
77%|#######7  | 106M/137M [00:23<00:17, 1.81Mb/s]
77%|#######7  | 106M/137M [00:23<00:14, 2.18Mb/s]
78%|#######7  | 106M/137M [00:23<00:12, 2.48Mb/s]
78%|#######7  | 107M/137M [00:23<00:10, 2.75Mb/s]
78%|#######8  | 107M/137M [00:23<00:10, 2.98Mb/s]
78%|#######8  | 107M/137M [00:24<00:17, 1.72Mb/s]
79%|#######8  | 108M/137M [00:24<00:17, 1.67Mb/s]
79%|#######8  | 108M/137M [00:24<00:18, 1.59Mb/s]
79%|#######8  | 108M/137M [00:24<00:24, 1.19Mb/s]
79%|#######9  | 108M/137M [00:24<00:23, 1.21Mb/s]
79%|#######9  | 109M/137M [00:25<00:17, 1.57Mb/s]
80%|#######9  | 109M/137M [00:25<00:14, 1.88Mb/s]
80%|########  | 110M/137M [00:25<00:10, 2.52Mb/s]
80%|########  | 110M/137M [00:25<00:10, 2.65Mb/s]
81%|########  | 110M/137M [00:25<00:08, 2.99Mb/s]
81%|########  | 111M/137M [00:25<00:08, 3.11Mb/s]
81%|########1 | 111M/137M [00:25<00:08, 3.08Mb/s]
81%|########1 | 111M/137M [00:25<00:08, 3.03Mb/s]
82%|########1 | 112M/137M [00:25<00:08, 3.01Mb/s]
82%|########1 | 112M/137M [00:26<00:08, 2.97Mb/s]
82%|########2 | 112M/137M [00:26<00:08, 3.03Mb/s]
82%|########2 | 113M/137M [00:26<00:08, 2.87Mb/s]
83%|########2 | 113M/137M [00:26<00:07, 3.12Mb/s]
83%|########2 | 113M/137M [00:26<00:16, 1.40Mb/s]
83%|########2 | 114M/137M [00:27<00:18, 1.28Mb/s]
84%|########3 | 114M/137M [00:27<00:09, 2.34Mb/s]
84%|########3 | 115M/137M [00:27<00:08, 2.55Mb/s]
84%|########4 | 115M/137M [00:27<00:07, 3.04Mb/s]
85%|########4 | 116M/137M [00:27<00:07, 3.00Mb/s]
85%|########4 | 116M/137M [00:27<00:06, 2.98Mb/s]
85%|########5 | 117M/137M [00:27<00:06, 3.04Mb/s]
85%|########5 | 117M/137M [00:27<00:06, 3.06Mb/s]
86%|########5 | 117M/137M [00:28<00:06, 3.12Mb/s]
86%|########5 | 118M/137M [00:28<00:06, 3.10Mb/s]
86%|########6 | 118M/137M [00:28<00:06, 3.13Mb/s]
86%|########6 | 118M/137M [00:28<00:13, 1.42Mb/s]
87%|########6 | 119M/137M [00:29<00:10, 1.72Mb/s]
87%|########6 | 119M/137M [00:29<00:08, 2.03Mb/s]
```

```
 87%|########7 | 119M/137M [00:29<00:07, 2.35Mb/s]
 88%|########7 | 120M/137M [00:29<00:05, 3.26Mb/s]
 88%|########8 | 121M/137M [00:29<00:04, 3.37Mb/s]
 88%|########8 | 121M/137M [00:29<00:04, 3.45Mb/s]
 89%|########8 | 121M/137M [00:29<00:04, 3.53Mb/s]
 89%|########8 | 122M/137M [00:29<00:04, 3.55Mb/s]
 89%|########9 | 122M/137M [00:29<00:04, 3.60Mb/s]
 89%|########9 | 122M/137M [00:29<00:04, 3.59Mb/s]
 90%|########9 | 123M/137M [00:30<00:03, 3.64Mb/s]
 90%|######### | 123M/137M [00:30<00:03, 3.62Mb/s]
 90%|######### | 124M/137M [00:30<00:03, 3.65Mb/s]
 91%|######### | 124M/137M [00:30<00:03, 3.86Mb/s]
 91%|######### | 125M/137M [00:30<00:03, 4.05Mb/s]
 91%|#########1| 125M/137M [00:30<00:02, 4.25Mb/s]
 92%|#########1| 126M/137M [00:30<00:02, 4.57Mb/s]
 92%|#########2| 126M/137M [00:30<00:02, 4.56Mb/s]
 92%|#########2| 126M/137M [00:31<00:03, 3.11Mb/s]
 94%|#########3| 128M/137M [00:31<00:01, 6.04Mb/s]
 94%|#########4| 129M/137M [00:31<00:01, 5.88Mb/s]
 95%|#########4| 130M/137M [00:31<00:01, 5.69Mb/s]
 95%|#########5| 130M/137M [00:31<00:01, 5.64Mb/s]
 96%|#########5| 131M/137M [00:31<00:01, 5.59Mb/s]
 96%|#########5| 131M/137M [00:31<00:00, 5.61Mb/s]
 96%|#########6| 132M/137M [00:31<00:00, 5.52Mb/s]
 97%|#########6| 133M/137M [00:31<00:00, 5.46Mb/s]
 97%|#########7| 133M/137M [00:32<00:00, 5.44Mb/s]
 98%|#########7| 134M/137M [00:32<00:00, 5.44Mb/s]
 98%|#########8| 134M/137M [00:32<00:00, 5.40Mb/s]
 99%|#########8| 135M/137M [00:32<00:00, 5.43Mb/s]
 99%|#########8| 135M/137M [00:32<00:00, 2.48Mb/s]
 99%|#########9| 136M/137M [00:33<00:00, 3.01Mb/s]
100%|#########9| 137M/137M [00:33<00:00, 3.57Mb/s]
100%|##########| 137M/137M [00:33<00:00, 4.13Mb/s]
[INFO] Beginning extraction
[INFO] Chromium extracted to: C:\Users\artia\AppData\Local\pyppeteer\pyppeteer\local-chrom
ium\588429
```

In [237…

```
import joblib
```

In [238…

```
joblib.dump(RF_grid.best_estimator_,'US_homePrice_Project.obj')
```

Out[238…

```
['US_homePrice_Project.obj']
```

END