In [170…

```
import pandas as pd
import pickle
import warnings
warnings.filterwarnings('ignore')
```

In [171…

```
a=pd.read_csv(r"C:\Users\reshma_koduri\OneDrive\Documents\archive (4)\House_Price.cs
```

In [172…

```
a
```

Out[172…

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Unnamed: 4	HOUSE_TYPE	Unnamed: 6	Unnamed: 7	Purpc
0	₹8.5 Cr	₹4.22 Lacs	6 BHK	6.0	BHK	Independent House	Independent	House	for s
1	₹45.0 L	₹23.83 K	3 BHK	3.0	BHK	Independent House	Independent	House	for s
2	₹1.35 Cr	₹67.02 K	3 BHK	3.0	BHK	Independent House	Independent	House	for s
3	₹60.0 L	₹31.77 K	5 BHK	5.0	BHK	Independent House	Independent	House	for s
4	₹52.0 L	₹27.54 K	4 BHK	4.0	BHK	Independent House	Independent	House	for s
...	...	...	...	...	...	...	...	...	
3963	₹13.0 L	₹6.88 K	2 BHK	2.0	BHK	Independent House	Independent	House	for s
3964	₹1.5 Cr	₹74.47 K	8 BHK	8.0	BHK	Independent House	Independent	House	for s
3965	₹50.0 L	₹26.48 K	5 BHK	5.0	BHK	Independent House	Independent	House	for s
3966	₹1.1 Cr	₹54.61 K	5 BHK	5.0	BHK	Independent House	Independent	House	for s
3967	₹30.0 L	₹15.89 K	2 BHK	2.0	BHK	Independent House	Independent	House	for s

3968 rows × 15 columns

In [173…

```
a.head(10)
```

Out[173…

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Unnamed: 4	HOUSE_TYPE	Unnamed: 6	Unnamed: 7	Purpose
0	₹8.5 Cr	₹4.22 Lacs	6 BHK	6.0	BHK	Independent House	Independent	House	for sale in
1	₹45.0 L	₹23.83 K	3 BHK	3.0	BHK	Independent House	Independent	House	for sale in

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Unnamed: 4	HOUSE_TYPE	Unnamed: 6	Unnamed: 7	Purpose
2	₹1.35 Cr	₹67.02 K	3 BHK	3.0	BHK	Independent House	Independent	House	for sale in
3	₹60.0 L	₹31.77 K	5 BHK	5.0	BHK	Independent House	Independent	House	for sale in
4	₹52.0 L	₹27.54 K	4 BHK	4.0	BHK	Independent House	Independent	House	for sale in
5	₹32.0 L	₹16.95 K	3 BHK	3.0	BHK	Independent House	Independent	House	for sale in
6	₹69.3 L	₹36.70 K	3 BHK	3.0	BHK	Independent House	Independent	House	for sale in
7	₹40.0 L	₹21.18 K	2 BHK	2.0	BHK	Independent House	Independent	House	for sale in
8	₹95.0 L	₹47.16 K	4 BHK	4.0	BHK	Independent House	Independent	House	for sale in
9	₹3.0 Cr	₹1.49 Lacs	9 BHK	9.0	BHK	Independent House	Independent	House	for sale in

In [174…

```
a.tail(10)
```

Out[174…

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Unnamed: 4	HOUSE_TYPE	Unnamed: 6	Unnamed: 7	Purpc
3958	₹42.0 L	₹22.24 K	4 BHK	4.0	BHK	Independent House	Independent	House	for s
3959	₹1.5 Cr	₹74.47 K	4 BHK	4.0	BHK	Independent House	Independent	House	for s
3960	₹1.5 Cr	₹74.47 K	3 BHK	3.0	BHK	Independent House	Independent	House	for s
3961	₹27.0 L	₹14.30 K	1 BHK	1.0	BHK	Independent House	Independent	House	for s
3962	₹31.0 L	₹16.42 K	3 BHK	3.0	BHK	Independent House	Independent	House	for s
3963	₹13.0 L	₹6.88 K	2 BHK	2.0	BHK	Independent House	Independent	House	for s
3964	₹1.5 Cr	₹74.47 K	8 BHK	8.0	BHK	Independent House	Independent	House	for s
3965	₹50.0 L	₹26.48 K	5 BHK	5.0	BHK	Independent House	Independent	House	for s
3966	₹1.1 Cr	₹54.61 K	5 BHK	5.0	BHK	Independent House	Independent	House	for s

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Unnamed: 4	HOUSE_TYPE	Unnamed: 6	Unnamed: 7	Purpc
3967	₹30.0 L	₹15.89 K	2 BHK	2.0	BHK	Independent House	Independent	House	for s

In [175…
```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3968 entries, 0 to 3967
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Flat_Price       3968 non-null   object 
 1   EMI_Starts       3968 non-null   object 
 2   BHK              3968 non-null   object 
 3   css-11nfaq3      3966 non-null   float64
 4   Unnamed: 4       3966 non-null   object 
 5   HOUSE_TYPE       3968 non-null   object 
 6   Unnamed: 6       3966 non-null   object 
 7   Unnamed: 7       3966 non-null   object 
 8   Purpose          3966 non-null   object 
 9   Location         3966 non-null   object 
 10  Area_Type        3968 non-null   object 
 11  Total_Sq.ft      3968 non-null   object 
 12  Price_per_sq.ft  3968 non-null   object 
 13  Owner_name       3958 non-null   object 
 14  Owner_type       3968 non-null   object 
dtypes: float64(1), object(14)
memory usage: 465.1+ KB
```

In [176…
```
a.describe()
```

Out[176…

	css-11nfaq3
count	3966.000000
mean	4.742436
std	2.440006
min	1.000000
25%	3.000000
50%	4.000000
75%	6.000000
max	10.500000

In [177…
```
b=a.drop(['Unnamed: 6','Unnamed: 4','Unnamed: 7','Purpose','Location','HOUSE_TYPE','
b
```

Out[177…

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Area_Type	Total_Sq.ft	Price_per_sq.ft
0	₹8.5 Cr	₹4.22 Lacs	6 BHK	6.0	Build Up Area	4200 sq.ft	₹20.24 K/sq.ft
1	₹45.0 L	₹23.83 K	3 BHK	3.0	Build Up Area	1400 sq.ft	₹3.21 K/sq.ft
2	₹1.35 Cr	₹67.02 K	3 BHK	3.0	Build Up Area	2500 sq.ft	₹5.40 K/sq.ft

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Area_Type	Total_Sq.ft	Price_per_sq.ft
3	₹60.0 L	₹31.77 K	5 BHK	5.0	Build Up Area	1100 sq.ft	₹5.45 K/sq.ft
4	₹52.0 L	₹27.54 K	4 BHK	4.0	Build Up Area	900 sq.ft	₹5.78 K/sq.ft
...	...	...	...	...	...	...	...
3963	₹13.0 L	₹6.88 K	2 BHK	2.0	Build Up Area	1500 sq.ft	₹866/sq.ft
3964	₹1.5 Cr	₹74.47 K	8 BHK	8.0	Build Up Area	2560 sq.ft	₹5.86 K/sq.ft
3965	₹50.0 L	₹26.48 K	5 BHK	5.0	Build Up Area	1900 sq.ft	₹2.63 K/sq.ft
3966	₹1.1 Cr	₹54.61 K	5 BHK	5.0	Build Up Area	2600 sq.ft	₹4.23 K/sq.ft
3967	₹30.0 L	₹15.89 K	2 BHK	2.0	Build Up Area	950 sq.ft	₹3.16 K/sq.ft

3968 rows × 7 columns

In [200…

```
b['Price_per_sq.ft'] = b['Price_per_sq.ft'].apply(lambda x: x.rstrip("/sq.ft"))
b['Price_per_sq.ft'] = b['Price_per_sq.ft'].apply(lambda x: x.lstrip('₹'))
b['Total_Sq.ft'] = b['Total_Sq.ft'].apply(lambda x: x.rstrip("/sq.ft"))
b.head()
```

Out[200…

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Area_Type	Total_Sq.ft	Price_per_sq.ft	total	bhk	
0	8.5	4.22 Lacs	6	6.0	Build Up Area	4200	20.24 K	35.0	35.0	8500000(
1	45.0	23.83 K	3	3.0	Build Up Area	1400	3.21 K	35.0	35.0	450000(
2	1.35	67.02 K	3	3.0	Build Up Area	2500	5.40 K	35.0	35.0	1350000(
3	60.0	31.77 K	5	5.0	Build Up Area	1100	5.45 K	35.0	35.0	600000(
4	52.0	27.54 K	4	4.0	Build Up Area	900	5.78 K	35.0	35.0	520000(

In [201…

```
b.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3968 entries, 0 to 3967
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Flat_Price       3968 non-null   object 
 1   EMI_Starts       3968 non-null   object 
 2   BHK              3968 non-null   object 
 3   css-11nfaq3      3968 non-null   float64
 4   Area_Type        3968 non-null   object 
 5   Total_Sq.ft      3968 non-null   object 
 6   Price_per_sq.ft  3968 non-null   object 
 7   total            3968 non-null   float64
 8   bhk              3968 non-null   float64
 9   rs               3968 non-null   float64
 10  emi              3968 non-null   float64
 11  price            3968 non-null   float64
```

```
dtypes: float64(6), object(6)
memory usage: 372.1+ KB
```

In [202…

```
b['total']=pd.to_numeric(b['Total_Sq.ft'], errors='coerce')
b['bhk']=pd.to_numeric(b['BHK'], errors='coerce')
```

In [203…

```
b['EMI_Starts'] = b['EMI_Starts'].apply(lambda x: x.lstrip('₹'))
b['Flat_Price'] = b['Flat_Price'].apply(lambda x: x.lstrip('₹'))
b['BHK'] = b['BHK'].apply(lambda x: x.rstrip('BHK'))
b.head()
```

Out[203…

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Area_Type	Total_Sq.ft	Price_per_sq.ft	total	bhk	
0	8.5	4.22 Lacs	6	6.0	Build Up Area	4200	20.24 K	4200.0	6.0	850000
1	45.0	23.83 K	3	3.0	Build Up Area	1400	3.21 K	1400.0	3.0	45000
2	1.35	67.02 K	3	3.0	Build Up Area	2500	5.40 K	2500.0	3.0	135000
3	60.0	31.77 K	5	5.0	Build Up Area	1100	5.45 K	1100.0	5.0	60000
4	52.0	27.54 K	4	4.0	Build Up Area	900	5.78 K	900.0	4.0	52000

In [204…

```
b['rs']=b['Flat_Price'].replace({'L':'*1e5','Cr':'*1e7'},regex=True).map(pd.eval)
b['emi']=b['EMI_Starts'].replace({'K':'*1e3','Lacs':'*1e5'},regex=True).map(pd.eval)
b['price']=b['Price_per_sq.ft'].replace({'K':'*1e3'},regex=True).map(pd.eval)
b
```

Out[204…

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Area_Type	Total_Sq.ft	Price_per_sq.ft	total	bhk	
0	8.5	4.22 Lacs	6	6.0	Build Up Area	4200	20.24 K	4200.0	6.0	8.5
1	45.0	23.83 K	3	3.0	Build Up Area	1400	3.21 K	1400.0	3.0	45.0
2	1.35	67.02 K	3	3.0	Build Up Area	2500	5.40 K	2500.0	3.0	1.3
3	60.0	31.77 K	5	5.0	Build Up Area	1100	5.45 K	1100.0	5.0	60.0
4	52.0	27.54 K	4	4.0	Build Up Area	900	5.78 K	900.0	4.0	52.0
...	...	...	...	...	...	...	...	...	...	
3963	13.0	6.88 K	2	2.0	Build Up Area	1500	866	1500.0	2.0	13.0
3964	1.5	74.47 K	8	8.0	Build Up Area	2560	5.86 K	2560.0	8.0	1.5
3965	50.0	26.48 K	5	5.0	Build Up Area	1900	2.63 K	1900.0	5.0	50.0

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Area_Type	Total_Sq.ft	Price_per_sq.ft	total	bhk	
3966	1.1	54.61 K	5	5.0	Build Up Area	2600	4.23 K	2600.0	5.0	1.1
3967	30.0	15.89 K	2	2.0	Build Up Area	950	3.16 K	950.0	2.0	30.0

3968 rows × 12 columns

In [205…

```
b.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3968 entries, 0 to 3967
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Flat_Price       3968 non-null   object 
 1   EMI_Starts       3968 non-null   object 
 2   BHK              3968 non-null   object 
 3   css-11nfaq3      3968 non-null   float64
 4   Area_Type        3968 non-null   object 
 5   Total_Sq.ft      3968 non-null   object 
 6   Price_per_sq.ft  3968 non-null   object 
 7   total            3966 non-null   float64
 8   bhk              3920 non-null   float64
 9   rs               3968 non-null   float64
 10  emi              3968 non-null   float64
 11  price            3968 non-null   float64
dtypes: float64(6), object(6)
memory usage: 372.1+ KB
```

In [206…

```
b['Flat_Price']=b['Flat_Price'].str[:-2]
#b['Total_Sq.ft']=b['Total_Sq.ft'].str[:-5]
```

In [207…

```
b
```

Out[207…

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Area_Type	Total_Sq.ft	Price_per_sq.ft	total	bhk	
0	8.	4.22 Lacs	6	6.0	Build Up Area	4200	20.24 K	4200.0	6.0	8.5
1	45	23.83 K	3	3.0	Build Up Area	1400	3.21 K	1400.0	3.0	45.0
2	1.3	67.02 K	3	3.0	Build Up Area	2500	5.40 K	2500.0	3.0	1.3
3	60	31.77 K	5	5.0	Build Up Area	1100	5.45 K	1100.0	5.0	60.0
4	52	27.54 K	4	4.0	Build Up Area	900	5.78 K	900.0	4.0	52.0
...	...	...	...	...	...	...	...	...	...	
3963	13	6.88 K	2	2.0	Build Up Area	1500	866	1500.0	2.0	13.0

	Flat_Price	EMI_Starts	BHK	css-11nfaq3	Area_Type	Total_Sq.ft	Price_per_sq.ft	total	bhk	
3964	1.	74.47 K	8	8.0	Build Up Area	2560	5.86 K	2560.0	8.0	1.5
3965	50	26.48 K	5	5.0	Build Up Area	1900	2.63 K	1900.0	5.0	50.0
3966	1.	54.61 K	5	5.0	Build Up Area	2600	4.23 K	2600.0	5.0	1.1
3967	30	15.89 K	2	2.0	Build Up Area	950	3.16 K	950.0	2.0	30.0

3968 rows × 12 columns

In [208…
```
b.isna().sum()
```

Out[208…
```
Flat_Price          0
EMI_Starts          0
BHK                 0
css-11nfaq3         0
Area_Type           0
Total_Sq.ft         0
Price_per_sq.ft     0
total               2
bhk                48
rs                  0
emi                 0
price               0
dtype: int64
```

In [209…
```
b.fillna(35,inplace=True)
```

In [237…
```
c=b.drop(['Flat_Price','EMI_Starts','Price_per_sq.ft','Area_Type','Total_Sq.ft','BHK
c
```

Out[237…

	css-11nfaq3	total	bhk	rs	emi	price
0	6.0	4200.0	6.0	8.50	422000.0	20240.0
1	3.0	1400.0	3.0	45.00	23830.0	3210.0
2	3.0	2500.0	3.0	1.35	67020.0	5400.0
3	5.0	1100.0	5.0	60.00	31770.0	5450.0
4	4.0	900.0	4.0	52.00	27540.0	5780.0
...	...	...	...	...	...	...
3963	2.0	1500.0	2.0	13.00	6880.0	866.0
3964	8.0	2560.0	8.0	1.50	74470.0	5860.0
3965	5.0	1900.0	5.0	50.00	26480.0	2630.0
3966	5.0	2600.0	5.0	1.10	54610.0	4230.0
3967	2.0	950.0	2.0	30.00	15890.0	3160.0

3968 rows × 6 columns

In [238…
```
d=pd.get_dummies(c,dtype=int)
d
```

Out[238…

	css-11nfaq3	total	bhk	rs	emi	price
0	6.0	4200.0	6.0	8.50	422000.0	20240.0
1	3.0	1400.0	3.0	45.00	23830.0	3210.0
2	3.0	2500.0	3.0	1.35	67020.0	5400.0
3	5.0	1100.0	5.0	60.00	31770.0	5450.0
4	4.0	900.0	4.0	52.00	27540.0	5780.0
...	...	...	...	...	...	...
3963	2.0	1500.0	2.0	13.00	6880.0	866.0
3964	8.0	2560.0	8.0	1.50	74470.0	5860.0
3965	5.0	1900.0	5.0	50.00	26480.0	2630.0
3966	5.0	2600.0	5.0	1.10	54610.0	4230.0
3967	2.0	950.0	2.0	30.00	15890.0	3160.0

3968 rows × 6 columns

In [239…
```
list(d)
```

Out[239…
```
['css-11nfaq3', 'total', 'bhk', 'rs', 'emi', 'price']
```

In [240…
```
y=d['price']
y
```

Out[240…
```
0       20240.0
1        3210.0
2        5400.0
3        5450.0
4        5780.0
         ...
3963      866.0
3964     5860.0
3965     2630.0
3966     4230.0
3967     3160.0
Name: price, Length: 3968, dtype: float64
```

In [241…
```
x=d.drop(['price'],axis=1)
x
```

Out[241…

	css-11nfaq3	total	bhk	rs	emi
0	6.0	4200.0	6.0	8.50	422000.0
1	3.0	1400.0	3.0	45.00	23830.0
2	3.0	2500.0	3.0	1.35	67020.0

	css-11nfaq3	total	bhk	rs	emi
3	5.0	1100.0	5.0	60.00	31770.0
4	4.0	900.0	4.0	52.00	27540.0
...	...	...	...	...	...
3963	2.0	1500.0	2.0	13.00	6880.0
3964	8.0	2560.0	8.0	1.50	74470.0
3965	5.0	1900.0	5.0	50.00	26480.0
3966	5.0	2600.0	5.0	1.10	54610.0
3967	2.0	950.0	2.0	30.00	15890.0

3968 rows × 5 columns

Linear Regression

In [242...]
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [243...]
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(x_train,y_train)
```

Out[243...]
```
▾ LinearRegression

LinearRegression()
```

In [244...]
```
ypred=reg.predict(x_test)
ypred
```

Out[244...]
```
array([5382.83289678, 5342.5902715 , 5407.43975008, ..., 4952.84079772,
       5662.43429697, 4659.89928401])
```

In [245...]
```
from sklearn.metrics import r2_score
r2_score(y_test,ypred)
```

Out[245...]
```
0.47051851206690365
```

In [246...]
```
from sklearn.metrics import mean_squared_error
mean_squared_error(ypred,y_test)
```

Out[246...]
```
12370167.082565164
```

In [247...]
```
Results= pd.DataFrame(columns=['Actual','Predicted'])
Results['Actual']=y_test
Results['Predicted']=ypred
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(10)
```

Out[247...

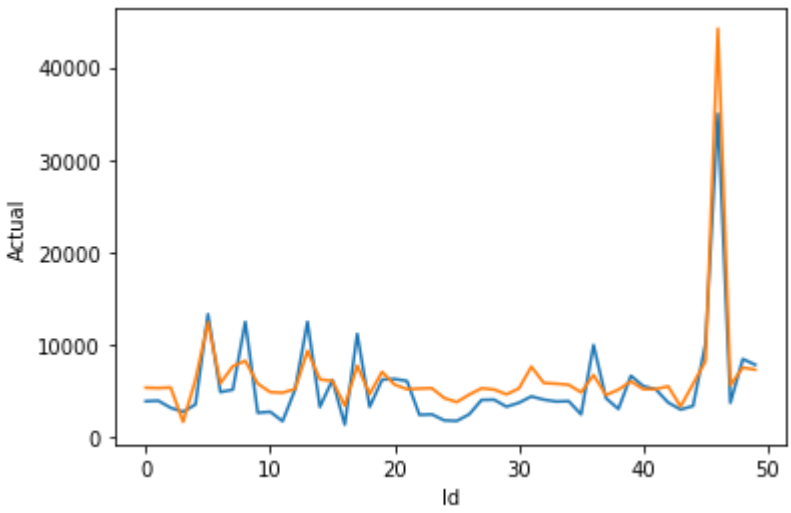
	index	Actual	Predicted	Id
0	2700	3920.0	5382.832897	0
1	2037	3980.0	5342.590272	1
2	315	3210.0	5407.439750	2
3	598	2780.0	1695.688109	3
4	3101	3570.0	6260.446319	4
5	2335	13330.0	12454.527455	5
6	318	4910.0	5867.097451	6
7	1398	5170.0	7699.935171	7
8	3297	12500.0	8287.203340	8
9	3378	2670.0	5819.749024	9

In [248...

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='Id',y='Actual',data=Results.head(50))
sns.lineplot(x='Id',y='Predicted',data=Results.head(50))
plt.plot()
```

Out[248...

```
[]
```



Ridge Regression

In [249...

```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Ridge
alpha = [1e-15, 1e-10, 1e-8, 1e-4, 1e-3,1e-2, 1, 5, 10, 20,30]
ridge=Ridge()
parameters={'alpha':alpha}
regressor=GridSearchCV(ridge,parameters)
regressor.fit(x_train,y_train)
```

Out[249…

GridSearchCV
estimator: Ridge
Ridge

In [250…
```
regressor.best_params_
```

Out[250…
```
{'alpha': 1e-15}
```

In [251…
```
ridge=Ridge(1e-15)
ridge.fit(x_train,y_train)
pred_ridge=ridge.predict(x_test)
pred_ridge
```

Out[251…
```
array([5382.83289678, 5342.5902715 , 5407.43975008, ..., 4952.84079772,
       5662.43429697, 4659.89928401])
```

In [252…
```
r2_score(y_test,pred_ridge)
```

Out[252…
```
0.4705185120669253
```

Lasso Regression

In [253…
```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Lasso
lasso=Lasso()
alpha = [1e-15, 1e-10, 1e-8, 1e-4, 1e-3,1e-2, 1, 5, 10, 20,30]
parameters={'alpha':alpha}
l_reg=GridSearchCV(lasso,parameters)
l_reg.fit(x_train,y_train)
```

Out[253…

GridSearchCV
estimator: Lasso
Lasso

In [254…
```
l_reg.best_params_
```

Out[254…
```
{'alpha': 1e-15}
```

In [255…
```
lasso=Lasso(1e-15)
lasso.fit(x_train,y_train)
pred_lasso=lasso.predict(x_test)
pred_lasso
```

Out[255…
```
array([5382.83289678, 5342.5902715 , 5407.43975008, ..., 4952.84079772,
       5662.43429697, 4659.89928401])
```

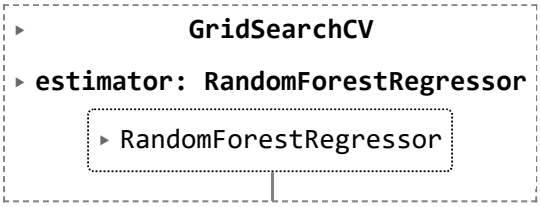
In [256…
```
r2_score(y_test,pred_lasso)
```

Out[256…
```
0.4705185120669252
```

RandomForest Regression

In [258…
```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestRegressor
reg=RandomForestRegressor()
n_estimators=[25,50,75,100,125,150,175,200]
criterion=['squared_error']
max_depth=[3,5,10]
parameters={'n_estimators': n_estimators,'criterion':criterion,'max_depth':max_depth
rfc_reg = GridSearchCV(reg, parameters)
rfc_reg.fit(x_train,y_train)
```

Out[258…



In [259…
```
rfc_reg.best_params_
```

Out[259…
```
{'criterion': 'squared_error', 'max_depth': 10, 'n_estimators': 125}
```

In [261…
```
reg=RandomForestRegressor(n_estimators=125,criterion='squared_error',max_depth=10)
reg.fit(x_train,y_train)
ypred=reg.predict(x_test)
ypred
```

Out[261…
```
array([3899.46279292, 3952.03653883, 3119.07769123, ..., 2946.87598071,
       4812.53567032, 2766.95056603])
```

In [262…
```
from sklearn.metrics import r2_score
r2_score(y_test,ypred)
```

Out[262…
```
0.9868867587586051
```

In [ ]: