

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing, svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
In [4]: df=pd.read_csv(r"C:\Users\jangidi veena\OneDrive\Documents\jupyter\used_cars_d
df
```

Out[4]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Se
...
7248	7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	
7249	7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	
7250	7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	
7251	7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	
7252	7252	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	

7253 rows × 14 columns



In [5]: `df.head()`

Out[5]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second

In [6]: `df.describe()`

Out[6]:

	S.No.	Year	Kilometers_Driven	Seats	Price
count	7253.000000	7253.000000	7.253000e+03	7200.000000	6019.000000
mean	3626.000000	2013.365366	5.869906e+04	5.279722	9.479468
std	2093.905084	3.254421	8.442772e+04	0.811660	11.187917
min	0.000000	1996.000000	1.710000e+02	0.000000	0.440000
25%	1813.000000	2011.000000	3.400000e+04	5.000000	3.500000
50%	3626.000000	2014.000000	5.341600e+04	5.000000	5.640000
75%	5439.000000	2016.000000	7.300000e+04	5.000000	9.950000
max	7252.000000	2019.000000	6.500000e+06	10.000000	160.000000

In [7]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7253 entries, 0 to 7252
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   S.No.                 7253 non-null   int64
1   Name                  7253 non-null   object
2   Location              7253 non-null   object
3   Year                  7253 non-null   int64
4   Kilometers_Driven     7253 non-null   int64
5   Fuel_Type             7253 non-null   object
6   Transmission          7253 non-null   object
7   Owner_Type            7253 non-null   object
8   Mileage               7251 non-null   object
9   Engine               7207 non-null   object
10  Power                7207 non-null   object
11  Seats                7200 non-null   float64
12  New_Price            1006 non-null   object
13  Price                6019 non-null   float64
dtypes: float64(2), int64(3), object(9)
memory usage: 793.4+ KB
```

In [8]: df.tail()

Out[8]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_1
7248	7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	
7249	7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	
7250	7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	
7251	7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	1
7252	7252	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	

```
In [9]: df.isna().any()
```

```
Out[9]: S.No.          False
        Name          False
        Location       False
        Year           False
        Kilometers_Driven False
        Fuel_Type       False
        Transmission    False
        Owner_Type      False
        Mileage         True
        Engine          True
        Power           True
        Seats           True
        New_Price       True
        Price           True
        dtype: bool
```

```
In [10]: df.shape
```

```
Out[10]: (7253, 14)
```

```
In [11]: df.isnull().sum()
```

```
Out[11]: S.No.          0
        Name          0
        Location       0
        Year           0
        Kilometers_Driven 0
        Fuel_Type       0
        Transmission    0
        Owner_Type      0
        Mileage         2
        Engine          46
        Power           46
        Seats           53
        New_Price       6247
        Price           1234
        dtype: int64
```

```
In [12]: df.fillna(value=0,inplace=True)
```

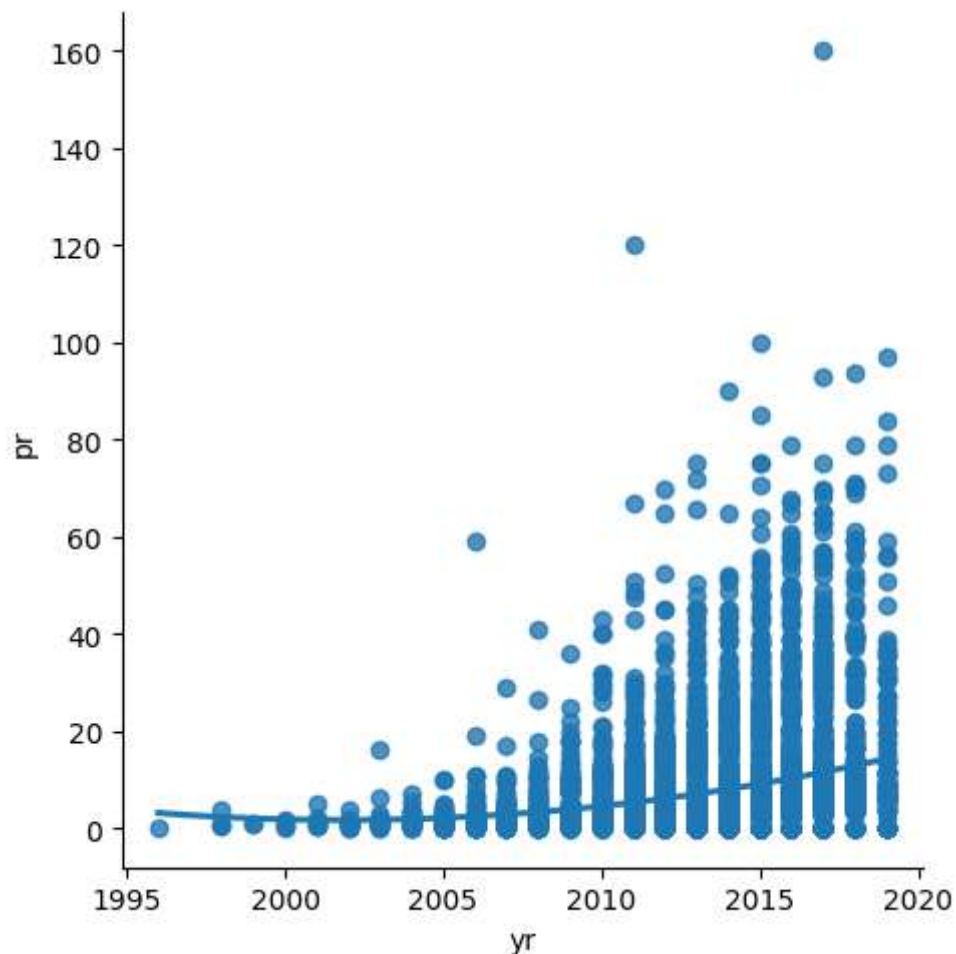
```
In [13]: df.isnull().sum()
```

```
Out[13]: S.No.          0
Name          0
Location      0
Year          0
Kilometers_Driven  0
Fuel_Type     0
Transmission  0
Owner_Type    0
Mileage       0
Engine        0
Power         0
Seats         0
New_Price     0
Price         0
dtype: int64
```

```
In [14]: df=df[['Year','Price']]
df.columns=['yr','pr']
```

```
In [15]: sns.lmplot(x="yr",y="pr",data=df,order=2,ci=None)
```

```
Out[15]: <seaborn.axisgrid.FacetGrid at 0x1a4cead9510>
```



```
In [16]: df.head(10)
```

```
Out[16]:
```

	yr	pr
0	2010	1.75
1	2015	12.50
2	2011	4.50
3	2012	6.00
4	2013	17.74
5	2012	2.35
6	2013	3.50
7	2016	17.50
8	2013	5.20
9	2012	1.95

```
In [18]: df.describe()
```

```
Out[18]:
```

	yr	pr
count	7253.000000	7253.000000
mean	2013.365366	7.866665
std	3.254421	10.796286
min	1996.000000	0.000000
25%	2011.000000	2.290000
50%	2014.000000	4.650000
75%	2016.000000	8.400000
max	2019.000000	160.000000

```
In [19]: df.fillna(method='ffill',inplace=True)
```

C:\Users\jangidi veena\AppData\Local\Temp\ipykernel_14680\4116506308.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df.fillna(method='ffill',inplace=True)
```

```
In [20]: x=np.array(df['yr']).reshape(-1,1)
y=np.array(df['pr']).reshape(-1,1)
```

```
In [21]: df.dropna(inplace=True)
```

C:\Users\jangidi veena\AppData\Local\Temp\ipykernel_14680\1379821321.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

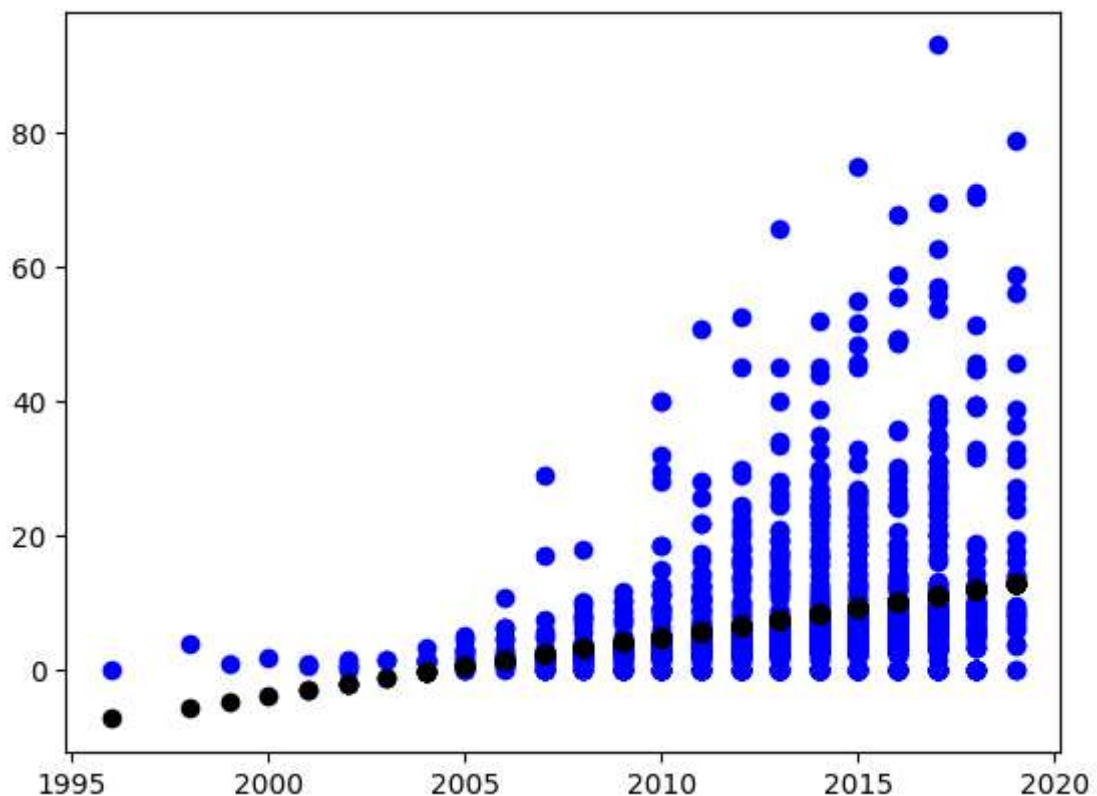
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df.dropna(inplace=True)
```

```
In [22]: X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
regr=LinearRegression()
regr.fit(X_train,y_train)
print(regr.score(X_test,y_test))
```

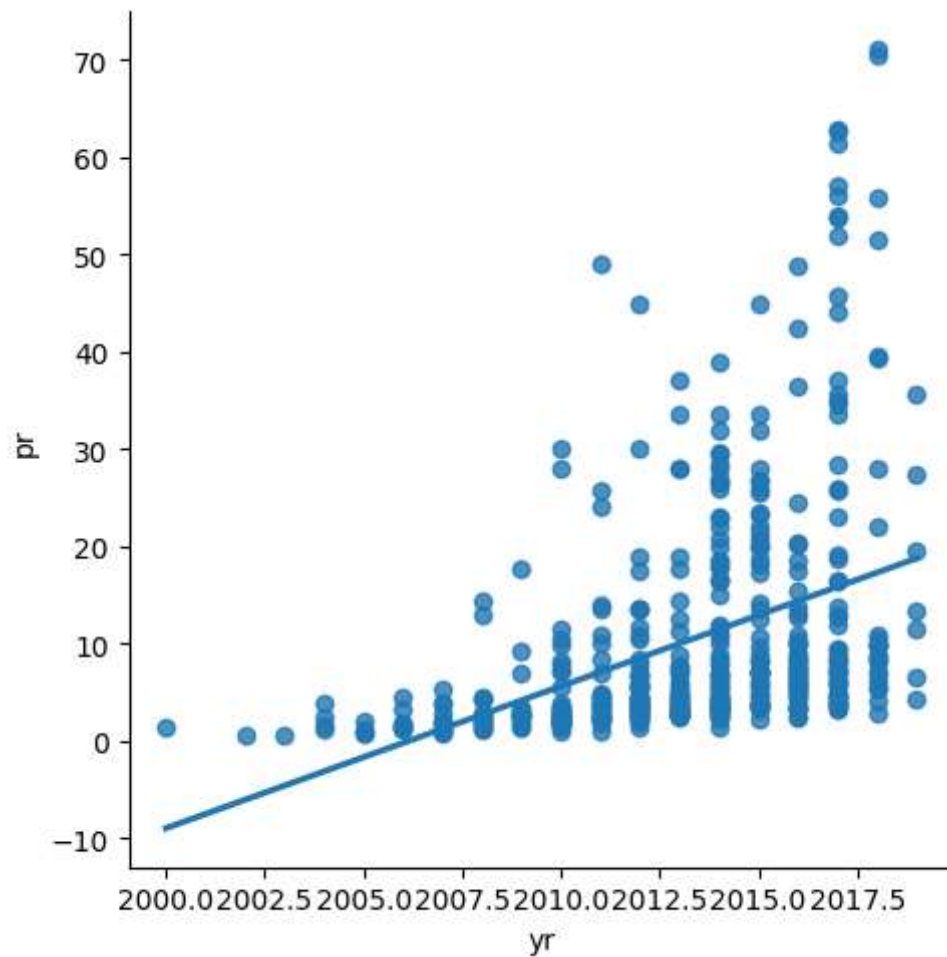
0.07278676245503346

```
In [23]: y_pred=regr.predict(X_test)
plt.scatter(X_test,y_test,color='b')
plt.scatter(X_test,y_pred,color='k')
plt.show()
```



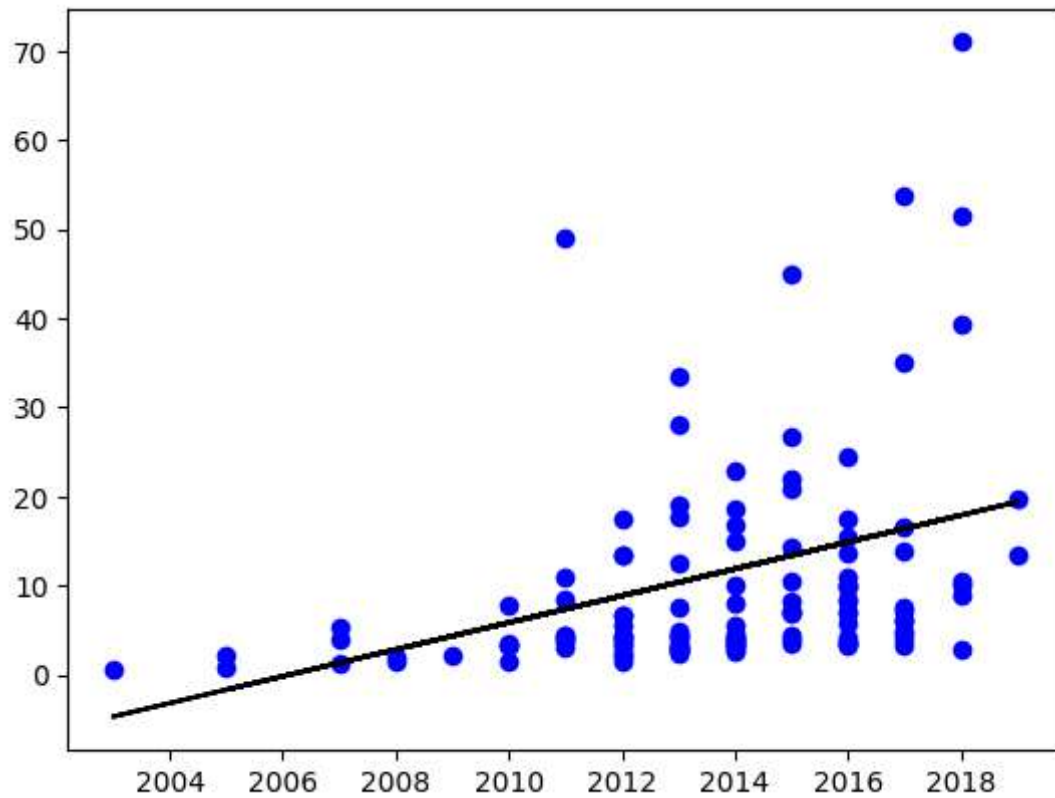
```
In [24]: df500=df[:][:500]  
sns.lmplot(x="yr",y="pr",data=df500,order=1,ci=None)
```

```
Out[24]: <seaborn.axisgrid.FacetGrid at 0x1a4cfc09250>
```




```
In [26]: df500.fillna(method='ffill',inplace=True)
X=np.array(df500['yr']).reshape(-1,1)
y=np.array(df500['pr']).reshape(-1,1)
df500.dropna(inplace=True)
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25)
regr=LinearRegression()
regr.fit(X_train,y_train)
print("Regression:",regr.score(X_test,y_test))
y_pred=regr.predict(X_test)
plt.scatter(X_test,y_test,color='b')
plt.plot(X_test,y_pred,color='k')
plt.show()
```

Regression: 0.1002500397691709



```
In [ ]: from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model=LinearRegression()
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
r2=r2_score(y_test,y_pred)
print("R2 score:",r2)
```