

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

```
In [2]: 1 df=pd.read_csv(r"C:\Users\91949\Downloads\used_cars_data.csv")
2 df
```

Out[2]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	NaN
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8.61 Lakh
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN
...
7248	7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	First	20.54 kmpl	1598 CC	103.6 bhp	5.0	NaN
7249	7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	First	17.21 kmpl	1197 CC	103.6 bhp	5.0	NaN
7250	7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5.0	NaN
7251	7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	Third	17.2 kmpl	1197 CC	103.6 bhp	5.0	NaN
7252	7252	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	First	10.0 kmpl	2148 CC	170 bhp	5.0	NaN

7253 rows × 14 columns

```
In [4]: 1 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 [5]:

1 df.describe()

Out[5]:

	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 [6]:

1 df.isnull().any()

Out[6]:

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 [7]:

1 df.isnull().sum()

Out[7]:

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 [8]:

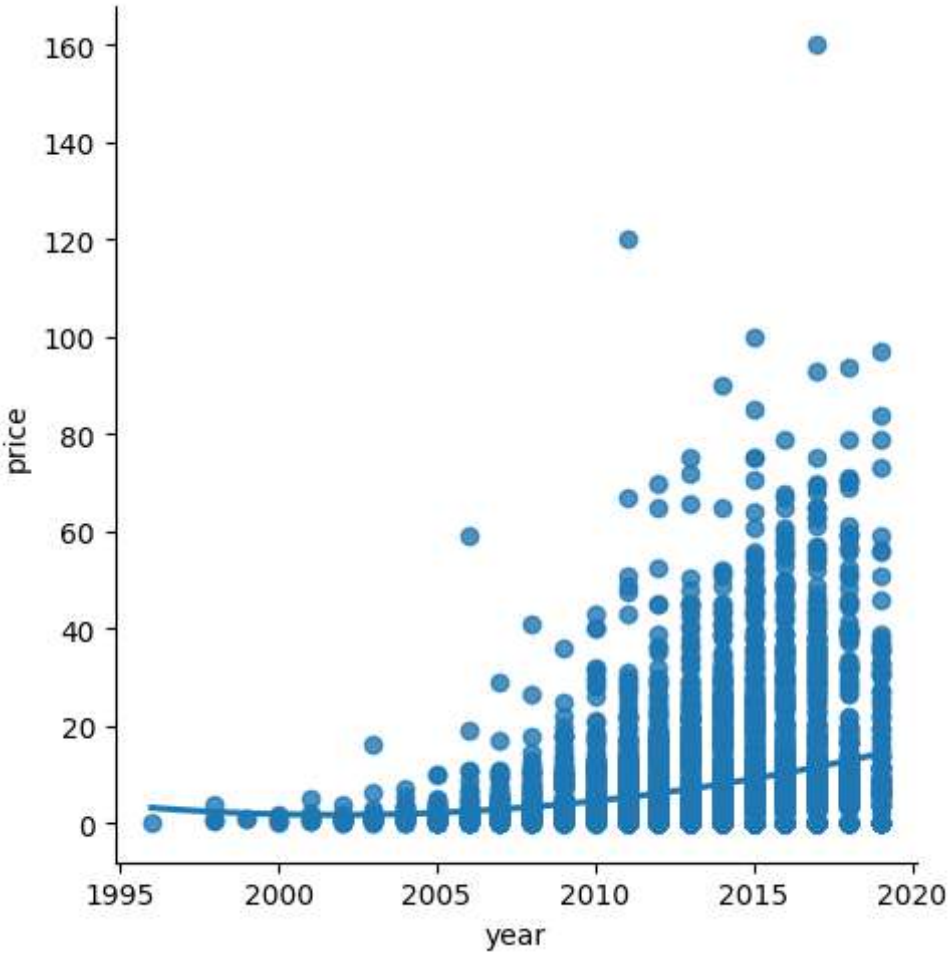
1 df.fillna(value=0,inplace=True)

In [9]:

1 df=df[['Year','Price']]
2 *#Taking only the selected two attributes from the dataset*
3 df.columns=['year','price']
4 *#Renaming the columns for easier writing of the code*

```
In [10]: 1 sns.lmplot(x='year',y='price',data=df,order=2,ci=None)
```

Out[10]: <seaborn.axisgrid.FacetGrid at 0x182f2bdccd0>



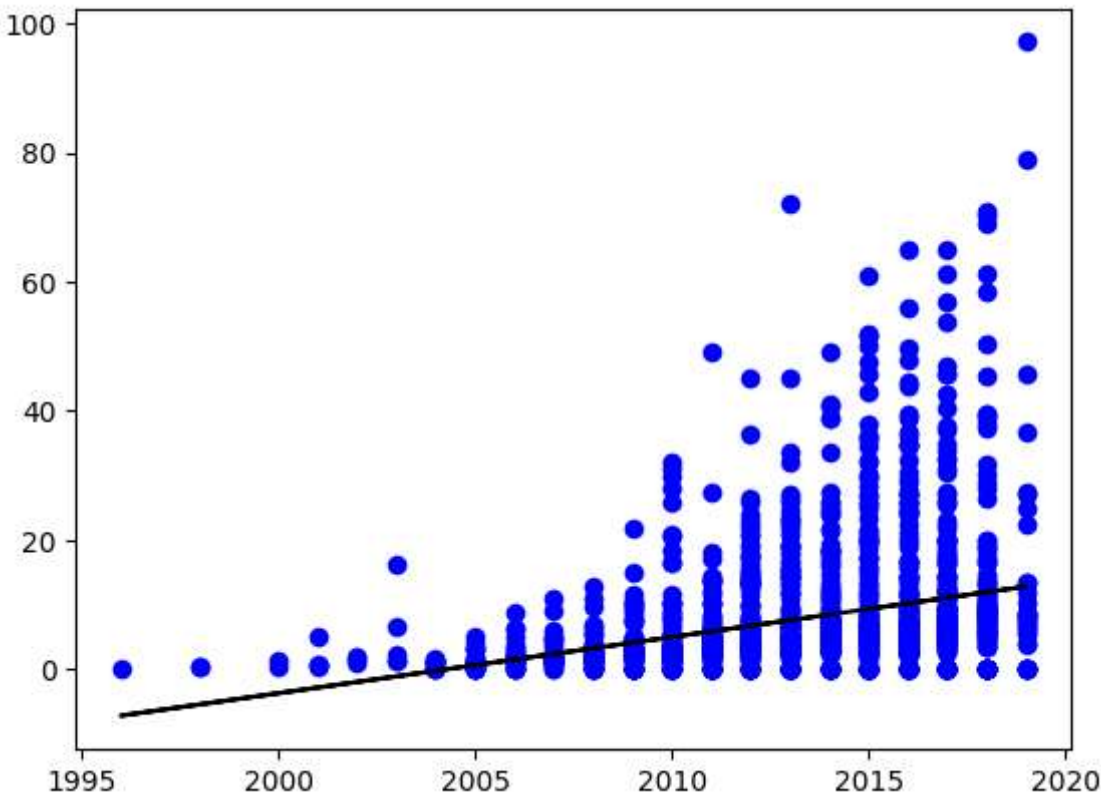
```
In [11]: 1 x=np.array(df['year']).reshape(-1,1)
```

```
In [12]: 1 y=np.array(df['price']).reshape(-1,1)
```

```
In [13]: 1 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
2 reg=LinearRegression()
3 reg.fit(x_train,y_train)
4 print(reg.score(x_test,y_test))
```

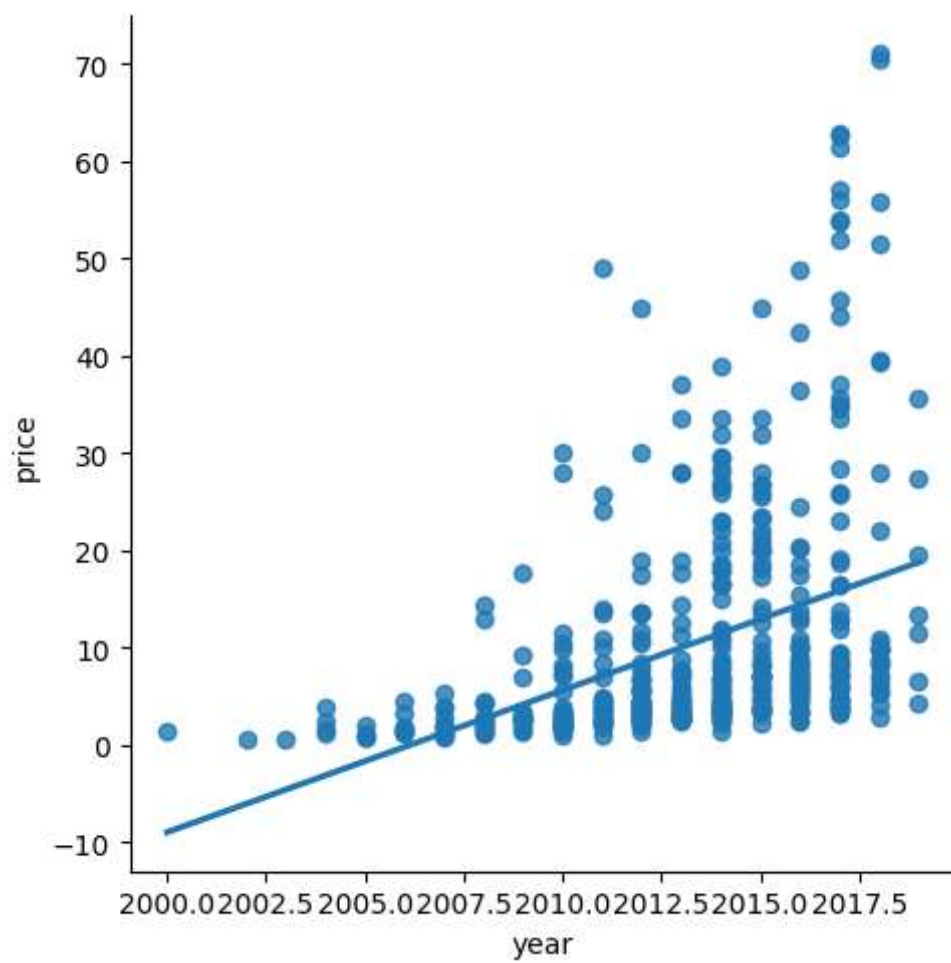
0.08126424995335535

```
In [14]: 1 y_pred=reg.predict(x_test)
2 plt.scatter(x_test,y_test,color='b')
3 plt.plot(x_test,y_pred,color='k')
4 plt.show()
```



```
In [15]: 1 df500=df[:][:500]
2 sns.lmplot(x='year',y='price',data=df500,order=1,ci=None)
```

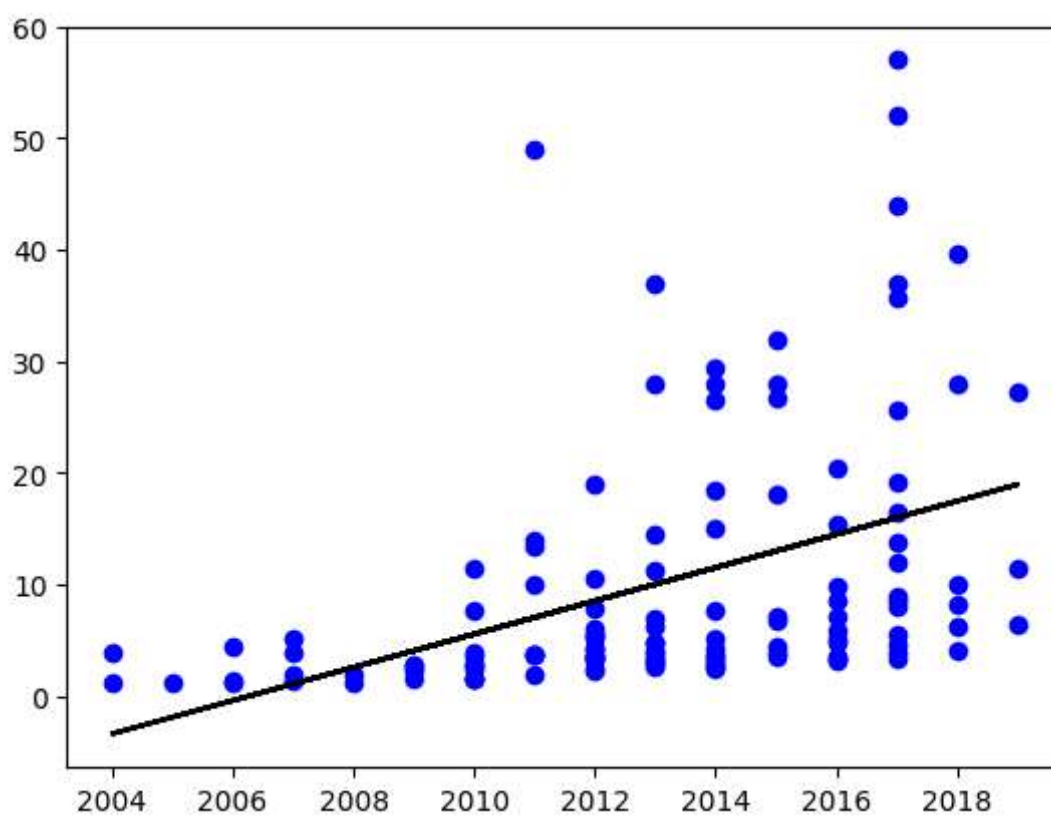
Out[15]: <seaborn.axisgrid.FacetGrid at 0x182dfaad410>



```
In [16]: 1 df500.fillna(method='ffill',inplace=True)
2 x=np.array(df500['year']).reshape(-1,1)
3 y=np.array(df500['price']).reshape(-1,1)
4 df500.dropna(inplace=True)
5 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
6 reg=LinearRegression()
7 reg.fit(x_train,y_train)
8 print("Regression:",reg.score(x_test,y_test))
9 y_pred=reg.predict(x_test)
10 plt.scatter(x_test,y_test,color='b')
11 plt.plot(x_test,y_pred,color='k')
12 plt.show
```

Regression: 0.16094700103342297

Out[16]: <function matplotlib.pyplot.show(close=None, block=None)>



```
In [17]: 1 from sklearn.linear_model import LinearRegression
2 from sklearn.metrics import r2_score
3 model=LinearRegression()
4 model.fit(x_train,y_train)
5 y_pred=model.predict(x_test)
6 r2=r2_score(y_test,y_pred)
7 print("R2 score: ",r2)
```

R2 score: 0.16094700103342297