

In [58]:

```
import pandas as pd
import numpy as np
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 [59]:

```
df=pd.read_csv(r"C:\Users\shaha\OneDrive\Desktop\Excel\used_cars_data.csv")
df
```

Out[59]:

	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
		Hyundai EON						

In [60]:

```
df.head()
```

Out[60]:

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

In [61]:

```
df.tail()
```

Out[61]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Own
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	

In [62]:

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 [63]:

df.describe()

Out[63]:

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

In [64]:

df.columns

Out[64]:

```
Index(['S.No.', 'Name', 'Location', 'Year', 'Kilometers_Driven', 'Fuel_Type',
      'Transmission',
      'Owner_Type', 'Mileage', 'Engine', 'Power', 'Seats', 'New_Price',
      'Price'],
      dtype='object')
```

In [65]:

```
df.shape
```

Out[65]:

```
(7253, 14)
```

In [66]:

```
df.isna().sum()
```

Out[66]:

```
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 [67]:

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

Out[67]:

```
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 [68]:

```
df['Price'].value_counts()
```

Out[68]:

Price	
4.50	88
5.50	84
3.50	82
4.25	73
3.25	71
3.00	68
6.50	64
2.50	63
4.00	56
4.75	53
3.75	52
6.00	52
5.25	51
5.00	47
7.50	46
2.75	46
5.75	45

In [69]:

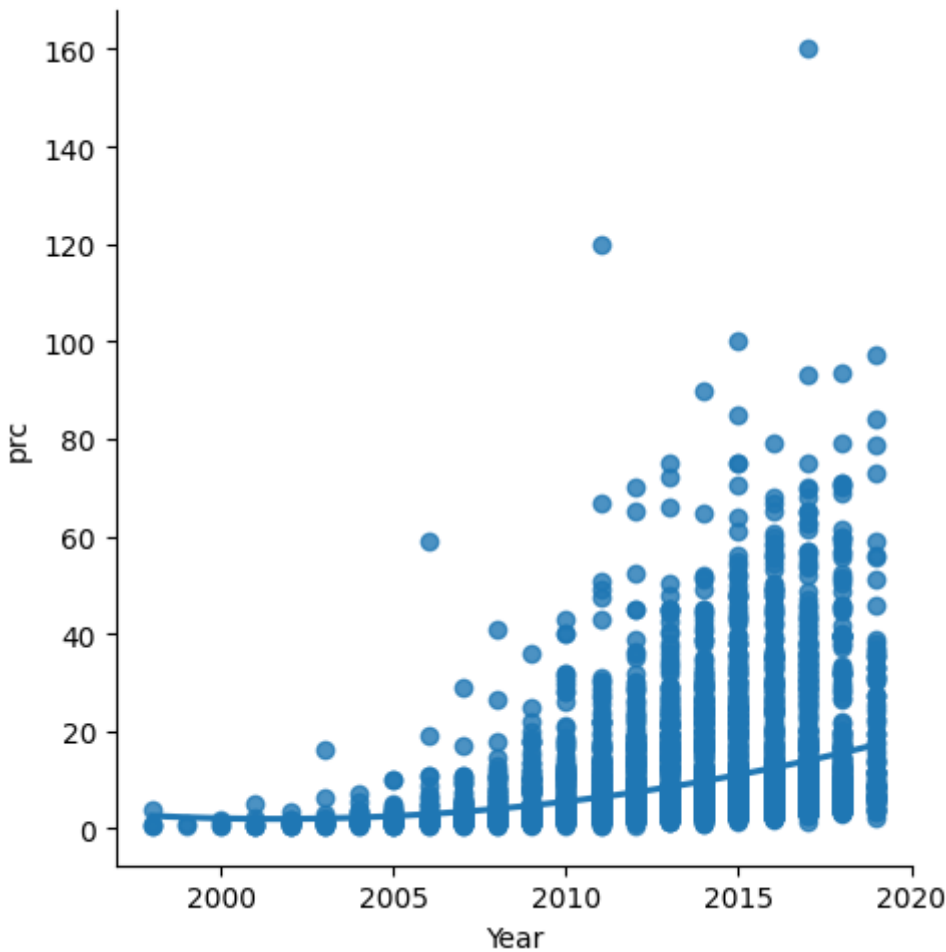
```
df=df[['Year','Price']]  
df.columns=['Year','prc']
```

In [70]:

```
sns.lmplot(x='Year',y='prc',data=df,order=2,ci=None)
```

Out[70]:

```
<seaborn.axisgrid.FacetGrid at 0x181c5d18580>
```



In [71]:

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

C:\Users\shaha\AppData\Local\Temp\ipykernel\_22724\4116506308.py:1: Setting WithCopyWarning:

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) ([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 [72]:

```
x = np.array(df['Year']).reshape(-1,1)
y = np.array(df['prc']).reshape(-1,1)
```

In [73]:

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

C:\Users\shaha\AppData\Local\Temp\ipykernel\_22724\658506679.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) ([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 [74]:

```
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))
```

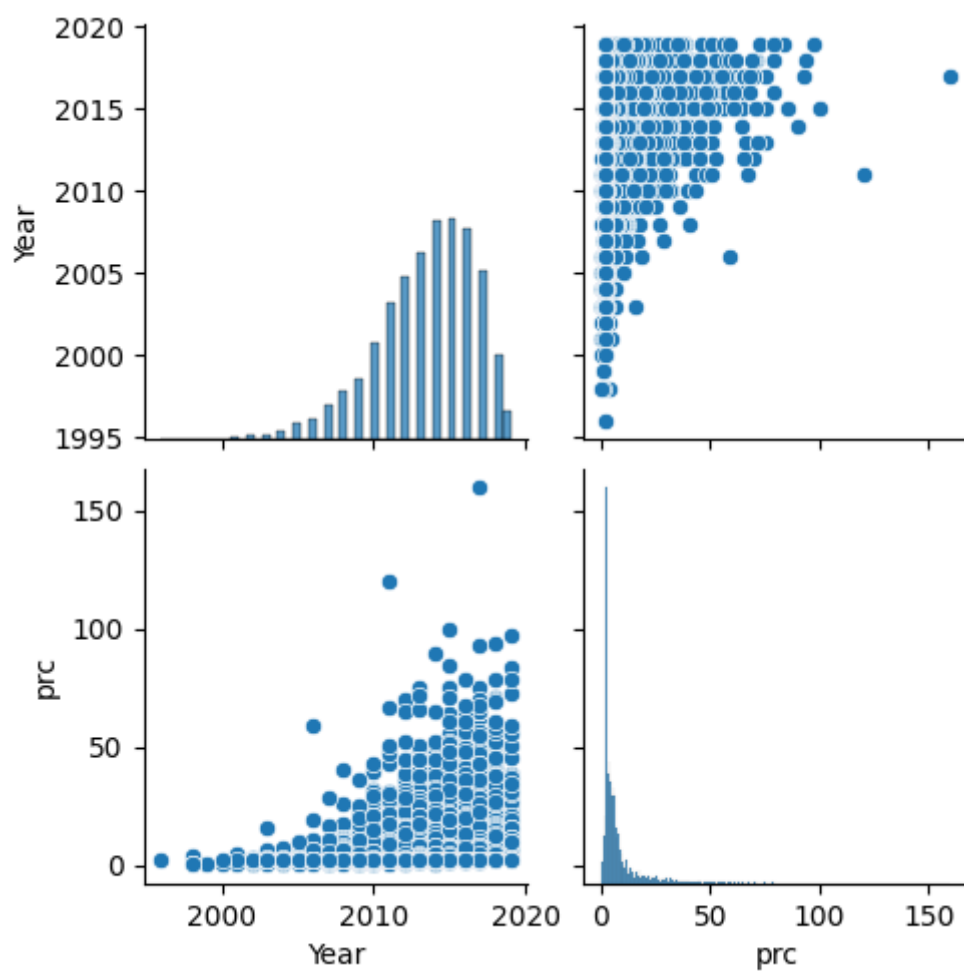
0.0717829532789932

In [75]:

```
sns.pairplot(df)
```

Out[75]:

<seaborn.axisgrid.PairGrid at 0x181c5d9ff10>



In [ ]: