Linear Regression

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn import preprocessing,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [2]:
```
df=pd.read_csv(r"C:\Users\yoshitha lakshmi\OneDrive\Desktop\python\used_cars_data.csv")
df
```

Out[2]:

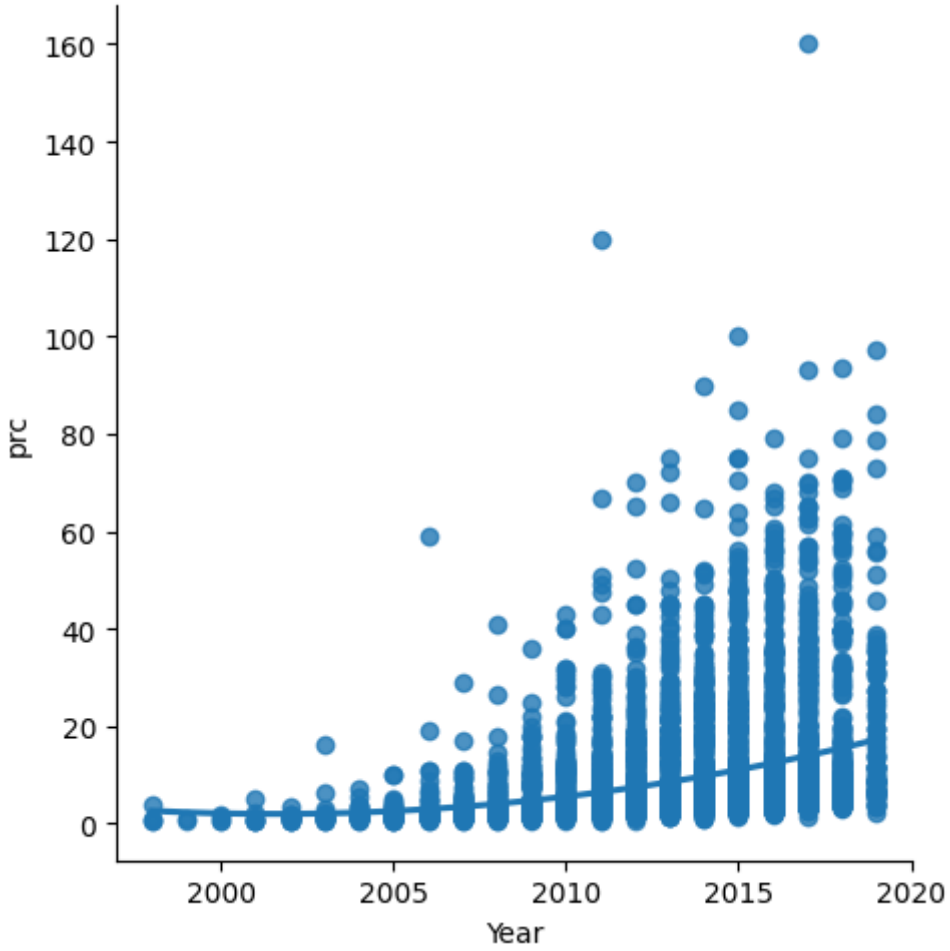
S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	NaN	1.75
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00
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	17.74
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	First	20.54 kmpl	1598 CC	103.6 bhp	5.0	NaN	NaN
7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	First	17.21 kmpl	1197 CC	103.6 bhp	5.0	NaN	NaN
7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5.0	NaN	NaN
7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	Third	17.2 kmpl	1197 CC	103.6 bhp	5.0	NaN	NaN
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	NaN

ws × 14 columns

In [3]:
```
df=df[['Year','Price']]
df.columns=['Year','prc']
```

In [4]:
```
1 sns.lmplot(x='Year',y='prc',data=df,order=2,ci=None)
```

Out[4]: <seaborn.axisgrid.FacetGrid at 0x2d686084be0>



In [5]:
```
1 df.describe()
```

Out[5]:

	Year	prc
count	7253.000000	6019.000000
mean	2013.365366	9.479468
std	3.254421	11.187917
min	1996.000000	0.440000
25%	2011.000000	3.500000
50%	2014.000000	5.640000
75%	2016.000000	9.950000
max	2019.000000	160.000000

In [6]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7253 entries, 0 to 7252
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Year    7253 non-null   int64  
 1   prc     6019 non-null   float64
dtypes: float64(1), int64(1)
memory usage: 113.5 KB
```

In [7]:
```
1 df.fillna(method='ffill',inplace=True)
```

```
C:\Users\yoshitha lakshmi\AppData\Local\Temp\ipykernel_30228\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 [8]:
```
1 x = np.array(df['Year']).reshape(-1,1)
2 y = np.array(df['prc']).reshape(-1,1)
```

In [9]:
```
1 df.dropna(inplace=True)
```

```
C:\Users\yoshitha lakshmi\AppData\Local\Temp\ipykernel_30228\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 [10]:
```
1 x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25)
2 regr = LinearRegression()
3 regr.fit(x_train,y_train)
4 print(regr.score(x_test,y_test))
```

```
0.07664483707100178
```

In [12]:
```
1 sns.pairplot(df)
```

Out[12]: <seaborn.axisgrid.PairGrid at 0x2d686086140>