

PROBLEM STATEMENT

To predict and analyze which fuel giving high mileage.By using linear regression

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 [2]:

```
df=pd.read_csv(r"C:\Users\Teju\Downloads\used_cars_data.csv")
df
```

Out[2]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owr
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	
...
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 [3]:

```
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 [4]:

```
df.head()
```

Out[4]:

	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 [5]:

```
df.tail()
```

Out[5]:

	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 [7]:

```
df.shape
```

Out[7]:

(7253, 14)

In [8]:

```
df.describe
```

Out[8]:

```
<bound method NDFrame.describe of      S.No.
Name      Location \
0          0          Maruti Wagon R LXI CNG      Mumbai
1          1      Hyundai Creta 1.6 CRDi SX Option      Pune
2          2          Honda Jazz V      Chennai
3          3      Maruti Ertiga VDI      Chennai
4          4      Audi A4 New 2.0 TDI Multitronic      Coimbatore
...      ...      ...      ...
7248      7248      Volkswagen Vento Diesel Trendline      Hyderabad
7249      7249      Volkswagen Polo GT TSI      Mumbai
7250      7250      Nissan Micra Diesel XV      Kolkata
7251      7251      Volkswagen Polo GT TSI      Pune
7252      7252      Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...      Kochi
```

```
      Year  Kilometers_Driven  Fuel_Type  Transmission  Owner_Type      Mileag
e \
0      2010          72000      CNG      Manual      First      26.6 km/k
g
1      2015          41000      Diesel      Manual      First      19.67 kmp
l
2      2011          46000      Petrol      Manual      First      18.2 kmp
l
3      2012          87000      Diesel      Manual      First      20.77 kmp
l
4      2013          40670      Diesel      Automatic      Second      15.2 kmp
l
...      ...      ...      ...      ...      ...
...
7248      2011          89411      Diesel      Manual      First      20.54 kmp
l
7249      2015          59000      Petrol      Automatic      First      17.21 kmp
l
7250      2012          28000      Diesel      Manual      First      23.08 kmp
l
7251      2013          52262      Petrol      Automatic      Third      17.2 kmp
l
7252      2014          72443      Diesel      Automatic      First      10.0 kmp
l
```

```
      Engine      Power  Seats  New_Price  Price
0      998 CC      58.16 bhp      5.0      NaN      1.75
1      1582 CC      126.2 bhp      5.0      NaN      12.50
2      1199 CC      88.7 bhp      5.0      8.61 Lakh      4.50
3      1248 CC      88.76 bhp      7.0      NaN      6.00
4      1968 CC      140.8 bhp      5.0      NaN      17.74
...      ...      ...      ...      ...      ...
7248      1598 CC      103.6 bhp      5.0      NaN      NaN
7249      1197 CC      103.6 bhp      5.0      NaN      NaN
7250      1461 CC      63.1 bhp      5.0      NaN      NaN
7251      1197 CC      103.6 bhp      5.0      NaN      NaN
7252      2148 CC      170 bhp      5.0      NaN      NaN
```

[7253 rows x 14 columns]>

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 [11]:

```
df=df[['Kilometers_Driven','Price']]
#Changing the column names
df.columns=['kilometers','price']
df
```

Out[11]:

	kilometers	price
0	72000	1.75
1	41000	12.50
2	46000	4.50
3	87000	6.00
4	40670	17.74
...
7248	89411	NaN
7249	59000	NaN
7250	28000	NaN
7251	52262	NaN
7252	72443	NaN

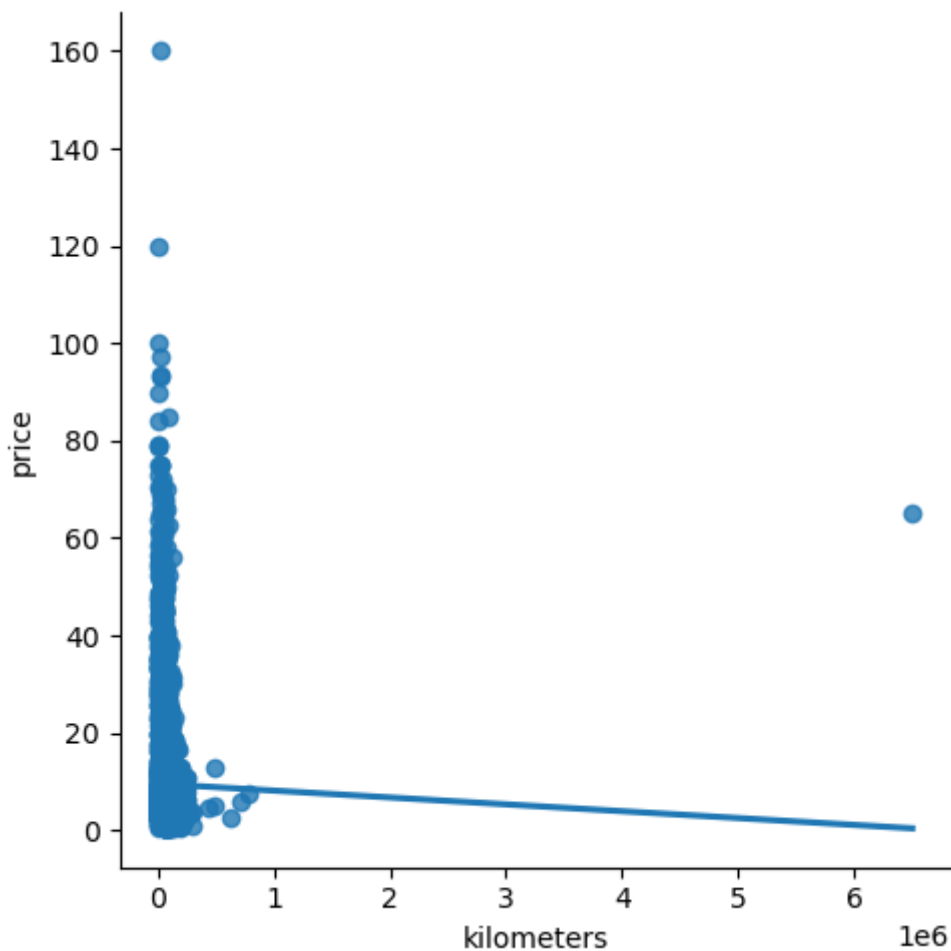
7253 rows × 2 columns

In [12]:

```
sns.lmplot(x="kilometers",y="price",data=df,order=1,ci=None)
```

Out[12]:

<seaborn.axisgrid.FacetGrid at 0x154c6c43fd0>



In [13]:

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

In [14]:

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

In [15]:

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

In [16]:

```
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))
```

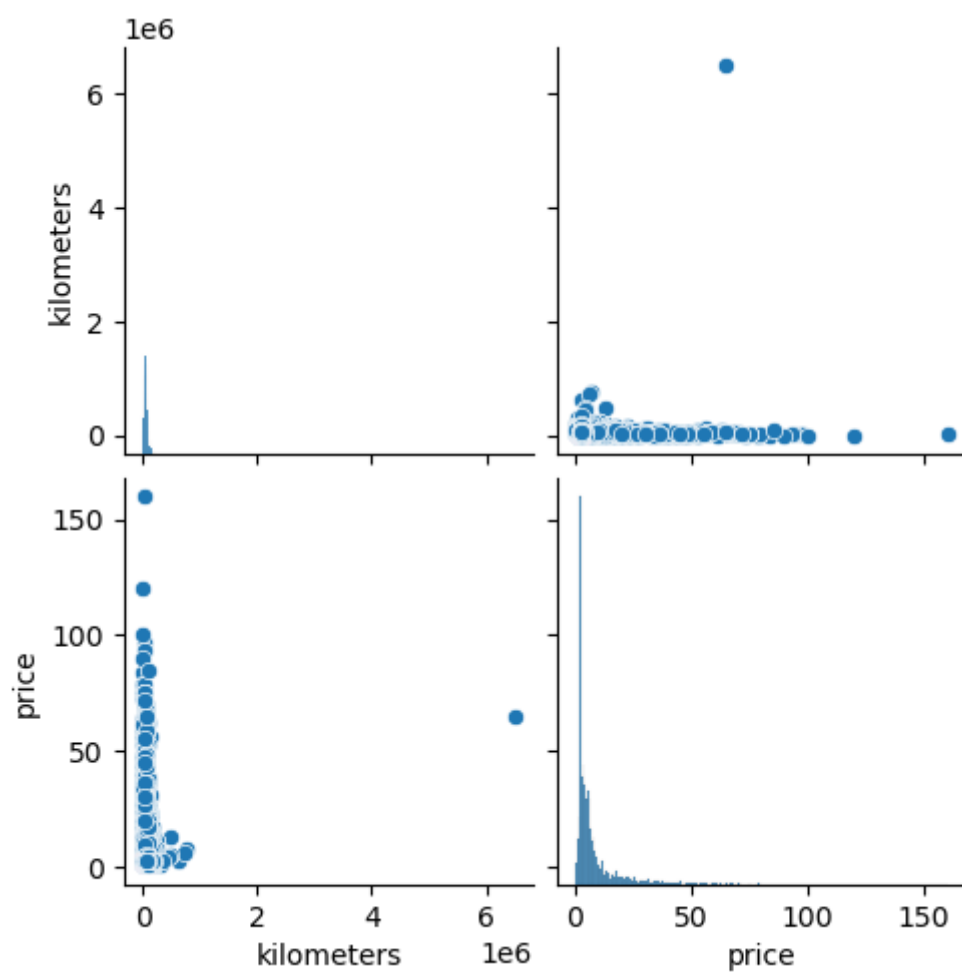
-0.0008618004292264558

In [17]:

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

Out[17]:

<seaborn.axisgrid.PairGrid at 0x154c800f880>



In []: