In [1]:

```
#Importing all the required libraries
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]:

```
#Reading the Dataset
df=pd.read_csv(r"C:\Users\munigreeshma\Downloads\used_cars_data.csv")
df
```

Out[2]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_P
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	I
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	I
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8.61 L
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	I
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	I
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7248	7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	First	20.54 kmpl	1598 CC	103.6 bhp	5.0	I
7249	7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	First	17.21 kmpl	1197 CC	103.6 bhp	5.0	I
7250	7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5.0	I
7251	7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	Third	17.2 kmpl	1197 CC	103.6 bhp	5.0	I
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	I

7253 rows × 14 columns

In [3]:
```
df = df[['Kilometers_Driven','Year']]
#Taking only selected two attributes from dataset
df.columns = ['kil','yr']
```

In [4]:
```
print('This Dataframe contains %d Rows and %d Columns'%(df.shape))
```

```
This Dataframe contains 7253 Rows and 2 Columns
```

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

Out[5]:

	kil	yr
0	72000	2010
1	41000	2015
2	46000	2011
3	87000	2012
4	40670	2013

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

Out[6]:

	kil	yr
7248	89411	2011
7249	59000	2015
7250	28000	2012
7251	52262	2013
7252	72443	2014

In [7]:
```
df.describe()
```

Out[7]:

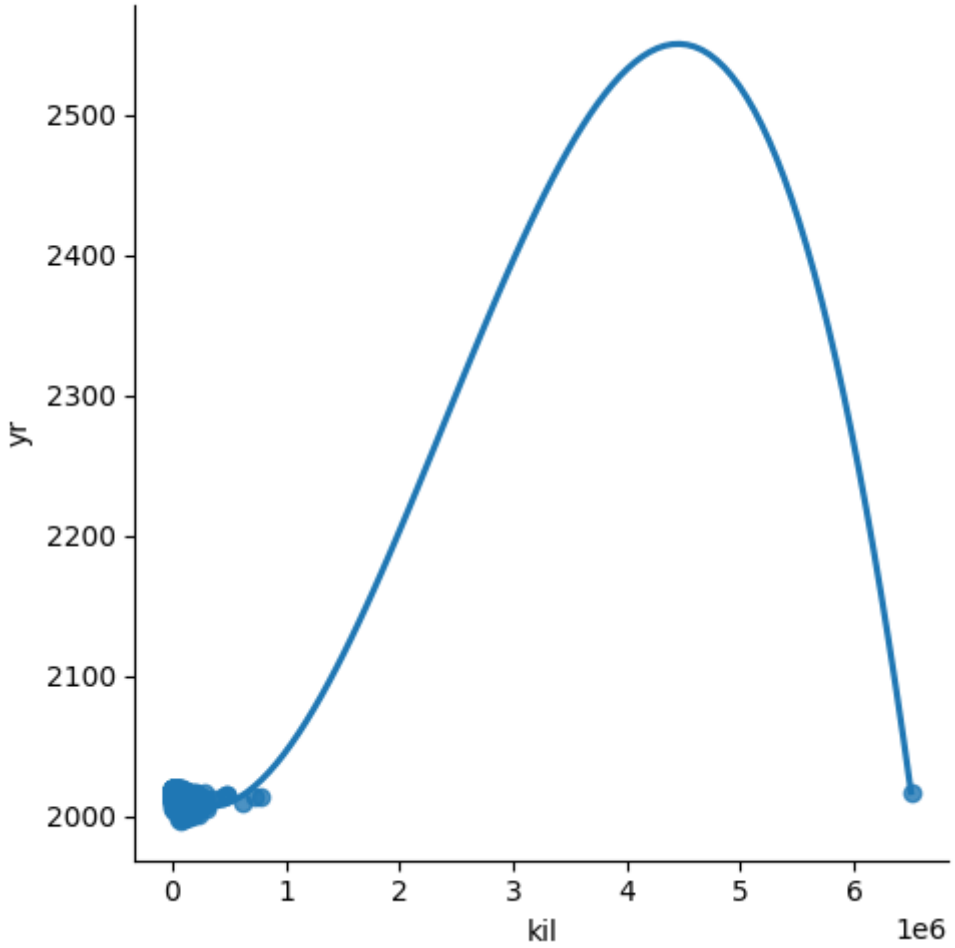
	kil	yr
count	7.253000e+03	7253.000000
mean	5.869906e+04	2013.365366
std	8.442772e+04	3.254421
min	1.710000e+02	1996.000000
25%	3.400000e+04	2011.000000
50%	5.341600e+04	2014.000000
75%	7.300000e+04	2016.000000
max	6.500000e+06	2019.000000

In [8]:
```
df.info()
```

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

In [9]:
```
#Step-3: Exploring the Data Scatter - plotting the data scatter
sns.lmplot(x="kil",y="yr", data = df, order = 3, ci = None)
```

Out[9]: <seaborn.axisgrid.FacetGrid at 0x1a6a383ead0>



In [11]:
```
#Data cleaning - Eliminating NaN OR missing input numbers
df.fillna(method ='ffill', inplace = True)
```

```
C:\Users\munigreeshma\AppData\Local\Temp\ipykernel_15396\3102774782.py:2: 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)
```

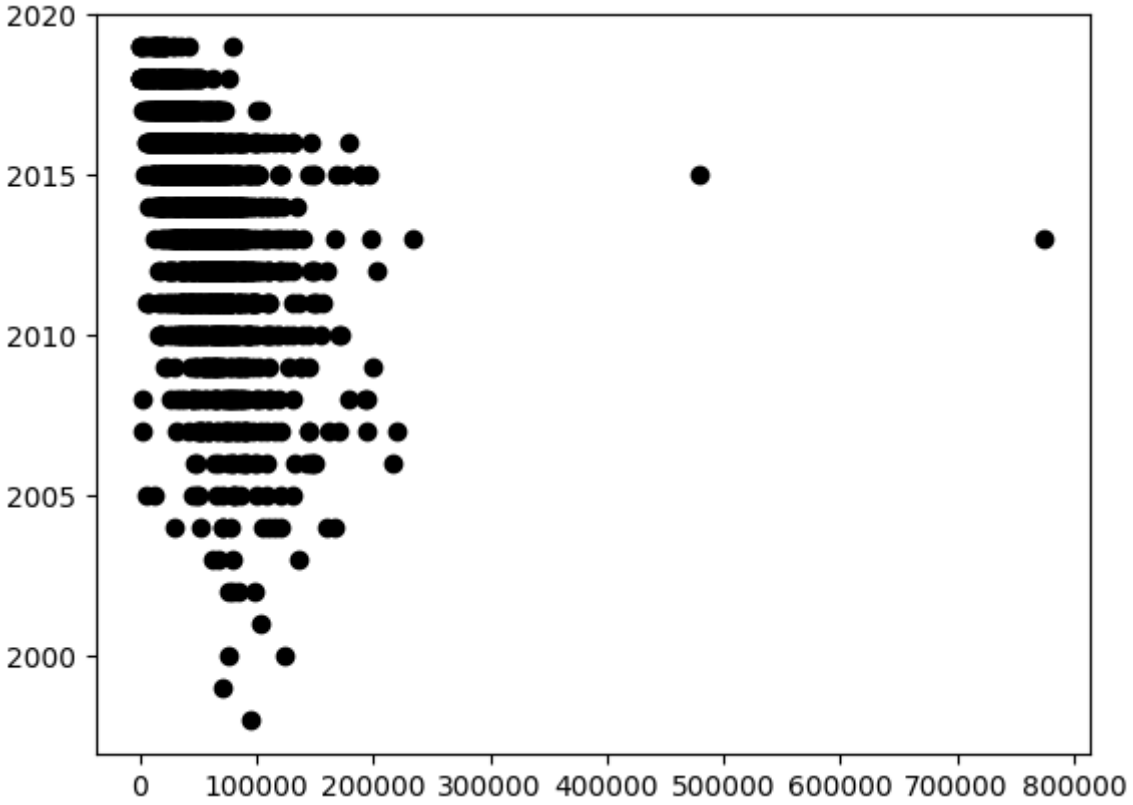
In [12]:
```
#Training Our Model
X = np.array(df['kil']).reshape(-1, 1)
y = np.array(df['yr']).reshape(-1, 1)
#Seperating the data into independent and dependent variables and convert
#Now each dataset contains only one coloumn
```

In [13]:
```
X_train,X_test,y_train,y_test = train_test_split(X, y, test_size = 0.25)
# Splitting the data into training data and test data
regr = LinearRegression()
regr.fit(X_train, y_train)
print(regr.score(X_test, y_test))
```

```
0.04670389595657254
```

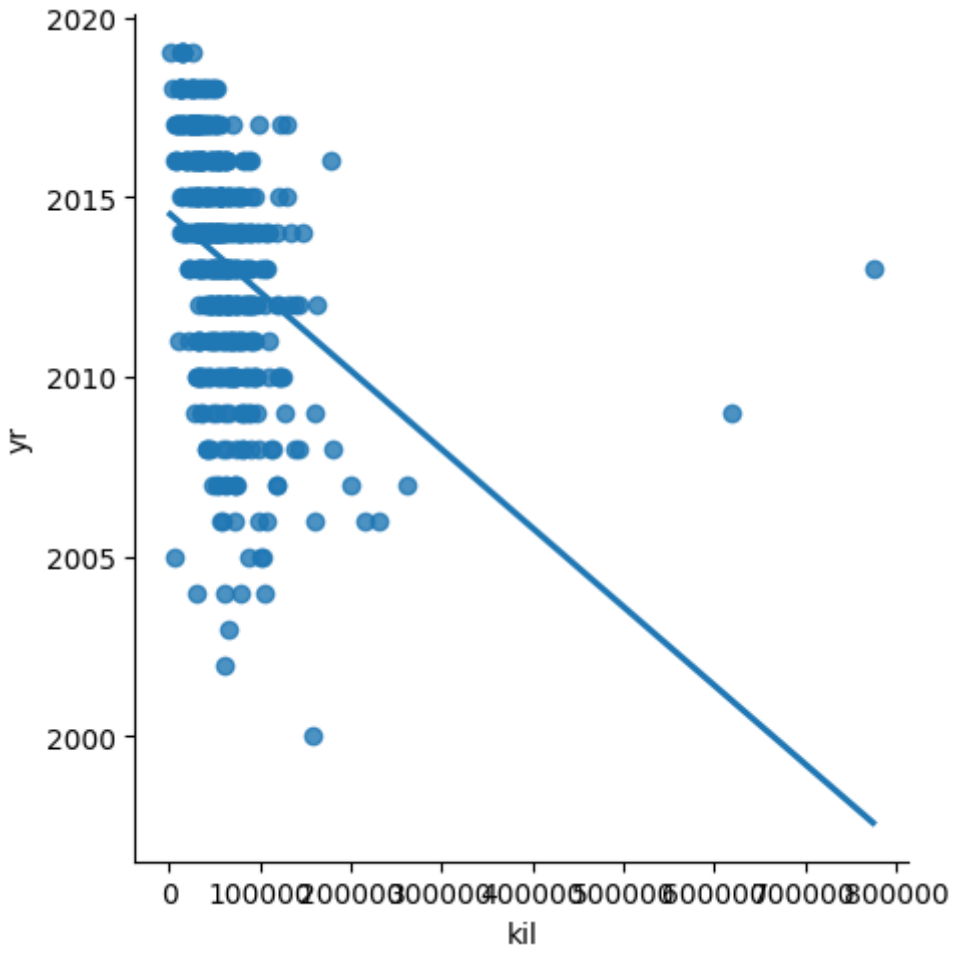
In [14]:

```
#Exploring Our Results
y_pred = regr.predict(X_test)
plt.scatter(X_test, y_test, color = 'k')
plt.show()
# Data scatter of predicted values
```



In [15]:
```
#Working with a smaller Dataset
df500 = df[:][:500]
# Selecting the 1st 500 rows of teh data
sns.lmplot(x = "kil", y = "yr", data = df500, order = 1, ci = None)
```

Out[15]: <seaborn.axisgrid.FacetGrid at 0x1a6a35a4f50>



In [16]:
```
sns.pairplot(df)
```

Out[16]: <seaborn.axisgrid.PairGrid at 0x1a6a38efa10>