In [18]:
```
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 [19]:
```
df=pd.read_csv(r"C:\Users\mouni\Downloads\used_cars_data.csv")
df
```

Out[19]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_
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	Se
...	...	...	...	...	...	...	...	
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 [20]:
```
df=df[['Kilometers_Driven', 'Price']]
df.columns=['kd', 'price']
```

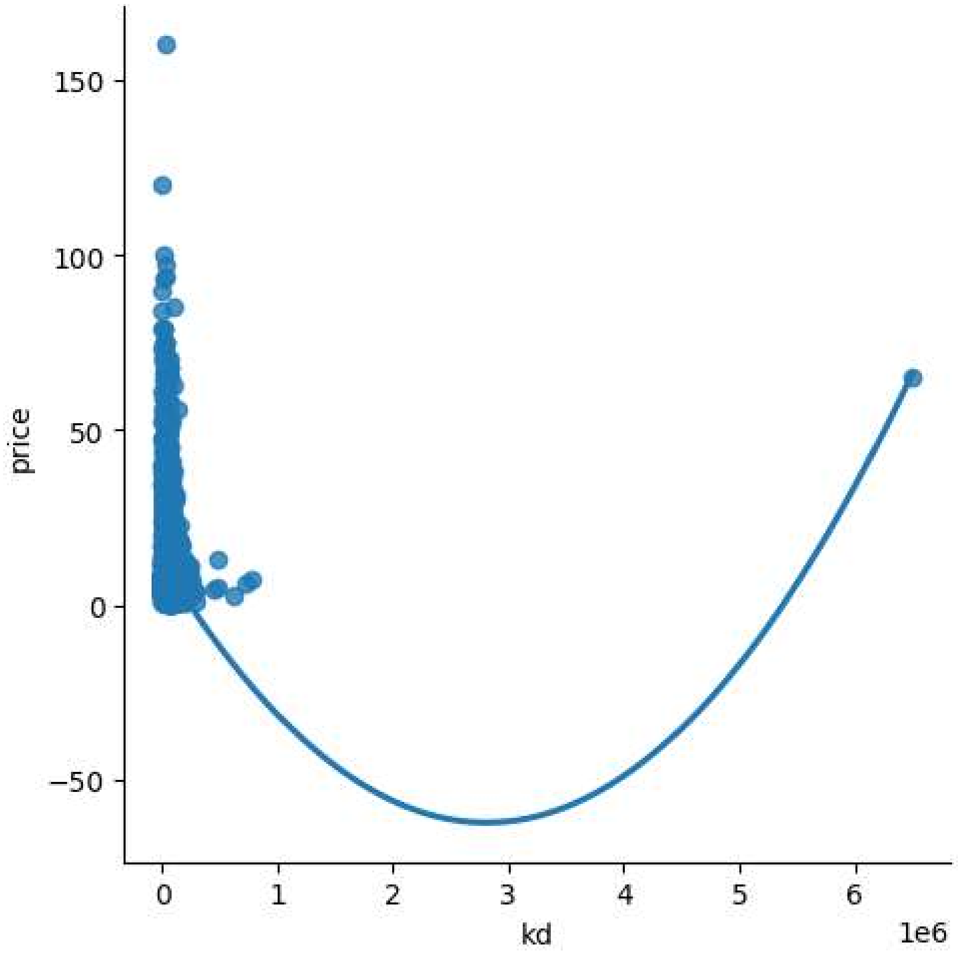
In [21]:
```
df.head(20)
```

Out[21]:

	kd	price
0	72000	1.75
1	41000	12.50
2	46000	4.50
3	87000	6.00
4	40670	17.74
5	75000	2.35
6	86999	3.50
7	36000	17.50
8	64430	5.20
9	65932	1.95
10	25692	9.95
11	60000	4.49
12	64424	5.60
13	72000	27.00
14	85000	17.50
15	110000	15.00
16	58950	5.40
17	25000	5.99
18	77469	6.34
19	78500	28.00

```
In [23]: sns.lmplot(x='kd',y='price',data=df,order=2,ci=None)

Out[23]: <seaborn.axisgrid.FacetGrid at 0x1faeccf5ab0>
```



```
In [24]: df.describe()

Out[24]:
```

	kd	price
count	7.253000e+03	6019.000000
mean	5.869906e+04	9.479468
std	8.442772e+04	11.187917
min	1.710000e+02	0.440000
25%	3.400000e+04	3.500000
50%	5.341600e+04	5.640000
75%	7.300000e+04	9.950000
max	6.500000e+06	160.000000

In [25]:
```
df.info()
```

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

In [26]:
```
df.fillna(method='ffill')
```

Out[26]:

	kd	price
0	72000	1.75
1	41000	12.50
2	46000	4.50
3	87000	6.00
4	40670	17.74
...	...	...
7248	89411	2.50
7249	59000	2.50
7250	28000	2.50
7251	52262	2.50
7252	72443	2.50

7253 rows × 2 columns

In [27]:
```
df.fillna(value=0,inplace=True)
```

```
C:\Users\mouni\AppData\Local\Temp\ipykernel_5504\1434098079.py:1: SettingWit
hCopyWarning:
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://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  df.fillna(value=0,inplace=True)
```

In [29]:
```
x=np.array(df['kd']).reshape(-1,1)
y=np.array(df['price']).reshape(-1,1)
```

In [30]:
```
df.dropna(inplace=True)
```

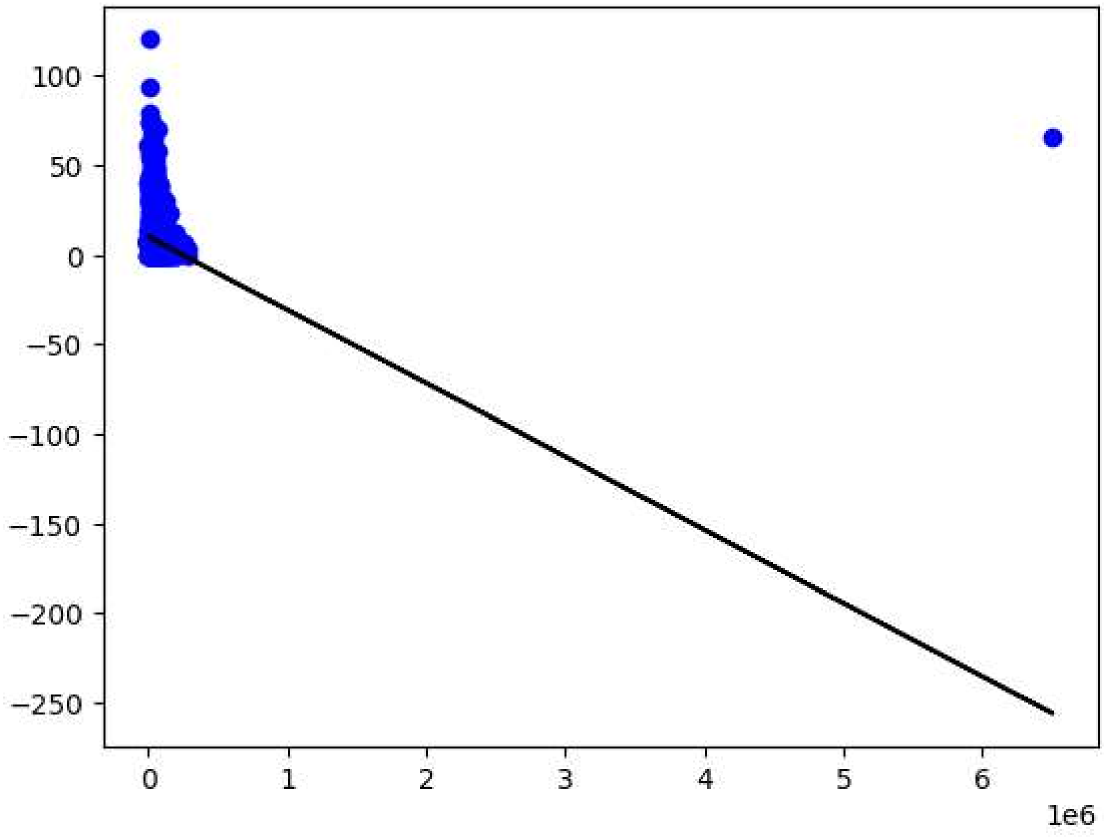
```
C:\Users\mouni\AppData\Local\Temp\ipykernel_5504\1379821321.py:1: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  df.dropna(inplace=True)
```

In [31]:
```
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))
```

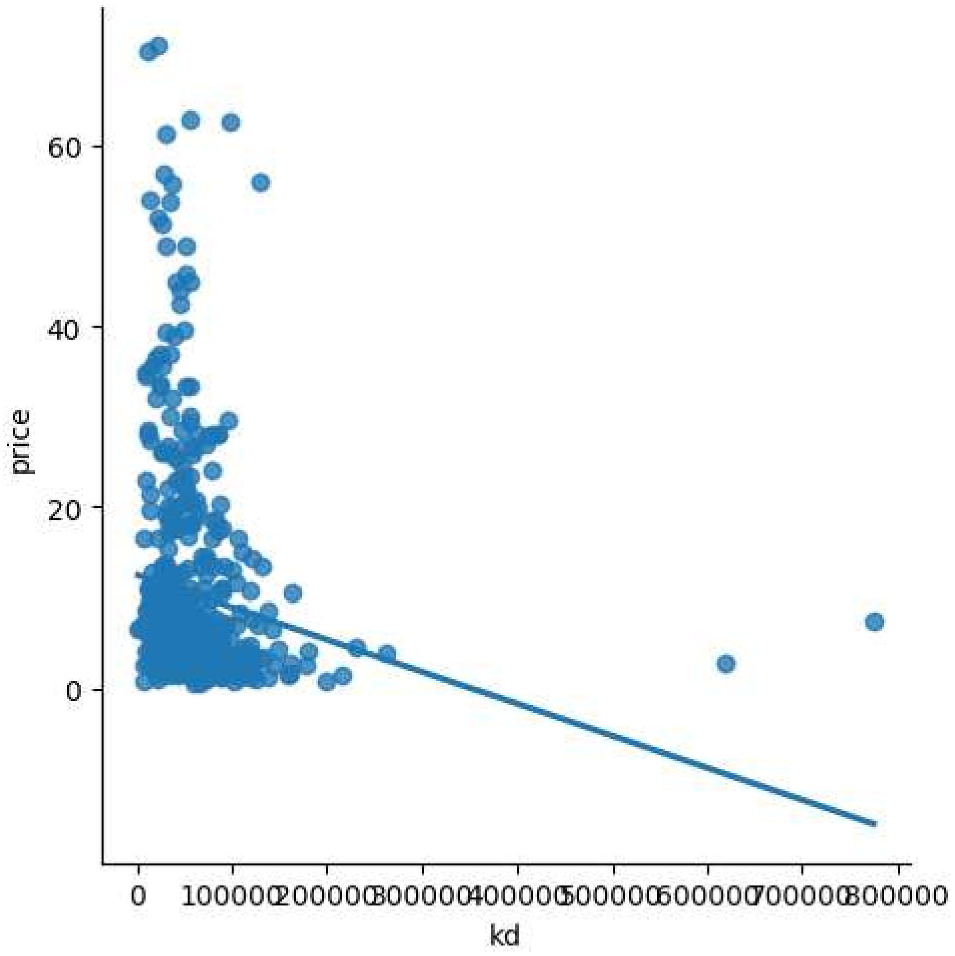
```
-0.41368708627835615
```

In [32]:
```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```



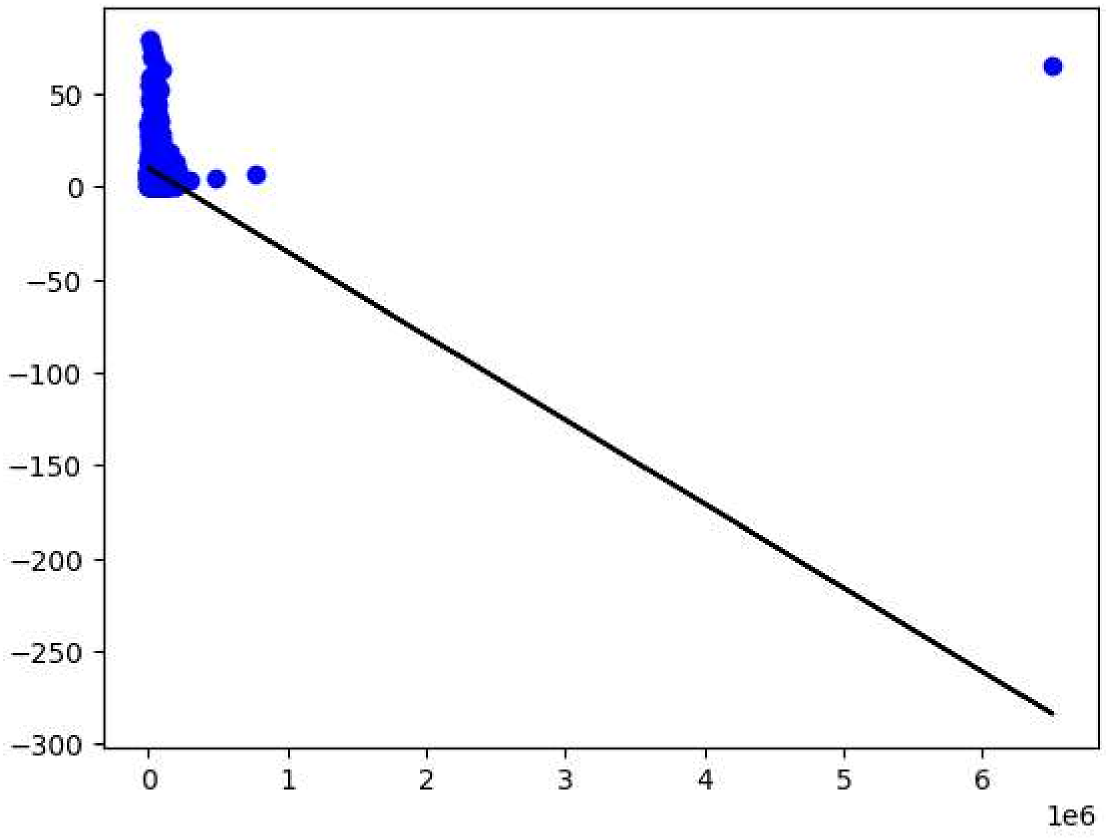
```
In [33]: df500=df[:][:500]
         sns.lmplot(x="kd",y="price",data=df500,order=1,ci=None)

Out[33]: <seaborn.axisgrid.FacetGrid at 0x1fae5f8baf0>
```



In [34]:
```
df500.dropna(inplace=True)
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("Regression:",regr.score(x_test,y_test))
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

```
Regression: -0.5417583905652401
```



In [35]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model=LinearRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
r2=r2_score(y_test,y_pred)
print("r2 score:",r2)
```

```
r2 score: -0.5417583905652401
```

In [ ]:

In [ ]:

```
```

In [ ]:

```
```