

```
In [14]: 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 [16]: df=pd.read_csv(r"C:\Users\arshiha\Downloads\fiat500_VehicleSelection_Dataset.csv")
df
```

```
Out[16]:
```

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon
0	1	lounge	51	882	25000	1	44.907242	8.611560
1	2	pop	51	1186	32500	1	45.666359	12.241890
2	3	sport	74	4658	142228	1	45.503300	11.417840
3	4	lounge	51	2739	160000	1	40.633171	17.634609
4	5	pop	73	3074	106880	1	41.903221	12.495650
...	...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	45.069679	7.704920
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870
1535	1536	pop	51	2223	60457	1	45.481541	9.413480
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270
1537	1538	pop	51	1766	54276	1	40.323410	17.568270

1538 rows × 9 columns



```
In [17]: df=df[['km','price']]
df.columns=['kms','prc']
```

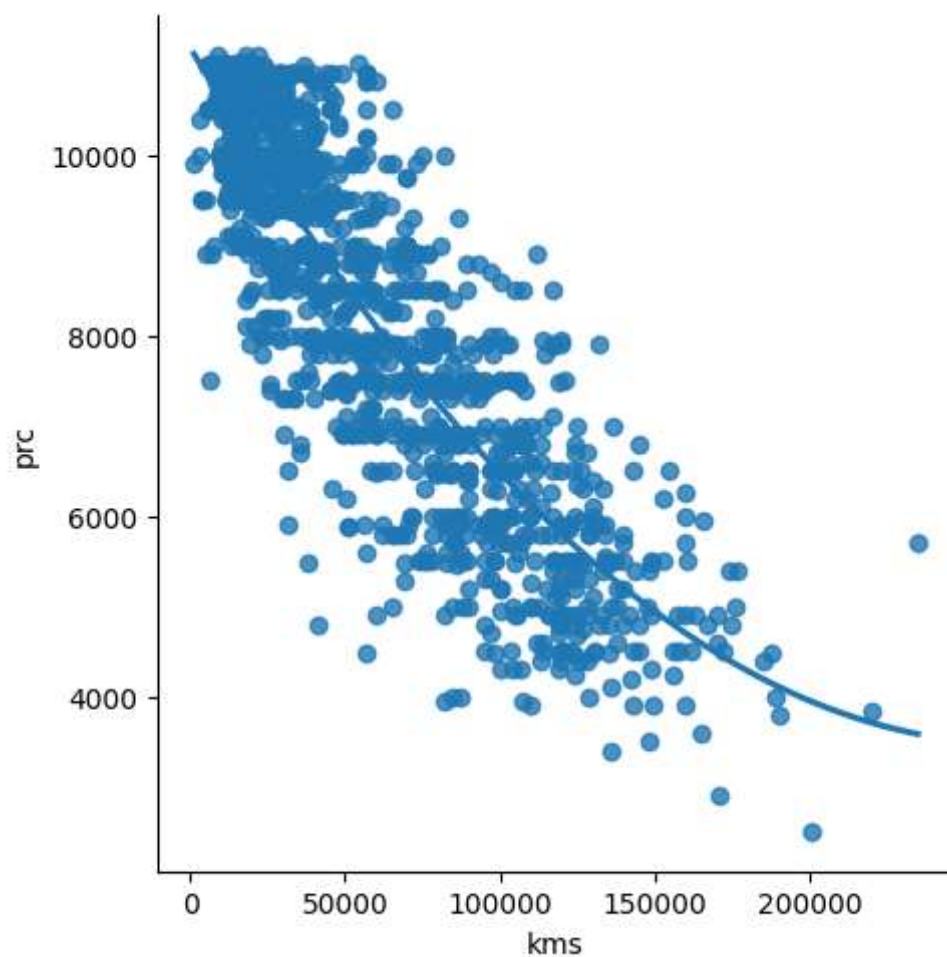
```
In [18]: df.head(10)
```

```
Out[18]:
```

	kms	prc
0	25000	8900
1	32500	8800
2	142228	4200
3	160000	6000
4	106880	5700
5	70225	7900
6	11600	10750
7	49076	9190
8	76000	5600
9	89000	6000

```
In [20]: sns.lmplot(x='kms',y='prc',data=df,order=2,ci=None)
```

```
Out[20]: <seaborn.axisgrid.FacetGrid at 0x2351a318610>
```



In [21]: `df.describe()`

Out[21]:

	kms	prc
count	1538.000000	1538.000000
mean	53396.011704	8576.003901
std	40046.830723	1939.958641
min	1232.000000	2500.000000
25%	20006.250000	7122.500000
50%	39031.000000	9000.000000
75%	79667.750000	10000.000000
max	235000.000000	11100.000000

In [22]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1538 entries, 0 to 1537
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  -
0    kms      1538 non-null    int64
1    prc      1538 non-null    int64
dtypes: int64(2)
memory usage: 24.2 KB
```

In [23]: `df.fillna(method="ffill",inplace=True)`

C:\Users\arshiha\AppData\Local\Temp\ipykernel\_4132\1844562654.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.fillna(method="ffill",inplace=True)
```

In [24]: `x=np.array(df['kms']).reshape(-1,1)`  
`y=np.array(df['prc']).reshape(-1,1)`

```
In [25]: df.dropna(inplace=True)
```

C:\Users\arshiha\AppData\Local\Temp\ipykernel\_4132\1379821321.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

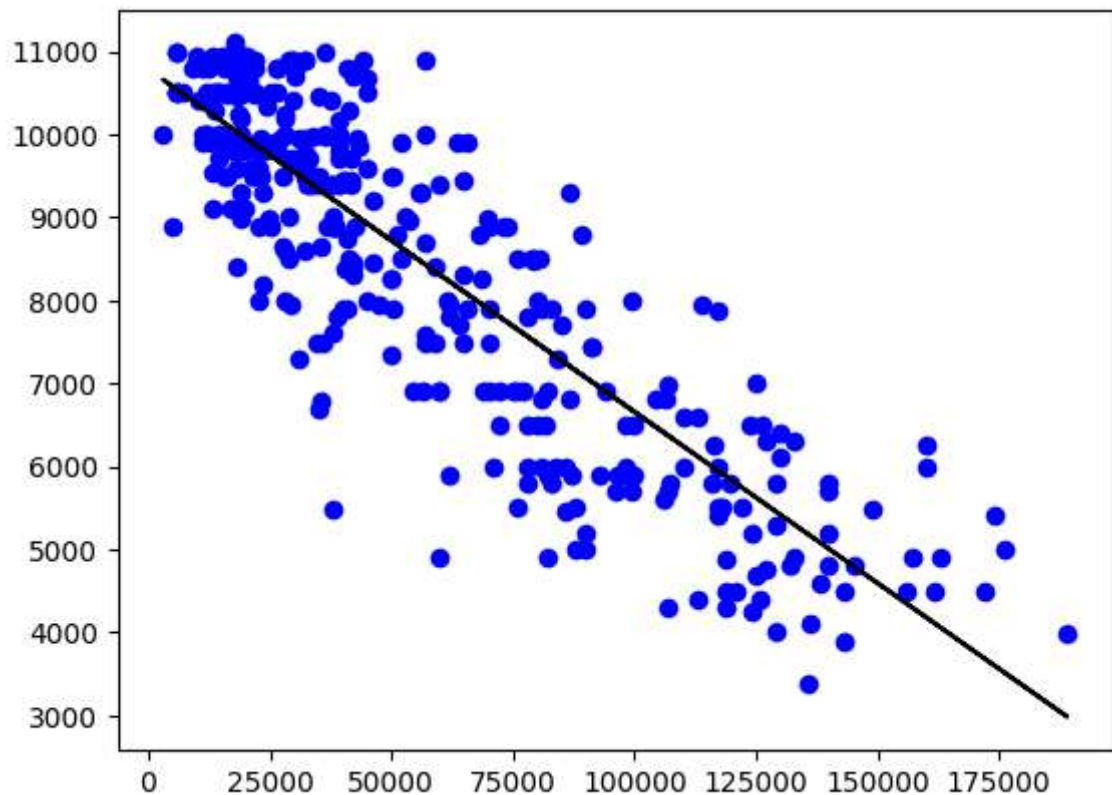
See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

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

```
In [26]: 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))
```

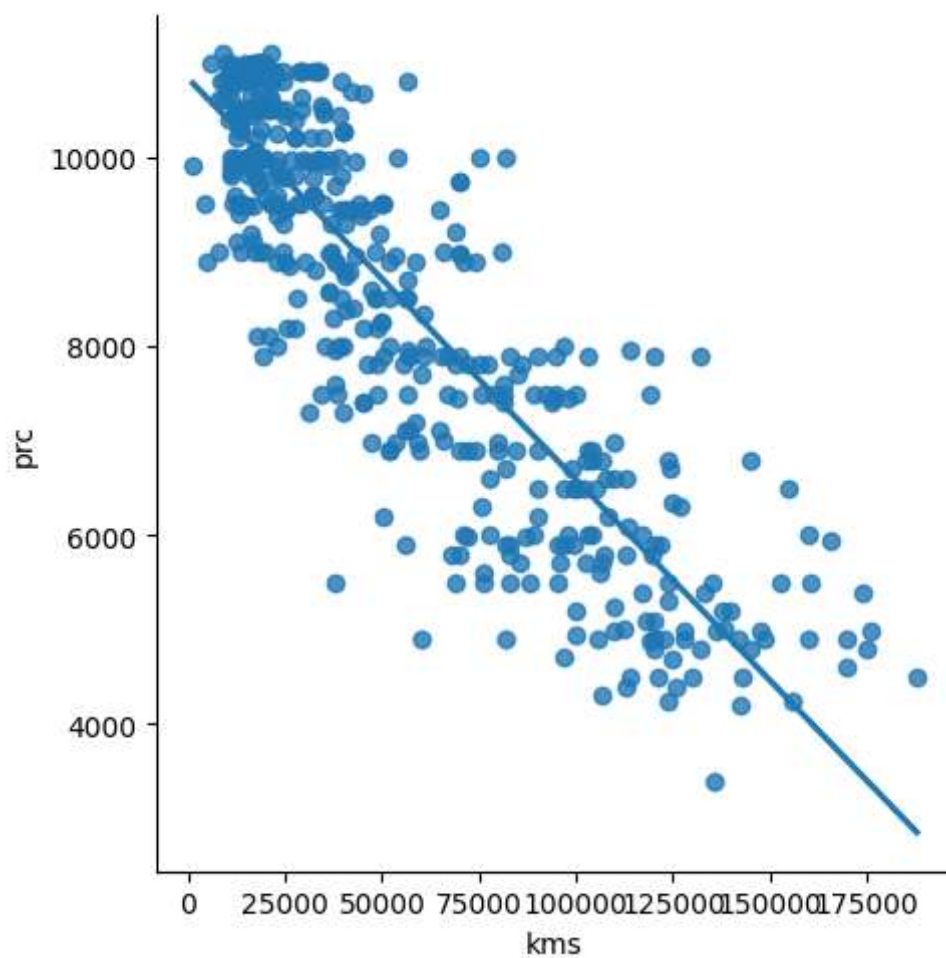
0.7638305392098184

```
In [28]: y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```



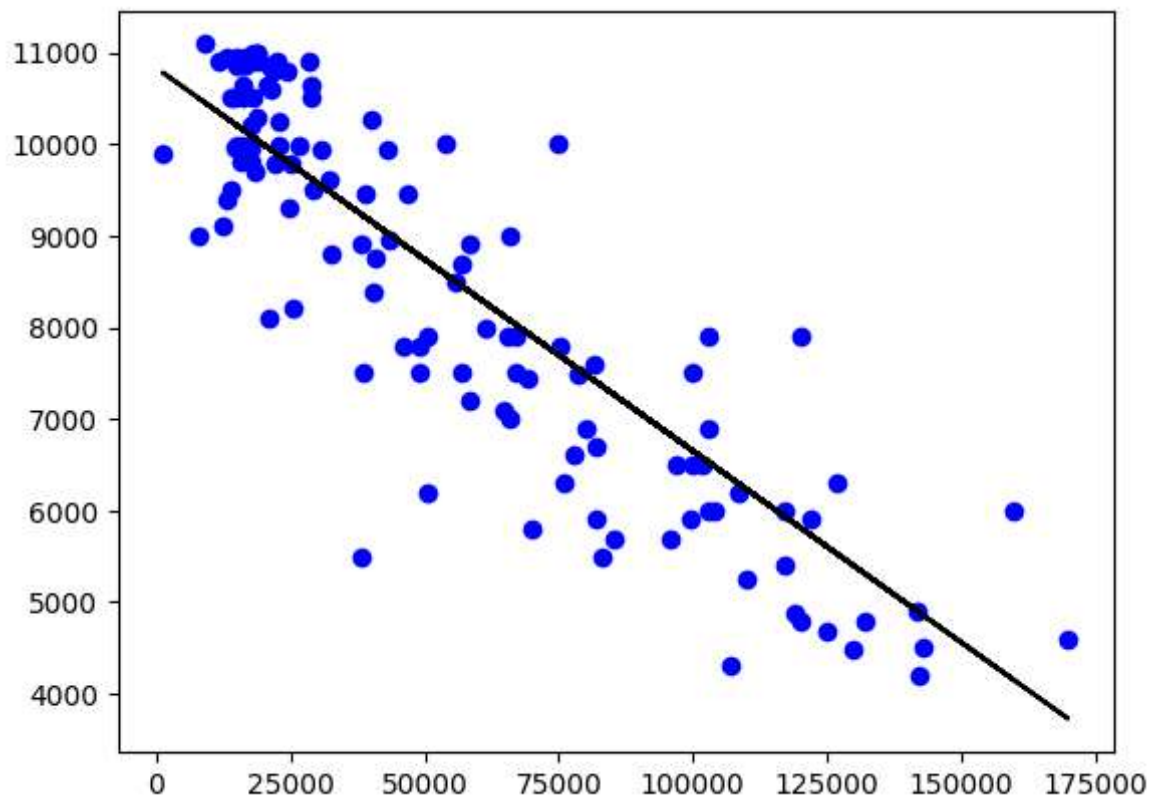
```
In [30]: df500=df[:][:500]  
sns.lmplot(x="kms",y="prc",data=df500,order=1,ci=None)
```

Out[30]: <seaborn.axisgrid.FacetGrid at 0x2351a41e810>



```
In [31]: df500.fillna(method='ffill',inplace=True)
x=np.array(df500['kms']).reshape(-1,1)
y=np.array(df500['prc']).reshape(-1,1)
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: 0.7782557903560138



```
In [32]: 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: 0.7782557903560138

In [ ]:

