

```
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\USER\Downloads\bottle.csv.zip")
df
```

C:\Users\USER\AppData\Local\Temp\ipykernel_9844\4196241061.py:1: DtypeWarning: Columns (47,73) have mixed types. Specify dtype option on import or set low_memory=False.
df=pd.read_csv(r"C:\Users\USER\Downloads\bottle.csv.zip")

Out[2]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	R_PHAEO	R_PRES	R_SAMP	I
0	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	10.500	33.4400	NaN	25.64900	NaN	...	NaN	0	NaN	
1	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	10.460	33.4400	NaN	25.65600	NaN	...	NaN	8	NaN	
2	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	10.460	33.4370	NaN	25.65400	NaN	...	NaN	10	NaN	
3	1	4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	10.450	33.4200	NaN	25.64300	NaN	...	NaN	19	NaN	
4	1	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	10.450	33.4210	NaN	25.64300	NaN	...	NaN	20	NaN	
...
864858	34404	864859	093.4 026.4	20-1611SR-MX-310-2239-09340264-0000A-7	0	18.744	33.4083	5.805	23.87055	108.74	...	0.18	0	NaN	

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	R_PHAEO	R_PRES	R_SAMP	I
864859	34404	864860	093.4 026.4	20- 1611SR- MX-310- 2239- 09340264- 0002A-3	2	18.744	33.4083	5.805	23.87072	108.74	...	0.18	2	4.0	
864860	34404	864861	093.4 026.4	20- 1611SR- MX-310- 2239- 09340264- 0005A-3	5	18.692	33.4150	5.796	23.88911	108.46	...	0.18	5	3.0	
864861	34404	864862	093.4 026.4	20- 1611SR- MX-310- 2239- 09340264- 0010A-3	10	18.161	33.4062	5.816	24.01426	107.74	...	0.31	10	2.0	
864862	34404	864863	093.4 026.4	20- 1611SR- MX-310- 2239- 09340264- 0015A-3	15	17.533	33.3880	5.774	24.15297	105.66	...	0.61	15	1.0	

864863 rows × 74 columns

```
In [3]: df = df[['Salnty','T_degC']]
df.columns=['Sal','Temp']
```

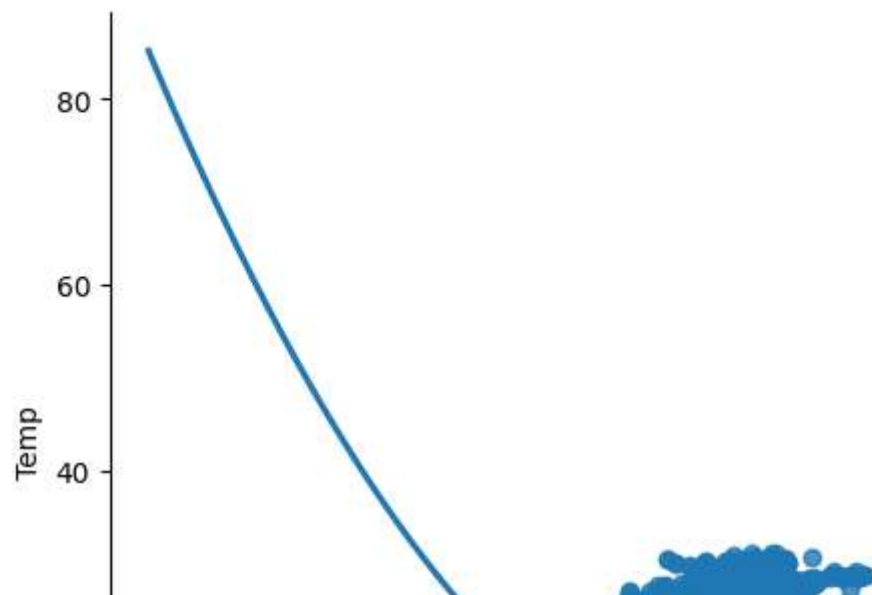
```
In [4]: df.head(10)
```

```
Out[4]:
```

	Sal	Temp
0	33.440	10.50
1	33.440	10.46
2	33.437	10.46
3	33.420	10.45
4	33.421	10.45
5	33.431	10.45
6	33.440	10.45
7	33.424	10.24
8	33.420	10.06
9	33.494	9.86

```
In [5]: sns.lmplot(x='Sal',y="Temp",data=df,order=2,ci=None)
```

```
Out[5]: <seaborn.axisgrid.FacetGrid at 0x2020d6cfe90>
```



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

Out[6]:

	Sal	Temp
count	817509.000000	853900.000000
mean	33.840350	10.799677
std	0.461843	4.243825
min	28.431000	1.440000
25%	33.488000	7.680000
50%	33.863000	10.060000
75%	34.196900	13.880000
max	37.034000	31.140000

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 864863 entries, 0 to 864862
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  ---
0    Sal      817509 non-null    float64
1    Temp     853900 non-null    float64
dtypes: float64(2)
memory usage: 13.2 MB
```

In [8]: `df.fillna(method='ffill',inplace=True)`

C:\Users\USER\AppData\Local\Temp\ipykernel_9844\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 [9]: x=np.array(df['Sal']).reshape(-1,1)
        y=np.array(df['Temp']).reshape(-1,1)
```

```
In [10]: df.dropna(inplace=True)
```

C:\Users\USER\AppData\Local\Temp\ipykernel_9844\1379821321.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

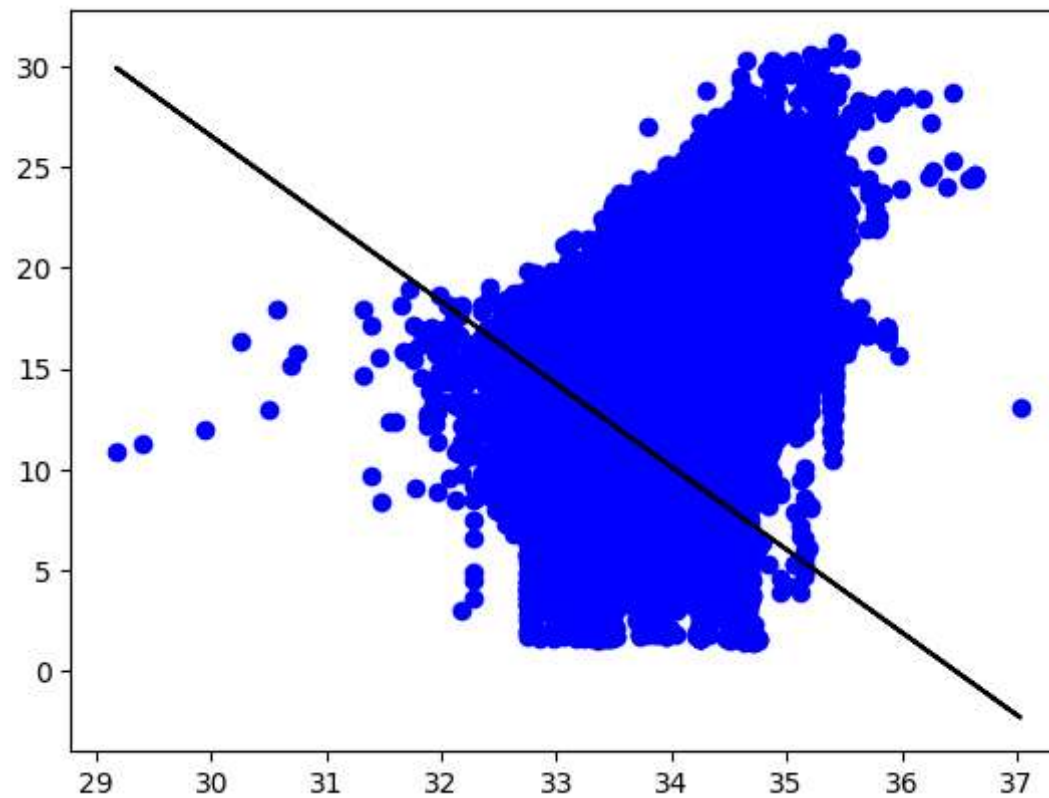
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
df.dropna(inplace=True)

```
In [11]: 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))
```

0.2068740407299181

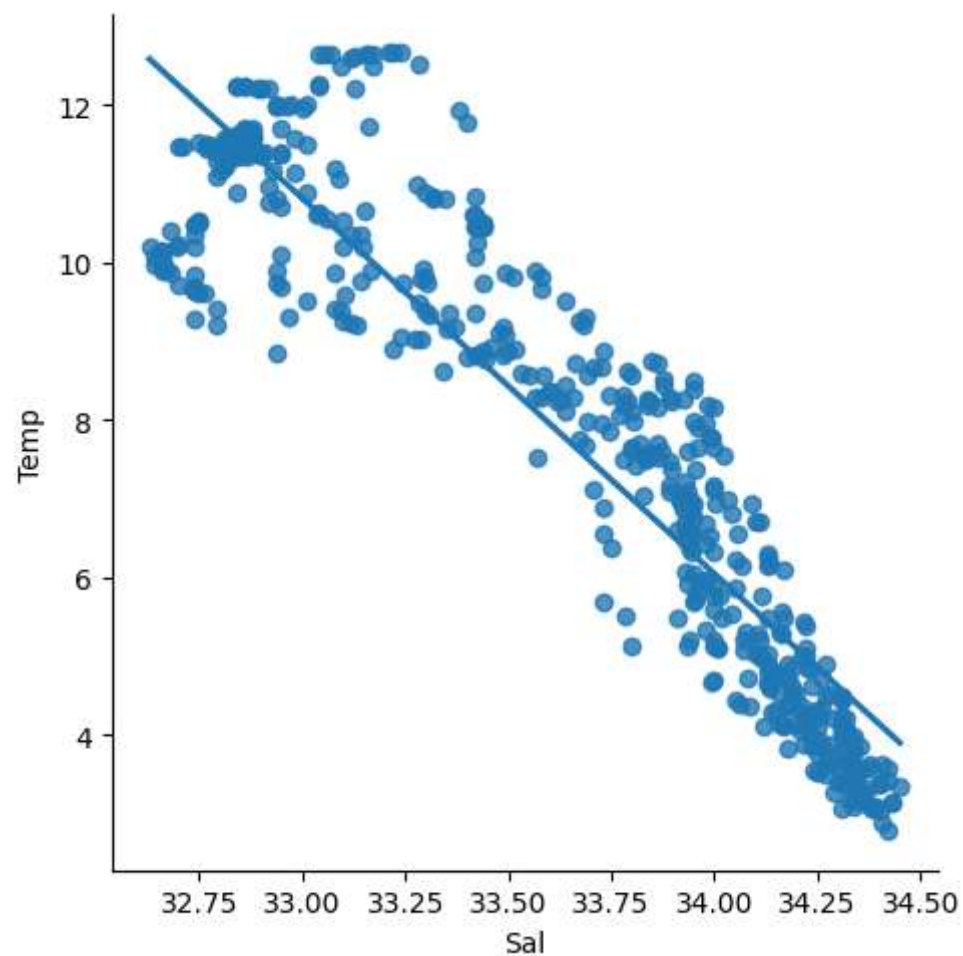
```
In [12]: y_pred=regr.predict(x_test)  
plt.scatter(x_test,y_test,color='b')  
plt.plot(x_test,y_pred,color='k')
```

Out[12]: [<matplotlib.lines.Line2D at 0x20214eeaf10>]




```
In [13]: df500=df[:][:500]  
sns.lmplot(x='Sal',y='Temp',data=df500,order=1,ci=None)
```

```
Out[13]: <seaborn.axisgrid.FacetGrid at 0x20214f16590>
```

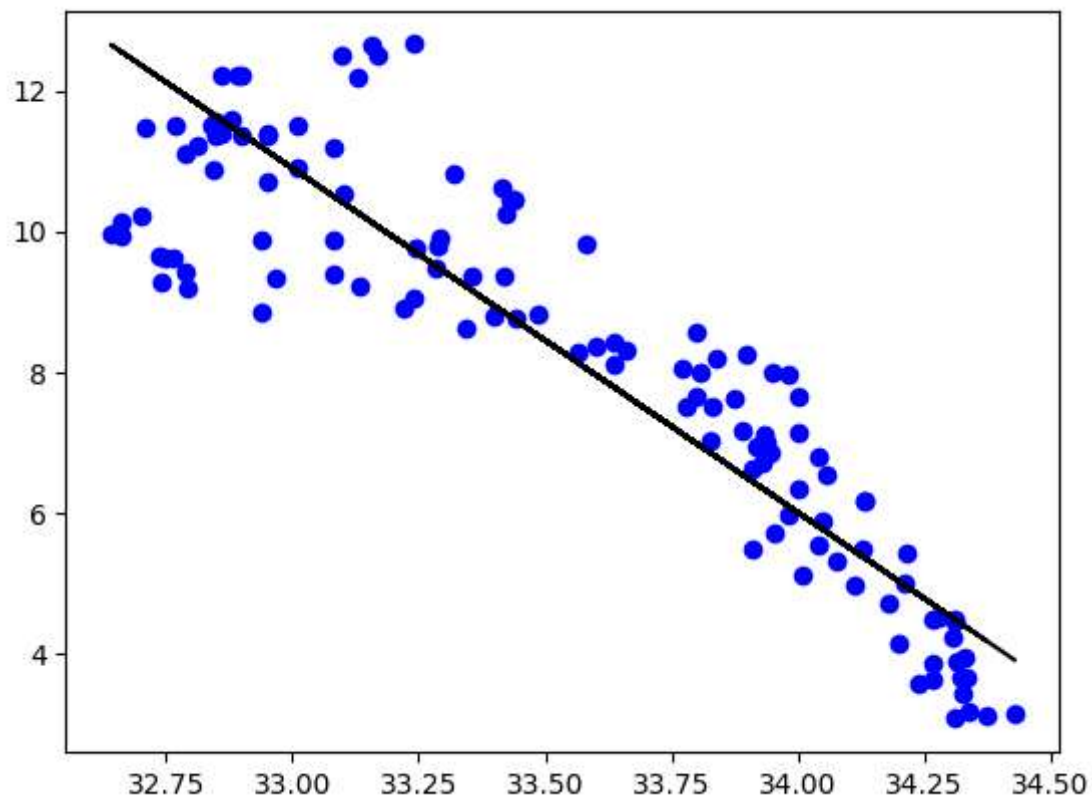


```
In [14]: from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model=LinearRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
r2=r2_score(y_test,y_pred)
print("R2 score:",r2)
```

R2 score: 0.2068740407299181

```
In [15]: df500.fillna(method='ffill',inplace=True)
x=np.array(df500['Sal']).reshape(-1,1)
y=np.array(df500['Temp']).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: 0.8002606873825874



```
In [16]: 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 [17]: df=pd.read_csv(r"C:\Users\USER\Downloads\fiat500_VehicleSelection_Dataset (1).csv")
df
```

```
Out[17]:
```

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	1	lounge	51	882	25000	1	44.907242	8.611560	8900
1	2	pop	51	1186	32500	1	45.666359	12.241890	8800
2	3	sport	74	4658	142228	1	45.503300	11.417840	4200
3	4	lounge	51	2739	160000	1	40.633171	17.634609	6000
4	5	pop	73	3074	106880	1	41.903221	12.495650	5700
...
1533	1534	sport	51	3712	115280	1	45.069679	7.704920	5200
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870	4600
1535	1536	pop	51	2223	60457	1	45.481541	9.413480	7500
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270	5990
1537	1538	pop	51	1766	54276	1	40.323410	17.568270	7900

1538 rows × 9 columns

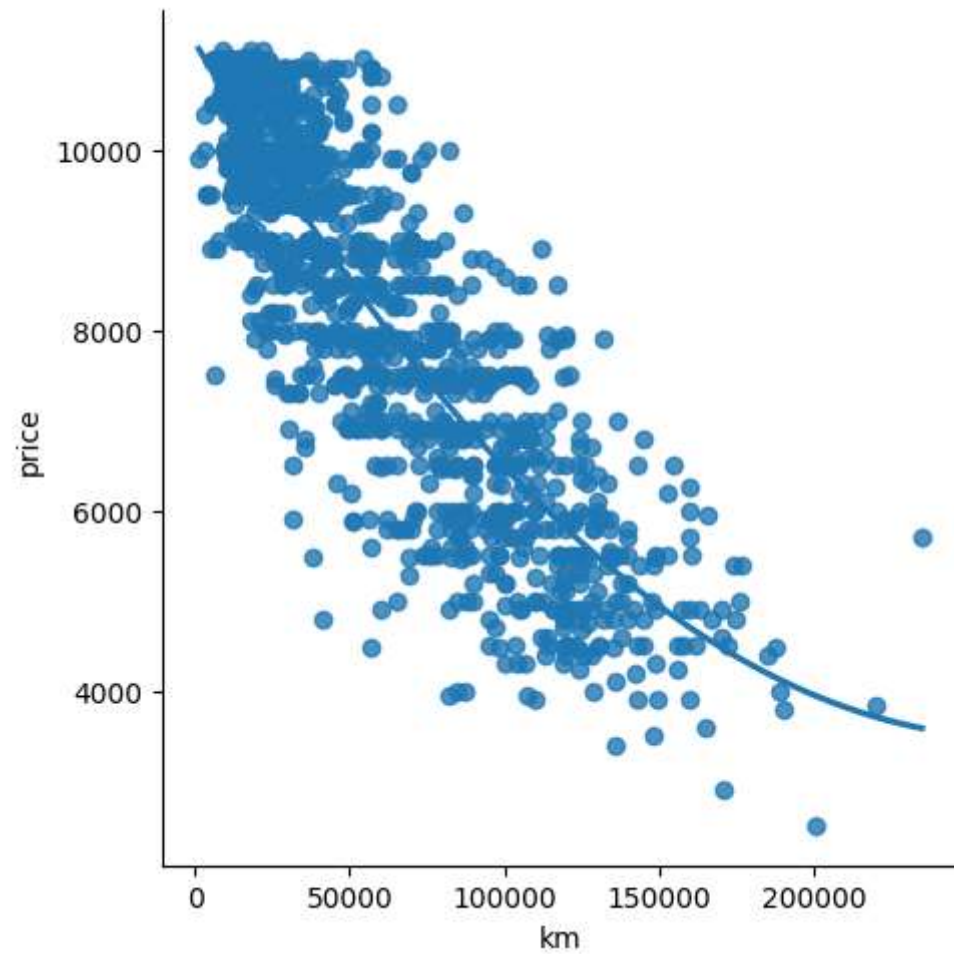
```
In [18]: df=df[['km','price']]
df.columns=['km','price']
df.head(10)
```

```
Out[18]:
```

	km	price
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 [19]: sns.lmplot(x='km',y="price",data=df,order=2,ci=None)
```

```
Out[19]: <seaborn.axisgrid.FacetGrid at 0x20218274ad0>
```



In [20]: `df.describe()`

Out[20]:

	km	price
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 [21]: `df.info()`

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

In [22]: `df.fillna(method="ffill",inplace=True)`

C:\Users\USER\AppData\Local\Temp\ipykernel_9844\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)
`df.fillna(method="ffill",inplace=True)`

```
In [28]: x=np.array(df['km']).reshape(-1,1)
         y=np.array(df['price']).reshape(-1,1)
```

```
In [29]: df.dropna(inplace=True)
```

C:\Users\USER\AppData\Local\Temp\ipykernel_9844\1379821321.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

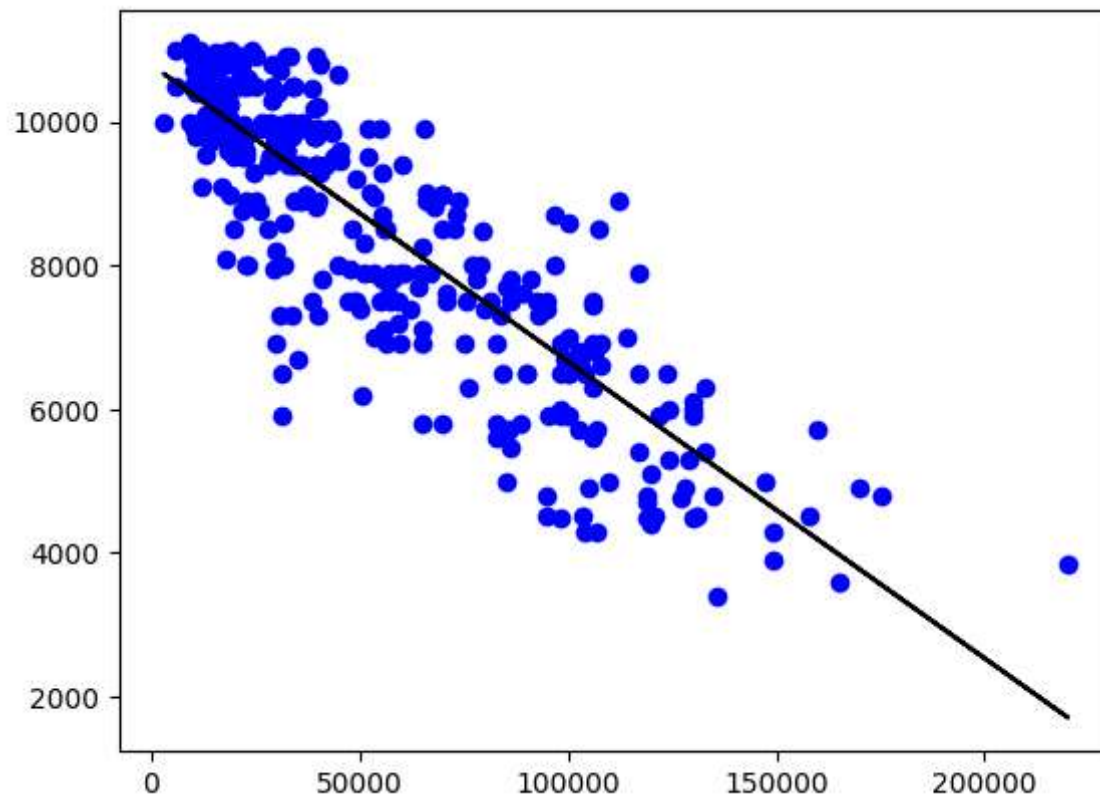
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
df.dropna(inplace=True)

```
In [30]: 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))
```

0.7607724472274502

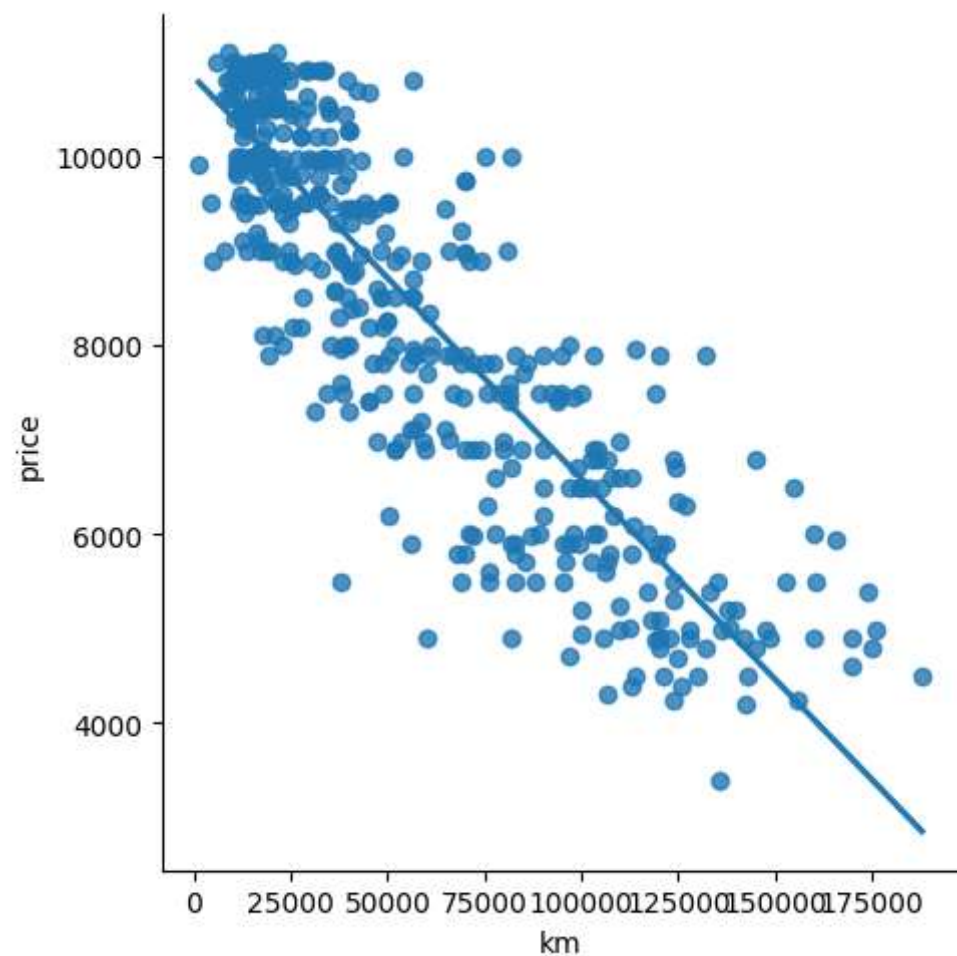

```
In [31]: y_pred=regr.predict(x_test)  
plt.scatter(x_test,y_test,color='b')  
plt.plot(x_test,y_pred,color='k')
```

Out[31]: [<matplotlib.lines.Line2D at 0x202182b6210>]



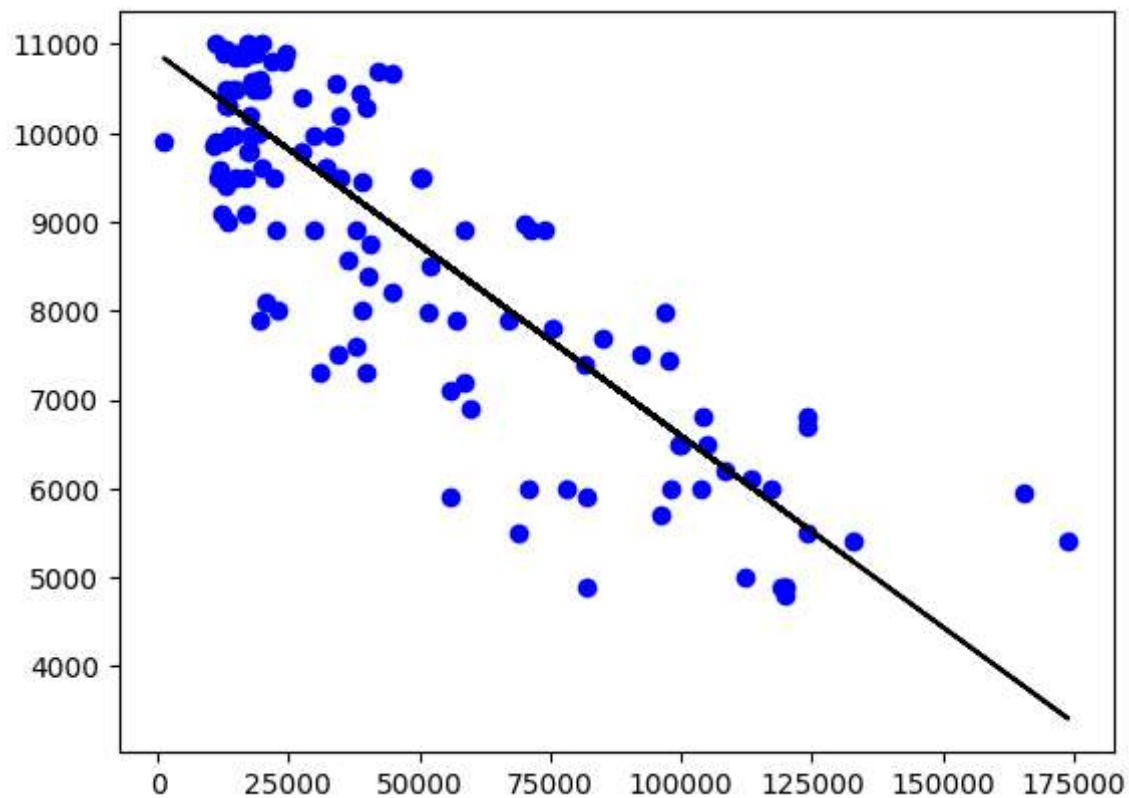
```
In [32]: df500=df[:][:500]  
sns.lmplot(x='km',y='price',data=df500,order=1,ci=None)
```

```
Out[32]: <seaborn.axisgrid.FacetGrid at 0x202170bf790>
```



```
In [33]: df500.fillna(method='ffill',inplace=True)
x=np.array(df500['km']).reshape(-1,1)
y=np.array(df500['price']).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: 0.7111666364484011



```
In [34]: 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_test2=r2_score(y_test,y_pred)
print("R2 score:",r2)
```

R2 score: 0.7111666364484011

In []:

In []: