In [4]:

```
dg=pd.read_csv(r"C:\Users\magam\Downloads\bottle.csv\bottle.csv")
dg.head(5)
```

```
C:\Users\magam\AppData\Local\Temp\ipykernel_12308\3556595680.py:1: Dtyp
eWarning: Columns (47,73) have mixed types. Specify dtype option on imp
ort or set low_memory=False.
  dg=pd.read_csv(r"C:\Users\magam\Downloads\bottle.csv\bottle.csv")
```

Out[4]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat
0	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	10.50	33.440	NaN	25.649	NaN
1	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	10.46	33.440	NaN	25.656	NaN
2	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	10.46	33.437	NaN	25.654	NaN
3	1	4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	10.45	33.420	NaN	25.643	NaN
4	1	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	10.45	33.421	NaN	25.643	NaN

5 rows × 74 columns

In [5]:

```
dg=dg[['Salnty','T_degC']]
dg.columns=['Sal','Temp']
dg.head(10)
```

Out[5]:

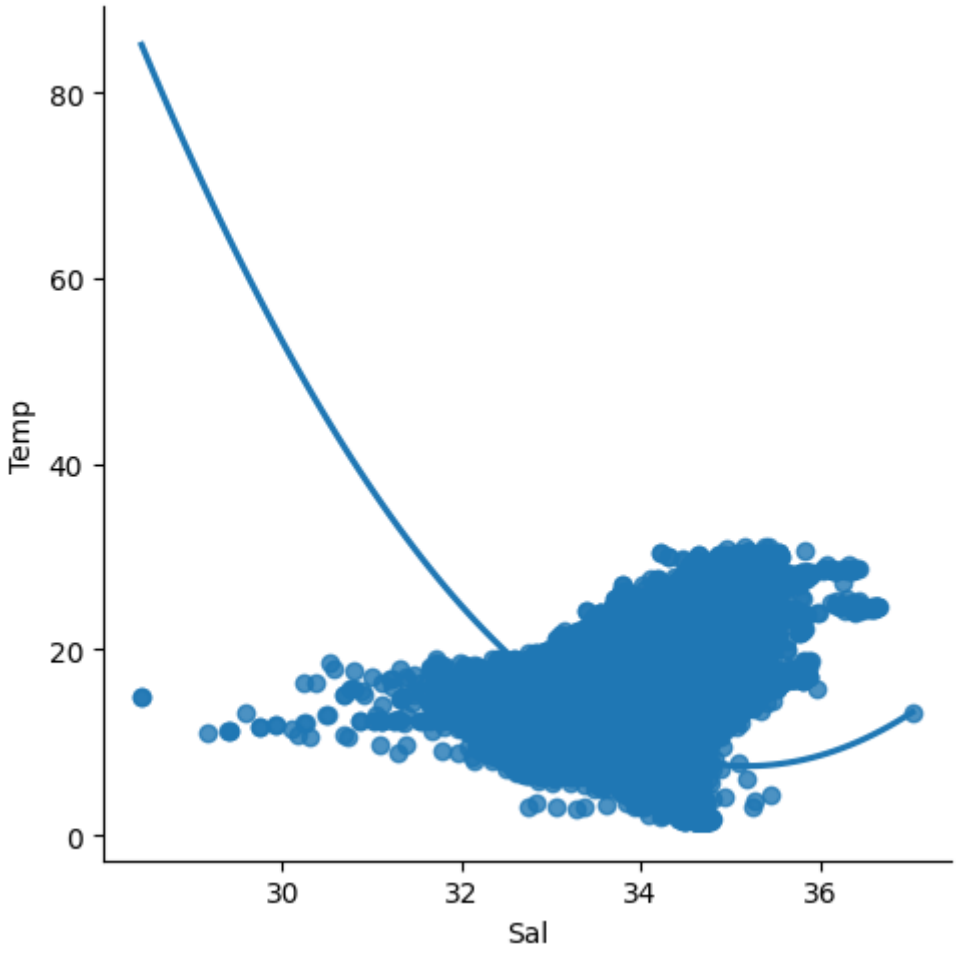
	Sal	Temp
0	33.440	10.50
1	33.440	10.46
2	33.437	10.46
3	33.420	10.45
4	33.421	10.45
5	33.431	10.45
6	33.440	10.45
7	33.424	10.24
8	33.420	10.06
9	33.494	9.86

In [6]:

```
#ploting the datascatter
sb.lmplot(x="Sal", y="Temp", data=dg, order=2, ci=None)
dg.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]:

```
print(dg.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
None
```

In [8]:

```
#data cleaning
dg.fillna(method='ffill',inplace=True)
```

In [9]:

```
#training our model
x=np.array(dg['Sal']).reshape(-1,1)
y=np.array(dg['Temp']).reshape(-1,1)
dg.dropna(inplace=True)
```

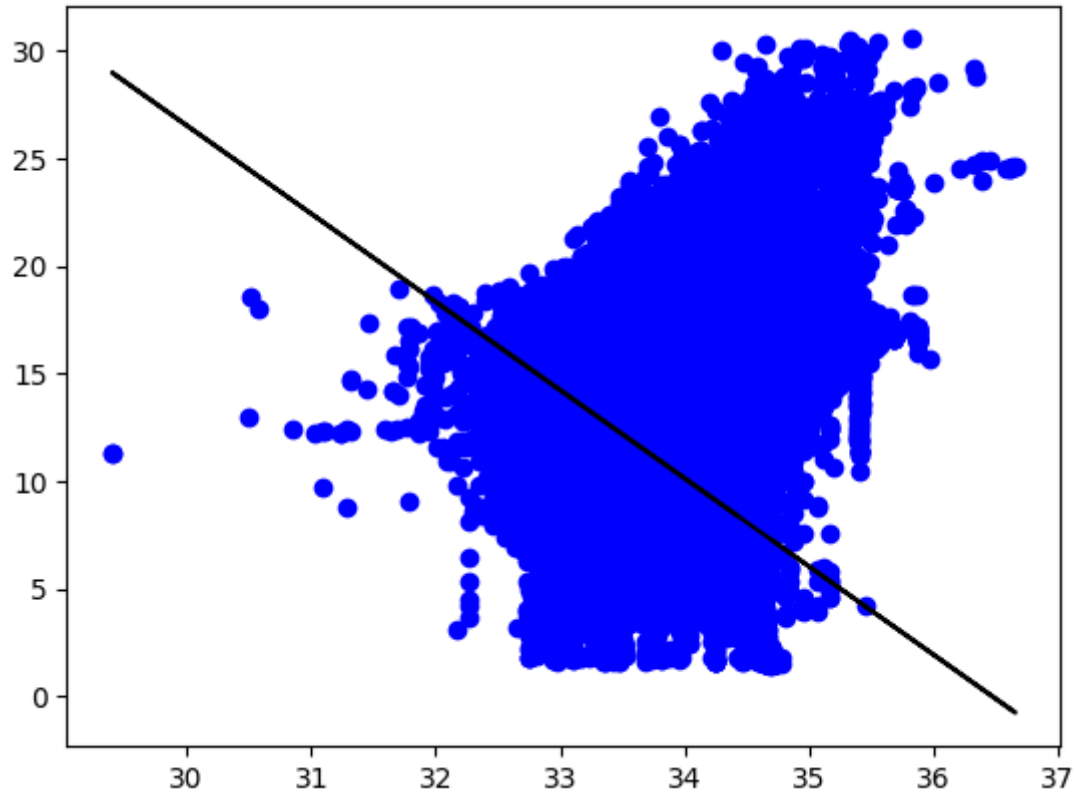
In [10]:

```
#splitting the data
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))
```

```
0.20529956613130818
```

In [11]:

```
#exploring our results
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

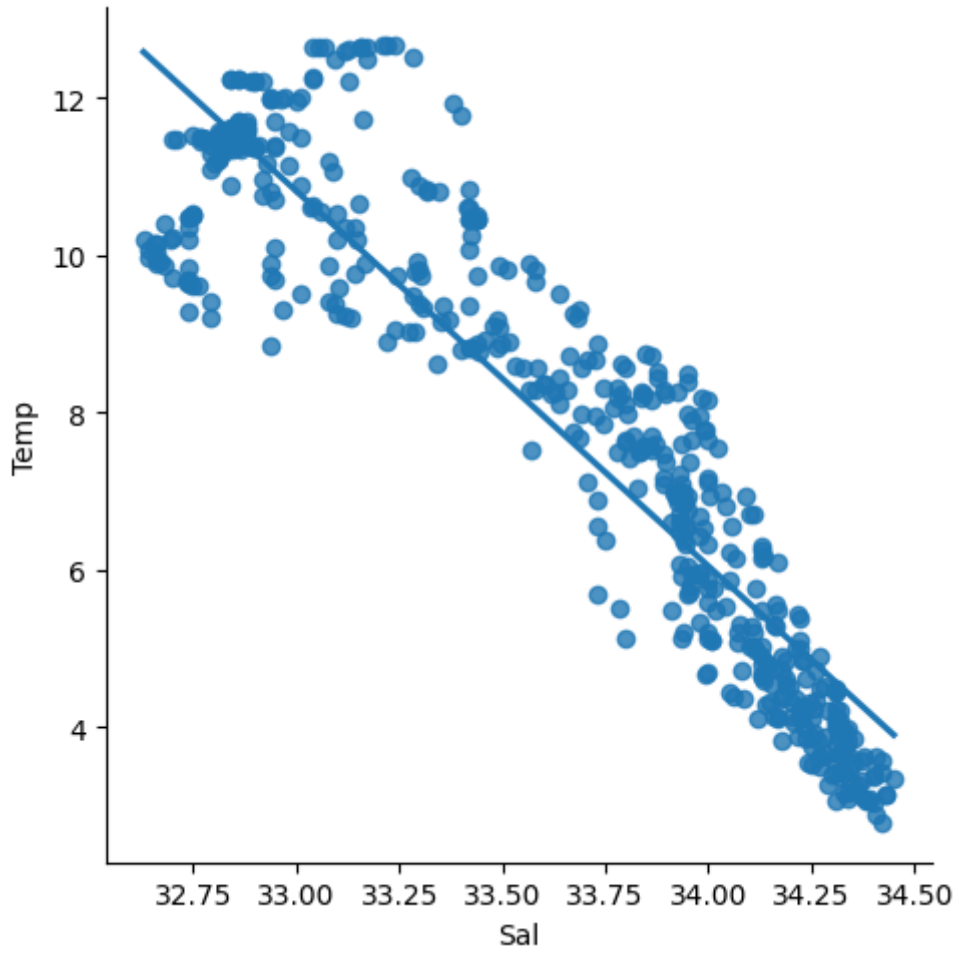


In [12]:

```
#Working with smaller dataset
dg500=dg[:][:500]
sb.lmplot(x='Sal',y='Temp',data=dg500,order=1,ci=None)
```

Out[12]:

```
<seaborn.axisgrid.FacetGrid at 0x1d5ed7b28d0>
```



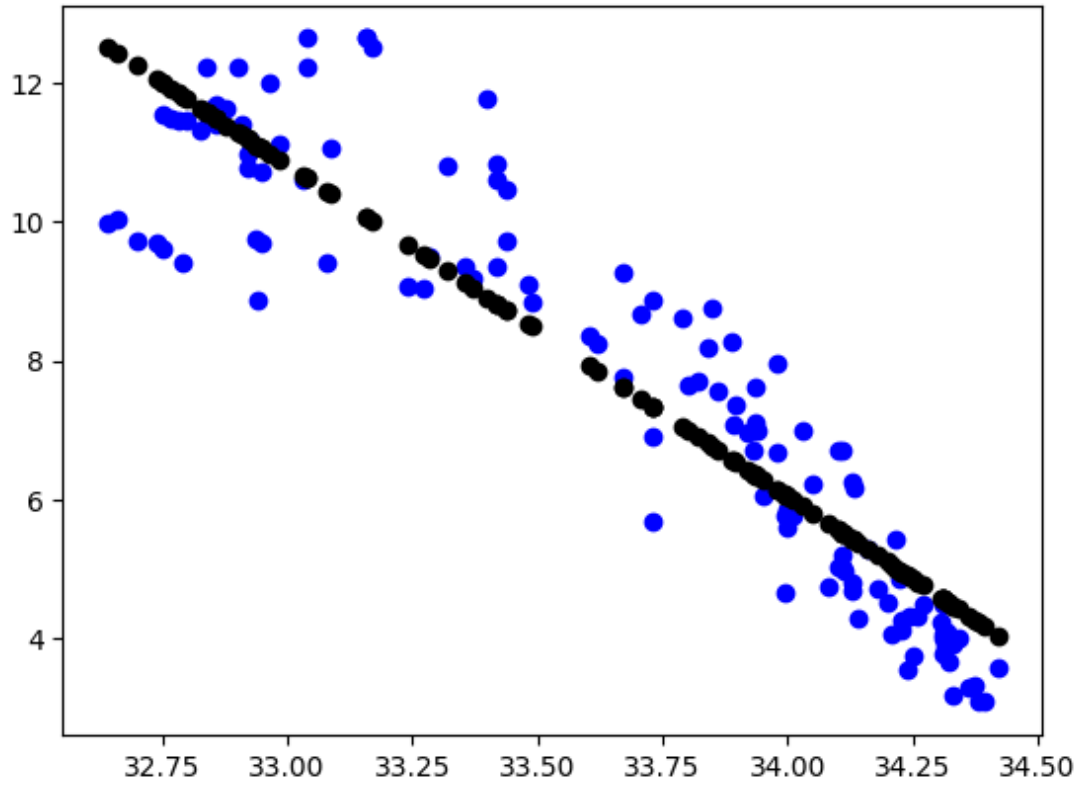
In [13]:

```
dg500.fillna(method='ffill',inplace=True)
x=np.array(dg500['Sal']).reshape(-1,1)
y=np.array(dg500['Temp']).reshape(-1,1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
```

In [14]:

```
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.scatter(x_test,y_pred,color='k')
plt.show()
```

```
Regression: 0.8504158261866646
```



In [25]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

In [26]:

```
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: 0.8504158261866646
```

analysis on vehicels

In [29]:

```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
from sklearn import preprocessing ,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
dv=pd.read_csv(r"C:\Users\magam\Downloads\fiat500_VehicleSelection_Dataset.csv")
dv.head(10)
```

Out[29]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lo
0	1	lounge	51	882	25000	1	44.907242	8.61156
1	2	pop	51	1186	32500	1	45.666359	12.24189
2	3	sport	74	4658	142228	1	45.503300	11.41784
3	4	lounge	51	2739	160000	1	40.633171	17.63460
4	5	pop	73	3074	106880	1	41.903221	12.49565
5	6	pop	74	3623	70225	1	45.000702	7.68227
6	7	lounge	51	731	11600	1	44.907242	8.61156
7	8	lounge	51	1521	49076	1	41.903221	12.49565
8	9	sport	73	4049	76000	1	45.548000	11.54947
9	10	sport	51	3653	89000	1	45.438301	10.99170

In [28]:

```
dv.shape
```

Out[28]:

```
(1538, 9)
```

In [30]:

```
dv=dv[['age_in_days','km']]
dv.columns=['age','km']
dv.head(10)
```

Out[30]:

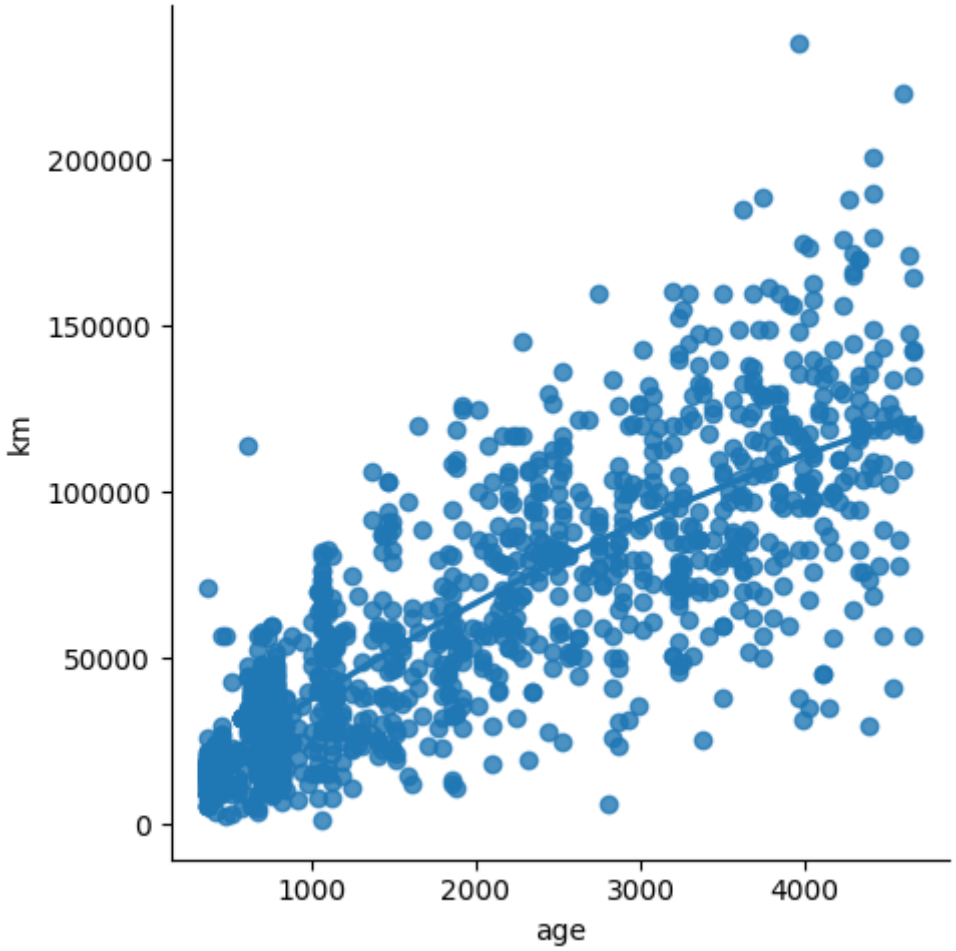
	age	km
0	882	25000
1	1186	32500
2	4658	142228
3	2739	160000
4	3074	106880
5	3623	70225
6	731	11600
7	1521	49076
8	4049	76000
9	3653	89000

In [31]:

```
sb.lmplot(x="age", y="km", data=dv, order=2, ci=None)
dv.describe()
```

Out[31]:

	age	km
count	1538.000000	1538.000000
mean	1650.980494	53396.011704
std	1289.522278	40046.830723
min	366.000000	1232.000000
25%	670.000000	20006.250000
50%	1035.000000	39031.000000
75%	2616.000000	79667.750000
max	4658.000000	235000.000000



In [32]:

```
dv.info()
```

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

In [33]:

```
#data cleaning
dv.fillna(method='ffill',inplace=True)
```

In [34]:

```
x=np.array(dv['age']).reshape(-1,1)
y=np.array(dv['km']).reshape(-1,1)
dv.dropna(inplace=True)
```

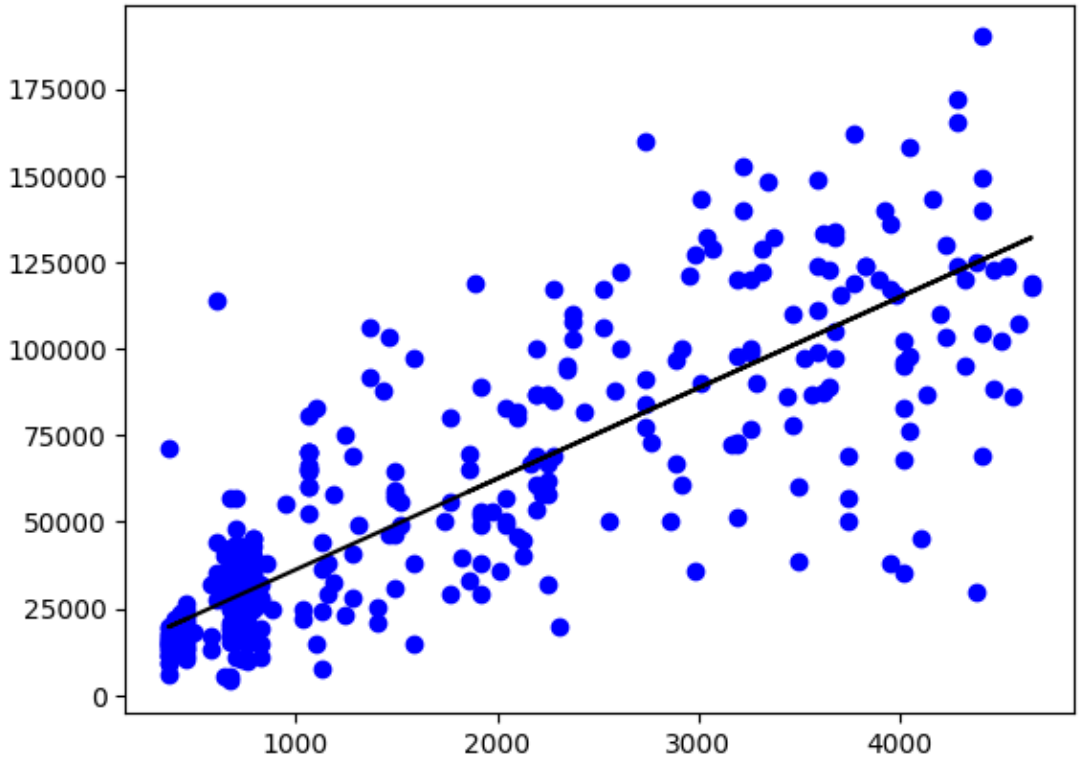
In [35]:

```
#splitting the data
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))
```

```
0.6728675548379355
```

In [36]:

```
#exploring our results
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

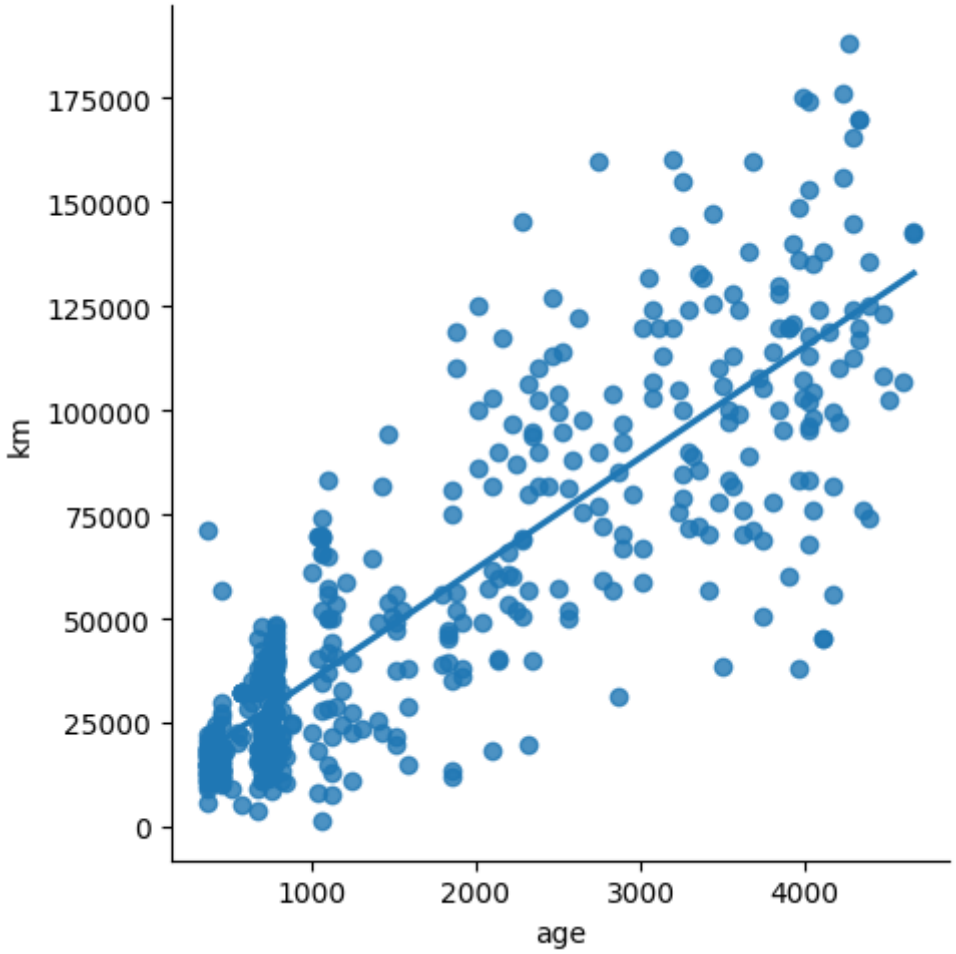


In [37]:

```
#Working with smaller dataset
dv500=dv[:][:500]
sb.lmplot(x='age',y='km',data=dv500,order=1,ci=None)
```

Out[37]:

```
<seaborn.axisgrid.FacetGrid at 0x1d5c7b3bc10>
```



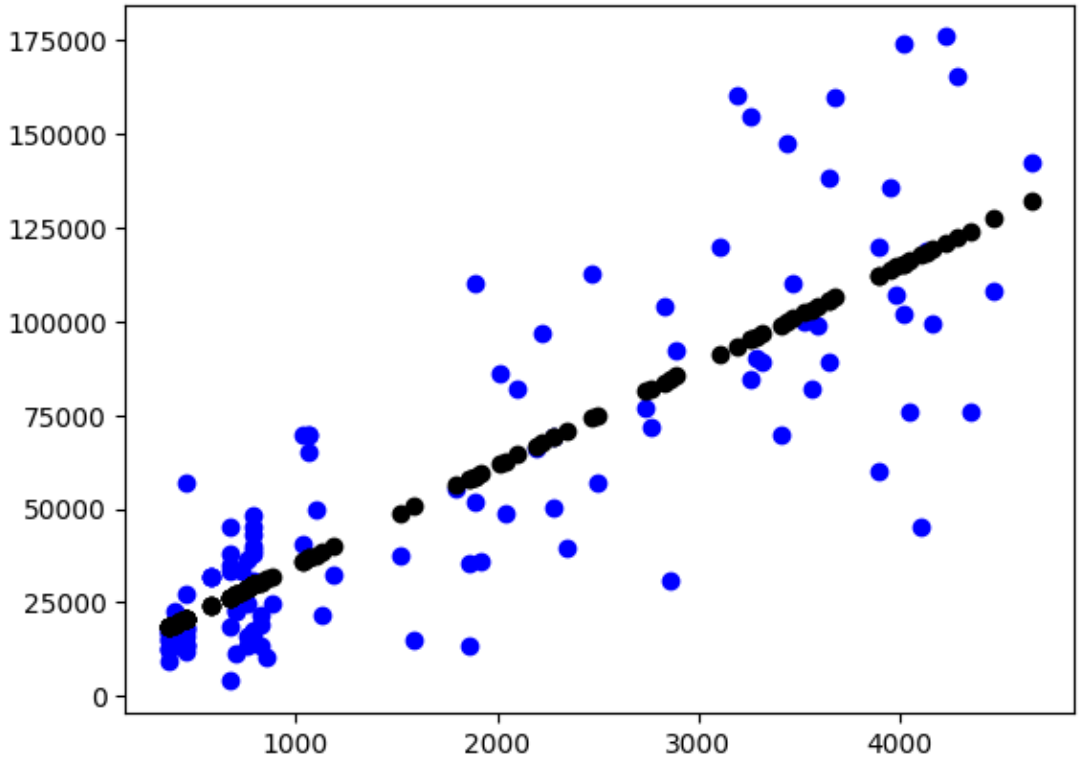
In [38]:

```
dv500.fillna(method='ffill',inplace=True)
x=np.array(dv500['age']).reshape(-1,1)
y=np.array(dv500['km']).reshape(-1,1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
```

In [39]:

```
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.scatter(x_test,y_pred,color='k')
plt.show()
```

```
Regression: 0.7193872957031895
```



In [40]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

In [41]:

```
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: 0.7193872957031895
```