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

Out[2]:

	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 [3]:
```
df=df[['age_in_days','km']]
df.columns=['age','km']
```

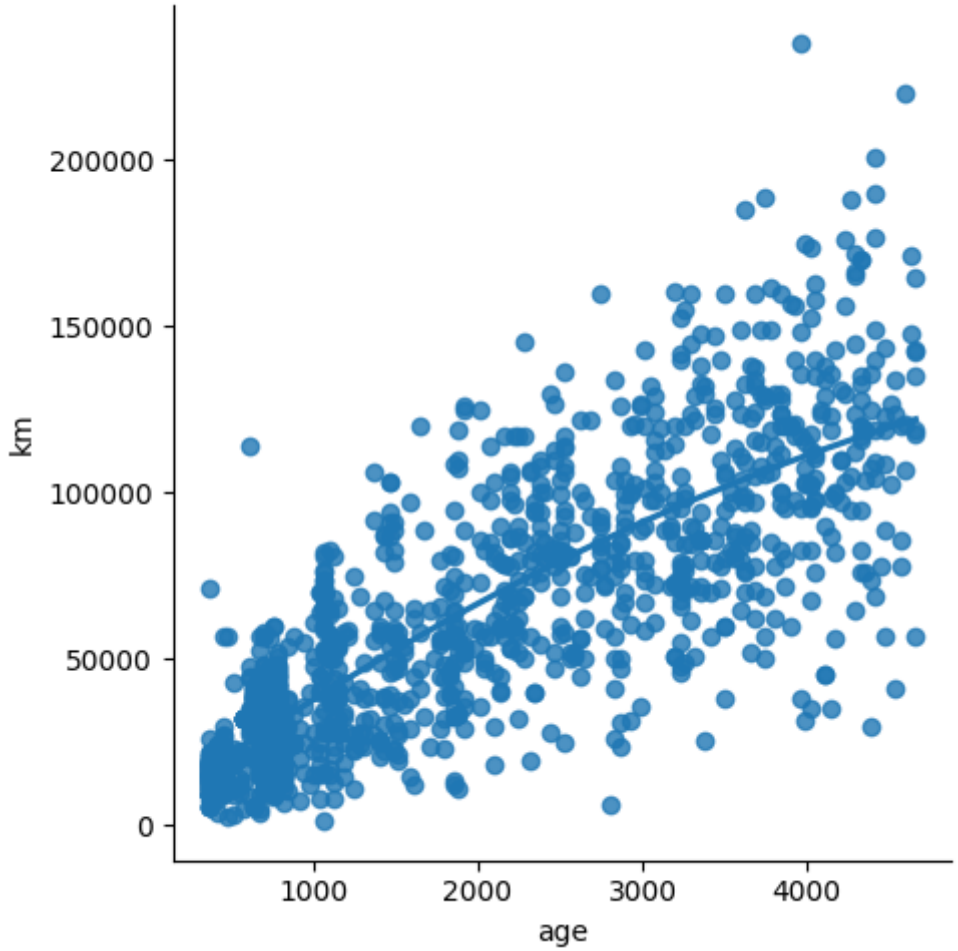
In [4]:
```
df.head(10)
```

Out[4]:

	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 [5]:
```
sns.lmplot(x ="age", y= "km", data = df,order = 2, ci = None)
```

Out[5]:
```
<seaborn.axisgrid.FacetGrid at 0x1af9ea27310>
```



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

Out[6]:

	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 [7]:
```
df.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 [8]:
```
df.fillna(method = 'ffill',inplace = True)
```

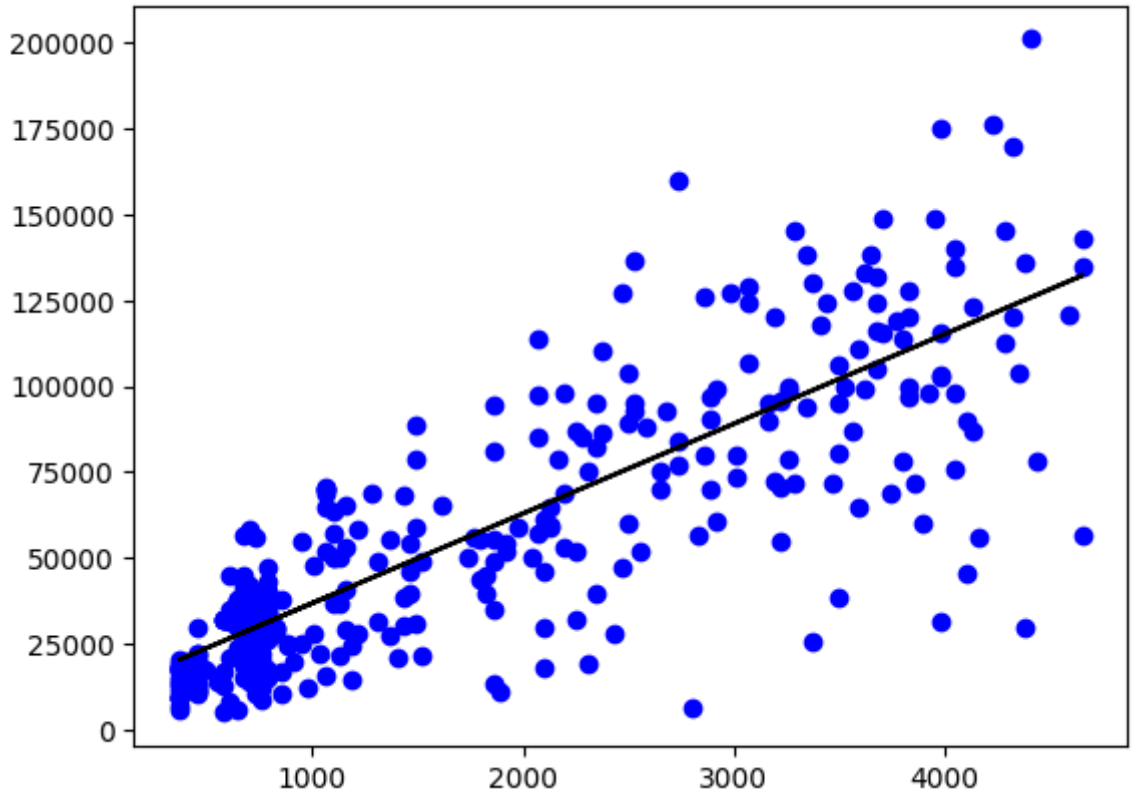
```
C:\Users\sweet\AppData\Local\Temp\ipykernel_16760\3028625988.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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['age']).reshape(-1, 1)
Y = np.array(df['km']).reshape(-1, 1)
```

In [10]:
```
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))
```

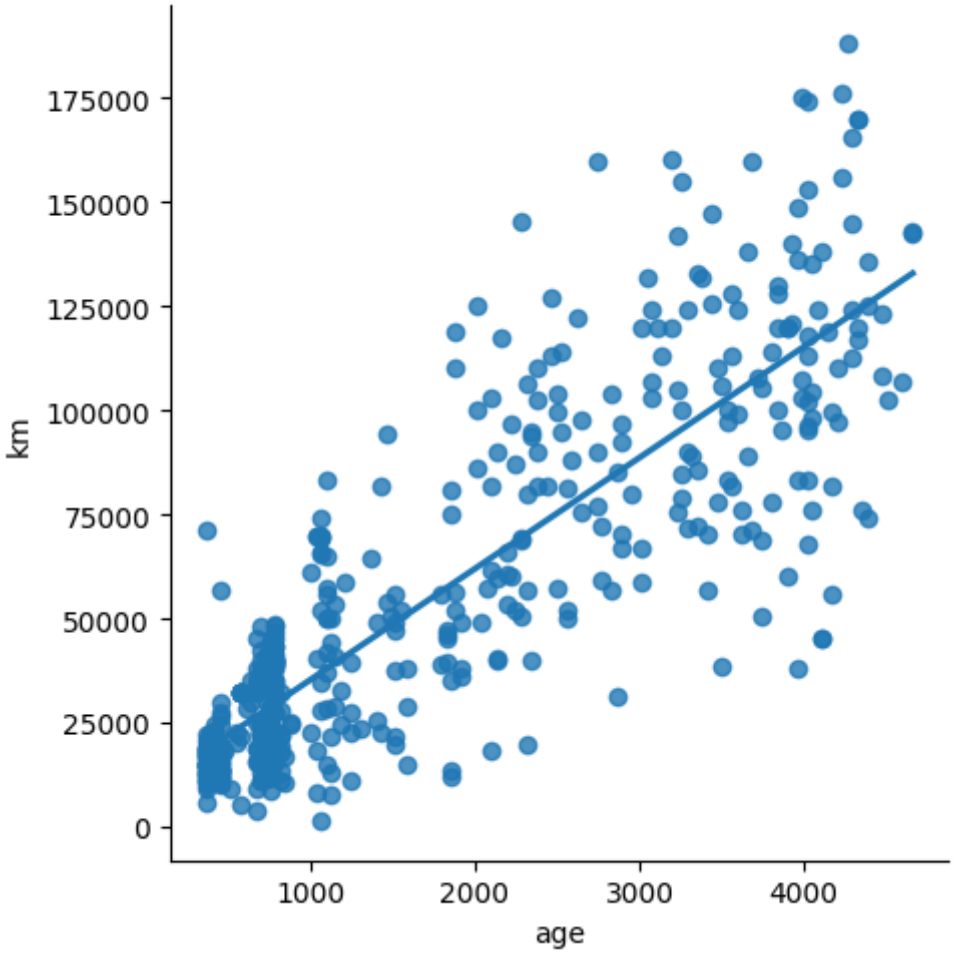
```
0.6897796921371845
```

In [11]:
```
y_pred=regr.predict(X_test)
plt.scatter(X_test, Y_test,color='b')
plt.plot(X_test,y_pred,color='k')
plt.show()
```



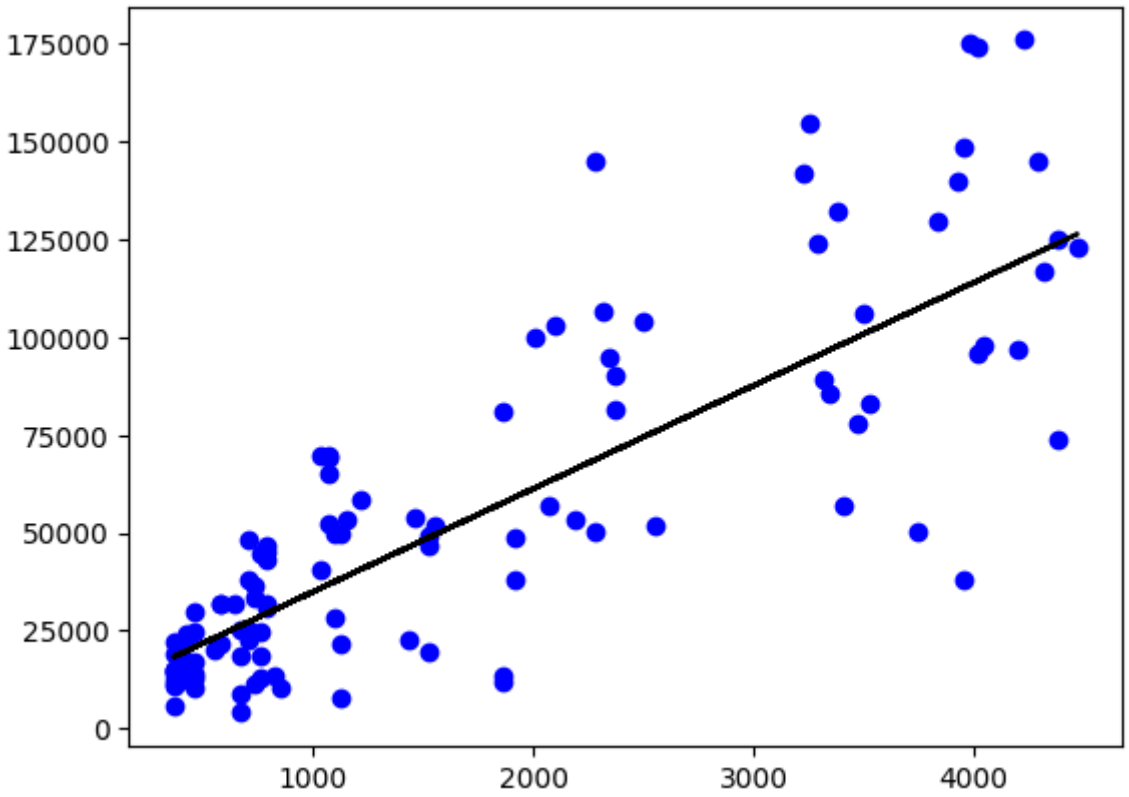
In [12]:
```
df500 = df[:][:500]
sns.lmplot(x ="age",y ="km", data = df500,order = 1,ci = None)
```

Out[12]: <seaborn.axisgrid.FacetGrid at 0x1af9eaa7f10>



In [13]:
```
df500.fillna(method = 'ffill',inplace = True)
X = np. array(df500['age']).reshape(-1, 1)
y = np.array(df500['km']).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: 0.7126781765533383
```



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: 0.7126781765533383
```