price prediction of car

In [1]:
```
import pandas as pd
import pickle
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
a=pd.read_csv("C:\\Users\\reshma_koduri\\OneDrive\\Documents\\fiat500 crt.csv")
a
```

Out[2]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	pric
0	1	lounge	51	882	25000	1	44.907242	8.611560	890
1	2	pop	51	1186	32500	1	45.666359	12.241890	880
2	3	sport	74	4658	142228	1	45.503300	11.417840	420
3	4	lounge	51	2739	160000	1	40.633171	17.634609	600
4	5	pop	73	3074	106880	1	41.903221	12.495650	570
...	...	...	...	...	...	...	...	...	.
1533	1534	sport	51	3712	115280	1	45.069679	7.704920	520
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870	460
1535	1536	pop	51	2223	60457	1	45.481541	9.413480	750
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270	599
1537	1538	pop	51	1766	54276	1	40.323410	17.568270	790

1538 rows × 9 columns

In [3]:
```
a.head(10)
```

Out[3]:

	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
5	6	pop	74	3623	70225	1	45.000702	7.682270	7900
6	7	lounge	51	731	11600	1	44.907242	8.611560	10750
7	8	lounge	51	1521	49076	1	41.903221	12.495650	9190
8	9	sport	73	4049	76000	1	45.548000	11.549470	5600
9	10	sport	51	3653	89000	1	45.438301	10.991700	6000

In [4]:

```
a.tail(10)
```

Out[4]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	pric
1528	1529	lounge	51	2861	126000	1	43.841980	10.51531	550
1529	1530	lounge	51	731	22551	1	38.122070	13.36112	990
1530	1531	lounge	51	670	29000	1	45.764648	8.99450	1080
1531	1532	sport	73	4505	127000	1	45.528511	9.59323	475
1532	1533	pop	51	1917	52008	1	45.548000	11.54947	990
1533	1534	sport	51	3712	115280	1	45.069679	7.70492	520
1534	1535	lounge	74	3835	112000	1	45.845692	8.66687	460
1535	1536	pop	51	2223	60457	1	45.481541	9.41348	750
1536	1537	lounge	51	2557	80750	1	45.000702	7.68227	599
1537	1538	pop	51	1766	54276	1	40.323410	17.56827	790

In [5]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1538 entries, 0 to 1537
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   ID               1538 non-null   int64  
 1   model            1538 non-null   object 
 2   engine_power     1538 non-null   int64  
 3   age_in_days      1538 non-null   int64  
 4   km               1538 non-null   int64  
 5   previous_owners  1538 non-null   int64  
 6   lat              1538 non-null   float64
 7   lon              1538 non-null   float64
 8   price            1538 non-null   int64  
dtypes: float64(2), int64(6), object(1)
memory usage: 108.3+ KB
```

In [6]:

```
a.describe()
```

Out[6]:

	ID	engine_power	age_in_days	km	previous_owners	lat	
count	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.0
mean	769.500000	51.904421	1650.980494	53396.011704	1.123537	43.541361	11.5
std	444.126671	3.988023	1289.522278	40046.830723	0.416423	2.133518	2.3
min	1.000000	51.000000	366.000000	1232.000000	1.000000	36.855839	7.2
25%	385.250000	51.000000	670.000000	20006.250000	1.000000	41.802990	9.5
50%	769.500000	51.000000	1035.000000	39031.000000	1.000000	44.394096	11.8
75%	1153.750000	51.000000	2616.000000	79667.750000	1.000000	45.467960	12.7

	ID	engine_power	age_in_days	km	previous_owners	lat	
max	1538.000000	77.000000	4658.000000	235000.000000	4.000000	46.795612	18.3

In [7]:
```
a.shape
```
Out[7]:
```
(1538, 9)
```

In [8]:
```
a["model"].unique()
```
Out[8]:
```
array(['lounge', 'pop', 'sport'], dtype=object)
```

In [9]:
```
a["engine_power"].unique()
```
Out[9]:
```
array([51, 74, 73, 62, 63, 66, 77, 58], dtype=int64)
```

In [10]:
```
a.groupby(["model"]).count()
```
Out[10]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
model								
lounge	1094	1094	1094	1094	1094	1094	1094	1094
pop	358	358	358	358	358	358	358	358
sport	86	86	86	86	86	86	86	86

In [11]:
```
b=a.drop(["lat","lon"],axis=1)
b
```
Out[11]:

	ID	model	engine_power	age_in_days	km	previous_owners	price
0	1	lounge	51	882	25000	1	8900
1	2	pop	51	1186	32500	1	8800
2	3	sport	74	4658	142228	1	4200
3	4	lounge	51	2739	160000	1	6000
4	5	pop	73	3074	106880	1	5700
...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	5200
1534	1535	lounge	74	3835	112000	1	4600
1535	1536	pop	51	2223	60457	1	7500
1536	1537	lounge	51	2557	80750	1	5990
1537	1538	pop	51	1766	54276	1	7900

1538 rows × 7 columns

In [12]:
```
b.shape
```
Out[12]:
```
(1538, 7)
```

In [13]:
```
c=pd.get_dummies(b,dtype=int)
c
```
Out[13]:

	ID	engine_power	age_in_days	km	previous_owners	price	model_lounge	model_pop
0	1	51	882	25000	1	8900	1	0
1	2	51	1186	32500	1	8800	0	1
2	3	74	4658	142228	1	4200	0	0
3	4	51	2739	160000	1	6000	1	0
4	5	73	3074	106880	1	5700	0	1
...	...	...	...	...	...	...	...	...
1533	1534	51	3712	115280	1	5200	0	0
1534	1535	74	3835	112000	1	4600	1	0
1535	1536	51	2223	60457	1	7500	0	1
1536	1537	51	2557	80750	1	5990	1	0
1537	1538	51	1766	54276	1	7900	0	1

1538 rows × 9 columns

In [14]:
```
b.shape
```
Out[14]:
```
(1538, 7)
```

In [15]:
```
c.shape
```
Out[15]:
```
(1538, 9)
```

linear regression

In [16]:
```
y=c["price"]
y
```
Out[16]:
```
0       8900
1       8800
2       4200
3       6000
4       5700
        ...
1533    5200
1534    4600
1535    7500
1536    5990
```

```
1537    7900
Name: price, Length: 1538, dtype: int64
```

In [17]:

```
x=c.drop('price',axis=1)
x
```

Out[17]:

	ID	engine_power	age_in_days	km	previous_owners	model_lounge	model_pop	model_
0	1	51	882	25000	1	1	0	
1	2	51	1186	32500	1	0	1	
2	3	74	4658	142228	1	0	0	
3	4	51	2739	160000	1	1	0	
4	5	73	3074	106880	1	0	1	
...	...	...	...	...	...	...	...	
1533	1534	51	3712	115280	1	0	0	
1534	1535	74	3835	112000	1	1	0	
1535	1536	51	2223	60457	1	0	1	
1536	1537	51	2557	80750	1	1	0	
1537	1538	51	1766	54276	1	0	1	

1538 rows × 8 columns

In [18]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [19]:

```
x_train.head(10)
```

Out[19]:

	ID	engine_power	age_in_days	km	previous_owners	model_lounge	model_pop	model_s
527	528	51	425	13111	1	1	0	
129	130	51	1127	21400	1	1	0	
602	603	51	2039	57039	1	0	1	
331	332	51	1155	40700	1	1	0	
323	324	51	425	16783	1	1	0	
1358	1359	51	762	29378	1	1	0	
522	523	51	425	18443	1	1	0	
584	585	51	397	11997	1	1	0	
1236	1237	51	2162	66900	1	1	0	
535	536	51	609	35000	1	0	0	

In [20]:

```
x_test.head(10)
```

Out[20]:

	ID	engine_power	age_in_days	km	previous_owners	model_lounge	model_pop	model_
481	482	51	3197	120000	2	0	1	
76	77	62	2101	103000	1	0	1	
1502	1503	51	670	32473	1	1	0	
669	670	51	913	29000	1	1	0	
1409	1410	51	762	18800	1	1	0	
1414	1415	51	762	39751	1	1	0	
1089	1090	51	882	33160	1	1	0	
1507	1508	51	701	17324	1	1	0	
970	971	51	701	29000	1	1	0	
1198	1199	51	1155	38000	1	1	0	

In [21]:

```
x_train
```

Out[21]:

	ID	engine_power	age_in_days	km	previous_owners	model_lounge	model_pop	model_
527	528	51	425	13111	1	1	0	
129	130	51	1127	21400	1	1	0	
602	603	51	2039	57039	1	0	1	
331	332	51	1155	40700	1	1	0	
323	324	51	425	16783	1	1	0	
...	...	...	...	...	...	...	...	
1130	1131	51	1127	24000	1	1	0	
1294	1295	51	852	30000	1	1	0	
860	861	51	3409	118000	1	0	1	
1459	1460	51	762	16700	1	1	0	
1126	1127	51	701	39207	1	1	0	

1030 rows × 8 columns

In [22]:

```
y_train
```

Out[22]:

```
527      9990
129      9500
602      7590
331      8750
323      9100
        ...
1130    10990
1294     9800
860      5500
1459     9990
```

```
1126     8900
Name: price, Length: 1030, dtype: int64
```

In [23]:
```
from sklearn.linear_model import LinearRegression
```

In [24]:
```
reg = LinearRegression()
reg.fit(x_train, y_train)
```

Out[24]:
```
LinearRegression()
```

In [25]:
```
ypred=reg.predict(x_test)
ypred
```

Out[25]:
```
array([ 5899.77964737,  7197.62924574,  9800.37740463,  9731.3210909 ,
        9979.99951756,  9596.52406572,  9643.73338578, 10050.33073776,
        9885.56012459,  9311.60470059, 10404.37854317,  7748.77326627,
        7673.72157177,  6503.76089728,  9604.27208331, 10408.72221178,
        9599.41735171,  7757.28075206,  4996.66853213, 10502.64231139,
       10414.2294308 , 10429.36133944,  7596.9398719 ,  9988.21510684,
        6952.25165624,  9065.16343409,  4772.288312  ,  6940.11525001,
        7801.10159922,  9604.24792145,  7282.28723062,  5261.84262409,
        5373.45316982,  5160.91018573,  8909.39762321,  5685.39083778,
        9831.30377571,  8261.38356615,  6212.08668723,  8480.76861437,
        9744.32206892,  6748.18993475,  9158.04117106, 10140.92032693,
        8698.12507001, 10374.43944919,  9131.23457001,  8911.76992913,
        7039.12870137,  9113.65847445,  9465.38254199, 10336.12580158,
       10085.66330384,  6792.95730931,  9762.59603398,  9371.43598336,
        9571.74666753, 10483.88602187,  9762.23136567,  7157.87417612,
       10042.51745284,  7023.97482239,  9920.89879869,  7152.16936549,
        6428.46049563,  9927.08893458,  9790.58469866,  8515.19767058,
        8440.66180158,  6547.80847762,  7785.13446318,  6823.91682676,
        8328.38997783, 10470.63288956,  7404.28602048,  8548.19586197,
        9812.55725973, 10065.20319918,  7298.91962975,  9480.04115956,
       10379.65950157,  8051.39314913, 10478.22278795,  3801.06120255,
       10317.81964081, 10488.36246325,  6219.09121691, 10397.82158707,
        6534.54144803,  9055.87119042, 10432.74090716,  9314.53037115,
        6767.36059542,  3328.81851306, 10131.23611482,  9823.8191059 ,
        6233.28720959,  5031.73921855,  9053.77470799,  9816.00882493,
        5471.75249123,  5668.84783459, 10132.45253753,  8045.73210461,
       10428.82182153,  6796.50688961,  6678.31755056,  5777.97901331,
        8824.42481646,  9902.53553708, 10454.4666007 ,  9398.45161112,
        9025.85855762, 10087.77691363, 10444.01649714, 10215.46161238,
        9752.38039204,  9291.77779835, 10333.61634118,  5337.64426142,
        9756.41053083,  6121.86669001,  9042.17650753, 10200.54823683,
        9232.30786194,  9886.85932572,  8364.88286712,  8407.62481862,
        7510.21897396, 10503.50280474, 10410.60417948, 10067.92142721,
       10213.10285619,  6824.40428783,  9584.31621133, 10477.42497298,
        9607.29479057,  7996.51950016,  9650.66260176,  7909.35045968,
       10458.58171516,  9178.92605043,  5787.24059264,  6674.54716343,
        8285.42271321, 10482.06648523,  9962.57285307,  9749.14008762,
       10653.60132894,  7532.70615215,  6735.67649881,  8012.07769702,
       10281.3946047 ,  8860.90114279,  8365.82696593,  9648.54454897,
        9763.59188019, 10082.49019892, 10347.81286764,  7128.13584657,
        9728.1405881 ,  6283.41989125,  7854.43531849,  9367.86479641,
        4980.95833992,  9358.48399276, 10030.84118879, 10120.70583821,
        6386.7781425 ,  9831.19828452,  9063.94195958,  5220.86002087,
        5517.11530327,  4473.45028636, 10252.36956549, 10046.61099803,
        5463.26923098,  8590.37673683,  7006.54516088, 10020.91145712,
       10124.06025085,  6060.6999267 ,  9715.11680734,  9659.99387728,
        9154.45966418,  9157.84122047, 10147.16712385,  9853.77923811,
        7356.6865774 ,  5143.33165723,  9432.18223147, 10215.68087954,
```

```
 5607.10314766, 10626.1169135 ,  6118.04289246,  9848.92992522,
 9811.94539615,  7791.16377684,  6602.45896948,  9944.68476715,
 8316.59336022,  9073.34594295,  6111.62189531, 10446.46448484,
 6398.14070518,  8624.12284224,  8311.16178629,  9753.10680417,
 8225.08420112, 10064.50630092, 10052.23409671, 10098.56264954,
10361.47837175,  8485.4409232 ,  6681.00431434,  9408.76426219,
 6545.38191593, 10390.44878073,  9040.08224603, 10429.27015352,
 9095.39717978,  9903.71419783,  8406.67033768,  9279.45427755,
10025.02420518,  8434.25683875,  4704.68348995, 10104.79296616,
10008.01172598, 10562.069171  , 10187.05199839,  4932.13977277,
 7227.14540005,  9620.93541143,  9818.10659892,  5635.77281584,
10077.80679352,  5119.34566584,  8365.39509055,  7449.02502112,
 7849.84539778,  9710.21844323,  8647.40656762, 10420.6870726 ,
 7170.58061309,  9714.02814666,  8032.32472193,  7425.56922823,
10482.16586168, 10406.3869895 ,  5401.79745107,  9122.0165486 ,
 9597.97229245, 10581.63654938, 10141.32739981,  9203.15899424,
 6077.61452709,  9664.23021907, 10449.21924068,  8879.17566437,
 8158.73192584,  9946.80868659,  9404.18636408,  9900.88642841,
10440.48296647, 10440.02664169,  9629.0783767 ,  8156.19315992,
10395.01726927, 10327.77641295,  8758.84752593,  8271.03867398,
 6835.60437927, 10186.61323004,  4826.62436194,  8770.88291508,
 5721.32362057, 10094.64625833,  8788.4342758 , 10004.3262471 ,
 9641.21860186, 10511.56009626, 10144.80816809,  9741.74797425,
 9729.42839754,  6812.16303264,  9625.06535478,  8533.36084724,
 6638.53573409, 10339.36287461, 10078.3890038 , 10183.66146605,
 5874.12498048,  8726.6643865 ,  9624.59229204,  5714.67253044,
 8353.32273194,  9625.861619  ,  4295.85638809, 10009.60170323,
 9841.44773564,  7801.52658197, 10083.71707973, 10504.7721317 ,
10222.57849347,  6851.54663842,  6526.51427642, 10399.21193868,
 8479.07219717,  5398.81635625,  9866.63997959,  4754.9680649 ,
 7829.92939939,  5396.20859662,  9874.32950027, 10449.65722248,
 9632.36166395,  8802.57321694,  7717.59639713,  4236.90408712,
 9835.47539985, 10349.56651647,  5778.39752839, 10191.79238956,
 9471.68820756,  8010.4707309 ,  5555.7111004 ,  9861.71891322,
10474.80105677,  6314.98093974,  9523.7453663 ,  9561.11041677,
10352.80965879,  9527.49307259,  9792.37364303,  9616.0065618 ,
 6807.53991198,  7903.11943666, 10337.32030094, 10357.37387919,
 7384.90841953,  9932.44462146, 10457.13633287, 10605.11604236,
10327.00132827,  9996.59007033,  9558.56544772,  7700.22947415,
 9295.3841376 , 10004.99198578,  9956.27420989,  9954.02181373,
 9380.89429211,  9625.02615488,  9693.54168863,  9866.33338396,
 8837.79737236,  6180.4995007 ,  6315.41170408,  8140.32142265,
 8558.30561171,  6544.66054253,  6827.17241535,  5507.41833662,
 8166.78045128,  9908.12458322,  7745.66342057,  9844.49790596,
10174.79553559,  5746.8293518 ,  9862.98243724, 10003.10167457,
 8085.76425129,  4501.09803781, 10661.76991323,  3794.99190937,
 9982.04980447, 10469.13214313,  5810.50066645,  5447.10033776,
10445.7927837 ,  6780.66231709,  8921.18579817, 10472.88930782,
 9444.52338801,  9952.09708594,  8461.18187543,  7991.57597229,
10424.15698606,  5416.92320084,  9853.81621966, 10181.69406325,
10299.81798176,  9479.02375558,  9180.38501329,  9790.07245428,
 5718.57999355,  4913.72264519,  4808.74310294,  9627.94079876,
 6155.05001405,  9841.089319  , 10027.56518619,  5026.85606469,
 7999.04161281,  9767.14523706,  5918.04692352, 10332.87612869,
 5451.3135132 ,  9681.50473606, 10135.98241131, 10150.92850481,
 9422.11906717,  4905.01131578,  5797.34608151,  7056.68769764,
 9971.18382668, 10360.15654753,  9987.70474839,  7743.36310516,
 8783.29617999,  9909.31017188, 10248.09893653,  9918.98006458,
 8415.08618207,  9360.42935199,  8470.45081947,  9791.13418786,
 9784.2656714 ,  9797.85968036,  6692.09111957,  7342.30281167,
 8743.41864477,  9923.03184648,  9751.23832604, 10429.33283903,
 8200.58522359,  6694.13167995,  9973.16744797,  8903.70063631,
 9984.58845839, 10225.15247789, 10294.02652381, 10034.72455704,
 9305.10621345,  9948.16523916,  9186.20845773, 10079.79764999,
```

```
         7793.82239102,  6071.31707428,  8749.4287838 , 10242.58466298,
         5670.08789955, 10031.63232869,  9657.45573603,  7583.19020833,
         9319.6055608 ,  7368.43448617, 10331.08272186, 10085.00458146,
        10531.97199511,  9969.43792134, 10045.80840799,  6272.3376062 ,
        10637.67447027,  9901.28901464, 10528.99812638,  9658.37727979,
         9706.42308428,  6229.17037014,  8088.31830318, 10346.59831789,
         6356.69367452,  7396.91036774, 10017.20542568,  6844.59813249,
         7887.68750992,  5288.3270576 ,  4549.21369484,  8665.20883653,
         6933.12654174,  7409.46613273,  6810.73951513,  7100.11749152,
         9914.83740258,  8836.58079762,  9389.49036004, 10345.04939513,
        10099.22772449, 10388.01465664,  9737.49969878,  6047.91809547,
         9772.54808451,  7667.85544351,  5578.21674253,  4932.47966028,
         9798.63085318,  9308.19800043, 10147.12672348,  6229.17451824,
         8639.82429626, 10384.38382728,  5122.50109744, 10075.09512522,
         6293.83314577,  9923.60261897,  8319.81828156, 10388.99473827])
```

In [26]:

```
from sklearn.metrics import r2_score
r2_score(y_test,ypred)
```

Out[26]:

```
0.8404533110612131
```

In [27]:

```
from sklearn.metrics import mean_squared_error
mean_squared_error(ypred,y_test)
```

Out[27]:

```
585925.1591527035
```

In [28]:

```
results=pd.DataFrame(columns=['Price','Predicted'])
results['Price']=y_test
results["Predicted"]=ypred
results=results.reset_index()
results['Id']=results.index
results.head(5)
```

Out[28]:

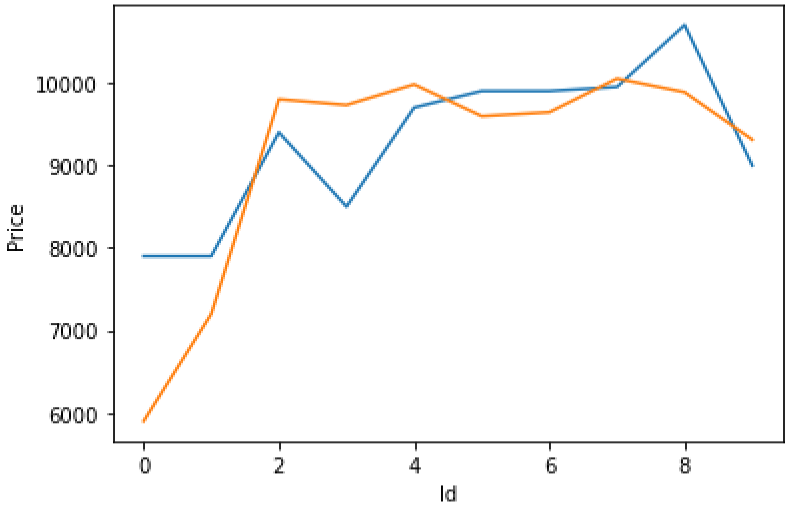
	index	Price	Predicted	Id
0	481	7900	5899.779647	0
1	76	7900	7197.629246	1
2	1502	9400	9800.377405	2
3	669	8500	9731.321091	3
4	1409	9700	9979.999518	4

In [29]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='Id',y='Price',data=results.head(10))
sns.lineplot(x='Id',y='Predicted',data=results.head(10))
plt.plot()
```

Out[29]:

```
[]
```



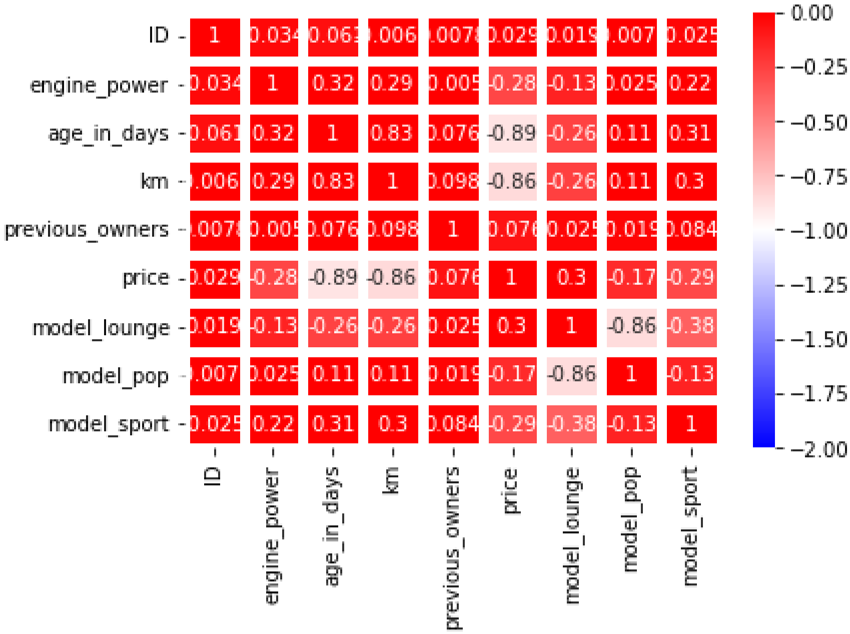
In [30]:
```
cor=c.corr()
cor
```

Out[30]:

	ID	engine_power	age_in_days	km	previous_owners	price	mode
ID	1.000000	-0.034059	-0.060753	-0.006537	0.007803	0.028516	
engine_power	-0.034059	1.000000	0.319190	0.285495	-0.005030	-0.277235	-
age_in_days	-0.060753	0.319190	1.000000	0.833890	0.075775	-0.893328	-
km	-0.006537	0.285495	0.833890	1.000000	0.097539	-0.859373	-
previous_owners	0.007803	-0.005030	0.075775	0.097539	1.000000	-0.076274	-
price	0.028516	-0.277235	-0.893328	-0.859373	-0.076274	1.000000	
model_lounge	0.019193	-0.133321	-0.259863	-0.255746	-0.024643	0.302299	
model_pop	-0.007142	0.024783	0.108327	0.109024	-0.019316	-0.167190	-
model_sport	-0.024718	0.217362	0.313276	0.303874	0.084129	-0.288706	-

In [31]:
```
import seaborn as sb
sb.heatmap(cor,vmax=0,vmin=-2,annot=True,linewidth=-5,cmap="bwr")
```

Out[31]: <AxesSubplot:>


ID
engine_power
age_in_days
km
previous_owners
price
model_lounge
model_pop
model_sport
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−1.75
−2.00