Car price prediction

July 25, 2024

1 CAR PRICE PREDICTION

1.1 Load the dataset

[180]:
```
import pandas as pd
import seaborn as sns
from sklearn.linear_model import LinearRegression
model = LinearRegression()
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

[181]:
```
car_df = pd.read_csv("C:/Users/deeps/OneDrive/Documents/WEBSTER/DATASET/
 ↪carprices.csv")
car_df
```

[181]:

	Car Model	Mileage	Sell Price($)	Age(yrs)
0	BMW X5	69000	18000	6
1	BMW X5	35000	34000	3
2	BMW X5	57000	26100	5
3	BMW X5	22500	40000	2
4	BMW X5	46000	31500	4
5	Audi A5	59000	29400	5
6	Audi A5	52000	32000	5
7	Audi A5	72000	19300	6
8	Audi A5	91000	12000	8
9	Mercedez Benz C class	67000	22000	6
10	Mercedez Benz C class	83000	20000	7
11	Mercedez Benz C class	79000	21000	7
12	Mercedez Benz C class	59000	33000	5
13	Mercedez Benz C class	58000	35000	4
14	Mercedez Benz C class	66000	23000	6
15	Mercedez Benz C class	80000	22000	7
16	Mercedez Benz C class	70000	21000	6
17	Mercedez Benz C class	85000	20000	8
18	Mercedez Benz C class	79000	25000	5
19	Audi A5	71000	19900	5

```
20                Audi A5    99000          15000         6
21                 BMW X5    35500          38000         2
22                 BMW X5    56000          30000         3
23                 BMW X5    30000          35000         3
24                Audi A5    50000          38000         4
```

1.2 Data Exploration

[182]:
```
car_df.dtypes
```

[182]:
```
Car Model        object
Mileage           int64
Sell Price($)     int64
Age(yrs)          int64
dtype: object
```

[183]:
```
car_df.describe()
```

[183]:
```
             Mileage  Sell Price($)   Age(yrs)
count     25.000000      25.000000  25.000000
mean   62840.000000   26408.000000   5.120000
std    19375.521498    7841.551717   1.691153
min    22500.000000   12000.000000   2.000000
25%    52000.000000   20000.000000   4.000000
50%    66000.000000   25000.000000   5.000000
75%    79000.000000   33000.000000   6.000000
max    99000.000000   40000.000000   8.000000
```

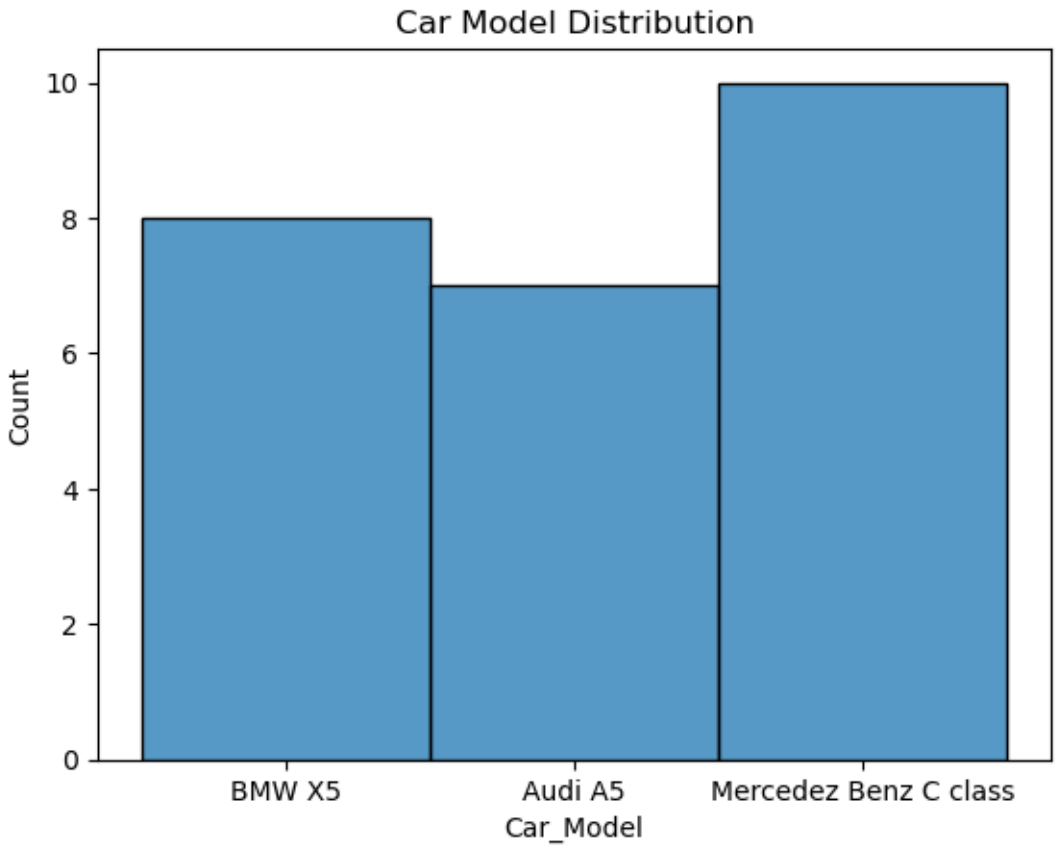
[184]:
```
car_df.groupby("Car Model").mean()
```

[184]:
```
                           Mileage  Sell Price($)  Age(yrs)
Car Model
Audi A5                70571.428571   23657.142857  5.571429
BMW X5                 43875.000000   31575.000000  3.500000
Mercedez Benz C class  72600.000000   24200.000000  6.100000
```

[185]:
```
car_df.groupby("Car Model").median()
```

[185]:
```
                       Mileage  Sell Price($)  Age(yrs)
Car Model
Audi A5                71000.0        19900.0       5.0
BMW X5                 40750.0        32750.0       3.0
Mercedez Benz C class  74500.0        22000.0       6.0
```

[186]:
```
car_df.rename(columns={'Car Model':'Car_Model'}, inplace=True)
sns.histplot(car_df.Car_Model)
plt.title('Car Model Distribution')
plt.show()
```



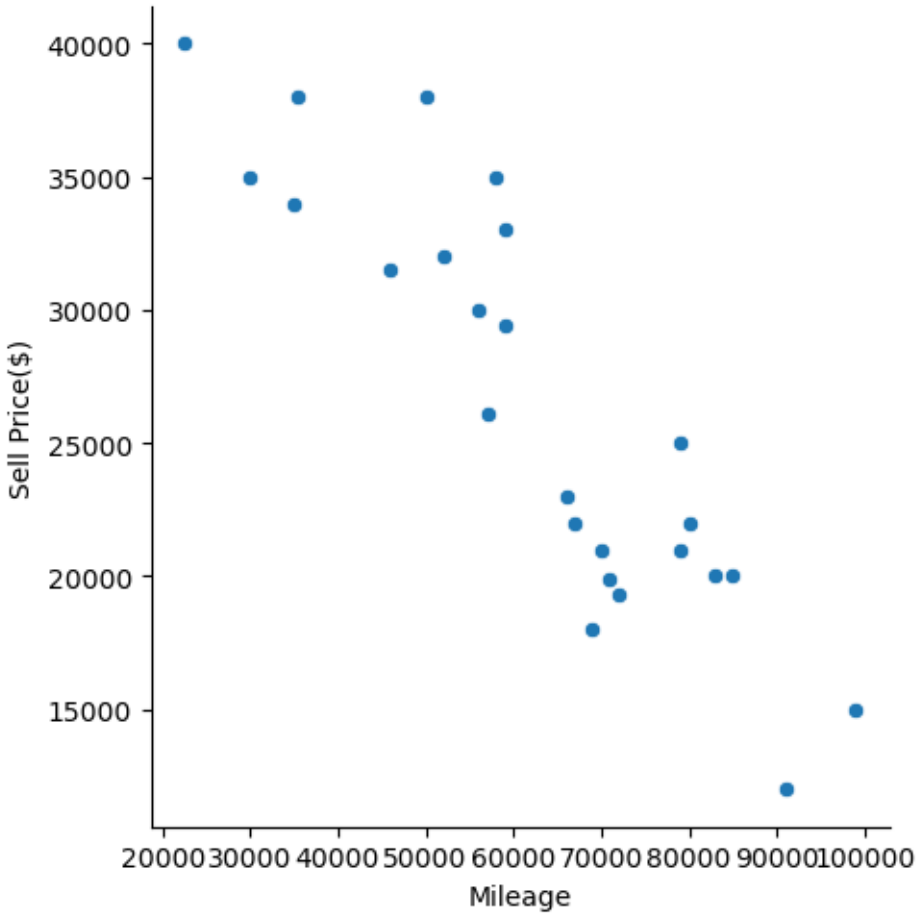
[187]:
```
car_df.value_counts()
```

[187]:
```
Car_Model              Mileage  Sell Price($)  Age(yrs)
Audi A5                50000    38000          4           1
BMW X5                 57000    26100          5           1
Mercedez Benz C class  83000    20000          7           1
                       80000    22000          7           1
                       79000    25000          5           1
                                21000          7           1
                       70000    21000          6           1
                       67000    22000          6           1
                       66000    23000          6           1
                       59000    33000          5           1
                       58000    35000          4           1
BMW X5                 69000    18000          6           1
                       56000    30000          3           1
Audi A5                52000    32000          5           1
BMW X5                 46000    31500          4           1
                       35500    38000          2           1
                       35000    34000          3           1
```

```
                          30000    35000          3           1
                          22500    40000          2           1
Audi A5                   99000    15000          6           1
                          91000    12000          8           1
                          72000    19300          6           1
                          71000    19900          5           1
                          59000    29400          5           1
Mercedez Benz C class     85000    20000          8           1
Name: count, dtype: int64
```

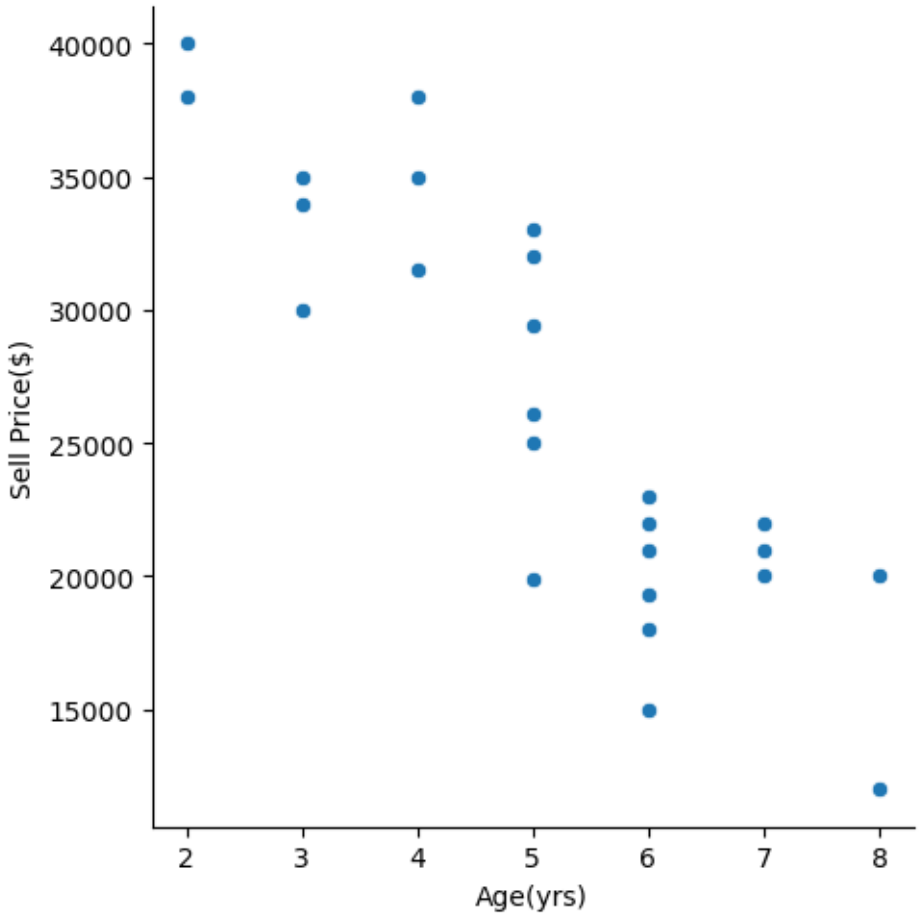
[188]:
```
sns.relplot(data=car_df, x='Mileage', y='Sell Price($)')
```

[188]:
```
<seaborn.axisgrid.FacetGrid at 0x168794e4a10>
```



[189]:
```
sns.relplot(data=car_df, x='Age(yrs)', y='Sell Price($)')
```

[189]:
```
<seaborn.axisgrid.FacetGrid at 0x1687956bb50>
```



1.3 Data Preparation

[190]:
```
dummies = pd.get_dummies(car_df['Car_Model'])
dummies
```

[190]:

	Audi A5	BMW X5	Mercedez Benz C class
0	False	True	False
1	False	True	False
2	False	True	False
3	False	True	False
4	False	True	False
5	True	False	False
6	True	False	False
7	True	False	False
8	True	False	False
9	False	False	True
10	False	False	True
11	False	False	True

```
12    False   False                  True
13    False   False                  True
14    False   False                  True
15    False   False                  True
16    False   False                  True
17    False   False                  True
18    False   False                  True
19     True   False                 False
20     True   False                 False
21    False    True                 False
22    False    True                 False
23    False    True                 False
24     True   False                 False
```

[191]:
```
# Assuming your DataFrame is named df
dummies = dummies.astype(int)
dummies
```

[191]:

	Audi A5	BMW X5	Mercedez Benz C class
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
5	1	0	0
6	1	0	0
7	1	0	0
8	1	0	0
9	0	0	1
10	0	0	1
11	0	0	1
12	0	0	1
13	0	0	1
14	0	0	1
15	0	0	1
16	0	0	1
17	0	0	1
18	0	0	1
19	1	0	0
20	1	0	0
21	0	1	0
22	0	1	0
23	0	1	0
24	1	0	0

[192]:
```
combined = pd.concat([car_df,dummies],axis='columns')
combined
```

[192]:

	Car_Model	Mileage	Sell Price($)	Age(yrs)	Audi A5	BMW X5	\
0	BMW X5	69000	18000	6	0	1	
1	BMW X5	35000	34000	3	0	1	
2	BMW X5	57000	26100	5	0	1	
3	BMW X5	22500	40000	2	0	1	
4	BMW X5	46000	31500	4	0	1	
5	Audi A5	59000	29400	5	1	0	
6	Audi A5	52000	32000	5	1	0	
7	Audi A5	72000	19300	6	1	0	
8	Audi A5	91000	12000	8	1	0	
9	Mercedez Benz C class	67000	22000	6	0	0	
10	Mercedez Benz C class	83000	20000	7	0	0	
11	Mercedez Benz C class	79000	21000	7	0	0	
12	Mercedez Benz C class	59000	33000	5	0	0	
13	Mercedez Benz C class	58000	35000	4	0	0	
14	Mercedez Benz C class	66000	23000	6	0	0	
15	Mercedez Benz C class	80000	22000	7	0	0	
16	Mercedez Benz C class	70000	21000	6	0	0	
17	Mercedez Benz C class	85000	20000	8	0	0	
18	Mercedez Benz C class	79000	25000	5	0	0	
19	Audi A5	71000	19900	5	1	0	
20	Audi A5	99000	15000	6	1	0	
21	BMW X5	35500	38000	2	0	1	
22	BMW X5	56000	30000	3	0	1	
23	BMW X5	30000	35000	3	0	1	
24	Audi A5	50000	38000	4	1	0	

	Mercedez Benz C class
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1

```
19                      0
20                      0
21                      0
22                      0
23                      0
24                      0
```

[193]:
```
combined.describe()
```

[193]:
```
             Mileage  Sell Price($)   Age(yrs)    Audi A5     BMW X5  \
count     25.000000      25.000000  25.000000  25.000000  25.000000
mean   62840.000000   26408.000000   5.120000   0.280000   0.320000
std    19375.521498    7841.551717   1.691153   0.458258   0.476095
min    22500.000000   12000.000000   2.000000   0.000000   0.000000
25%    52000.000000   20000.000000   4.000000   0.000000   0.000000
50%    66000.000000   25000.000000   5.000000   0.000000   0.000000
75%    79000.000000   33000.000000   6.000000   1.000000   1.000000
max    99000.000000   40000.000000   8.000000   1.000000   1.000000

       Mercedez Benz C class
count                   25.0
mean                     0.4
std                      0.5
min                      0.0
25%                      0.0
50%                      0.0
75%                      1.0
max                      1.0
```

[194]:
```
final_car = combined.drop(['Car_Model'], axis='columns')
final_car
```

[194]:

	Mileage	Sell Price($)	Age(yrs)	Audi A5	BMW X5	Mercedez Benz C class
0	69000	18000	6	0	1	0
1	35000	34000	3	0	1	0
2	57000	26100	5	0	1	0
3	22500	40000	2	0	1	0
4	46000	31500	4	0	1	0
5	59000	29400	5	1	0	0
6	52000	32000	5	1	0	0
7	72000	19300	6	1	0	0
8	91000	12000	8	1	0	0
9	67000	22000	6	0	0	1
10	83000	20000	7	0	0	1
11	79000	21000	7	0	0	1
12	59000	33000	5	0	0	1
13	58000	35000	4	0	0	1

14	66000	23000	6	0	0	1
15	80000	22000	7	0	0	1
16	70000	21000	6	0	0	1
17	85000	20000	8	0	0	1
18	79000	25000	5	0	0	1
19	71000	19900	5	1	0	0
20	99000	15000	6	1	0	0
21	35500	38000	2	0	1	0
22	56000	30000	3	0	1	0
23	30000	35000	3	0	1	0
24	50000	38000	4	1	0	0

1.4 Model building

[195]:
```
### USing entire dataset for model building.
```

[196]:
```
## Dropping sell price column from predictors as it is a target variable.
X = final_car.drop('Sell Price($)', axis='columns')
X
```

[196]:

	Mileage	Age(yrs)	Audi A5	BMW X5	Mercedez Benz C class
0	69000	6	0	1	0
1	35000	3	0	1	0
2	57000	5	0	1	0
3	22500	2	0	1	0
4	46000	4	0	1	0
5	59000	5	1	0	0
6	52000	5	1	0	0
7	72000	6	1	0	0
8	91000	8	1	0	0
9	67000	6	0	0	1
10	83000	7	0	0	1
11	79000	7	0	0	1
12	59000	5	0	0	1
13	58000	4	0	0	1
14	66000	6	0	0	1
15	80000	7	0	0	1
16	70000	6	0	0	1
17	85000	8	0	0	1
18	79000	5	0	0	1
19	71000	5	1	0	0
20	99000	6	1	0	0
21	35500	2	0	1	0
22	56000	3	0	1	0
23	30000	3	0	1	0
24	50000	4	1	0	0

[197]:
```
Y  = final_car[['Sell Price($)']]
Y
```

[197]:

	Sell Price($)
0	18000
1	34000
2	26100
3	40000
4	31500
5	29400
6	32000
7	19300
8	12000
9	22000
10	20000
11	21000
12	33000
13	35000
14	23000
15	22000
16	21000
17	20000
18	25000
19	19900
20	15000
21	38000
22	30000
23	35000
24	38000

[198]:
```
## fitting linear regression
model.fit(X,Y)
```

[198]:
```
LinearRegression()
```

[199]:
```
model.predict(X)
```

[199]:
```
array([[18678.01574345],
       [35262.93930597],
       [24398.27900102],
       [41127.1691161 ],
       [29830.6091535 ],
       [28283.26364269],
       [30298.79537843],
       [22275.06728   ],
       [12274.20629038],
       [26038.93198822],
```

```
       [19166.93631022],
       [20318.66873065],
       [30607.46282538],
       [33160.46192679],
       [26326.86509333],
       [20030.73562554],
       [25175.13267291],
       [16326.0041037 ],
       [24848.80072326],
       [24828.06638142],
       [14500.87344215],
       [37384.03874972],
       [29216.34409875],
       [36702.6048315 ],
       [33139.72758495]])
```

1.4.1 Tell me the score (accuracy) of your model

[200]:
```
model.score(X,Y)
```

[200]:
```
0.9006591109002584
```

1.4.2 1) Predict price of a mercedez benz that is 5 yr old with mileage 45000

[201]:
```
model.predict([[45000,5,0,0,1]])
```

[201]:
```
array([[34638.52629686]])
```

1.4.3 2) Predict price of a BMW X5 that is 7 yr old with mileage 96000.

[202]:
```
model.predict([[86000,7,0,1,0]])
```

[202]:
```
array([[11518.08696034]])
```

1.4.4 3) Predict price of a Audi A5 car that is 4 yr old with mileage 80000.

[203]:
```
model.predict([[80000,4,1,0,0]])
```

[203]:
```
array([[24501.73443177]])
```

1.4.5 3) Predict price of a mercedez benz car that is 1 yr old with mileage 50000.

[204]:
```
model.predict([[50000,1,0,0,1]])
```

[204]:
```
array([[42259.12475656]])
```

[ ]:
```

```