In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
data=pd.read_csv('BMW_Car_Sales_Classification.csv')
```

In [3]:
```
#data=data.drop('Sales_Classification',axis=1)
```

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Model                 50000 non-null  object 
 1   Year                  50000 non-null  int64  
 2   Region                50000 non-null  object 
 3   Color                 50000 non-null  object 
 4   Fuel_Type             50000 non-null  object 
 5   Transmission          50000 non-null  object 
 6   Engine_Size_L         50000 non-null  float64
 7   Mileage_KM            50000 non-null  int64  
 8   Price_USD             50000 non-null  int64  
 9   Sales_Volume          50000 non-null  int64  
 10  Sales_Classification  50000 non-null  object 
dtypes: float64(1), int64(4), object(6)
memory usage: 4.2+ MB
```

In [5]:
```
data= pd.get_dummies(data,columns=['Sales_Classification','Transmission','Fuel_Type
```

In [6]:
```
data.head()
```

Out[6]:

	Year	Engine_Size_L	Mileage_KM	Price_USD	Sales_Volume	Sales_Classification_High	Sa
0	2016	3.5	151748	98740	8300	True	
1	2013	1.6	121671	79219	3428	False	
2	2022	4.5	10991	113265	6994	False	
3	2024	1.7	27255	60971	4047	False	
4	2020	2.1	122131	49898	3080	False	

5 rows × 36 columns

In [7]:
```
data.describe()
```

Out[7]:

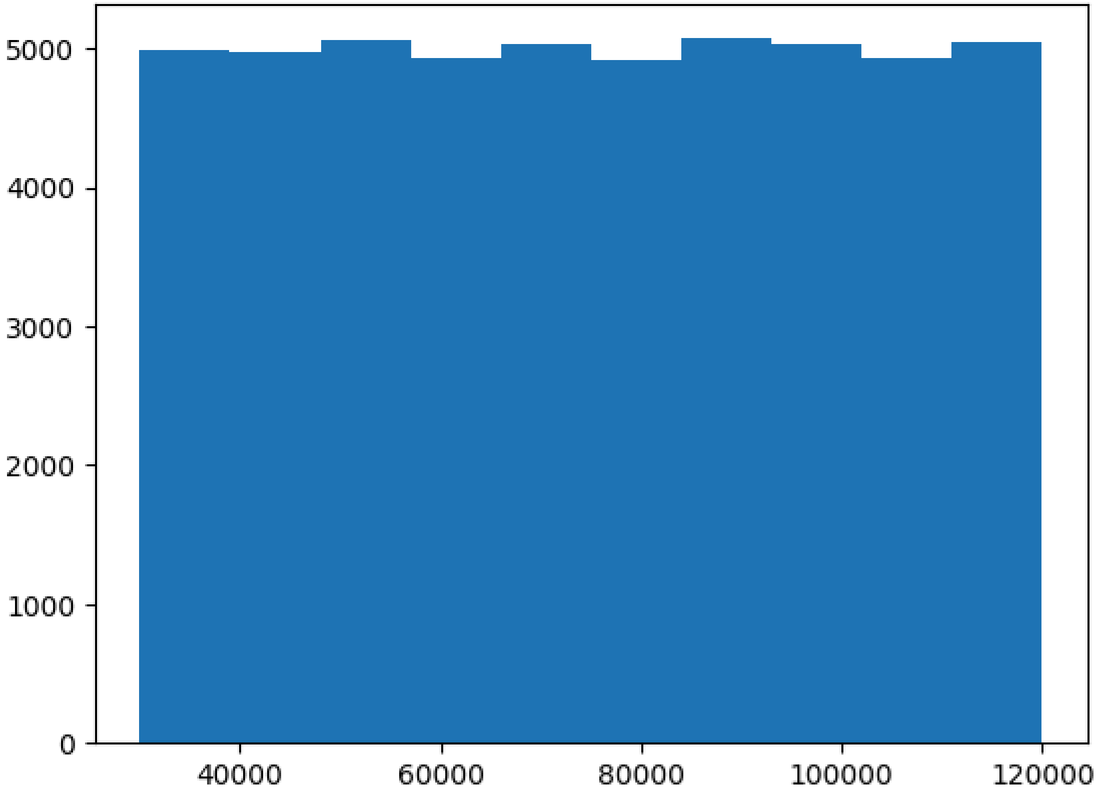
	Year	Engine_Size_L	Mileage_KM	Price_USD	Sales_Volume
count	50000.000000	50000.000000	50000.000000	50000.000000	50000.000000
mean	2017.015700	3.247180	100307.203140	75034.600900	5067.514680
std	4.324459	1.009078	57941.509344	25998.248882	2856.767125
min	2010.000000	1.500000	3.000000	30000.000000	100.000000
25%	2013.000000	2.400000	50178.000000	52434.750000	2588.000000
50%	2017.000000	3.200000	100388.500000	75011.500000	5087.000000
75%	2021.000000	4.100000	150630.250000	97628.250000	7537.250000
max	2024.000000	5.000000	199996.000000	119998.000000	9999.000000

In [8]:

```
plt.hist(data['Price_USD'])
```

Out[8]:

```
(array([4990., 4974., 5056., 4931., 5038., 4921., 5072., 5028., 4936.,
        5054.]),
 array([ 30000. ,  38999.8,  47999.6,  56999.4,  65999.2,  74999. ,
         83998.8,  92998.6, 101998.4, 110998.2, 119998. ]),
 <BarContainer object of 10 artists>)
```

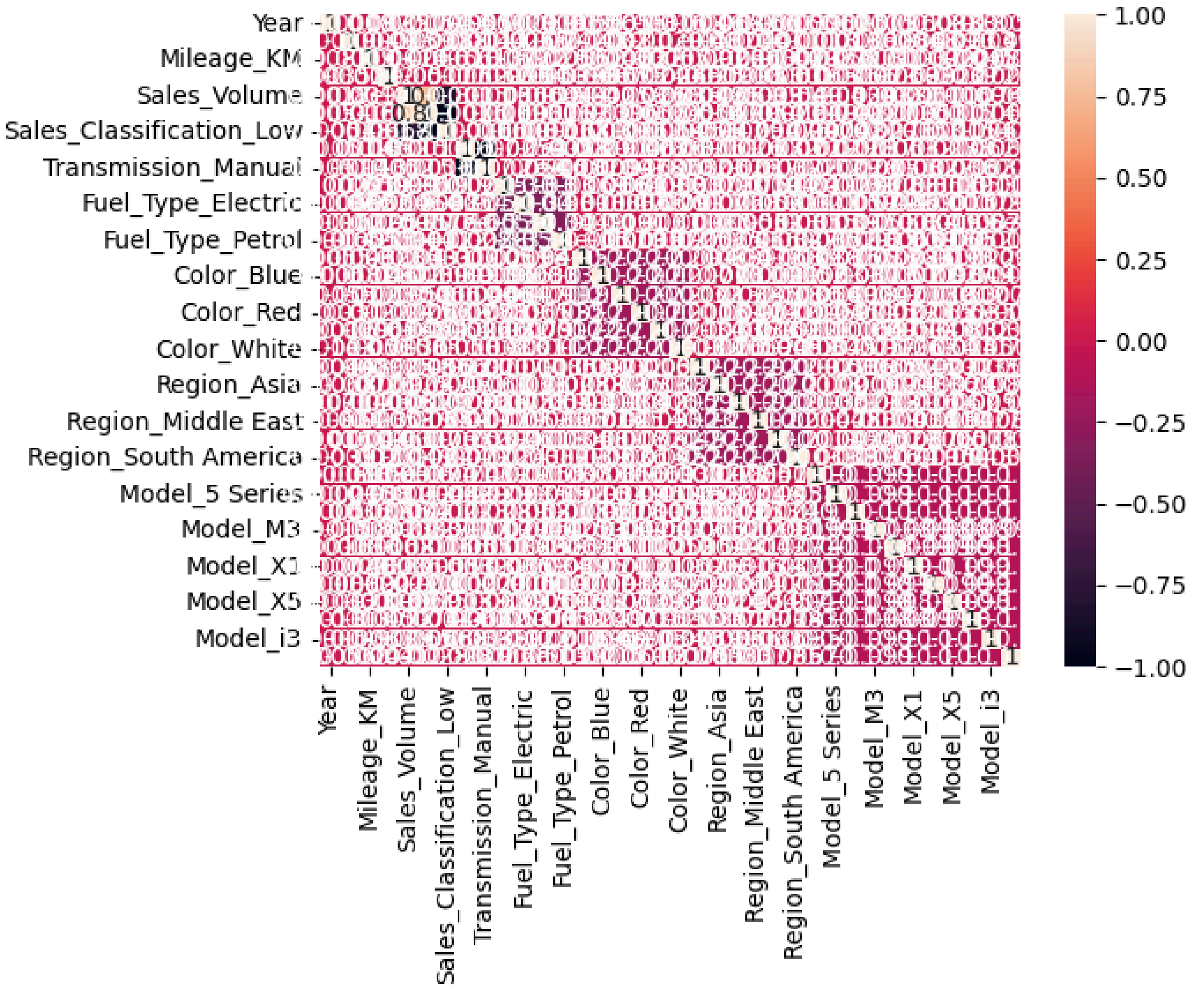


In [9]:

```
sns.heatmap(data.corr(),annot=True)
```

Out[9]:

```
<Axes: >
```



In [10]:
```
#sns.pairplot(data)
```

In [11]:
```
x=data.drop('Price_USD',axis=1)
y=data['Price_USD']
```

In [12]:
```
x
```

Out[12]:

	Year	Engine_Size_L	Mileage_KM	Sales_Volume	Sales_Classification_High	Sales_Cla
0	2016	3.5	151748	8300	True	
1	2013	1.6	121671	3428	False	
2	2022	4.5	10991	6994	False	
3	2024	1.7	27255	4047	False	
4	2020	2.1	122131	3080	False	
...	...	...	...	...	...	
49995	2014	4.6	151030	8182	True	
49996	2023	4.2	147396	9816	True	
49997	2010	4.5	174939	8280	True	
49998	2020	3.8	3379	9486	True	
49999	2020	3.3	171003	1764	False	

50000 rows × 35 columns

In [13]:

```
y
```

Out[13]:

```
0         98740
1         79219
2        113265
3         60971
4         49898
          ...  
49995     42932
49996     48714
49997     46126
49998     58566
49999     77492
Name: Price_USD, Length: 50000, dtype: int64
```

In [14]:

```
from sklearn.model_selection import train_test_split
```

In [15]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=23)
```

In [16]:

```
from sklearn.linear_model import LinearRegression
```

In [17]:

```
lm=LinearRegression()
```

In [18]:

```
lm.fit(x_train,y_train)
```

```
Out[18]:  ▾ LinearRegression  i ?
          ▶ Parameters
```

```
In [19]: lm.intercept_
```

```
Out[19]: np.float64(-11224.909669668006)
```

```
In [20]: lm.coef_
```

```
Out[20]: array([ 4.24126815e+01,  1.89967550e+02, -2.84274649e-03,  3.18948016e-02,
                -1.05206128e+02,  1.05206128e+02,  1.16595260e+02, -1.16595260e+02,
                -9.06046530e+01,  3.95268207e+02, -3.05013646e+02,  3.50092234e-01,
                -1.98122556e+02,  4.34024316e+02, -1.33125912e+01, -1.93396990e+02,
                -4.91516709e+02,  4.62324530e+02, -1.33310722e+02,  6.52763473e+02,
                -1.61882107e+02, -2.85927177e+02, -1.54395104e+02,  8.27516374e+01,
                 2.94138016e+02, -1.07911494e+01,  3.95824290e+02, -1.79699298e+02,
                -6.90159342e+02,  5.88820823e+02, -9.13124947e+01, -2.22120044e+02,
                -4.53282952e+02,  1.42682954e+02,  2.25899197e+02])
```

```
In [21]: c=pd.DataFrame(lm.coef_,x.columns,columns=['Price_USD'])
```

```
In [22]: c
```

Out[22]:

	Price_USD
Year	42.412682
Engine_Size_L	189.967550
Mileage_KM	-0.002843
Sales_Volume	0.031895
Sales_Classification_High	-105.206128
Sales_Classification_Low	105.206128
Transmission_Automatic	116.595260
Transmission_Manual	-116.595260
Fuel_Type_Diesel	-90.604653
Fuel_Type_Electric	395.268207
Fuel_Type_Hybrid	-305.013646
Fuel_Type_Petrol	0.350092
Color_Black	-198.122556
Color_Blue	434.024316
Color_Grey	-13.312591
Color_Red	-193.396990
Color_Silver	-491.516709
Color_White	462.324530
Region_Africa	-133.310722
Region_Asia	652.763473
Region_Europe	-161.882107
Region_Middle East	-285.927177
Region_North America	-154.395104
Region_South America	82.751637
Model_3 Series	294.138016
Model_5 Series	-10.791149
Model_7 Series	395.824290
Model_M3	-179.699298
Model_M5	-690.159342
Model_X1	588.820823

	Price_USD
Model_X3	-91.312495
Model_X5	-222.120044
Model_X6	-453.282952
Model_i3	142.682954
Model_i8	225.899197

```
In [23]: pr=lm.predict(x_test)
```

```
In [24]: pr
```

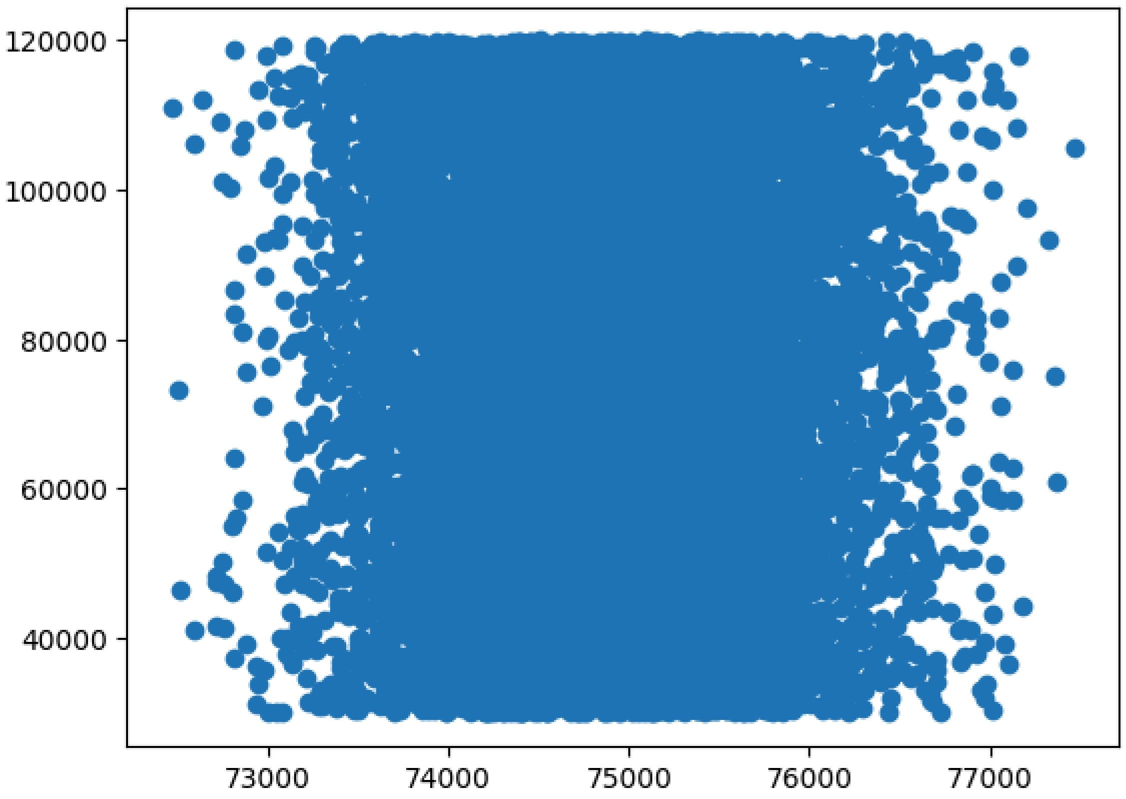
```
Out[24]: array([73500.79639859, 74633.70418724, 74779.94638861, ...,
                73620.55249048, 75150.90617265, 74552.66215489], shape=(15000,))
```

```
In [25]: y_test
```

```
Out[25]: 49466    104395
         11621    108196
         39058     86604
         10033    101480
         22076     59951
                   ...
         35962     71658
         1710      41403
         37523    119878
         14167     92089
         21600     30170
         Name: Price_USD, Length: 15000, dtype: int64
```

```
In [26]: plt.scatter(x=pr,y=y_test)
```

```
Out[26]: <matplotlib.collections.PathCollection at 0x2d72183c500>
```



```
In [27]: from sklearn import metrics
```

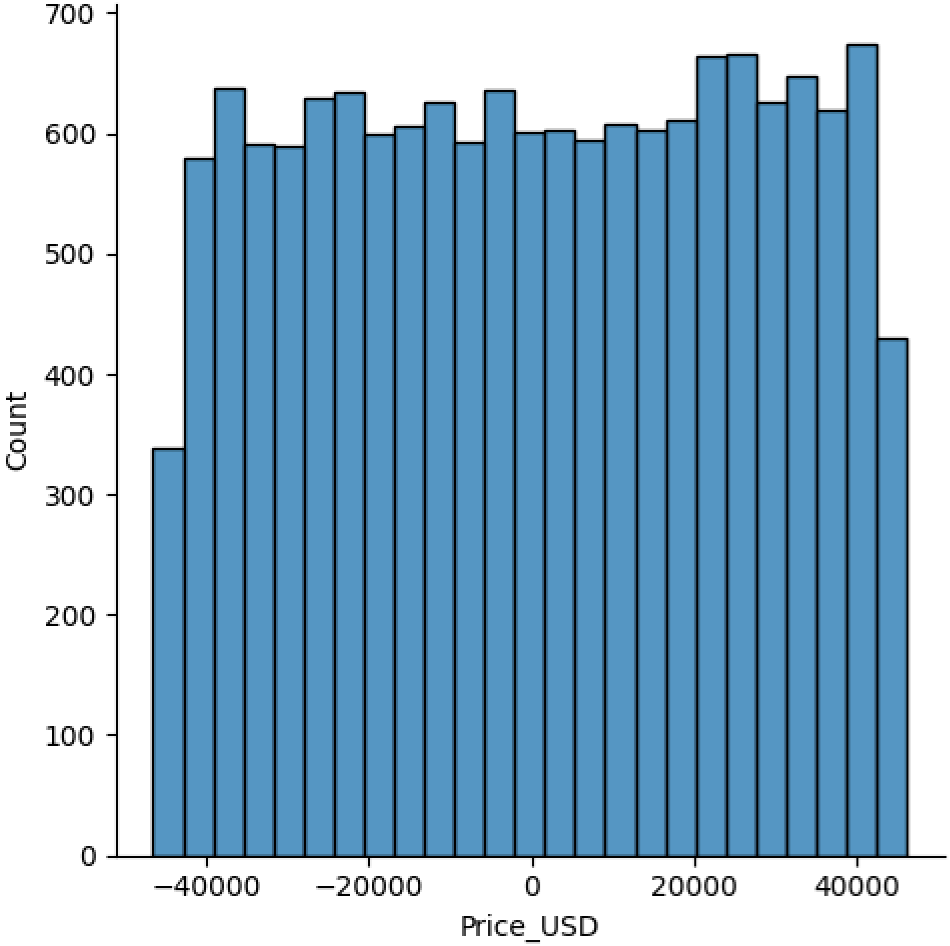
```
In [28]: metrics.mean_absolute_error(y_test,pr)

Out[28]: 22660.005257327874
```

```
In [29]: metrics.mean_squared_error(y_test,pr)

Out[29]: 681578803.0890763
```

```
In [30]: np.sqrt(metrics.mean_squared_error(y_test,pr))

Out[30]: np.float64(26107.064237272567)
```

```
In [31]: metrics.r2_score(y_test,pr)

Out[31]: -0.0012332612175953717
```

```
In [32]: sns.displot(y_test-pr,bins=25)

Out[32]: <seaborn.axisgrid.FacetGrid at 0x2d721479850>
```


700
600
500
400
300
200
100
0
Count
−40000
−20000
0
20000
40000
Price_USD