In [1]:
```python
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns
```

In [2]:
```python
df1 = pd.read_csv('Training_Data_Set.csv')
df1.head(3)
```

Out[2]:

| | ID | Maker | model | Location | Distance | Owner Type | manufacture_year | Age of car | engine_displacement | engine_power | body_type | Vroom Audit Rating |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 11100001 | skoda | octavia | Ahmedabad | NaN | Second | 1964.0 | 55 | 1964 | 147.0 | compact | 8 |
| **1** | 11100002 | fiat | panda | Ahmedabad | 27750.0 | Third | 2012.0 | 7 | 1242 | 51.0 | NaN | 6 |
| **2** | 11100003 | bmw | x1 | Hyderabad | 46000.0 | Third | 2014.0 | 5 | 1995 | 105.0 | NaN | 7 |

In [3]:
```python
df = df1.drop(['ID'] , axis = 1)
df.head(3)
```

Out[3]:

| | Maker | model | Location | Distance | Owner Type | manufacture_year | Age of car | engine_displacement | engine_power | body_type | Vroom Audit Rating | transmissic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | skoda | octavia | Ahmedabad | NaN | Second | 1964.0 | 55 | 1964 | 147.0 | compact | 8 | ma |
| **1** | fiat | panda | Ahmedabad | 27750.0 | Third | 2012.0 | 7 | 1242 | 51.0 | NaN | 6 | ma |
| **2** | bmw | x1 | Hyderabad | 46000.0 | Third | 2014.0 | 5 | 1995 | 105.0 | NaN | 7 | au |

In [4]:
```python
(df.describe())
```

Out[4]:

| | Distance | manufacture_year | Age of car | engine_displacement | engine_power | Vroom Audit Rating | Price |
|---|---|---|---|---|---|---|---|
| **count** | 5.230400e+04 | 53513.000000 | 53515.000000 | 53515.000000 | 52076.000000 | 53515.000000 | 5.351500e+04 |
| **mean** | 9.454626e+04 | 2010.408032 | 8.591890 | 1904.049014 | 100.448345 | 5.998374 | 1.098084e+06 |
| **std** | 2.755617e+05 | 4.650367 | 4.650322 | 1496.564596 | 45.330622 | 1.418336 | 8.441565e+05 |
| **min** | 0.000000e+00 | 1934.000000 | 3.000000 | 14.000000 | 10.000000 | 4.000000 | 3.000000e+00 |
| **25%** | 1.549000e+04 | 2008.000000 | 5.000000 | 1395.000000 | 73.000000 | 5.000000 | 5.051812e+05 |
| **50%** | 6.552000e+04 | 2011.000000 | 8.000000 | 1896.000000 | 91.000000 | 6.000000 | 8.854552e+05 |
| **75%** | 1.356410e+05 | 2014.000000 | 11.000000 | 1995.000000 | 125.000000 | 7.000000 | 1.477829e+06 |
| **max** | 9.899800e+06 | 2016.000000 | 85.000000 | 32000.000000 | 896.000000 | 8.000000 | 2.212078e+07 |

In [5]:
```python
df.shape
```

Out[5]:
```
(53515, 16)
```

In [6]:
```python
for column in df.columns:
    if df[column].dtype =='object':
        print(column.upper(),':',df[column].nunique())
        print(df[column].value_counts().sort_values())
        print('\n')
```

```
MAKER : 8
maserati       38
fiat         1845
hyundai      2240
nissan       5485
bmw          7178
audi         7326
toyota       7840
skoda       21563
Name: Maker, dtype: int64


MODEL : 23
tt            903
juke          955
citigo       1120
q7           1245
roomster     1322
rapid        1409
aygo         1486
avensis      1512
auris        1666
micra        1676
```

```
coupe        1710
q3           1736
panda        1769
yeti         1898
x5           1979
q5           2039
i30          2047
x1           2420
x3           2779
qashqai      2854
yaris        3176
superb       3195
octavia     12619
Name: model, dtype: int64


LOCATION : 11
Ahmedabad     4770
Hyderabad     4804
Delhi         4822
Chennai       4834
Mumbai        4860
Pune          4862
Kolkata       4867
Jaipur        4870
Bangalore     4877
Kochi         4969
Coimbatore    4974
Name: Location, dtype: int64


OWNER TYPE : 4
Fourth & Above    13349
Second            13365
Third             13395
First             13406
Name: Owner Type, dtype: int64


BODY_TYPE : 2
van           9
compact    4127
Name: body_type, dtype: int64


TRANSMISSION : 2
auto    16781
man     36734
Name: transmission, dtype: int64


DOOR_COUNT : 7
1           2
6           8
3         185
2        4348
None     7534
5        7630
4       33808
Name: door_count, dtype: int64


SEAT_COUNT : 10
1           1
8           1
9           2
6          23
3         109
2         725
7         852
4        4467
None     8511
5       38824
Name: seat_count, dtype: int64


FUEL_TYPE : 2
petrol    25956
diesel    27559
Name: fuel_type, dtype: int64
```

In [7]:

```
df = pd.get_dummies(df , columns = ['Maker','model','Location','Owner Type','body_type','transmission','door_co
df
```

Out[7]:

| | Distance | manufacture_year | Age of car | engine_displacement | engine_power | Vroom Audit Rating | Price | Maker_bmw | Maker_fiat | Maker_hyundai | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | NaN | 1964.0 | 55 | 1964 | 147.0 | 8 | 543764.25 | 0 | 0 | 0 | ... |
| **1** | 27750.0 | 2012.0 | 7 | 1242 | 51.0 | 6 | 401819.25 | 0 | 1 | 0 | ... |
| **2** | 46000.0 | 2014.0 | 5 | 1995 | 105.0 | 7 | 2392855.50 | 1 | 0 | 0 | ... |
| **3** | 43949.0 | 2011.0 | 8 | 1618 | 140.0 | 7 | 958606.50 | 0 | 0 | 0 | ... |
| **4** | 59524.0 | 2012.0 | 7 | 2993 | 180.0 | 7 | 3085561.50 | 1 | 0 | 0 | ... |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **53510** | 29334.0 | 2014.0 | 5 | 1598 | 77.0 | 4 | 1342996.50 | 0 | 0 | 0 | ... |
| **53511** | 223631.0 | 2009.0 | 10 | 1900 | 77.0 | 8 | 510732.75 | 0 | 0 | 0 | ... |
| **53512** | 25500.0 | 2015.0 | 4 | 1995 | 105.0 | 4 | 2008123.50 | 1 | 0 | 0 | ... |
| **53513** | 1195500.0 | 2011.0 | 8 | 11950 | 93.0 | 5 | 874352.25 | 0 | 0 | 0 | ... |
| **53514** | 142000.0 | 2008.0 | 11 | 2993 | 173.0 | 4 | 1576610.25 | 1 | 0 | 0 | ... |

53515 rows × 67 columns

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53515 entries, 0 to 53514
Data columns (total 67 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Distance                   52304 non-null  float64
 1   manufacture_year           53513 non-null  float64
 2   Age of car                 53515 non-null  int64  
 3   engine_displacement        53515 non-null  int64  
 4   engine_power               52076 non-null  float64
 5   Vroom Audit Rating         53515 non-null  int64  
 6   Price                      53515 non-null  float64
 7   Maker_bmw                  53515 non-null  uint8  
 8   Maker_fiat                 53515 non-null  uint8  
 9   Maker_hyundai              53515 non-null  uint8  
 10  Maker_maserati             53515 non-null  uint8  
 11  Maker_nissan               53515 non-null  uint8  
 12  Maker_skoda                53515 non-null  uint8  
 13  Maker_toyota               53515 non-null  uint8  
 14  model_avensis              53515 non-null  uint8  
 15  model_aygo                 53515 non-null  uint8  
 16  model_citigo               53515 non-null  uint8  
 17  model_coupe                53515 non-null  uint8  
 18  model_i30                  53515 non-null  uint8  
 19  model_juke                 53515 non-null  uint8  
 20  model_micra                53515 non-null  uint8  
 21  model_octavia              53515 non-null  uint8  
 22  model_panda                53515 non-null  uint8  
 23  model_q3                   53515 non-null  uint8  
 24  model_q5                   53515 non-null  uint8  
 25  model_q7                   53515 non-null  uint8  
 26  model_qashqai              53515 non-null  uint8  
 27  model_rapid                53515 non-null  uint8  
 28  model_roomster             53515 non-null  uint8  
 29  model_superb               53515 non-null  uint8  
 30  model_tt                   53515 non-null  uint8  
 31  model_x1                   53515 non-null  uint8  
 32  model_x3                   53515 non-null  uint8  
 33  model_x5                   53515 non-null  uint8  
 34  model_yaris                53515 non-null  uint8  
 35  model_yeti                 53515 non-null  uint8  
 36  Location_Bangalore         53515 non-null  uint8  
 37  Location_Chennai           53515 non-null  uint8  
 38  Location_Coimbatore        53515 non-null  uint8  
 39  Location_Delhi             53515 non-null  uint8  
 40  Location_Hyderabad         53515 non-null  uint8  
 41  Location_Jaipur            53515 non-null  uint8  
 42  Location_Kochi             53515 non-null  uint8  
 43  Location_Kolkata           53515 non-null  uint8  
 44  Location_Mumbai            53515 non-null  uint8  
 45  Location_Pune              53515 non-null  uint8  
 46  Owner Type_Fourth & Above  53515 non-null  uint8  
 47  Owner Type_Second          53515 non-null  uint8  
 48  Owner Type_Third           53515 non-null  uint8  
 49  body_type_van              53515 non-null  uint8  
 50  transmission_man           53515 non-null  uint8  
 51  door_count_2               53515 non-null  uint8  
 52  door_count_3               53515 non-null  uint8  
 53  door_count_4               53515 non-null  uint8  
 54  door_count_5               53515 non-null  uint8  
 55  door_count_6               53515 non-null  uint8  
 56  door_count_None            53515 non-null  uint8  
 57  seat_count_2               53515 non-null  uint8  
 58  seat_count_3               53515 non-null  uint8  
 59  seat_count_4               53515 non-null  uint8  
 60  seat_count_5               53515 non-null  uint8  
 61  seat_count_6               53515 non-null  uint8  
 62  seat_count_7               53515 non-null  uint8  
 63  seat_count_8               53515 non-null  uint8  
 64  seat_count_9               53515 non-null  uint8  
 65  seat_count_None            53515 non-null  uint8  
 66  fuel_type_petrol           53515 non-null  uint8  
dtypes: float64(4), int64(3), uint8(60)
memory usage: 5.9 MB
```

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
Distance               1211
manufacture_year          2
Age of car                0
engine_displacement       0
engine_power           1439
                       ... 
seat_count_7              0
seat_count_8              0
seat_count_9              0
seat_count_None           0
fuel_type_petrol          0
Length: 67, dtype: int64
```

In [10]:
```
df = df.fillna(df.mean())
```

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
Distance                0
manufacture_year        0
Age of car              0
engine_displacement     0
engine_power            0
                       ..
seat_count_7            0
seat_count_8            0
seat_count_9            0
seat_count_None         0
fuel_type_petrol        0
Length: 67, dtype: int64
```

In [12]:
```
df.duplicated().sum()
```

Out[12]:
```
114
```

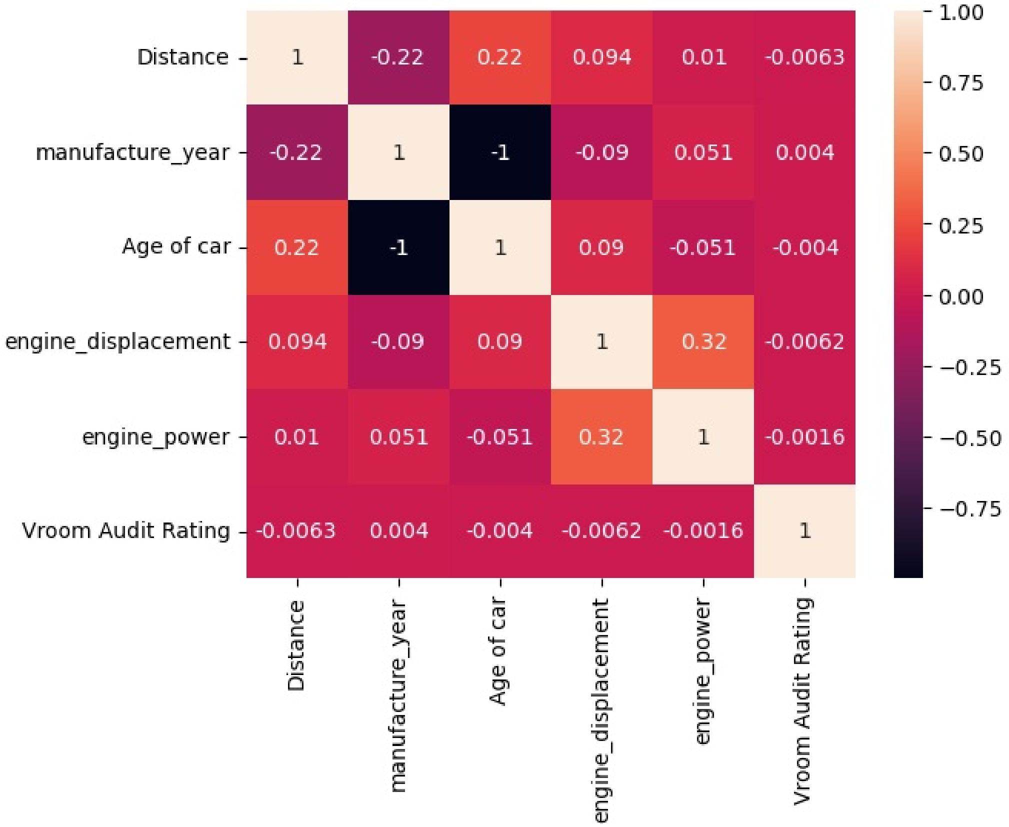
In [13]:
```
df1=df.drop_duplicates()
df1.head(2)
```

Out[13]:

| | Distance | manufacture_year | Age of car | engine_displacement | engine_power | Vroom Audit Rating | Price | Maker_bmw | Maker_fiat | Maker_hyundai | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 94546.262446 | 1964.0 | 55 | 1964 | 147.0 | 8 | 543764.25 | 0 | 0 | 0 | ... |
| **1** | 27750.000000 | 2012.0 | 7 | 1242 | 51.0 | 6 | 401819.25 | 0 | 1 | 0 | ... |

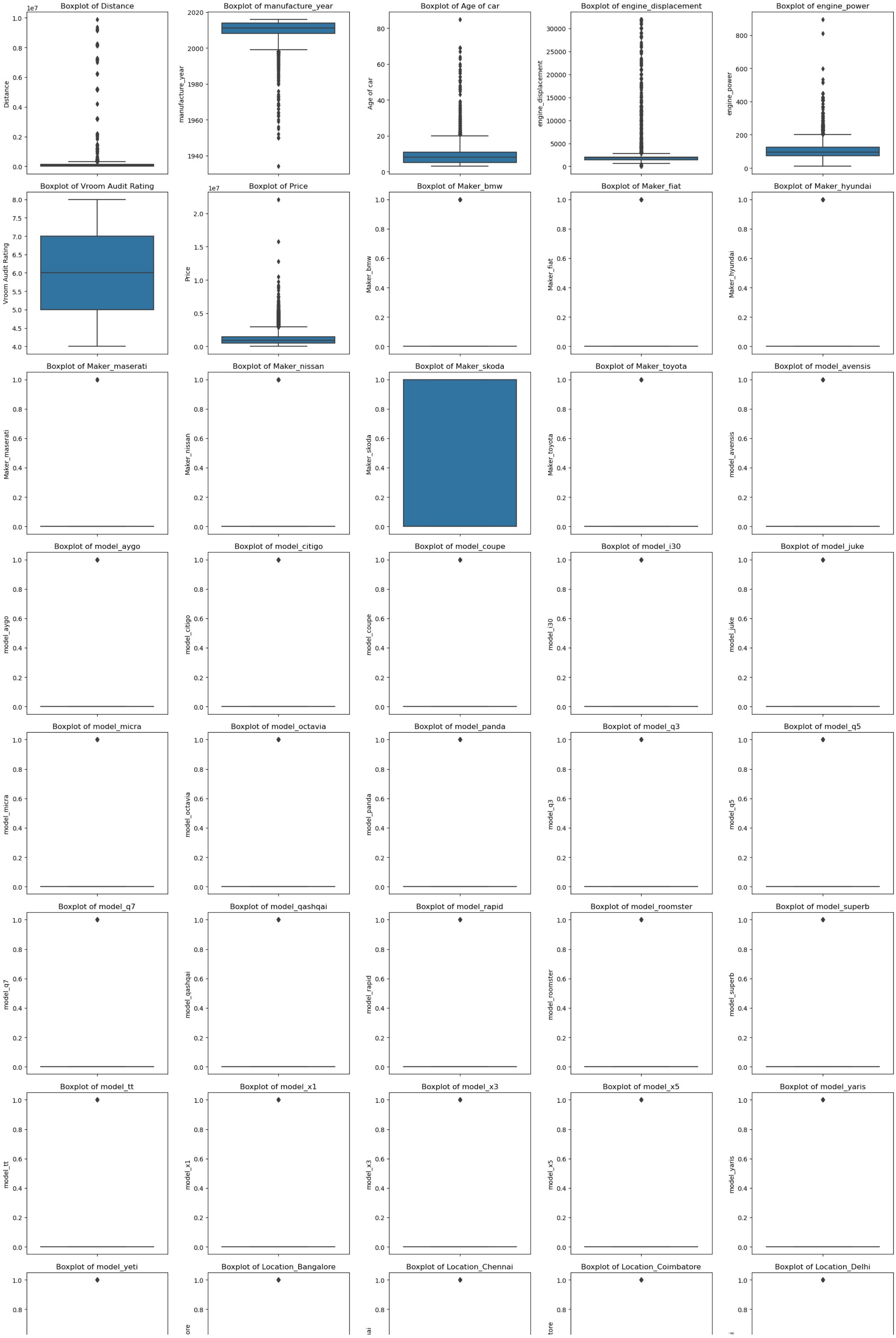
2 rows × 67 columns

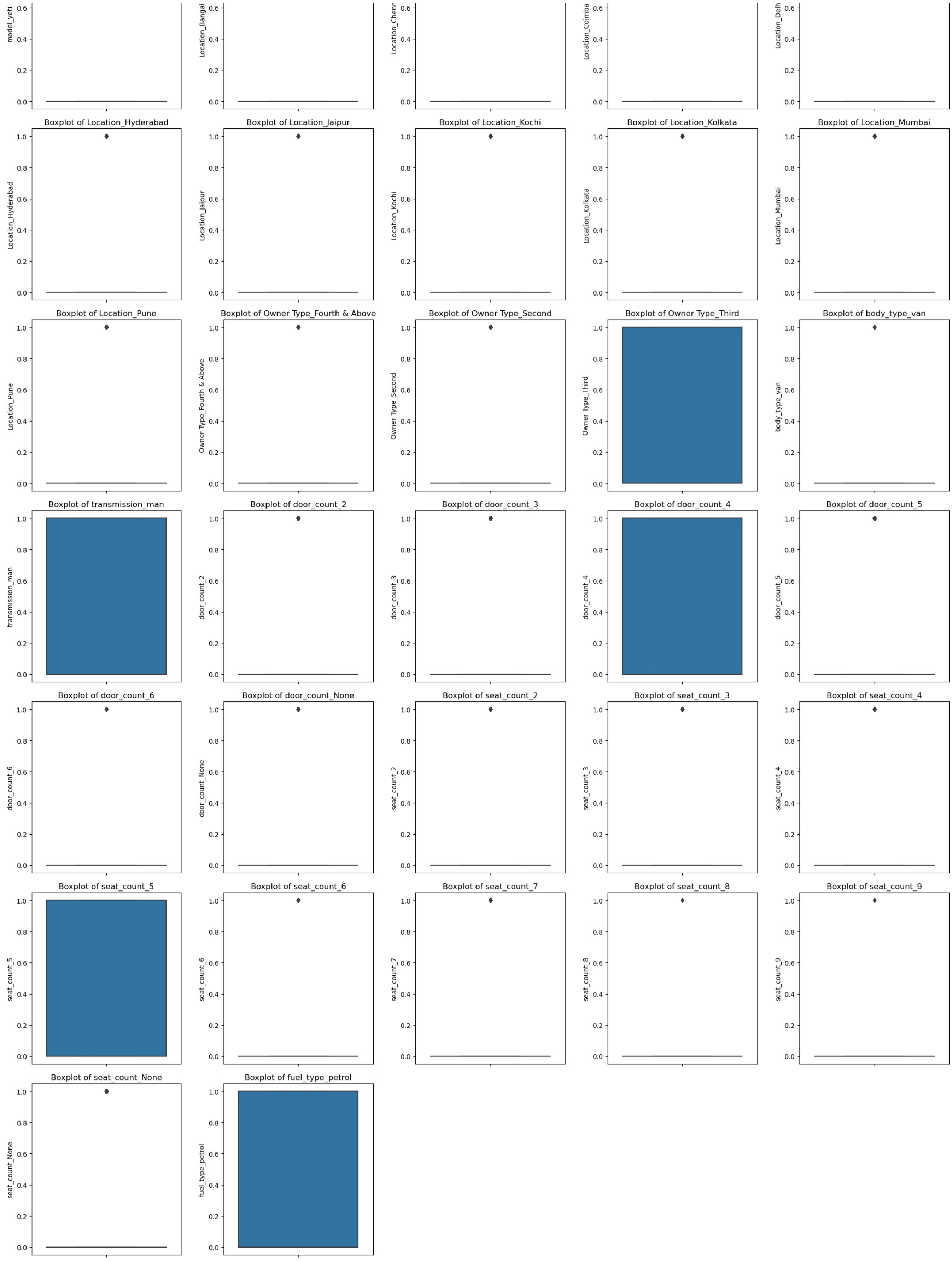
In [14]:
```
sns.heatmap(df1.iloc[:, 0:6].corr(),annot=True)
plt.show()
```




In [15]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming df1 is your DataFrame
feature_list = df1.columns

# Calculate the number of rows and columns needed
num_plots = len(feature_list)
rows = (num_plots // 5) + (1 if num_plots % 5 else 0)
cols = 5

plt.figure(figsize=(20, rows * 4))
```

```
for i in range(num_plots):
    plt.subplot(rows, cols, i + 1)
    sns.boxplot(y=df1[feature_list[i]], data=df1)
    plt.title('Boxplot of {}'.format(feature_list[i]))

plt.tight_layout()
plt.show()
```






In [16]:

```
def remove_outlier(col):
    Q1,Q3=col.quantile([0.25,0.75])
    IQR=Q3-Q1
    lower_range= Q1-(1.5*IQR)
    upper_range= Q3+(1.5*IQR)
    return lower_range,upper_range
```

In [17]:

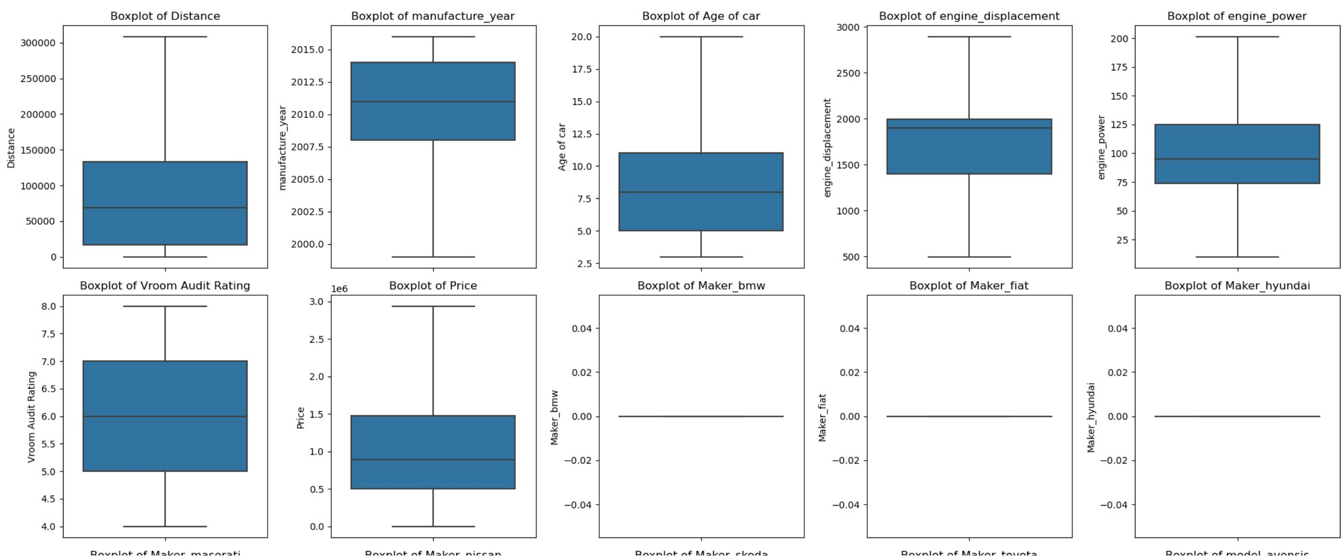
```
for i in feature_list:
    LL, UL = remove_outlier(df1[i])
    df1[i] = np.where(df1[i] > UL,UL, df1[i])
    df1[i] = np.where(df1[i] < LL,LL,df1[i])
```

```
/var/folders/y7/v00504rn5c5dgvzgtlt4bkcr0000gn/T/ipykernel_3490/610677943.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

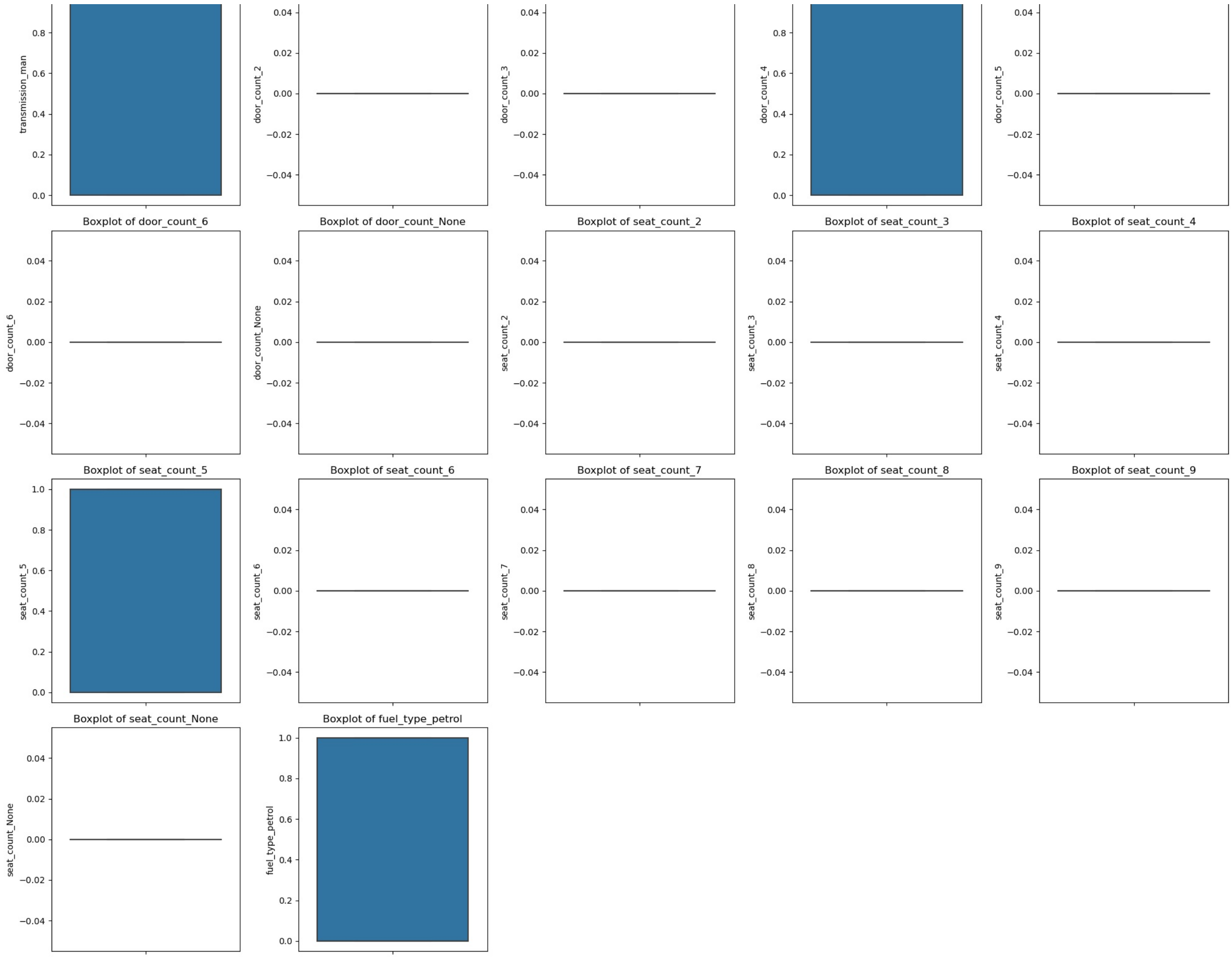
```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df1[i] = np.where(df1[i] < LL,LL,df1[i])
/var/folders/y7/v00504rn5c5dgvzgtlt4bkcr0000gn/T/ipykernel_3490/610677943.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df1[i] = np.where(df1[i] > UL,UL, df1[i])
/var/folders/y7/v00504rn5c5dgvzgtlt4bkcr0000gn/T/ipykernel_3490/610677943.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df1[i] = np.where(df1[i] < LL,LL,df1[i])
/var/folders/y7/v00504rn5c5dgvzgtlt4bkcr0000gn/T/ipykernel_3490/610677943.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df1[i] = np.where(df1[i] > UL,UL, df1[i])
/var/folders/y7/v00504rn5c5dgvzgtlt4bkcr0000gn/T/ipykernel_3490/610677943.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df1[i] = np.where(df1[i] < LL,LL,df1[i])
/var/folders/y7/v00504rn5c5dgvzgtlt4bkcr0000gn/T/ipykernel_3490/610677943.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df1[i] = np.where(df1[i] > UL,UL, df1[i])
/var/folders/y7/v00504rn5c5dgvzgtlt4bkcr0000gn/T/ipykernel_3490/610677943.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df1[i] = np.where(df1[i] < LL,LL,df1[i])
```

In [18]:

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming df1 is your DataFrame
feature_list = df1.columns

# Calculate the number of rows and columns needed
num_plots = len(feature_list)
rows = (num_plots // 5) + (1 if num_plots % 5 else 0)
cols = 5

plt.figure(figsize=(20, rows * 4))

for i in range(num_plots):
    plt.subplot(rows, cols, i + 1)
    sns.boxplot(y=df1[feature_list[i]], data=df1)
    plt.title('Boxplot of {}'.format(feature_list[i]))

plt.tight_layout()
plt.show()
```




Maker_maserati
Maker_nissan
Maker_skoda
Maker_toyota
model_avensis
Boxplot of model_aygo
model_aygo
Boxplot of model_citigo
model_citigo
Boxplot of model_coupe
model_coupe
Boxplot of model_i30
model_i30
Boxplot of model_juke
model_juke
Boxplot of model_micra
model_micra
Boxplot of model_octavia
model_octavia
Boxplot of model_panda
model_panda
Boxplot of model_q3
model_q3
Boxplot of model_q5
model_q5
Boxplot of model_q7
model_q7
Boxplot of model_qashqai
model_qashqai
Boxplot of model_rapid
model_rapid
Boxplot of model_roomster
model_roomster
Boxplot of model_superb
model_superb
Boxplot of model_tt
model_tt
Boxplot of model_x1
model_x1
Boxplot of model_x3
model_x3
Boxplot of model_x5
model_x5
Boxplot of model_yaris
model_yaris
Boxplot of model_yeti
model_yeti
Boxplot of Location_Bangalore
Location_Bangalore
Boxplot of Location_Chennai
Location_Chennai
Boxplot of Location_Coimbatore
Location_Coimbatore
Boxplot of Location_Delhi
Location_Delhi
Boxplot of Location_Hyderabad
Location_Hyderabad
Boxplot of Location_Jaipur
Location_Jaipur
Boxplot of Location_Kochi
Location_Kochi
Boxplot of Location_Kolkata
Location_Kolkata
Boxplot of Location_Mumbai
Location_Mumbai
Boxplot of Location_Pune
Location_Pune
Boxplot of Owner Type_Fourth & Above
Owner Type_Fourth & Above
Boxplot of Owner Type_Second
Owner Type_Second
Boxplot of Owner Type_Third
Owner Type_Third
Boxplot of body_type_van
body_type_van
Boxplot of transmission_man
Boxplot of door_count_2
Boxplot of door_count_3
Boxplot of door_count_4
Boxplot of door_count_5




## Pair plot

In [19]:
```
#sns.pairplot(df1 , diag_kind = 'kde')
```

## Train Test split

In [20]:
```
# Copy all the predictor variable into X dataframe
X = df1.drop('Price', axis=1)

# Copy target into the y dataframe.
y = df1[['Price']]
```

In [21]:
```
# Split X and y into training and test set in 75:25 ratio
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.25, random_state=1)
```

In [22]:
```
import statsmodels.api as sm
```

In [23]:
```
X_train=sm.add_constant(X_train)
X_test=sm.add_constant(X_test)
```

In [24]:
```
model = sm.OLS(y_train,X_train).fit()
model.summary()
```

Out[24]:

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Price | **R-squared:** | 0.790 |
| **Model:** | OLS | **Adj. R-squared:** | 0.790 |
| **Method:** | Least Squares | **F-statistic:** | 1.255e+04 |
| **Date:** | Wed, 05 Jun 2024 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 08:04:04 | **Log-Likelihood:** | -5.6646e+05 |
| **No. Observations:** | 40050 | **AIC:** | 1.133e+06 |
| **Df Residuals:** | 40037 | **BIC:** | 1.133e+06 |
| **Df Model:** | 12 | | |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **const** | -5.881e+07 | 1.47e+08 | -0.400 | 0.690 | -3.47e+08 | 2.3e+08 |
| **Distance** | -3.2001 | 0.036 | -89.621 | 0.000 | -3.270 | -3.130 |
| **manufacture_year** | 2.962e+04 | 7.29e+04 | 0.406 | 0.685 | -1.13e+05 | 1.73e+05 |
| **Age of car** | -2.463e+04 | 7.29e+04 | -0.338 | 0.735 | -1.68e+05 | 1.18e+05 |
| **engine_displacement** | 254.1679 | 6.738 | 37.723 | 0.000 | 240.962 | 267.374 |
| **engine_power** | 6632.8889 | 78.444 | 84.556 | 0.000 | 6479.138 | 6786.640 |
| **Vroom Audit Rating** | 963.0607 | 1182.810 | 0.814 | 0.416 | -1355.275 | 3281.396 |
| **Maker_bmw** | 3.57e-08 | 2.9e-09 | 12.308 | 0.000 | 3e-08 | 4.14e-08 |
| **Maker_fiat** | 3.856e-09 | 3.1e-10 | 12.438 | 0.000 | 3.25e-09 | 4.46e-09 |
| **Maker_hyundai** | -1.203e-08 | 9.75e-10 | -12.343 | 0.000 | -1.39e-08 | -1.01e-08 |
| **Maker_maserati** | -2.032e-10 | 1.77e-11 | -11.452 | 0.000 | -2.38e-10 | -1.68e-10 |
| **Maker_nissan** | -2.264e-11 | 2.05e-12 | -11.021 | 0.000 | -2.67e-11 | -1.86e-11 |
| **Maker_skoda** | -1.416e+05 | 3740.872 | -37.854 | 0.000 | -1.49e+05 | -1.34e+05 |
| **Maker_toyota** | 0 | 0 | nan | nan | 0 | 0 |
| **model_avensis** | 0 | 0 | nan | nan | 0 | 0 |
| **model_aygo** | 0 | 0 | nan | nan | 0 | 0 |
| **model_citigo** | 0 | 0 | nan | nan | 0 | 0 |
| **model_coupe** | 0 | 0 | nan | nan | 0 | 0 |
| **model_i30** | 0 | 0 | nan | nan | 0 | 0 |
| **model_juke** | 0 | 0 | nan | nan | 0 | 0 |
| **model_micra** | 0 | 0 | nan | nan | 0 | 0 |
| **model_octavia** | 0 | 0 | nan | nan | 0 | 0 |
| **model_panda** | 0 | 0 | nan | nan | 0 | 0 |
| **model_q3** | 0 | 0 | nan | nan | 0 | 0 |
| **model_q5** | 0 | 0 | nan | nan | 0 | 0 |
| **model_q7** | 0 | 0 | nan | nan | 0 | 0 |
| **model_qashqai** | 0 | 0 | nan | nan | 0 | 0 |
| **model_rapid** | 0 | 0 | nan | nan | 0 | 0 |
| **model_roomster** | 0 | 0 | nan | nan | 0 | 0 |
| **model_superb** | 0 | 0 | nan | nan | 0 | 0 |
| **model_tt** | 0 | 0 | nan | nan | 0 | 0 |
| **model_x1** | 0 | 0 | nan | nan | 0 | 0 |
| **model_x3** | 0 | 0 | nan | nan | 0 | 0 |
| **model_x5** | 0 | 0 | nan | nan | 0 | 0 |
| **model_yaris** | 0 | 0 | nan | nan | 0 | 0 |
| **model_yeti** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Bangalore** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Chennai** | 0 | 0 | nan | nan | 0 | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Location_Coimbatore** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Delhi** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Hyderabad** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Jaipur** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Kochi** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Kolkata** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Mumbai** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Pune** | 0 | 0 | nan | nan | 0 | 0 |
| **Owner Type_Fourth & Above** | 0 | 0 | nan | nan | 0 | 0 |
| **Owner Type_Second** | 0 | 0 | nan | nan | 0 | 0 |
| **Owner Type_Third** | -243.6636 | 3864.482 | -0.063 | 0.950 | -7818.138 | 7330.811 |
| **body_type_van** | 0 | 0 | nan | nan | 0 | 0 |
| **transmission_man** | -2.246e+05 | 4440.331 | -50.585 | 0.000 | -2.33e+05 | -2.16e+05 |
| **door_count_2** | 0 | 0 | nan | nan | 0 | 0 |
| **door_count_3** | 0 | 0 | nan | nan | 0 | 0 |
| **door_count_4** | -8.997e+04 | 4011.925 | -22.427 | 0.000 | -9.78e+04 | -8.21e+04 |
| **door_count_5** | 0 | 0 | nan | nan | 0 | 0 |
| **door_count_6** | 0 | 0 | nan | nan | 0 | 0 |
| **door_count_None** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_2** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_3** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_4** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_5** | 6.139e+04 | 4130.664 | 14.863 | 0.000 | 5.33e+04 | 6.95e+04 |
| **seat_count_6** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_7** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_8** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_9** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_None** | 0 | 0 | nan | nan | 0 | 0 |
| **fuel_type_petrol** | -1.971e+05 | 4201.344 | -46.921 | 0.000 | -2.05e+05 | -1.89e+05 |

| | | | |
|---|---|---|---|
| **Omnibus:** | 6020.661 | **Durbin-Watson:** | 2.009 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 16656.071 |
| **Skew:** | 0.818 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 5.702 | **Cond. No.** | 1.36e+16 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.76e-18. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

In [25]:
```
# Lets check the VIP(variation inflation factor)of the predictors
from statsmodels.stats.outliers_influence import variance_inflation_factor

vif_series1 = pd.Series(
```

```
    [variance_inflation_factor(X_train.values, i) for i in range(X_train.shape[1])],
     index=X_train.columns,
)
vif_file = "VIF values: \n\n{}\n".format(vif_series1)
print(vif_file)
```

```
/Users/sachinssharma/anaconda3/lib/python3.11/site-packages/statsmodels/regression/linear_model.py:1781: Runtim
eWarning: invalid value encountered in scalar divide
  return 1 - self.ssr/self.centered_tss
VIF values:

const                    7.682852e+09
Distance                 2.604749e+00
manufacture_year         3.463081e+04
Age of car               3.463332e+04
engine_displacement      4.206017e+00
                             ...
seat_count_7                      NaN
seat_count_8                      NaN
seat_count_9                      NaN
seat_count_None                   NaN
fuel_type_petrol         1.563505e+00
Length: 67, dtype: float64
```

In [26]:
```
my_df = vif_series1.to_frame(name='col_names')
my_df.to_excel('vif.xlsx')
```

In [27]:
```
my_df.head(10)
```

Out[27]:

| | col_names |
|---|---|
| **const** | 7.682852e+09 |
| **Distance** | 2.604749e+00 |
| **manufacture_year** | 3.463081e+04 |
| **Age of car** | 3.463332e+04 |
| **engine_displacement** | 4.206017e+00 |
| **engine_power** | 3.461206e+00 |
| **Vroom Audit Rating** | 1.000134e+00 |
| **Maker_bmw** | NaN |
| **Maker_fiat** | NaN |
| **Maker_hyundai** | NaN |

In [28]:
```
#1) Removing predictor 'engine_displacement' as VIF>2
X_train2=X_train.drop(["engine_displacement"], axis=1)
olsmod_1=sm.OLS(y_train, X_train2)
olsres_1=olsmod_1.fit()
print(
     "R-squared:",
    np.round(olsres_1.rsquared, 3),
    "\nAdjusted R-squared:",
    np.round(olsres_1.rsquared_adj, 3),
)
```

```
R-squared: 0.783
Adjusted R-squared: 0.782
```

In [29]:
```
0.790-0.783
```

Out[29]:
```
0.007000000000000006
```

In [30]:
```
#1) Removing predictor 'Age of car' as VIF>2
X_train2=X_train.drop(["Age of car"], axis=1)
olsmod_1=sm.OLS(y_train, X_train2)
olsres_1=olsmod_1.fit()
print(
     "R-squared:",
    np.round(olsres_1.rsquared, 3),
    "\nAdjusted R-squared:",
    np.round(olsres_1.rsquared_adj, 3),
)
```

```
R-squared: 0.79
Adjusted R-squared: 0.79
```

In [31]:
```
0.790-0.79
```

Out[31]:
```
0.0
```

In [32]:
```
#1) Removing predictor 'manufacture_year' as VIF>2
X_train2=X_train.drop(["manufacture_year"], axis=1)
```

```
olsmod_1=sm.OLS(y_train, X_train2)
olsres_1=olsmod_1.fit()
print(
     "R-squared:",
    np.round(olsres_1.rsquared, 3),
    "\nAdjusted R-squared:",
    np.round(olsres_1.rsquared_adj, 3),
)
```

```
R-squared: 0.79
Adjusted R-squared: 0.79
```

In [33]:
```
#1) Removing predictor 'Distance ' as VIF>2
X_train2=X_train.drop(["Distance"], axis=1)
olsmod_1=sm.OLS(y_train, X_train2)
olsres_1=olsmod_1.fit()
print(
     "R-squared:",
    np.round(olsres_1.rsquared, 3),
    "\nAdjusted R-squared:",
    np.round(olsres_1.rsquared_adj, 3),
)
```

```
R-squared: 0.748
Adjusted R-squared: 0.748
```

In [34]:
```
0.790-0.748 #Don't remove
```

Out[34]:
```
0.04200000000000004
```

In [35]:
```
#1) Removing predictor 'engine_power' as VIF>2
X_train2=X_train.drop(["engine_power"], axis=1)
olsmod_1=sm.OLS(y_train, X_train2)
olsres_1=olsmod_1.fit()
print(
     "R-squared:",
    np.round(olsres_1.rsquared, 3),
    "\nAdjusted R-squared:",
    np.round(olsres_1.rsquared_adj, 3),
)
```

```
R-squared: 0.753
Adjusted R-squared: 0.752
```

In [36]:
```
0.790-0.753 # don't remove
```

Out[36]:
```
0.03700000000000003
```

In [37]:
```
#As we are observing the multicollinearity and no such diffrence in removing manufacture_year , Age of car , en
# so remove and run regression model
```

## Dropping Multicolinear columns

In [38]:
```
X_train = X_train.drop(["manufacture_year"], axis = 1)
```

In [39]:
```
olsmod_5 = sm.OLS(y_train,X_train)
olsres_5 = olsmod_5.fit()
olsres_5.summary()
```

Out[39]:

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Price | **R-squared:** | 0.790 |
| **Model:** | OLS | **Adj. R-squared:** | 0.790 |
| **Method:** | Least Squares | **F-statistic:** | 1.369e+04 |
| **Date:** | Wed, 05 Jun 2024 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 08:04:09 | **Log-Likelihood:** | -5.6646e+05 |
| **No. Observations:** | 40050 | **AIC:** | 1.133e+06 |
| **Df Residuals:** | 40038 | **BIC:** | 1.133e+06 |
| **Df Model:** | 11 | | |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **const** | 1e+06 | 1.37e+04 | 72.895 | 0.000 | 9.73e+05 | 1.03e+06 |
| **Distance** | -3.2001 | 0.036 | -89.621 | 0.000 | -3.270 | -3.130 |
| **Age of car** | -5.425e+04 | 621.554 | -87.289 | 0.000 | -5.55e+04 | -5.3e+04 |
| **engine_displacement** | 254.1727 | 6.738 | 37.724 | 0.000 | 240.967 | 267.379 |
| **engine_power** | 6632.8456 | 78.443 | 84.557 | 0.000 | 6479.096 | 6786.595 |
| **Vroom Audit Rating** | 964.6015 | 1182.792 | 0.816 | 0.415 | -1353.698 | 3282.901 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Maker_bmw** | -2.141e-09 | 1.39e-09 | -1.546 | 0.122 | -4.86e-09 | 5.74e-10 |
| **Maker_fiat** | 1.325e-09 | 8.61e-10 | 1.539 | 0.124 | -3.63e-10 | 3.01e-09 |
| **Maker_hyundai** | -9.028e-10 | 6.21e-10 | -1.454 | 0.146 | -2.12e-09 | 3.14e-10 |
| **Maker_maserati** | -3.339e-13 | 2.45e-13 | -1.363 | 0.173 | -8.14e-13 | 1.46e-13 |
| **Maker_nissan** | -4.951e-13 | 5.14e-12 | -0.096 | 0.923 | -1.06e-11 | 9.59e-12 |
| **Maker_skoda** | -1.416e+05 | 3740.810 | -37.853 | 0.000 | -1.49e+05 | -1.34e+05 |
| **Maker_toyota** | 0 | 0 | nan | nan | 0 | 0 |
| **model_avensis** | 0 | 0 | nan | nan | 0 | 0 |
| **model_aygo** | 0 | 0 | nan | nan | 0 | 0 |
| **model_citigo** | 0 | 0 | nan | nan | 0 | 0 |
| **model_coupe** | 0 | 0 | nan | nan | 0 | 0 |
| **model_i30** | 0 | 0 | nan | nan | 0 | 0 |
| **model_juke** | 0 | 0 | nan | nan | 0 | 0 |
| **model_micra** | 0 | 0 | nan | nan | 0 | 0 |
| **model_octavia** | 0 | 0 | nan | nan | 0 | 0 |
| **model_panda** | 0 | 0 | nan | nan | 0 | 0 |
| **model_q3** | 0 | 0 | nan | nan | 0 | 0 |
| **model_q5** | 0 | 0 | nan | nan | 0 | 0 |
| **model_q7** | 0 | 0 | nan | nan | 0 | 0 |
| **model_qashqai** | 0 | 0 | nan | nan | 0 | 0 |
| **model_rapid** | 0 | 0 | nan | nan | 0 | 0 |
| **model_roomster** | 0 | 0 | nan | nan | 0 | 0 |
| **model_superb** | 0 | 0 | nan | nan | 0 | 0 |
| **model_tt** | 0 | 0 | nan | nan | 0 | 0 |
| **model_x1** | 0 | 0 | nan | nan | 0 | 0 |
| **model_x3** | 0 | 0 | nan | nan | 0 | 0 |
| **model_x5** | 0 | 0 | nan | nan | 0 | 0 |
| **model_yaris** | 0 | 0 | nan | nan | 0 | 0 |
| **model_yeti** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Bangalore** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Chennai** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Coimbatore** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Delhi** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Hyderabad** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Jaipur** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Kochi** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Kolkata** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Mumbai** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Pune** | 0 | 0 | nan | nan | 0 | 0 |
| **Owner Type_Fourth & Above** | 0 | 0 | nan | nan | 0 | 0 |
| **Owner Type_Second** | 0 | 0 | nan | nan | 0 | 0 |
| **Owner Type_Third** | -239.5045 | 3864.428 | -0.062 | 0.951 | -7813.873 | 7334.864 |
| **body_type_van** | 0 | 0 | nan | nan | 0 | 0 |
| **transmission_man** | -2.246e+05 | 4440.243 | -50.587 | 0.000 | -2.33e+05 | -2.16e+05 |
| **door_count_2** | 0 | 0 | nan | nan | 0 | 0 |
| **door_count_3** | 0 | 0 | nan | nan | 0 | 0 |
| **door_count_4** | -8.998e+04 | 4011.878 | -22.428 | 0.000 | -9.78e+04 | -8.21e+04 |
| **door_count_5** | 0 | 0 | nan | nan | 0 | 0 |
| **door_count_6** | 0 | 0 | nan | nan | 0 | 0 |
| **door_count_None** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_2** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_3** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_4** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_5** | 6.139e+04 | 4130.620 | 14.863 | 0.000 | 5.33e+04 | 6.95e+04 |
| **seat_count_6** | 0 | 0 | nan | nan | 0 | 0 |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **seat_count_7** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_8** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_9** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_None** | 0 | 0 | nan | nan | 0 | 0 |
| **fuel_type_petrol** | -1.971e+05 | 4201.233 | -46.920 | 0.000 | -2.05e+05 | -1.89e+05 |

| | | | |
|---|---|---|---|
| **Omnibus:** | 6020.713 | **Durbin-Watson:** | 2.009 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 16655.937 |
| **Skew:** | 0.818 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 5.702 | **Cond. No.** | 1.36e+16 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.76e-18. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

In [40]:
```
#by removing engine_displacement and age of car we are getting 0.783 and 0.745 respectively so remove only manu
```

In [41]:
```
#So Remove only manufacturer year
```

In [42]:
```
#columns = Vroom Audit Rating,Maker_bmw,Maker_fiat,Maker_hyundai,Maker_maserati,Maker_nissan,Owner Type_Third a
```

In [43]:
```
x_train5 = X_train.drop(['Vroom Audit Rating','Maker_bmw','Maker_fiat','Maker_hyundai','Maker_maserati','Maker_
olsmod_6 = sm.OLS(y_train,x_train5)
olsres_6 = olsmod_6.fit()
olsres_6.summary()
```

Out[43]:

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Price | **R-squared:** | 0.790 |
| **Model:** | OLS | **Adj. R-squared:** | 0.790 |
| **Method:** | Least Squares | **F-statistic:** | 1.674e+04 |
| **Date:** | Wed, 05 Jun 2024 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 08:04:10 | **Log-Likelihood:** | -5.6646e+05 |
| **No. Observations:** | 40050 | **AIC:** | 1.133e+06 |
| **Df Residuals:** | 40040 | **BIC:** | 1.133e+06 |
| **Df Model:** | 9 | | |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **const** | 1.006e+06 | 1.17e+04 | 85.729 | 0.000 | 9.83e+05 | 1.03e+06 |
| **Distance** | -3.2001 | 0.036 | -89.626 | 0.000 | -3.270 | -3.130 |
| **Age of car** | -5.425e+04 | 621.532 | -87.292 | 0.000 | -5.55e+04 | -5.3e+04 |
| **engine_displacement** | 254.1769 | 6.737 | 37.727 | 0.000 | 240.972 | 267.382 |
| **engine_power** | 6632.8515 | 78.441 | 84.559 | 0.000 | 6479.106 | 6786.598 |
| **Maker_skoda** | -1.416e+05 | 3740.706 | -37.854 | 0.000 | -1.49e+05 | -1.34e+05 |
| **Maker_toyota** | -6.021e-10 | 4.01e-11 | -15.023 | 0.000 | -6.81e-10 | -5.24e-10 |
| **model_avensis** | -4.284e-10 | 3.03e-11 | -14.123 | 0.000 | -4.88e-10 | -3.69e-10 |
| **model_aygo** | 6.391e-13 | 2.39e-13 | 2.677 | 0.007 | 1.71e-13 | 1.11e-12 |
| **model_citigo** | 0 | 0 | nan | nan | 0 | 0 |
| **model_coupe** | 0 | 0 | nan | nan | 0 | 0 |
| **model_i30** | 0 | 0 | nan | nan | 0 | 0 |
| **model_juke** | 0 | 0 | nan | nan | 0 | 0 |
| **model_micra** | 0 | 0 | nan | nan | 0 | 0 |
| **model_octavia** | 0 | 0 | nan | nan | 0 | 0 |
| **model_panda** | 0 | 0 | nan | nan | 0 | 0 |
| **model_q3** | 0 | 0 | nan | nan | 0 | 0 |
| **model_q5** | 0 | 0 | nan | nan | 0 | 0 |
| **model_q7** | 0 | 0 | nan | nan | 0 | 0 |
| **model_qashqai** | 0 | 0 | nan | nan | 0 | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **model_rapid** | 0 | 0 | nan | nan | 0 | 0 |
| **model_roomster** | 0 | 0 | nan | nan | 0 | 0 |
| **model_superb** | 0 | 0 | nan | nan | 0 | 0 |
| **model_tt** | 0 | 0 | nan | nan | 0 | 0 |
| **model_x1** | 0 | 0 | nan | nan | 0 | 0 |
| **model_x3** | 0 | 0 | nan | nan | 0 | 0 |
| **model_x5** | 0 | 0 | nan | nan | 0 | 0 |
| **model_yaris** | 0 | 0 | nan | nan | 0 | 0 |
| **model_yeti** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Bangalore** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Chennai** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Coimbatore** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Delhi** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Hyderabad** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Jaipur** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Kochi** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Kolkata** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Mumbai** | 0 | 0 | nan | nan | 0 | 0 |
| **Location_Pune** | 0 | 0 | nan | nan | 0 | 0 |
| **Owner Type_Fourth & Above** | 0 | 0 | nan | nan | 0 | 0 |
| **Owner Type_Second** | 0 | 0 | nan | nan | 0 | 0 |
| **body_type_van** | 0 | 0 | nan | nan | 0 | 0 |
| **transmission_man** | -2.246e+05 | 4440.076 | -50.585 | 0.000 | -2.33e+05 | -2.16e+05 |
| **door_count_2** | 0 | 0 | nan | nan | 0 | 0 |
| **door_count_3** | 0 | 0 | nan | nan | 0 | 0 |
| **door_count_4** | -8.999e+04 | 4011.542 | -22.434 | 0.000 | -9.79e+04 | -8.21e+04 |
| **door_count_5** | 0 | 0 | nan | nan | 0 | 0 |
| **door_count_6** | 0 | 0 | nan | nan | 0 | 0 |
| **door_count_None** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_2** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_3** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_4** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_5** | 6.14e+04 | 4130.527 | 14.866 | 0.000 | 5.33e+04 | 6.95e+04 |
| **seat_count_6** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_7** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_8** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_9** | 0 | 0 | nan | nan | 0 | 0 |
| **seat_count_None** | 0 | 0 | nan | nan | 0 | 0 |
| **fuel_type_petrol** | -1.971e+05 | 4201.104 | -46.925 | 0.000 | -2.05e+05 | -1.89e+05 |

| | | | |
|---|---|---|---|
| **Omnibus:** | 6022.274 | **Durbin-Watson:** | 2.009 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 16663.303 |
| **Skew:** | 0.819 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 5.703 | **Cond. No.** | 1.36e+16 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.76e-18. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

After dropping the features causes multicolinearity and the statistical insignificant ones, Our model performance hasn't dropped sharply. this shows that these varibales did not have much predictive power.

For linear Regression we need to check if the following assumptions hold

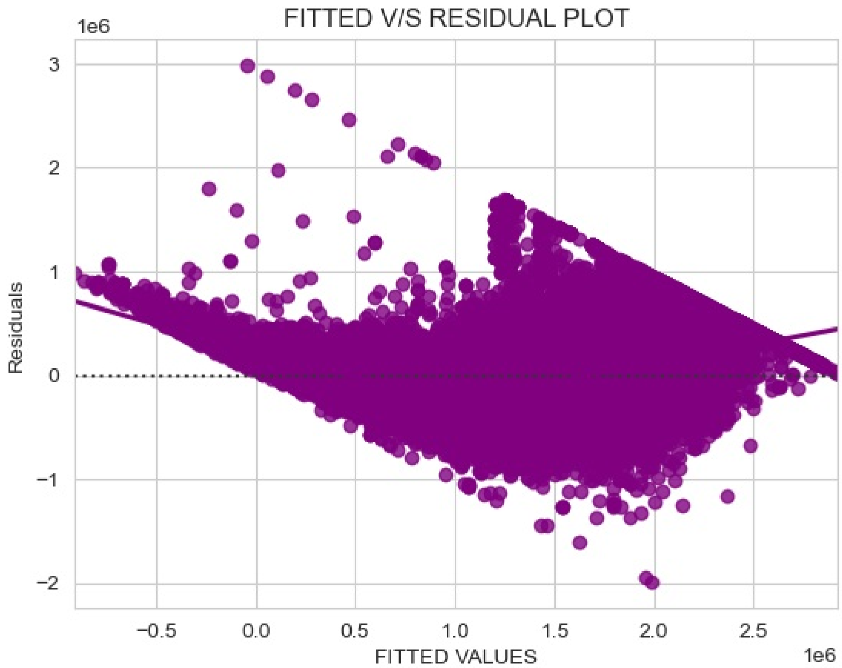
1. Linearity
2. independence
3. Homoscedasticity
4. Normality of error terms
5. No strong multicolinearity

In [44]:
```
df_pred = pd.DataFrame()

df_pred["Actual values"] = y_train.values.flatten() #actual values
df_pred["Fitted values"] = olsres_6.fittedvalues.values # predicted values
df_pred["Residuals"] = olsres_6.resid.values # residuals (actual-fitted)

df_pred.head()
```

Out[44]:

| | Actual values | Fitted values | Residuals |
|---|---|---|---|
| 0 | 817741.500 | 7.875369e+05 | 30204.635729 |
| 1 | 2938661.625 | 2.520180e+06 | 418481.249276 |
| 2 | 2893229.250 | 2.136881e+06 | 756347.908289 |
| 3 | 74262.000 | -2.064326e+04 | 94905.255319 |
| 4 | 1727234.250 | 1.767253e+06 | -40018.473918 |

In [45]:
```
# let us plot the fitted values vs residuals
sns.set_style("whitegrid")
sns.residplot(
    data = df_pred , x = "Fitted values" , y = "Residuals" , color = "purple" , lowess = True
)
plt.xlabel("FITTED VALUES")
plt.ylabel("Residuals")
plt.title("FITTED V/S RESIDUAL PLOT")
plt.show()
```




In [46]:
```
# it is having no pattern that means it is linearity in nature . assumed that linearity and independence of pre
```

## Test for normality

In [47]:
```
from scipy import stats
stats.shapiro(df_pred["Residuals"])
#null hypothesis : it is normally distributed
```

```
/Users/sachinssharma/anaconda3/lib/python3.11/site-packages/scipy/stats/_morestats.py:1816: UserWarning: p-valu
e may not be accurate for N > 5000.
  warnings.warn("p-value may not be accurate for N > 5000.")
```

Out[47]:
```
ShapiroResult(statistic=0.9637130498886108, pvalue=0.0)
```

score is less than 0.05 so reject null hypothesis, it is not normal as per the shapiro's test

## Test for Homoscedasticity

In [48]:
```
import statsmodels.stats.api as sms
sms.het_goldfeldquandt(df_pred['Residuals'] , x_train5)[1]
```

Out[48]:
```
0.7863209998227594
```

since p value is more than 0.05 we can say residuals are homoscedastic

#

In [49]:
```
# dropping columns from the test data that are not there in the above analysis, since by dropping both columns,
```

In [50]:
```
x_test2 = X_test.drop(['manufacture_year','Vroom Audit Rating','Maker_bmw','Maker_fiat','Maker_hyundai','Maker_
x_test2.head(2)
```

Out[50]:

| | const | Distance | Age of car | engine_displacement | engine_power | Maker_skoda | Maker_toyota | model_avensis | model_aygo | model_citigo | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **40755** | 1.0 | 155000.0 | 14.0 | 1995.0 | 110.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| **53337** | 1.0 | 199963.0 | 7.0 | 1968.0 | 81.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |

2 rows × 59 columns

In [51]:
```
# let's make the predictions on the test set
y_pred_test = olsres_6.predict(x_test2)
y_pred_train = olsres_6.predict(x_train5)
```

In [52]:
```
# to check model performance
from sklearn.metrics import mean_absolute_error , mean_squared_error
```

In [53]:
```
#let's  check the RMSE on the train data
rmse1 = np.sqrt(mean_squared_error(y_train,y_pred_train))
rmse1
```

Out[53]:
```
336013.3899990611
```

In [54]:
```
#let's check the RMSE on the test data
rmse2 = np.sqrt(mean_squared_error(y_test , y_pred_test))
rmse2
```

Out[54]:
```
329590.62845699233
```

more or less both values are similar so the model is accurate