Analysis of Variance and Linear Models

Video : https://vimeo.com/930516655/4f44843f37?share=copy

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as sm
import scipy.stats as stats
import numpy as np
import seaborn as sns
import sklearn.feature_selection as skl
```

In [2]:
```
cars = pd.read_csv('/Users/zanderbonnet/Desktop/GCU/DSC_510/DataSets/cars.csv')
cars.head()
```

Out[2]:

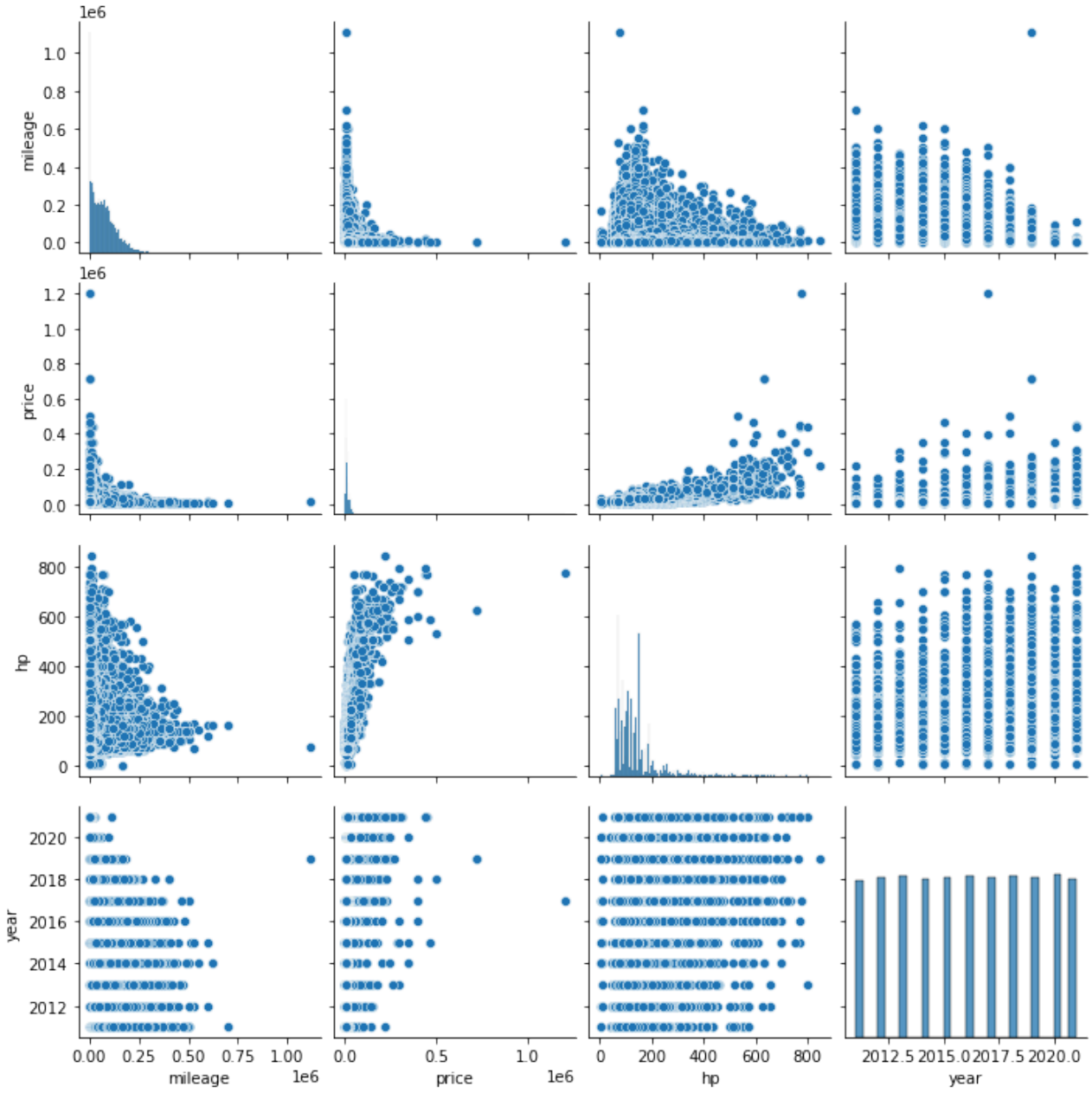
	mileage	make	model	fuel	gear	offerType	price	hp	year
0	235000	BMW	316	Diesel	Manual	Used	6800	116.0	2011
1	92800	Volkswagen	Golf	Gasoline	Manual	Used	6877	122.0	2011
2	149300	SEAT	Exeo	Gasoline	Manual	Used	6900	160.0	2011
3	96200	Renault	Megane	Gasoline	Manual	Used	6950	110.0	2011
4	156000	Peugeot	308	Gasoline	Manual	Used	6950	156.0	2011

In [3]:
```
cars.dropna(inplace= True)
cars.head()
```

Out[3]:

	mileage	make	model	fuel	gear	offerType	price	hp	year
0	235000	BMW	316	Diesel	Manual	Used	6800	116.0	2011
1	92800	Volkswagen	Golf	Gasoline	Manual	Used	6877	122.0	2011
2	149300	SEAT	Exeo	Gasoline	Manual	Used	6900	160.0	2011
3	96200	Renault	Megane	Gasoline	Manual	Used	6950	110.0	2011
4	156000	Peugeot	308	Gasoline	Manual	Used	6950	156.0	2011

In [4]:
```
sns.pairplot(cars)
plt.show()
```



In [5]:

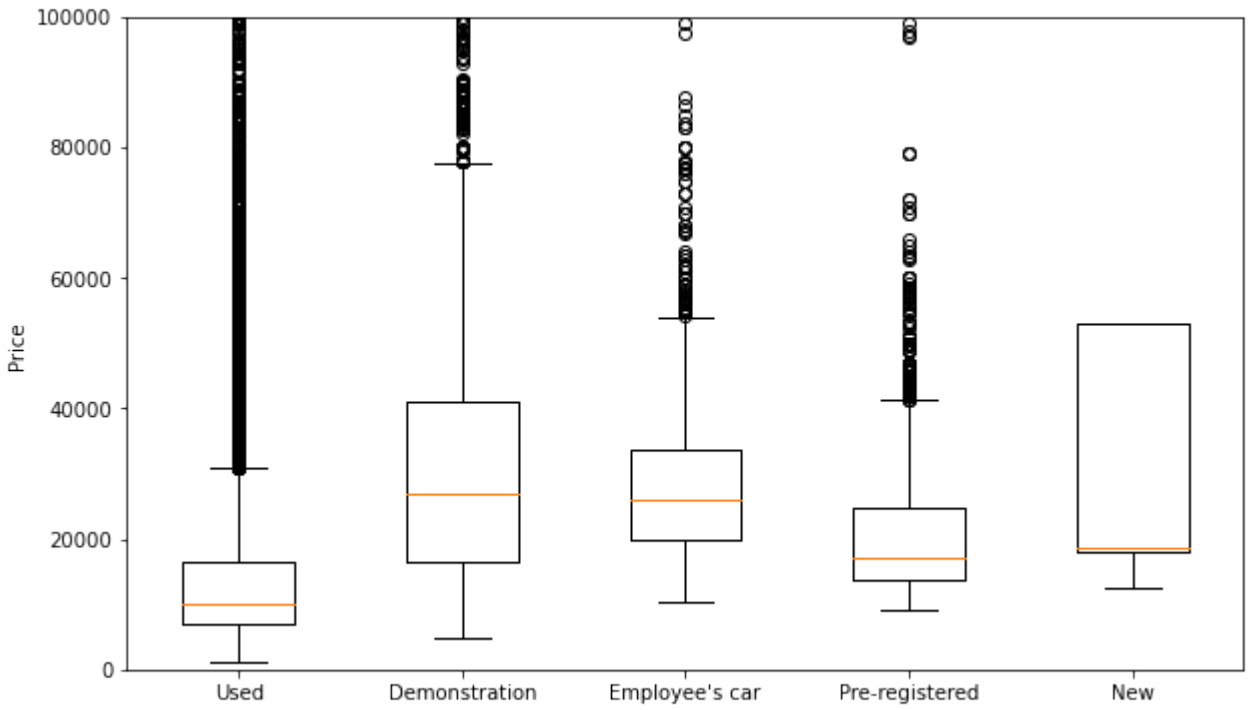
```
sns.heatmap(cars.corr(), annot = True, cmap="RdYlGn")
plt.show()
```

Topic 1

In [6]:

```
types = cars['offerType'].unique()
cost = []
for t in types:
    cost.append(cars[cars['offerType'] == t]['price'])
    print(t, np.mean(cars[cars['offerType'] == t]['price']))
print(stats.f_oneway(*cost))
plt.figure(figsize= (10,6))
plt.boxplot(cost, labels=types)
plt.ylim(0, 100000)
plt.ylabel('Price')
plt.show()
```

```
Used 14759.733338355683
Demonstration 34859.888132709486
Employee's car 30439.959713518354
Pre-registered 21134.68822254335
New 66123.76923076923
F_onewayResult(statistic=884.5325347552229, pvalue=0.0)
```



There is a significant difference in the true means of the price in the various offer types of cars

In [7]:

```
print('Difference in price by offer type')
makes = cars['make'].unique()
notdif = []
for m in makes:
    c = cars[cars['make'] == m]
    otype = c['offerType'].unique()
    l = []
    for o in otype:
        l.append(c[c['offerType'] == o]['price'])
    if len(l) > 1:
        print(m, 'P-Value:',stats.f_oneway(*l).pvalue)
```

```
        if stats.f_oneway(*l).pvalue > .1:
            notdif.append(m)
    else:
        print(m, 'Only', o)
```

```
Difference in price by offer type
BMW P-Value: 5.4645469487878734e-64
Volkswagen P-Value: 3.140739373395429e-289
SEAT P-Value: 9.493670562267e-147
Renault P-Value: 4.6458026705846207e-179
Peugeot P-Value: 1.9419792680469057e-66
Toyota P-Value: 3.1493719404887623e-41
Opel P-Value: 2.824444759260457e-263
Mazda P-Value: 1.391069799646565e-70
Ford P-Value: 1.178071089522876e-281
Mercedes-Benz P-Value: 1.9137002302311035e-37
Chevrolet P-Value: 5.202896599338879e-32
Audi P-Value: 3.306817110853567e-112
Fiat P-Value: 6.168598591498945e-205
Kia P-Value: 1.4830559231454102e-69
Dacia P-Value: 7.6688943450952285e-81
MINI P-Value: 2.846609115754667e-15
Hyundai P-Value: 1.9203959970524309e-174
Skoda P-Value: 4.349195806273358e-200
Citroen P-Value: 7.786242912759917e-26
Infiniti Only Used
Suzuki P-Value: 2.3829432778974185e-58
SsangYong P-Value: 2.169724373575228e-07
smart P-Value: 1.641455170464057e-27
Cupra P-Value: 0.011975507573209594
Volvo P-Value: 1.2043484702759885e-50
Jaguar P-Value: 3.2779813295507243e-07
Porsche P-Value: 4.43291968545244e-06
Nissan P-Value: 9.755749759778855e-75
Honda P-Value: 0.023406268097448096
Mitsubishi P-Value: 2.2230698828082806e-15
Lexus P-Value: 0.00020539933409712116
Jeep P-Value: 7.099502923815881e-11
Maserati Only Used
Bentley P-Value: 0.6387003098156387
Land P-Value: 2.229384787065092e-11
Alfa P-Value: 4.627912938003946e-15
Subaru P-Value: 4.5850729751894475e-11
Dodge P-Value: 0.4149527787687358
Microcar Only Used
Lamborghini P-Value: 0.25418393079214974
Lada P-Value: 1.9171486465744496e-09
Tesla Only Used
Chrysler Only Used
McLaren P-Value: 0.1013372147937004
Aston P-Value: 0.24685007639429998
Rolls-Royce Only Used
Lancia Only Used
Abarth P-Value: 0.0005195352919048654
DS P-Value: 0.3322366063107912
Daihatsu Only Used
Ligier Only Used
Ferrari P-Value: 0.9876576265780621
Aixam Only Used
Zhidou Only Demonstration
```

```
Morgan Only Used
Maybach Only Used
RAM Only Used
Alpina P-Value: 0.1931359881116224
Polestar Only Used
Brilliance Only Used
Piaggio Only Used
FISKER Only Used
Others Only Used
Cadillac P-Value: 0.31343491361237624
Iveco Only Used
Isuzu Only Used
Corvette P-Value: 0.0753746486499622
Baic Only Used
DFSK Only Used
Estrima Only Used
Alpine Only Demonstration
```

In [8]:

```
nd = cars[cars['make'].isin(notdif)].sort_values('make')
notdif
```

Out[8]:

```
['Bentley',
 'Dodge',
 'Lamborghini',
 'McLaren',
 'Aston',
 'DS',
 'Ferrari',
 'Alpina',
 'Cadillac']
```

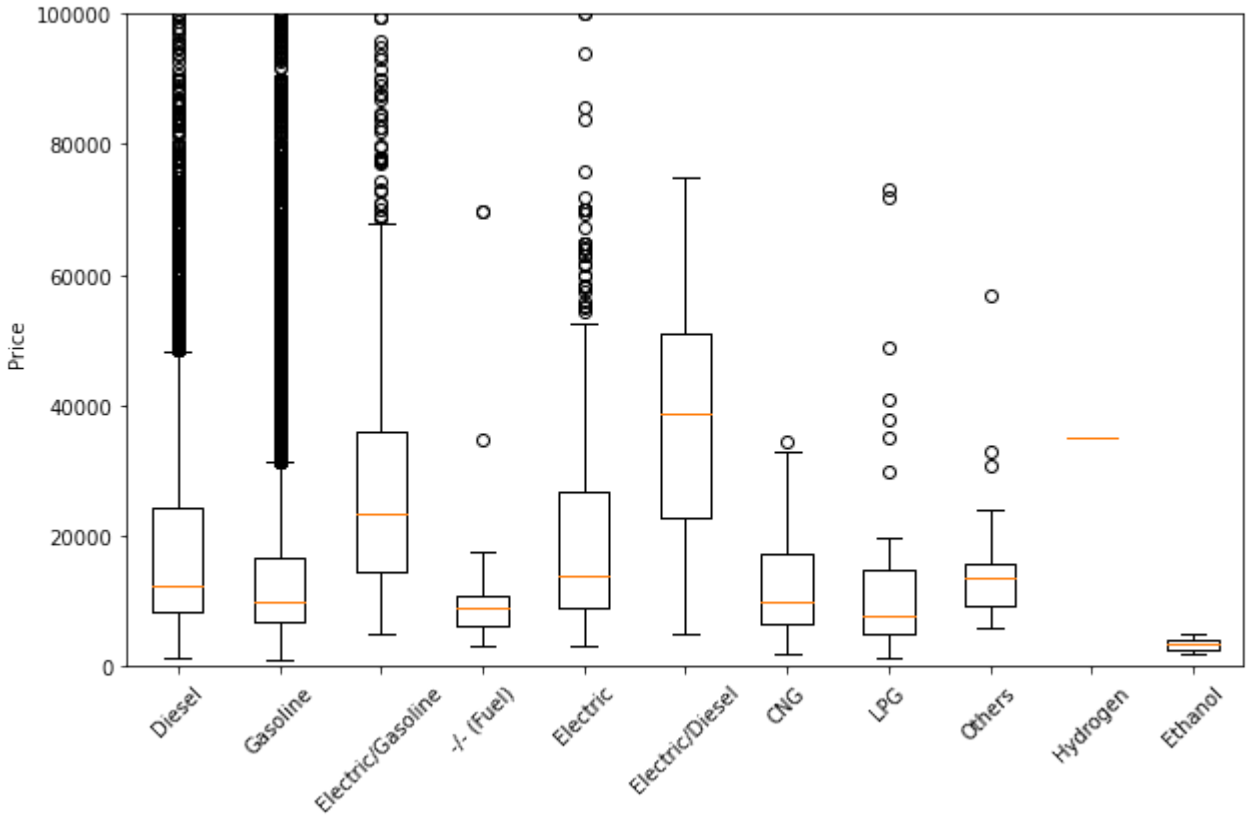
When analyzing the difference of price of offer type in every car brand it is found that a handful of brands have the same value across all types. This shows that these car brands tend to hold their value across all sales in the dataset.

In [9]:

```
types = cars['fuel'].unique()
cost = []
for t in types:
    cost.append(cars[cars['fuel'] == t]['price'])
    print(t, np.mean(cars[cars['fuel'] == t]['price']))
print(stats.f_oneway(*cost))
plt.figure(figsize= (10,6))
plt.boxplot(cost, labels=types)
plt.xticks(rotation = 45)
plt.ylim(0, 100000)
plt.ylabel('Price')
plt.show()
```

```
Diesel 18126.771116089076
Gasoline 15065.131685236769
Electric/Gasoline 29698.1815008726
-/- (Fuel) 15213.545454545454
Electric 23162.860816944023
Electric/Diesel 37605.018867924526
CNG 12602.172413793103
LPG 11396.368
Others 14468.63829787234
Hydrogen 34990.0
```

```
Ethanol 3450.0
F_onewayResult(statistic=98.24199443388889, pvalue=1.8423612429045121e-202)
```

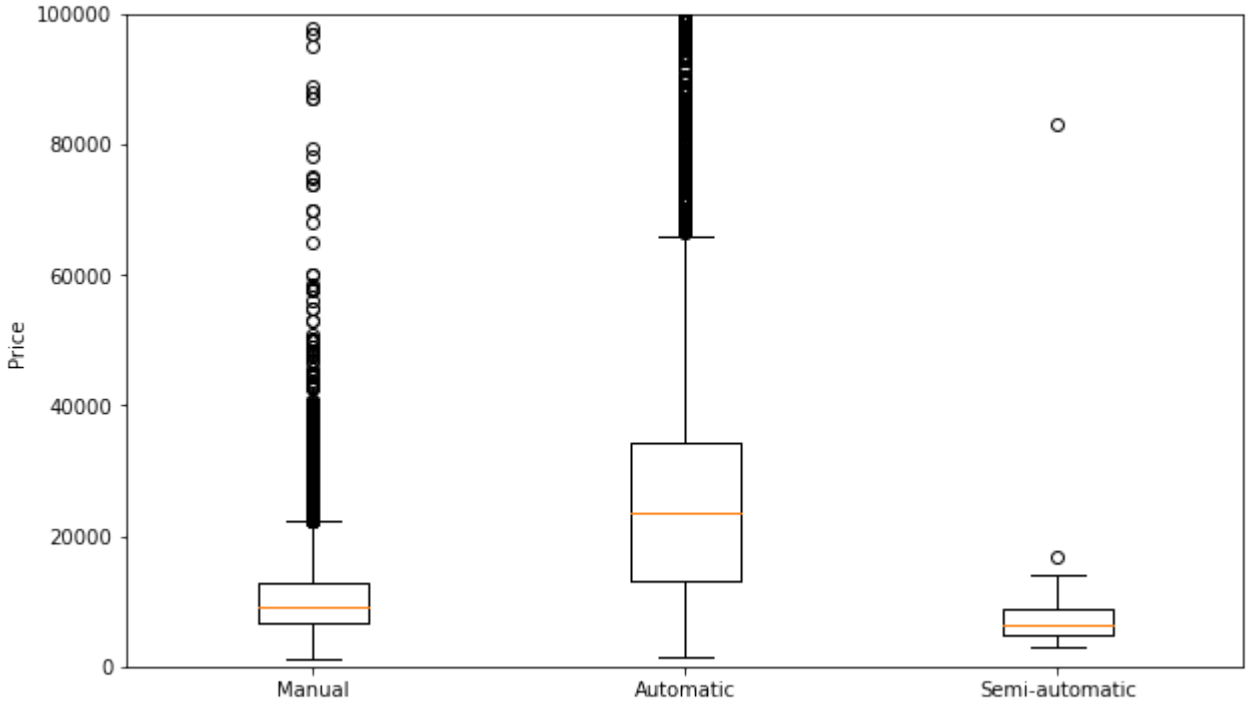


There is a significant difference in the price of cars based on their fuel type.

In [10]:

```
types = cars['gear'].unique()
cost = []
for t in types:
    cost.append(cars[cars['gear'] == t]['price'])
    print(t, np.mean(cars[cars['gear'] == t]['price']))
print(stats.f_oneway(*cost))
plt.figure(figsize= (10,6))
plt.boxplot(cost, labels=types)
plt.ylim(0, 100000)
plt.ylabel('Price')
plt.show()
```

```
Manual 10569.332739450329
Automatic 28158.78352222081
Semi-automatic 8424.982142857143
F_onewayResult(statistic=5302.455493353005, pvalue=0.0)
```



There is a significant difference in the price of cars based on their gear type.

Topic 2

In [11]:
```
mod = sm.ols('price ~ mileage + make + fuel + gear + offerType + hp + year', dat
mod.summary()
```

Out[11]:

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.782
Model:	OLS	**Adj. R-squared:**	0.782
Method:	Least Squares	**F-statistic:**	1857.
Date:	Wed, 03 Apr 2024	**Prob (F-statistic):**	0.00
Time:	21:45:10	**Log-Likelihood:**	-4.8482e+05
No. Observations:	46071	**AIC:**	9.698e+05
Df Residuals:	45981	**BIC:**	9.706e+05
Df Model:	89		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.879e+06	4.33e+04	-43.375	0.000	-1.96e+06	-1.79e+06
make[T.Aixam]	1.387e+04	6518.211	2.127	0.033	1089.790	2.66e+04
make[T.Alfa]	4947.7639	1585.698	3.120	0.002	1839.772	8055.756
make[T.Alpina]	7740.1407	3169.351	2.442	0.015	1528.163	1.4e+04
make[T.Alpine]	2.817e+04	9108.937	3.092	0.002	1.03e+04	4.6e+04

make[T.Aston]	8.47e+04	2161.528	39.184	0.000	8.05e+04	8.89e+04
make[T.Audi]	7820.7205	1388.714	5.632	0.000	5098.820	1.05e+04
make[T.BMW]	3863.1035	1390.396	2.778	0.005	1137.905	6588.302
make[T.Baic]	3162.9500	6513.401	0.486	0.627	-9603.417	1.59e+04
make[T.Bentley]	1.258e+05	2963.558	42.434	0.000	1.2e+05	1.32e+05
make[T.Brilliance]	1217.7113	9145.074	0.133	0.894	-1.67e+04	1.91e+04
make[T.Cadillac]	-5041.1869	3673.142	-1.372	0.170	-1.22e+04	2158.229
make[T.Chevrolet]	3121.8028	1503.564	2.076	0.038	174.795	6068.811
make[T.Chrysler]	-6796.3904	4260.024	-1.595	0.111	-1.51e+04	1553.323
make[T.Citroen]	4612.8293	1407.233	3.278	0.001	1854.631	7371.027
make[T.Corvette]	1.28e+04	5392.871	2.374	0.018	2232.763	2.34e+04
make[T.Cupra]	34.4250	1774.110	0.019	0.985	-3442.857	3511.707
make[T.DFSK]	855.9873	6514.215	0.131	0.895	-1.19e+04	1.36e+04
make[T.DS]	9181.1282	2639.472	3.478	0.001	4007.722	1.44e+04
make[T.Dacia]	3607.3937	1417.715	2.545	0.011	828.649	6386.138
make[T.Daihatsu]	7899.3331	3163.356	2.497	0.013	1699.106	1.41e+04
make[T.Dodge]	-1878.8025	2373.761	-0.791	0.429	-6531.411	2773.806
make[T.Estrima]	1.24e+04	6525.245	1.900	0.057	-388.423	2.52e+04
make[T.FISKER]	1.849e+04	9115.057	2.028	0.043	622.387	3.64e+04
make[T.Ferrari]	2.271e+05	3080.543	73.732	0.000	2.21e+05	2.33e+05
make[T.Fiat]	5404.1558	1393.538	3.878	0.000	2672.799	8135.513
make[T.Ford]	4073.2980	1383.172	2.945	0.003	1362.260	6784.336
make[T.Honda]	5753.1198	1529.933	3.760	0.000	2754.428	8751.812
make[T.Hyundai]	4295.4599	1391.022	3.088	0.002	1569.036	7021.884
make[T.Infiniti]	-2368.1570	2942.668	-0.805	0.421	-8135.832	3399.518
make[T.Isuzu]	3006.9604	9108.221	0.330	0.741	-1.48e+04	2.09e+04
make[T.Iveco]	1.083e+04	5379.297	2.013	0.044	282.396	2.14e+04
make[T.Jaguar]	5436.7977	1596.272	3.406	0.001	2308.080	8565.515
make[T.Jeep]	5059.7424	1550.993	3.262	0.001	2019.772	8099.713
make[T.Kia]	4035.5957	1403.470	2.875	0.004	1284.772	6786.419
make[T.Lada]	4517.5955	2122.785	2.128	0.033	356.904	8678.287
make[T.Lamborghini]	2.109e+05	3327.979	63.363	0.000	2.04e+05	2.17e+05
make[T.Lancia]	3386.6445	2581.707	1.312	0.190	-1673.542	8446.831
make[T.Land]	2.02e+04	1548.365	13.046	0.000	1.72e+04	2.32e+04
make[T.Lexus]	1.019e+04	1957.055	5.205	0.000	6349.734	1.4e+04
make[T.Ligier]	1.458e+04	4260.709	3.421	0.001	6226.550	2.29e+04

make[T.MINI]	5408.3497	1438.618	3.759	0.000	2588.635	8228.064
make[T.Maserati]	2.279e+04	2945.949	7.736	0.000	1.7e+04	2.86e+04
make[T.Maybach]	5.271e+05	6521.249	80.835	0.000	5.14e+05	5.4e+05
make[T.Mazda]	3691.4868	1415.922	2.607	0.009	916.258	6466.716
make[T.McLaren]	1.152e+05	3066.891	37.566	0.000	1.09e+05	1.21e+05
make[T.Mercedes-Benz]	9392.3839	1390.497	6.755	0.000	6666.988	1.21e+04
make[T.Microcar]	1.347e+04	3051.291	4.415	0.000	7492.342	1.95e+04
make[T.Mitsubishi]	3542.0663	1448.079	2.446	0.014	703.809	6380.323
make[T.Morgan]	4.241e+04	6513.093	6.511	0.000	2.96e+04	5.52e+04
make[T.Nissan]	4656.1161	1415.231	3.290	0.001	1882.242	7429.990
make[T.Opel]	3566.4934	1382.738	2.579	0.010	856.306	6276.681
make[T.Others]	-2810.8159	9108.220	-0.309	0.758	-2.07e+04	1.5e+04
make[T.Peugeot]	4751.8519	1400.668	3.393	0.001	2006.521	7497.183
make[T.Piaggio]	1.228e+04	5378.396	2.283	0.022	1735.619	2.28e+04
make[T.Polestar]	3193.7515	4726.070	0.676	0.499	-6069.419	1.25e+04
make[T.Porsche]	3.532e+04	1504.290	23.477	0.000	3.24e+04	3.83e+04
make[T.RAM]	-5491.5989	6528.873	-0.841	0.400	-1.83e+04	7305.093
make[T.Renault]	3898.0968	1387.252	2.810	0.005	1179.061	6617.133
make[T.Rolls-Royce]	1.165e+05	5388.377	21.617	0.000	1.06e+05	1.27e+05
make[T.SEAT]	5074.6305	1391.367	3.647	0.000	2347.529	7801.732
make[T.Skoda]	5559.6136	1386.320	4.010	0.000	2842.406	8276.821
make[T.SsangYong]	100.2935	2021.845	0.050	0.960	-3862.554	4063.141
make[T.Subaru]	4970.9561	1821.796	2.729	0.006	1400.208	8541.704
make[T.Suzuki]	3955.8181	1456.589	2.716	0.007	1100.880	6810.756
make[T.Tesla]	-8439.4356	2893.994	-2.916	0.004	-1.41e+04	-2767.162
make[T.Toyota]	5133.5632	1401.266	3.664	0.000	2387.059	7880.067
make[T.Volkswagen]	7290.0216	1381.223	5.278	0.000	4582.803	9997.241
make[T.Volvo]	6850.1592	1414.877	4.842	0.000	4076.978	9623.341
make[T.Zhidou]	3675.5976	9116.300	0.403	0.687	-1.42e+04	2.15e+04
make[T.smart]	6112.2046	1411.454	4.330	0.000	3345.732	8878.677
fuel[T.CNG]	-849.1791	2108.965	-0.403	0.687	-4982.783	3284.425
fuel[T.Diesel]	784.3942	1936.842	0.405	0.685	-3011.847	4580.635
fuel[T.Electric]	1620.5789	1964.630	0.825	0.409	-2230.127	5471.284
fuel[T.Electric/Diesel]	3550.1015	2299.969	1.544	0.123	-957.873	8058.076
fuel[T.Electric/Gasoline]	542.3352	1956.402	0.277	0.782	-3292.244	4376.914
fuel[T.Ethanol]	501.6679	6655.041	0.075	0.940	-1.25e+04	1.35e+04

fuel[T.Gasoline]	-1026.4867	1935.698	-0.530	0.596	-4820.485	2767.512
fuel[T.Hydrogen]	1.243e+04	9212.416	1.350	0.177	-5622.661	3.05e+04
fuel[T.LPG]	564.1310	2103.017	0.268	0.789	-3557.816	4686.078
fuel[T.Others]	-1558.4670	2339.632	-0.666	0.505	-6144.181	3027.247
gear[T.Manual]	-274.2263	116.020	-2.364	0.018	-501.626	-46.826
gear[T.Semi-automatic]	832.7004	1214.171	0.686	0.493	-1547.093	3212.494
offerType[T.Employee's car]	-4109.7257	332.088	-12.375	0.000	-4760.623	-3458.829
offerType[T.New]	-2560.6249	2523.131	-1.015	0.310	-7506.002	2384.752
offerType[T.Pre-registered]	-5198.8630	259.519	-20.033	0.000	-5707.525	-4690.201
offerType[T.Used]	-5521.4991	211.862	-26.062	0.000	-5936.752	-5106.246
mileage	-0.0559	0.001	-52.783	0.000	-0.058	-0.054
hp	145.7470	0.843	172.970	0.000	144.095	147.399
year	932.5685	21.411	43.556	0.000	890.603	974.534

Omnibus:	121801.610	**Durbin-Watson:**	1.580
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	11971677478.159
Skew:	30.846	**Prob(JB):**	0.00
Kurtosis:	2499.530	**Cond. No.**	9.79e+07

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.79e+07. This might indicate that there are strong multicollinearity or other numerical problems.

Fuel type does not have a significant p-value in the model, so it is not significant in predicting the price.

In [12]:
```
#eliminate fuel as it is not significant
mod = sm.ols('price ~ mileage + make + gear + offerType + hp + year', data = car
mod.summary()
```

Out[12]:

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.781
Model:	OLS	**Adj. R-squared:**	0.780
Method:	Least Squares	**F-statistic:**	2074.
Date:	Wed, 03 Apr 2024	**Prob (F-statistic):**	0.00
Time:	21:45:11	**Log-Likelihood:**	-4.8498e+05
No. Observations:	46071	**AIC:**	9.701e+05
Df Residuals:	45991	**BIC:**	9.708e+05

Df Model:	79
Covariance Type:	nonrobust

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.034e+06	4.24e+04	-47.953	0.000	-2.12e+06	-1.95e+06
make[T.Aixam]	1.592e+04	6538.555	2.435	0.015	3107.267	2.87e+04
make[T.Alfa]	5137.7621	1590.528	3.230	0.001	2020.302	8255.223
make[T.Alpina]	7794.3400	3180.068	2.451	0.014	1561.357	1.4e+04
make[T.Alpine]	2.806e+04	9140.645	3.069	0.002	1.01e+04	4.6e+04
make[T.Aston]	8.438e+04	2168.428	38.912	0.000	8.01e+04	8.86e+04
make[T.Audi]	8329.2819	1392.421	5.982	0.000	5600.115	1.11e+04
make[T.BMW]	4394.0701	1394.105	3.152	0.002	1661.603	7126.537
make[T.Baic]	2978.0275	6535.980	0.456	0.649	-9832.596	1.58e+04
make[T.Bentley]	1.256e+05	2973.496	42.246	0.000	1.2e+05	1.31e+05
make[T.Brilliance]	2827.8802	9139.362	0.309	0.757	-1.51e+04	2.07e+04
make[T.Cadillac]	-5262.6347	3685.649	-1.428	0.153	-1.25e+04	1961.296
make[T.Chevrolet]	3414.7576	1507.965	2.264	0.024	459.123	6370.392
make[T.Chrysler]	-6859.0954	4271.381	-1.606	0.108	-1.52e+04	1512.878
make[T.Citroen]	4991.0585	1411.200	3.537	0.000	2225.085	7757.032
make[T.Corvette]	1.255e+04	5411.417	2.319	0.020	1942.728	2.32e+04
make[T.Cupra]	-67.9905	1778.957	-0.038	0.970	-3554.773	3418.792
make[T.DFSK]	385.3104	6536.750	0.059	0.953	-1.24e+04	1.32e+04
make[T.DS]	1.013e+04	2647.072	3.826	0.000	4939.589	1.53e+04
make[T.Dacia]	3883.9169	1420.873	2.733	0.006	1098.983	6668.851
make[T.Daihatsu]	7896.1524	3174.035	2.488	0.013	1674.995	1.41e+04
make[T.Dodge]	-1974.6180	2376.619	-0.831	0.406	-6632.828	2683.592
make[T.Estrima]	1.476e+04	6538.690	2.258	0.024	1948.855	2.76e+04
make[T.FISKER]	2.023e+04	9142.182	2.212	0.027	2307.244	3.81e+04
make[T.Ferrari]	2.268e+05	3090.757	73.377	0.000	2.21e+05	2.33e+05
make[T.Fiat]	5720.3095	1397.188	4.094	0.000	2981.799	8458.820
make[T.Ford]	4538.9224	1386.971	3.273	0.001	1820.437	7257.408
make[T.Honda]	5903.1148	1534.675	3.846	0.000	2895.127	8911.103
make[T.Hyundai]	4572.2801	1394.968	3.278	0.001	1838.121	7306.439
make[T.Infiniti]	-2006.6101	2952.401	-0.680	0.497	-7793.362	3780.142
make[T.Isuzu]	5017.7449	9139.165	0.549	0.583	-1.29e+04	2.29e+04
make[T.Iveco]	1.129e+04	5397.817	2.091	0.037	706.221	2.19e+04

make[T.Jaguar]	5926.8371	1600.895	3.702	0.000	2789.058	9064.616
make[T.Jeep]	5515.9990	1555.371	3.546	0.000	2467.447	8564.551
make[T.Kia]	4341.1348	1407.465	3.084	0.002	1582.482	7099.788
make[T.Lada]	4658.2097	2129.691	2.187	0.029	483.982	8832.437
make[T.Lamborghini]	2.106e+05	3339.128	63.063	0.000	2.04e+05	2.17e+05
make[T.Lancia]	3468.6879	2590.283	1.339	0.181	-1608.307	8545.683
make[T.Land]	2.123e+04	1551.743	13.681	0.000	1.82e+04	2.43e+04
make[T.Lexus]	1.083e+04	1952.003	5.546	0.000	7000.645	1.47e+04
make[T.Ligier]	1.614e+04	4273.995	3.775	0.000	7758.915	2.45e+04
make[T.MINI]	5664.2706	1442.802	3.926	0.000	2836.355	8492.186
make[T.Maserati]	2.281e+04	2955.868	7.718	0.000	1.7e+04	2.86e+04
make[T.Maybach]	5.268e+05	6543.825	80.511	0.000	5.14e+05	5.4e+05
make[T.Mazda]	3988.7123	1420.012	2.809	0.005	1205.466	6771.958
make[T.McLaren]	1.149e+05	3077.103	37.326	0.000	1.09e+05	1.21e+05
make[T.Mercedes-Benz]	9806.3329	1394.298	7.033	0.000	7073.487	1.25e+04
make[T.Microcar]	1.513e+04	3059.557	4.946	0.000	9134.745	2.11e+04
make[T.Mitsubishi]	3691.7545	1452.254	2.542	0.011	845.314	6538.195
make[T.Morgan]	4.263e+04	6535.671	6.523	0.000	2.98e+04	5.54e+04
make[T.Nissan]	5019.1862	1419.325	3.536	0.000	2237.288	7801.084
make[T.Opel]	3905.8879	1386.649	2.817	0.005	1188.034	6623.742
make[T.Others]	-2105.1380	9139.828	-0.230	0.818	-2e+04	1.58e+04
make[T.Peugeot]	5138.7922	1404.587	3.659	0.000	2385.781	7891.804
make[T.Piaggio]	1.324e+04	5396.664	2.454	0.014	2664.289	2.38e+04
make[T.Polestar]	5381.9979	4727.551	1.138	0.255	-3884.076	1.46e+04
make[T.Porsche]	3.534e+04	1508.320	23.433	0.000	3.24e+04	3.83e+04
make[T.RAM]	-5328.3162	6538.273	-0.815	0.415	-1.81e+04	7486.801
make[T.Renault]	4422.5921	1390.910	3.180	0.001	1696.386	7148.798
make[T.Rolls-Royce]	1.161e+05	5406.927	21.477	0.000	1.06e+05	1.27e+05
make[T.SEAT]	5288.1422	1395.326	3.790	0.000	2553.282	8023.002
make[T.Skoda]	5810.6813	1390.317	4.179	0.000	3085.637	8535.725
make[T.SsangYong]	869.1886	2027.676	0.429	0.668	-3105.087	4843.465
make[T.Subaru]	5654.8083	1826.694	3.096	0.002	2074.460	9235.156
make[T.Suzuki]	4195.0608	1460.334	2.873	0.004	1332.783	7057.338
make[T.Tesla]	-6624.1495	2873.989	-2.305	0.021	-1.23e+04	-991.086
make[T.Toyota]	5430.8275	1403.397	3.870	0.000	2680.148	8181.507
make[T.Volkswagen]	7712.6265	1385.049	5.568	0.000	4997.908	1.04e+04

make[T.Volvo]	7654.6574	1418.116	5.398	0.000	4875.127	1.04e+04
make[T.Zhidou]	5926.2443	9141.860	0.648	0.517	-1.2e+04	2.38e+04
make[T.smart]	6177.9277	1415.467	4.365	0.000	3403.591	8952.264
gear[T.Manual]	-732.3576	112.497	-6.510	0.000	-952.854	-511.861
gear[T.Semi-automatic]	605.2981	1217.963	0.497	0.619	-1781.928	2992.524
offerType[T.Employee's car]	-4189.5382	332.715	-12.592	0.000	-4841.664	-3537.412
offerType[T.New]	-2818.1593	2531.857	-1.113	0.266	-7780.639	2144.321
offerType[T.Pre-registered]	-5356.4192	259.798	-20.618	0.000	-5865.627	-4847.212
offerType[T.Used]	-5612.1263	211.310	-26.559	0.000	-6026.297	-5197.955
mileage	-0.0483	0.001	-50.843	0.000	-0.050	-0.046
hp	145.8160	0.842	173.222	0.000	144.166	147.466
year	1009.0877	20.989	48.077	0.000	967.949	1050.226

Omnibus:	121239.309	**Durbin-Watson:**	1.583
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	11616605268.145
Skew:	30.472	**Prob(JB):**	0.00
Kurtosis:	2462.224	**Cond. No.**	9.55e+07

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.55e+07. This might indicate that there are
strong multicollinearity or other numerical problems.

The gearing of the car is also not significant in the price prediction model so it can be removed.

In [13]:
```
#eliminate gear as it is not significant
mod = sm.ols('price ~ mileage + make + offerType + hp + year', data = cars).fit(
mod.summary()
```

Out[13]:

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.781
Model:	OLS	**Adj. R-squared:**	0.780
Method:	Least Squares	**F-statistic:**	2125.
Date:	Wed, 03 Apr 2024	**Prob (F-statistic):**	0.00
Time:	21:45:11	**Log-Likelihood:**	-4.8500e+05
No. Observations:	46071	**AIC:**	9.702e+05
Df Residuals:	45993	**BIC:**	9.708e+05
Df Model:	77		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.075e+06	4.19e+04	-49.497	0.000	-2.16e+06	-1.99e+06
make[T.Aixam]	1.691e+04	6539.717	2.586	0.010	4093.459	2.97e+04
make[T.Alfa]	5333.4781	1590.955	3.352	0.001	2215.182	8451.774
make[T.Alpina]	7844.6411	3181.482	2.466	0.014	1608.887	1.41e+04
make[T.Alpine]	2.761e+04	9144.477	3.019	0.003	9686.849	4.55e+04
make[T.Aston]	8.411e+04	2169.016	38.780	0.000	7.99e+04	8.84e+04
make[T.Audi]	8666.1287	1392.087	6.225	0.000	5937.617	1.14e+04
make[T.BMW]	4730.2986	1393.770	3.394	0.001	1998.487	7462.110
make[T.Baic]	3333.2138	6538.678	0.510	0.610	-9482.696	1.61e+04
make[T.Bentley]	1.254e+05	2974.622	42.154	0.000	1.2e+05	1.31e+05
make[T.Brilliance]	2996.3730	9143.413	0.328	0.743	-1.49e+04	2.09e+04
make[T.Cadillac]	-5161.0732	3687.266	-1.400	0.162	-1.24e+04	2066.025
make[T.Chevrolet]	3590.4311	1508.393	2.380	0.017	633.958	6546.904
make[T.Chrysler]	-6410.1461	4272.734	-1.500	0.134	-1.48e+04	1964.479
make[T.Citroen]	5200.3746	1411.469	3.684	0.000	2433.872	7966.877
make[T.Corvette]	1.18e+04	5412.612	2.180	0.029	1191.586	2.24e+04
make[T.Cupra]	265.8883	1779.016	0.149	0.881	-3221.011	3752.788
make[T.DFSK]	1021.6613	6538.947	0.156	0.876	-1.18e+04	1.38e+04
make[T.DS]	1.047e+04	2647.729	3.955	0.000	5282.318	1.57e+04
make[T.Dacia]	3986.7776	1421.421	2.805	0.005	1200.771	6772.784
make[T.Daihatsu]	8166.9111	3175.177	2.572	0.010	1943.514	1.44e+04
make[T.Dodge]	-1929.9288	2377.672	-0.812	0.417	-6590.204	2730.346
make[T.Estrima]	1.573e+04	6539.948	2.405	0.016	2908.192	2.85e+04
make[T.FISKER]	2.045e+04	9146.206	2.236	0.025	2525.537	3.84e+04
make[T.Ferrari]	2.263e+05	3091.184	73.205	0.000	2.2e+05	2.32e+05
make[T.Fiat]	5890.4037	1397.569	4.215	0.000	3151.146	8629.661
make[T.Ford]	4714.9538	1387.330	3.399	0.001	1995.765	7434.143
make[T.Honda]	6145.3326	1534.907	4.004	0.000	3136.891	9153.774
make[T.Hyundai]	4791.3815	1395.185	3.434	0.001	2056.798	7525.965
make[T.Infiniti]	-1547.7818	2952.881	-0.524	0.600	-7335.474	4239.910
make[T.Isuzu]	5017.2567	9143.254	0.549	0.583	-1.29e+04	2.29e+04
make[T.Iveco]	1.135e+04	5400.224	2.101	0.036	762.863	2.19e+04
make[T.Jaguar]	6282.4278	1600.681	3.925	0.000	3145.068	9419.788
make[T.Jeep]	5914.5860	1554.863	3.804	0.000	2867.031	8962.141
make[T.Kia]	4557.8648	1407.703	3.238	0.001	1798.745	7316.984

make[T.Lada]	4768.7141	2130.576	2.238	0.025	592.753	8944.676
make[T.Lamborghini]	2.101e+05	3339.798	62.906	0.000	2.04e+05	2.17e+05
make[T.Lancia]	3680.9102	2591.233	1.421	0.155	-1397.948	8759.768
make[T.Land]	2.161e+04	1551.364	13.927	0.000	1.86e+04	2.46e+04
make[T.Lexus]	1.125e+04	1951.780	5.765	0.000	7427.207	1.51e+04
make[T.Ligier]	1.716e+04	4272.995	4.016	0.000	8786.853	2.55e+04
make[T.MINI]	5862.9368	1443.122	4.063	0.000	3034.394	8691.479
make[T.Maserati]	2.298e+04	2957.086	7.770	0.000	1.72e+04	2.88e+04
make[T.Maybach]	5.266e+05	6546.614	80.434	0.000	5.14e+05	5.39e+05
make[T.Mazda]	4207.5249	1420.249	2.963	0.003	1423.815	6991.234
make[T.McLaren]	1.145e+05	3077.847	37.186	0.000	1.08e+05	1.2e+05
make[T.Mercedes-Benz]	1.02e+04	1393.609	7.319	0.000	7468.811	1.29e+04
make[T.Microcar]	1.611e+04	3057.241	5.269	0.000	1.01e+04	2.21e+04
make[T.Mitsubishi]	3919.7037	1452.480	2.699	0.007	1072.819	6766.588
make[T.Morgan]	4.254e+04	6538.579	6.505	0.000	2.97e+04	5.54e+04
make[T.Nissan]	5256.7904	1419.487	3.703	0.000	2474.573	8039.008
make[T.Opel]	4077.0823	1387.023	2.939	0.003	1358.496	6795.669
make[T.Others]	-1935.4692	9143.879	-0.212	0.832	-1.99e+04	1.6e+04
make[T.Peugeot]	5334.1482	1404.898	3.797	0.000	2580.527	8087.770
make[T.Piaggio]	1.353e+04	5398.900	2.505	0.012	2944.289	2.41e+04
make[T.Polestar]	5434.0467	4729.660	1.149	0.251	-3836.160	1.47e+04
make[T.Porsche]	3.542e+04	1508.948	23.472	0.000	3.25e+04	3.84e+04
make[T.RAM]	-5221.7839	6541.179	-0.798	0.425	-1.8e+04	7599.029
make[T.Renault]	4677.4619	1390.986	3.363	0.001	1951.107	7403.816
make[T.Rolls-Royce]	1.159e+05	5409.184	21.418	0.000	1.05e+05	1.26e+05
make[T.SEAT]	5503.0824	1395.560	3.943	0.000	2767.763	8238.402
make[T.Skoda]	6067.7674	1390.381	4.364	0.000	3342.600	8792.935
make[T.SsangYong]	1194.2789	2027.968	0.589	0.556	-2780.569	5169.127
make[T.Subaru]	5982.0531	1826.816	3.275	0.001	2401.466	9562.641
make[T.Suzuki]	4384.4880	1460.695	3.002	0.003	1521.502	7247.474
make[T.Tesla]	-6734.3937	2875.224	-2.342	0.019	-1.24e+04	-1098.910
make[T.Toyota]	5762.0330	1403.100	4.107	0.000	3011.935	8512.131
make[T.Volkswagen]	7996.8977	1384.981	5.774	0.000	5282.313	1.07e+04
make[T.Volvo]	8038.6277	1417.526	5.671	0.000	5260.255	1.08e+04
make[T.Zhidou]	6833.1969	9144.887	0.747	0.455	-1.11e+04	2.48e+04
make[T.smart]	6808.4508	1412.705	4.819	0.000	4039.527	9577.374

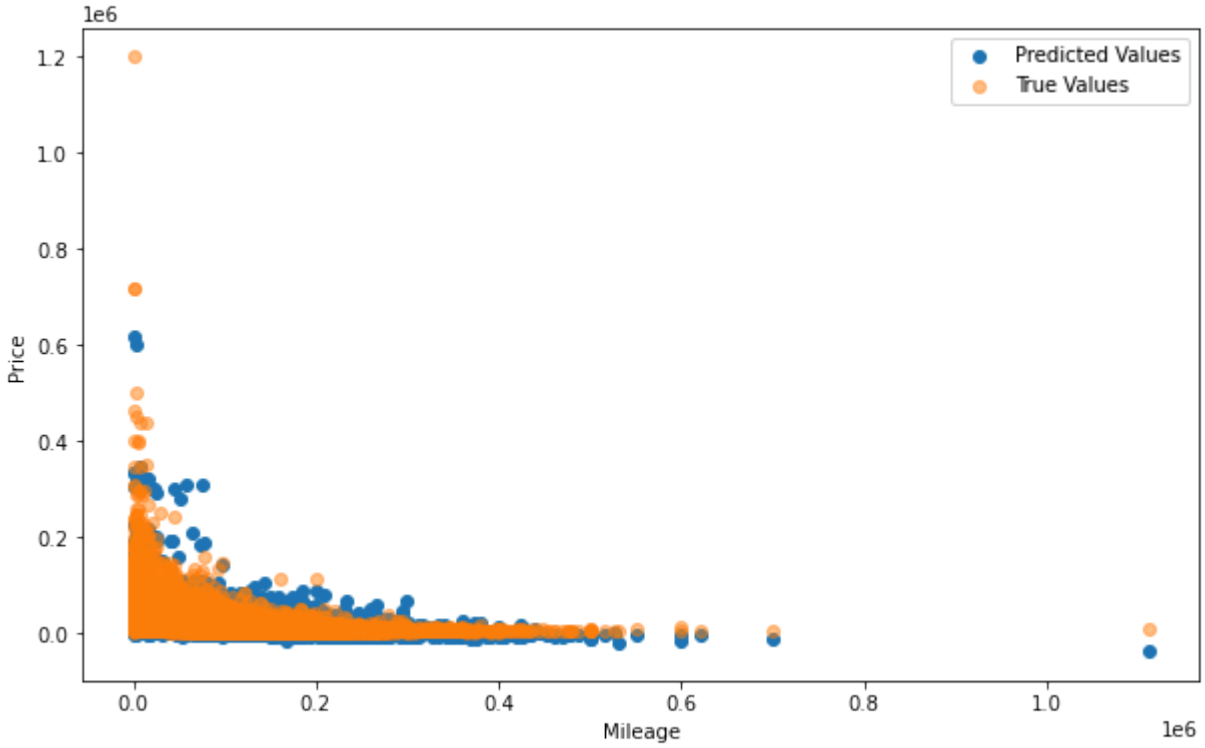
offerType[T.Employee's car]	-4192.4481	332.863	-12.595	0.000	-4844.865	-3540.032
offerType[T.New]	-2904.9221	2532.955	-1.147	0.251	-7869.554	2059.709
offerType[T.Pre-registered]	-5420.3776	259.728	-20.869	0.000	-5929.448	-4911.307
offerType[T.Used]	-5625.3998	211.392	-26.611	0.000	-6039.731	-5211.068
mileage	-0.0482	0.001	-50.701	0.000	-0.050	-0.046
hp	147.9862	0.773	191.397	0.000	146.471	149.502
year	1029.0661	20.758	49.575	0.000	988.381	1069.751

Omnibus:	121046.158	**Durbin-Watson:**	1.586
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	11523451997.497
Skew:	30.343	**Prob(JB):**	0.00
Kurtosis:	2452.344	**Cond. No.**	9.44e+07

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.44e+07. This might indicate that there are strong multicollinearity or other numerical problems.

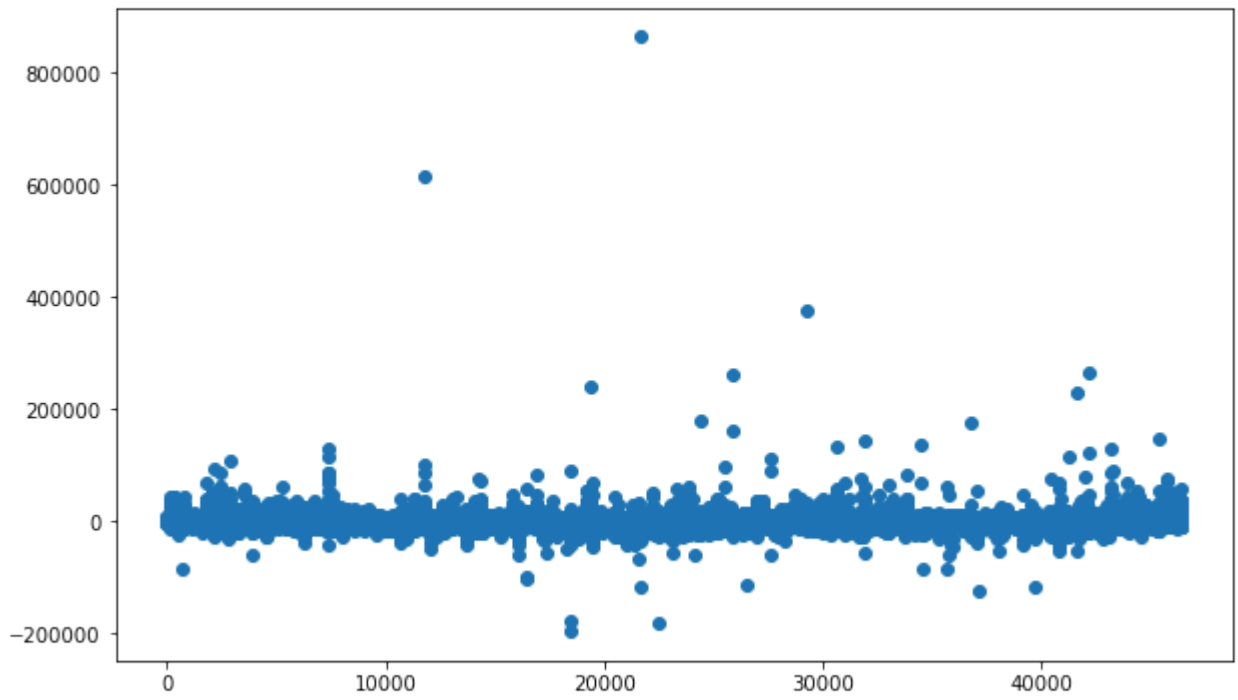
In [14]:

```
pred = mod.predict(cars)
plt.figure(figsize = (10,6))
plt.scatter(cars['mileage'], pred, label = 'Predicted Values')
plt.scatter(cars['mileage'], cars['price'], label = 'True Values', alpha = .5)
plt.xlabel('Mileage'), plt.ylabel('Price')
plt.legend()
plt.show()
```



In [15]:

```
plt.figure(figsize = (10,6))
plt.scatter(mod.resid.index,mod.resid)
plt.show()
```



The final model is based on the make, mileage, year, hp and offer type.

Make- This is dependent on the make of the vehicle. This makes sense as a significant predicto, because a luxuary car brand is going to cost more then a standard car brand.

mileage - -0.0483 This means that there is a negative .04 impact on the cars price for every mile that the car has. This makes sense as people are going to pay less for cars with more miles.

year - 1029.0661 There is a positive impact of 1029 for every year that the car is newer. Newer cars tend to be more expensive so this is a way to account for that.

hp - 147.986 There is a possitive 147 impact on the price for every hp that the car has. More powerful cars tend to have a higher price tag.

offer type - This is dependent on the offer type, but used has the largest impact on the price and new has the smallest impact on the price. We would expect this beacue new cars are going to be more expensive then their used counter part.

The final model is significant with a F statistic of 2125 leading to a 0 p-value, and a statistacly significant intercept.

The final model fits the assumptions of random residuals, independint variables, and equal variance of the predictors.

Topic 3

Assumptions/Limitations of ANOVA

The data must be normally distributed, be free of major outliers, and the variances of the samples must be similar. This limits the amount of data that you can work on using ANOVA. If we are to not meet these assumptions we might misleading and incorrect results. We can use data transformation and outlier removal to attemopt to correct any discrepencies in the data, but this may not always work. The bigest threat to ANOVA is how it is heavily influenced by outliers. Outliers can skew the means very significantly, so ANOVA will give a false positive if the means are skewed dramatically.

Assumptions/Limitations of Linear Regression

Similar to ANOVA the data must be normally distributed, be free of major outliers, and the variances of the samples must be similar. On top of these assumptions it must also have a linear relationship, and the residuals must have a random distrobution. This is the main limitation of linear regression. There are so many assumptions that it requires a perfect dataset. We can try and meet these assumptions by manipulatinfg the data, but somtimes we need to take a nonlinear approach.

References:

Germany Cars Dataset.(2021). Kaggle [Dataset]. https://www.kaggle.com/datasets/ander289386/cars-germany.

Rogel-Salazar, J. (2023). Statistics and data visualisation with python. CRC Press

https://www.kaggle.com/datasets/ander289386/cars-germany