Model For used car price prediction

In [2]:
```
import pandas as pd
import numpy as np
```

Stage 1

1.1 Reading the file

1.2 Exploratory data analysis (to check correlation)

1.3 Identifing Input and Output

1.1 Reading the file

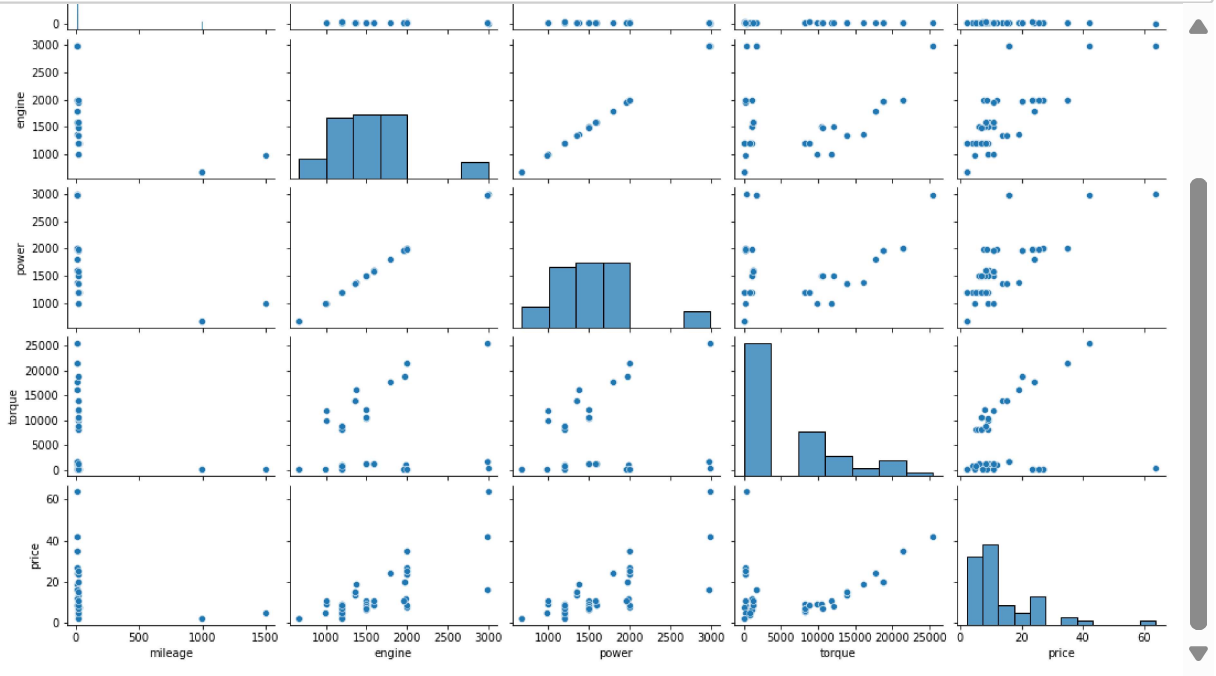
In [27]:
```
data=pd.read_csv(r"C:\Users\Param\Desktop\new used cars.csv")
print(data)
```

	mileage	engine	power	torque	price
0	7.81	2996	2996	333	63.75
1	17.40	999	999	9863	8.99
2	20.68	1995	1995	188	23.75
3	16.50	1353	1353	13808	13.56
4	14.67	1798	1798	17746	24.00
5	18.70	1199	1199	887	5.45
6	18.90	1197	1197	8186	5.12
7	15.80	1591	1591	1213	9.30
8	13.50	2987	2987	25479	42.00
9	17.00	1198	1198	1085	8.02
10	17.40	1497	1497	1176	10.95
11	16.42	1498	1498	10455	8.99
12	18.88	1995	1995	184	7.45
13	18.15	998	998	11835	10.95
14	18.90	1197	1197	8186	5.78
15	21.00	1197	1197	8180	8.95
16	18.88	1995	1995	184	8.50
17	22.69	1995	1995	190	23.50
18	17.00	1497	1497	12136	7.95
19	14.10	1368	1368	16077	18.95
20	12.40	1996	1996	21501	35.00
21	16.10	1197	1197	85	2.09
22	18.00	1497	1497	1173	6.25
23	998.00	671	671	90	2.12
24	18.53	1968	1968	18774	20.00
25	15.29	1591	1591	1213	8.35
26	21.40	1197	1197	831	5.03
27	17.10	1496	1496	10594	6.95
28	12.55	2982	2982	1685	16.00
29	16.80	1353	1353	13808	15.22
30	13.50	1999	1999	177	26.95
31	18.60	1197	1197	8186	6.70
32	17.10	1956	1956	170	10.75
33	20.68	1995	1995	188	25.50
34	22.56	1197	1197	8850	8.38
35	17.60	1582	1582	1262	10.90
36	17.80	1198	1198	867	3.95
37	26.49	1199	1199	72	7.25
38	1498.00	986	986	200	4.75
39	10.98	1984	1984	1144	11.95
40	12.40	1996	1996	21501	35.00
41	16.10	1197	1197	85	2.09
42	18.00	1497	1497	1173	6.25
43	998.00	671	671	90	2.12
44	18.53	1968	1968	18774	20.00
45	15.29	1591	1591	1213	8.35
46	21.40	1197	1197	831	5.03
47	17.10	1496	1496	10594	6.95
48	12.55	2982	2982	1685	16.00
49	16.80	1353	1353	13808	15.22
50	13.50	1999	1999	177	26.95
51	18.60	1197	1197	8186	6.70
52	17.10	1956	1956	170	10.75
53	20.68	1995	1995	188	25.50

```
54     22.56     1197    1197     8850    8.38
55     17.60     1582    1582     1262   10.90
```

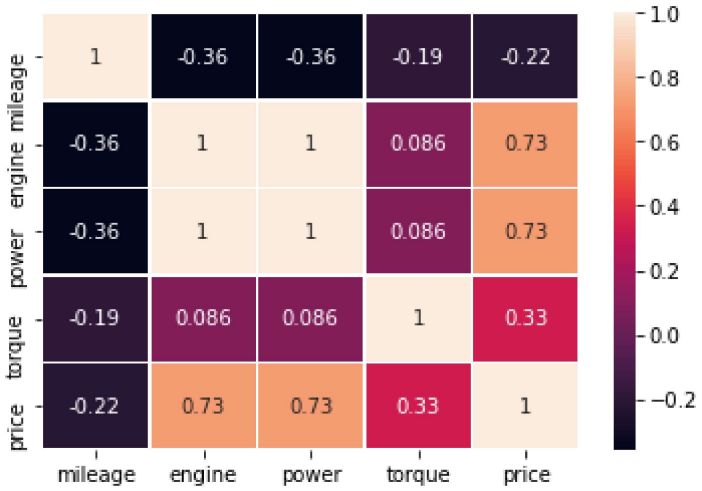
1.2 EDA (Exploratory Data analysis)

In [28]:
```
import seaborn as sns
sns.pairplot(data,height=2,aspect=1.5)
```



In [29]:
```
sns.heatmap(data.corr(), annot=True, linewidth=0.6)
```

Out[29]: <AxesSubplot:>



Stage 2: Splitting the Data

In [30]:
```
X=data.drop(['price','mileage'],axis='columns')
print(X)
Y=data.drop(['mileage','engine','power','torque'],axis='columns')
print(Y)
```

```
    engine  power  torque
0     2996   2996     333
1      999    999    9863
2     1995   1995     188
3     1353   1353   13808
4     1798   1798   17746
5     1199   1199     887
6     1197   1197    8186
7     1591   1591    1213
8     2987   2987   25479
9     1198   1198    1085
10    1497   1497    1176
11    1498   1498   10455
12    1995   1995     184
13     998    998   11835
14    1197   1197    8186
15    1197   1197    8180
16    1995   1995     184
17    1995   1995     190
```

Stage 3:Splitting

In [31]:
```
from sklearn.model_selection import train_test_split
X_train,X_test, Y_train, Y_test=train_test_split(X,Y,test_size=0.3,random_state
print(X_train)
print(X_test)
print(Y_train)
print(Y_test)
```

```
    engine  power  torque
46    1197   1197     831
17    1995   1995     190
54    1197   1197    8850
41    1197   1197      85
49    1353   1353   13808
12    1995   1995     184
30    1999   1999     177
39    1984   1984    1144
16    1995   1995     184
2     1995   1995     188
25    1591   1591    1213
19    1368   1368   16077
6     1197   1197    8186
14    1197   1197    8186
36    1198   1198     867
24    1968   1968   18774
5     1199   1199     887
38     986    986     200
```

Stage 3: Fitting of model or Equation

In [32]:
```
from sklearn.linear_model import LinearRegression
equation=LinearRegression()
```

In [33]:
```
equation.fit(X_train,Y_train)
```

Out[33]:
```
LinearRegression()
```

In [34]:
```
equation.intercept_
```

Out[34]:
```
array([-10.35620098])
```

In [35]:
```
equation.coef_
```

Out[35]:
```
array([[0.0061953 , 0.0061953 , 0.00057644]])
```

In [36]:
```
Y_test_predicted=equation.predict(X_test)
Y_test_predicted
```

Out[36]:
```
array([[ 9.97319504],
       [14.51464047],
       [14.47141888],
       [13.97780955],
       [ 9.19407068],
       [14.28692331],
       [26.95799426],
       [14.36773926],
       [ 7.70741826],
       [ 8.86869095],
       [27.5638701 ],
       [ 5.11317462],
       [ 8.87042026],
       [ 8.83176352],
       [14.28692331],
       [13.97780955],
       [-1.99022811]])
```

Stage 4: Evaluation of errors and R2

In [37]:
```
from sklearn import metrics
MAE=metrics.mean_absolute_error(Y_test, Y_test_predicted)
print(MAE)
```

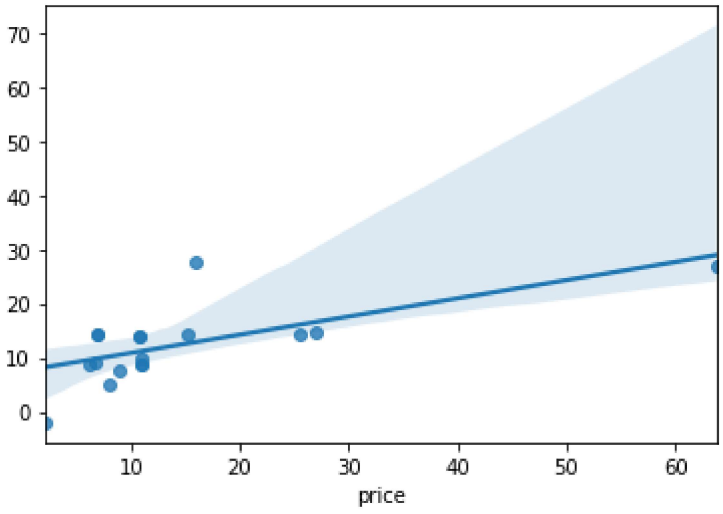
```
6.60815064654816
```

In [38]:
```
R2=metrics.r2_score(Y_test, Y_test_predicted)
R2
```

Out[38]:
```
0.4002569463316845
```

In [39]:
```
sns.regplot(x=Y_test, y=Y_test_predicted)
```

Out[39]:
```
<AxesSubplot:xlabel='price'>
```



In [ ]: