In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge,RidgeCV
from sklearn.linear_model import Lasso
from sklearn.preprocessing import StandardScaler
```

In [2]:

```
data=pd.read_csv(r"C:\Users\Prathyusha\Downloads\fiat500_VehicleSelection_Dataset.csv")
data
```

Out[2]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	
0	1	lounge	51	882	25000	1	44.907242	8.611
1	2	pop	51	1186	32500	1	45.666359	12.241
2	3	sport	74	4658	142228	1	45.503300	11.417
3	4	lounge	51	2739	160000	1	40.633171	17.634
4	5	pop	73	3074	106880	1	41.903221	12.495
...	...	...	...	...	...	...	...	
1533	1534	sport	51	3712	115280	1	45.069679	7.704
1534	1535	lounge	74	3835	112000	1	45.845692	8.666
1535	1536	pop	51	2223	60457	1	45.481541	9.413
1536	1537	lounge	51	2557	80750	1	45.000702	7.682
1537	1538	pop	51	1766	54276	1	40.323410	17.568

1538 rows × 9 columns

In [3]:

```
data.head(10)
```

Out[3]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon
0	1	lounge	51	882	25000	1	44.907242	8.611560
1	2	pop	51	1186	32500	1	45.666359	12.241890
2	3	sport	74	4658	142228	1	45.503300	11.417840
3	4	lounge	51	2739	160000	1	40.633171	17.634609
4	5	pop	73	3074	106880	1	41.903221	12.495650
5	6	pop	74	3623	70225	1	45.000702	7.682270
6	7	lounge	51	731	11600	1	44.907242	8.611560
7	8	lounge	51	1521	49076	1	41.903221	12.495650
8	9	sport	73	4049	76000	1	45.548000	11.549470
9	10	sport	51	3653	89000	1	45.438301	10.991700

In [4]:

```
data.tail(10)
```

Out[4]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	l
1528	1529	lounge	51	2861	126000	1	43.841980	10.515
1529	1530	lounge	51	731	22551	1	38.122070	13.361
1530	1531	lounge	51	670	29000	1	45.764648	8.994
1531	1532	sport	73	4505	127000	1	45.528511	9.593
1532	1533	pop	51	1917	52008	1	45.548000	11.549
1533	1534	sport	51	3712	115280	1	45.069679	7.704
1534	1535	lounge	74	3835	112000	1	45.845692	8.666
1535	1536	pop	51	2223	60457	1	45.481541	9.413
1536	1537	lounge	51	2557	80750	1	45.000702	7.682
1537	1538	pop	51	1766	54276	1	40.323410	17.568

In [5]:

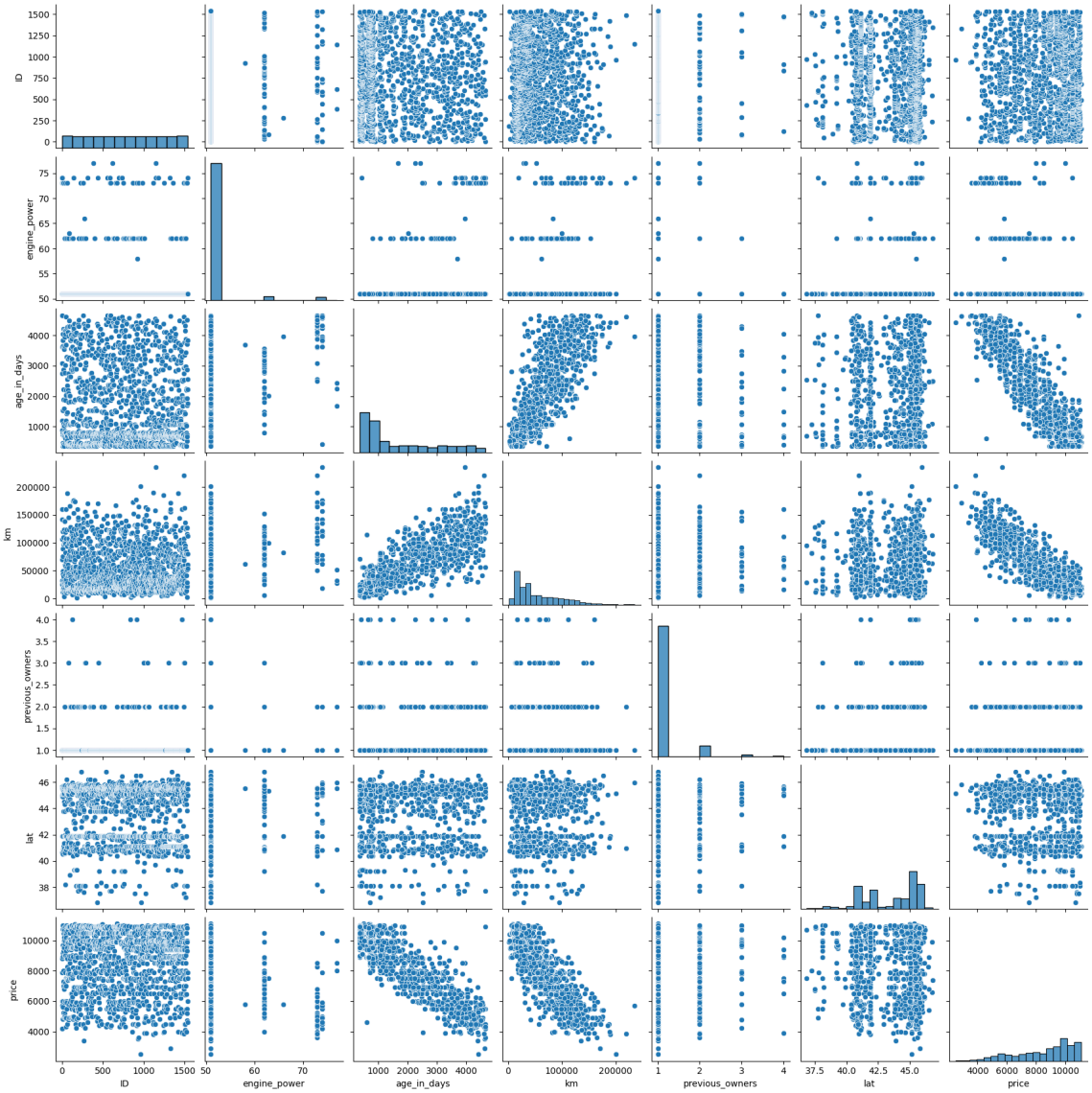
```
data.isnull().sum()
```

Out[5]:

```
ID                0
model             0
engine_power      0
age_in_days       0
km                0
previous_owners   0
lat               0
lon               0
price             0
dtype: int64
```

In [6]:

```
data.drop(columns=["lon","model"],inplace=True)
sns.pairplot(data)
data.price=np.log(data.price)
```



In [7]:

```
features=data.columns[0:2]
target=data.columns[-1]
x=data[features].values
y=data[target].values
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=17)
print("The dimenision of x_train is {}".format(x_train.shape))
print("The dimenision of x_test is {}".format(x_test.shape))
```

```
The dimenision of x_train is (1076, 2)
The dimenision of x_test is (462, 2)
```

In [8]:

```
lr=LinearRegression()
lr.fit(x_train,y_train)
actual=y_test
train_score_lr=lr.score(x_train,y_train)
test_score_lr=lr.score(x_test,y_test)
print("\n LinearRegression model \n")
print("The train score for lr model is {}".format(train_score_lr))
print("The test score for lr model is {}".format(test_score_lr))
```

```
 LinearRegression model

The train score for lr model is 0.07906758951709658
The test score for lr model is 0.08573839649638293
```

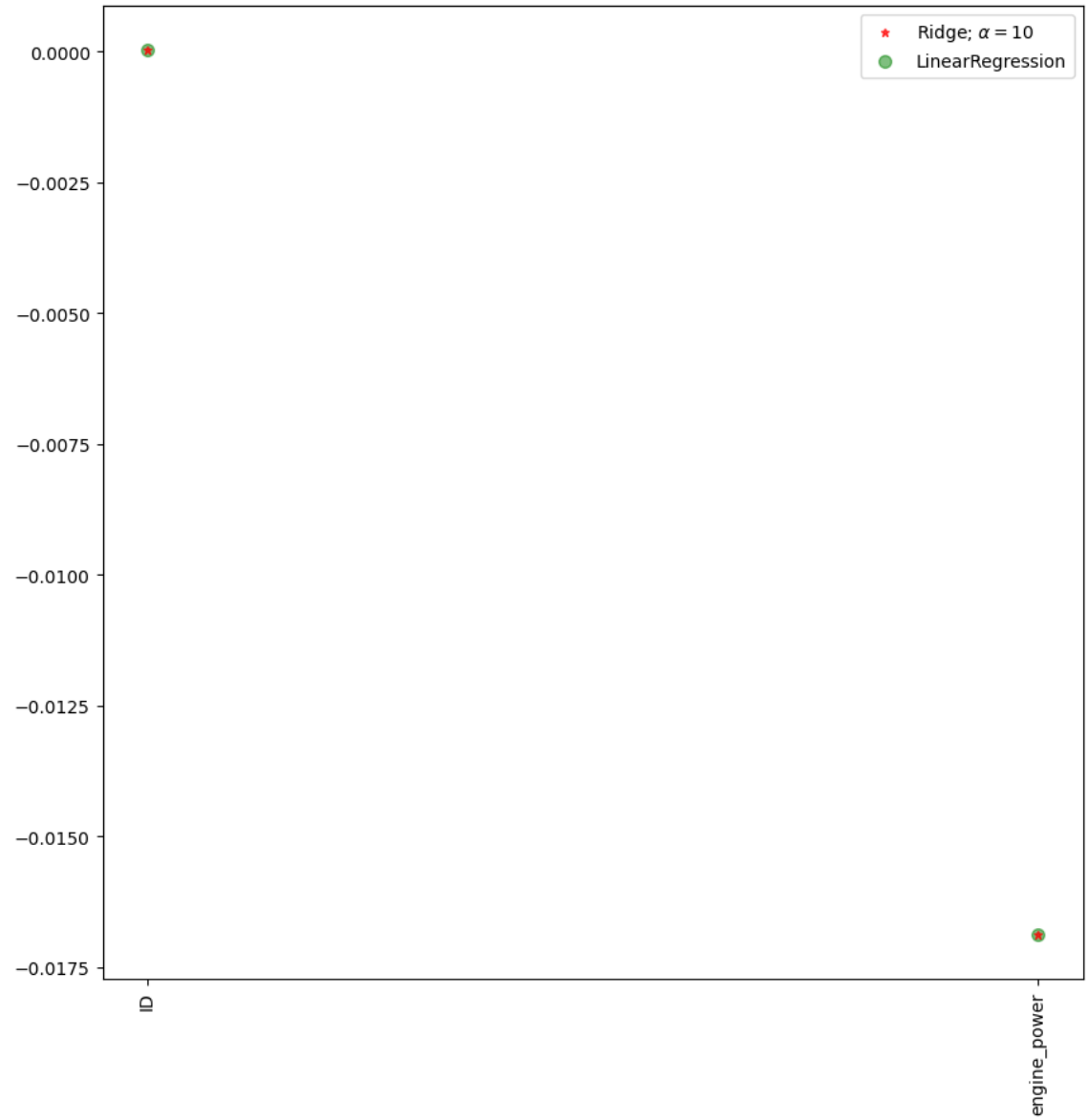
In [9]:

```
ridgeReg=Ridge(alpha=10)
ridgeReg.fit(x_train,y_train)
train_score_ridge=ridgeReg.score(x_train,y_train)
test_score_ridge=ridgeReg.score(x_test,y_test)
print("\nRidge Model:\n")
print("the train score for ridgemodel is {}".format(train_score_ridge))
print("the test score for ridgemodel is {}".format(train_score_ridge))
```

```
Ridge Model:

the train score for ridgemodel is 0.07906756718900732
the test score for ridgemodel is 0.07906756718900732
```

In [10]:

```
plt.figure(figsize=(10,10))
plt.plot(features,ridgeReg.coef_,alpha=0.7,linestyle='none',marker='*',markersize=5,colo
plt.plot(features,lr.coef_,alpha=0.5,linestyle='none',marker='o',markersize=7,color='gre
plt.xticks(rotation=90)
plt.legend()
plt.show()
```



In [11]:

```
print("\n Lasso model: \n")
lasso=Lasso(alpha=10)
lasso.fit(x_train,y_train)
train_score_ls=lasso.score(x_train,y_train)
test_score_ls=lasso.score(x_test,y_test)

print("the train score for is {}".format(train_score_ls))
print("the test score for is {}".format(train_score_ls))
```

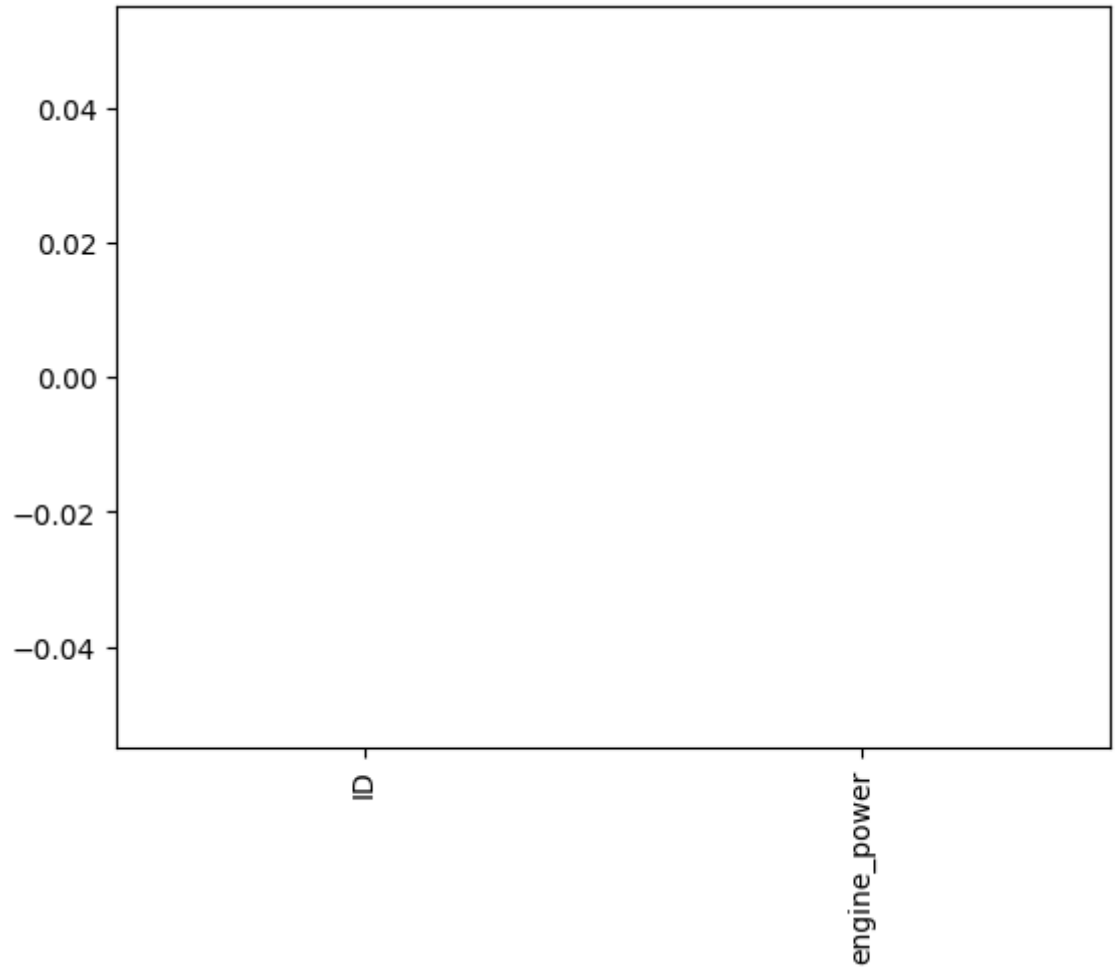
```
 Lasso model:

the train score for is 0.0
the test score for is 0.0
```

In [12]:

```
pd.Series(lasso.coef_,features).sort_values(ascending=True).plot(kind="bar")
```

Out[12]:

```
<Axes: >
```



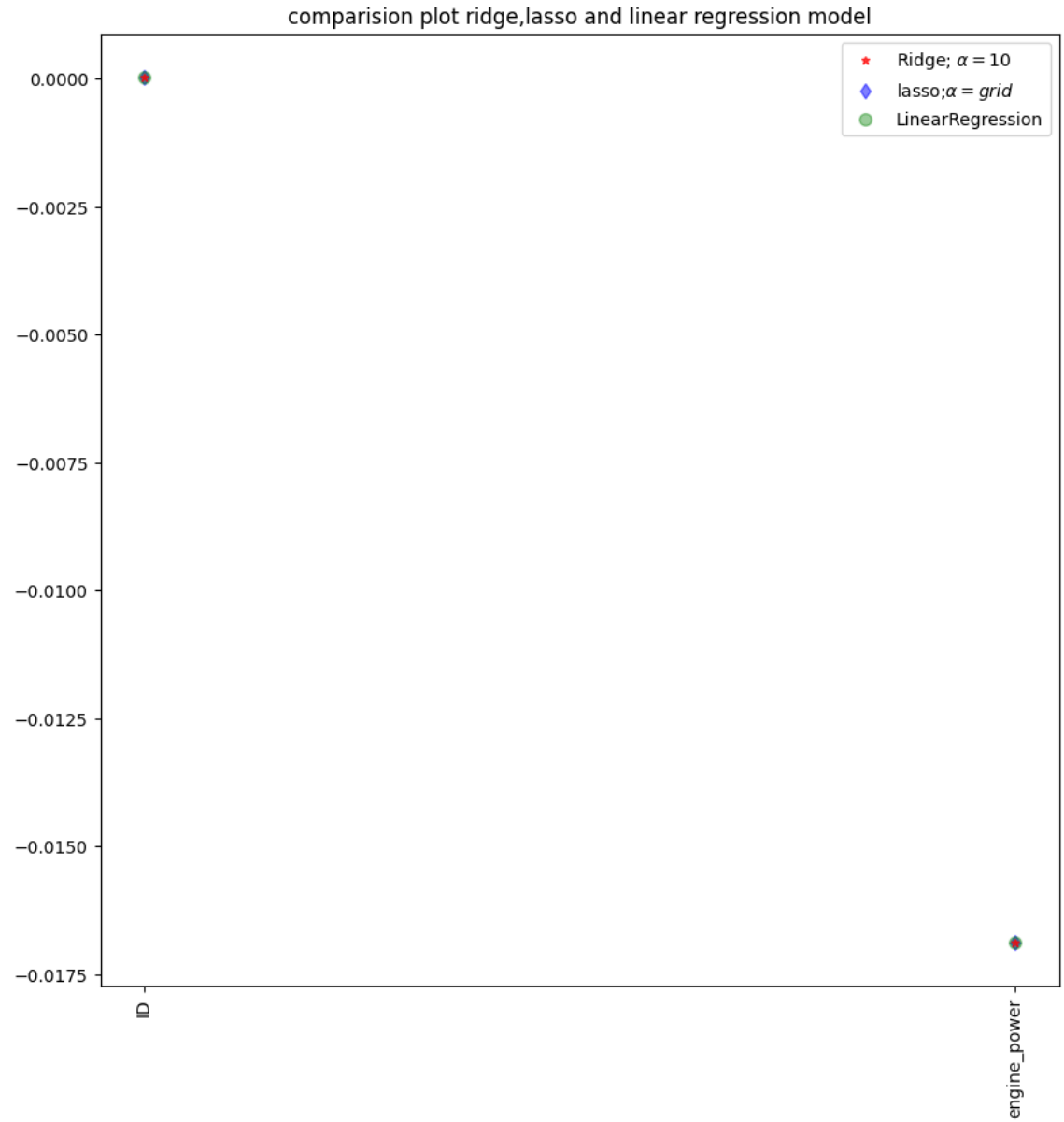
In [13]:

```
from sklearn.linear_model import LassoCV
lasso_cv=LassoCV(alphas=[0.0001,0.001,0.01,0.1,1,10],random_state=0).fit(x_train,y_train
print(lasso_cv.score(x_train,y_train))
print(lasso_cv.score(x_test,y_test))
```

```
0.07906758043914308
0.08572254951768365
```

In [14]:

```
plt.figure(figsize=(10,10))
plt.plot(features,ridgeReg.coef_,alpha=0.7,linestyle='none',marker='*',markersize=5,colo
plt.plot(lasso_cv.coef_,alpha=0.5,linestyle='none',marker='d',markersize=6,color='blue',
plt.plot(features,lr.coef_,alpha=0.4,linestyle='none',marker='o',markersize=7,color='gre
plt.xticks(rotation=90)
plt.legend()
plt.title("comparision plot ridge,lasso and linear regression model")
plt.show()
```



In [15]:

```
from sklearn.linear_model import RidgeCV
ridge_cv=RidgeCV(alphas=[0.0001,0.001,0.01,0.1,1,10]).fit(x_train,y_train)
print("the train score for ridgemodel is {}".format(ridge_cv.score(x_train,y_train)))
print("the test score for ridgemodel is {}".format(ridge_cv.score(x_train,y_train)))
```

```
the train score for ridgemodel is 0.07906756718900754
the test score for ridgemodel is 0.07906756718900754
```