In [1]:

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

In [2]:

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

Out[2]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	1	lounge	51	882	25000	1	44.907242	8.611560	8900
1	2	pop	51	1186	32500	1	45.666359	12.241890	8800
2	3	sport	74	4658	142228	1	45.503300	11.417840	4200
3	4	lounge	51	2739	160000	1	40.633171	17.634609	6000
4	5	pop	73	3074	106880	1	41.903221	12.495650	5700
...	...	...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	45.069679	7.704920	5200
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870	4600
1535	1536	pop	51	2223	60457	1	45.481541	9.413480	7500
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270	5990
1537	1538	pop	51	1766	54276	1	40.323410	17.568270	7900

1538 rows × 9 columns

In [3]:

```
df.columns
```

Out[3]:

```
Index(['ID', 'model', 'engine_power', 'age_in_days', 'km', 'previous_owners',
       'lat', 'lon', 'price'],
      dtype='object')
```

In [4]:

```
x=df[['ID','engine_power','age_in_days','km','previous_owners','lat','lon']]
y=df['price']
```

In [5]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
regr=LinearRegression()
regr.fit(x_train,y_train)
print(regr.score(x_test,y_test))
```

```
0.8442339248908175
```

In [6]:

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

Out[6]:

```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [7]:

```
coeff_df=pd.DataFrame(lm.coef_,x.columns,columns=['coefficient'])
coeff_df
```

Out[7]:

	coefficient
ID	-0.055110
engine_power	11.077477
age_in_days	-0.900816
km	-0.017422
previous_owners	7.287449
lat	37.713807
lon	5.681090

In [8]:

```
features = ['ID','engine_power','age_in_days','km','previous_owners',
 'lat','lon']
```

In [9]:

```
target=['price']
```

In [10]:

```
from sklearn.linear_model import Ridge,RidgeCV,Lasso
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

In [11]:

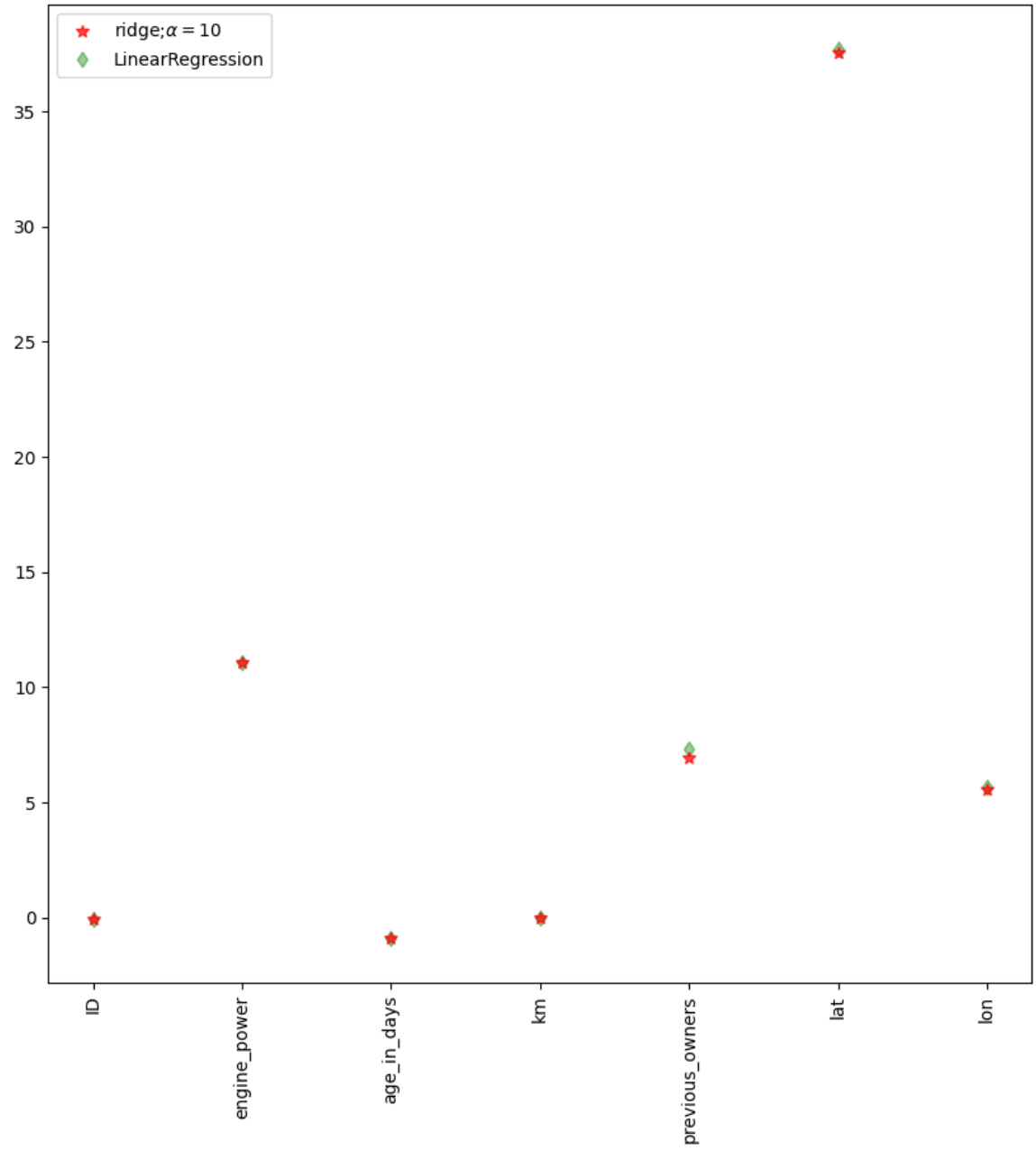
```
ridgeReg = Ridge(alpha=10)
ridgeReg.fit(x_train,y_train)
train_score_ridge = ridgeReg.score(x_train,y_train)
test_score_ridge = ridgeReg.score(x_test,y_test)
print('\nRidge model\n')
print('Train score for ridge model is {}'.format(train_score_ridge))
print('Test score for ridge model is {}'.format(test_score_ridge))
```

```
Ridge model

Train score for ridge model is 0.8421192197724451
Test score for ridge model is 0.8442304722432865
```

In [14]:

```
t.figure(figsize=(10,10))
t.plot(features,ridgeReg.coef_,alpha=0.7,linestyle='none',marker='*',markersize=7,color='Red',label=r'ridge;$\alpha=10$',zorder=7)
t.plot(features,lm.coef_,alpha=0.4,linestyle='none',color='green',marker='d',markersize=6,label='LinearRegression')
t.xticks(rotation=90)
t.legend()
t.show()
```



In [15]:

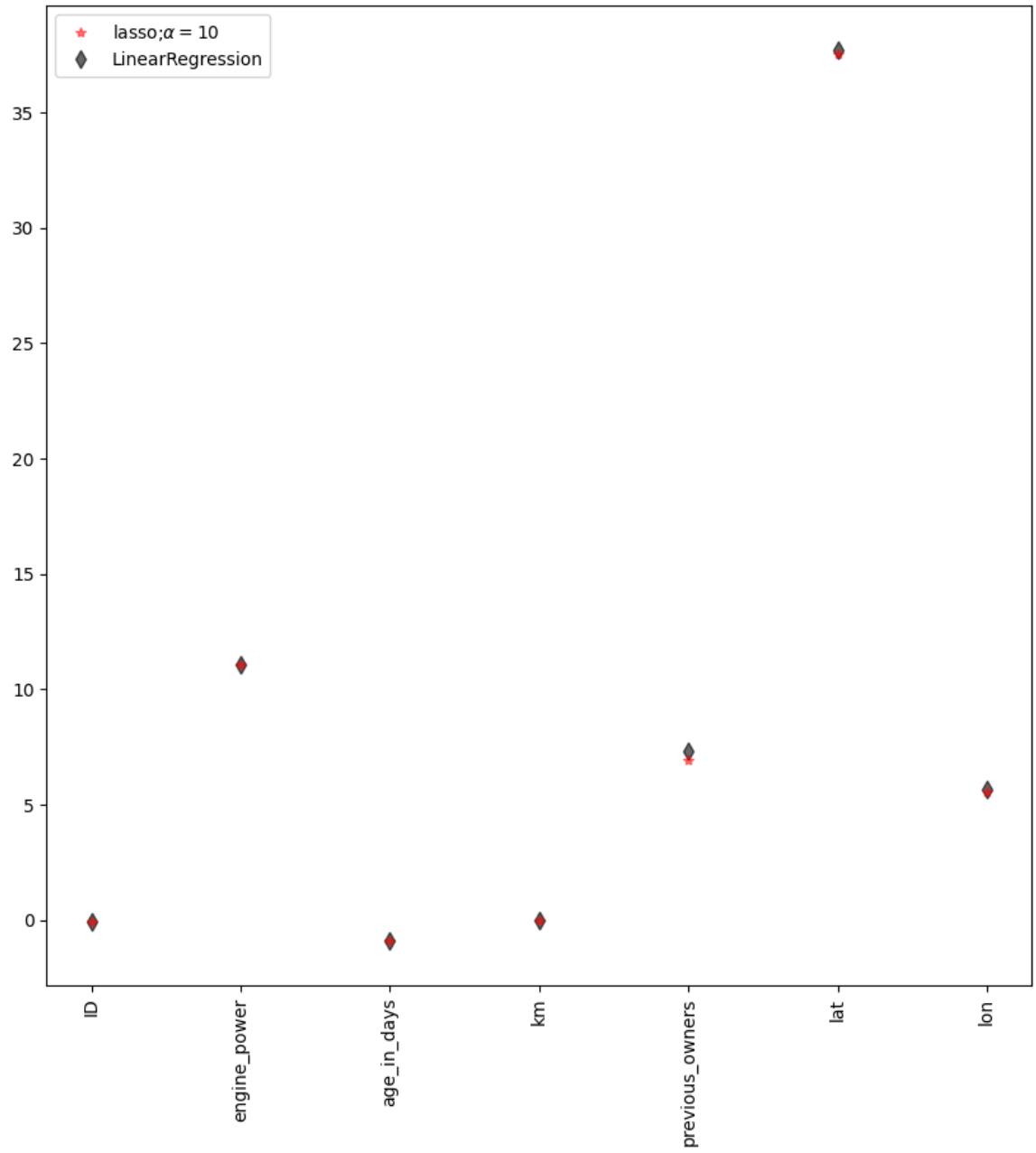
```
lassoReg=Lasso(alpha=10)
lassoReg.fit(x_train,y_train)
train_score_lasso=lassoReg.score(x_train,y_train)
test_score_lasso=lassoReg.score(x_test,y_test)
print('\nLasso Model\n')
print('Train score for lasso model is {}'.format(train_score_lasso))
print('Test score for lasso model is {}'.format(test_score_lasso))
```

```
Lasso Model

Train score for lasso model is 0.8420908335130204
Test score for lasso model is 0.8441608351601755
```

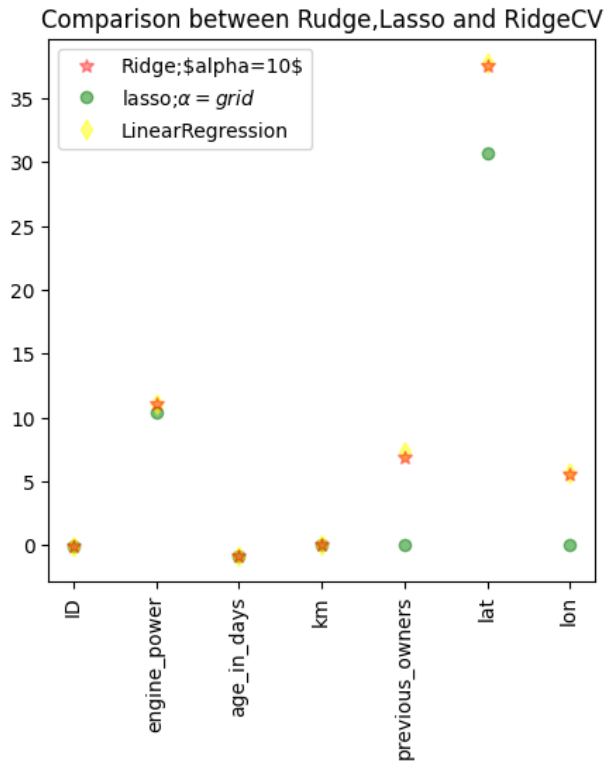
In [16]:

```
t.figure(figsize=(10,10))
t.plot(features,ridgeReg.coef_,alpha=0.5,linestyle='none',marker='*',markersize=6,color='red',label=r'lasso;$\alpha=10$',zorder=7)
t.plot(features,lm.coef_,alpha=0.6,linestyle='none',marker='d',markersize=7,color='k',label='LinearRegression')
t.xticks(rotation=90)
t.legend()
t.show()
```



In [17]:

```
comaprison between Ridge,Lasso and ridgeCV
t.figure(figsize=(5,5))
t.plot(features,ridgeReg.coef_,alpha=0.4,linestyle='none',marker='*',markersize=7,color='red',label=r'Ridge;\$alpha=10$',zorder=7)
t.plot(features,lassoReg.coef_,alpha=0.5,linestyle='none',marker='o',markersize=6,color='green',label=r'lasso;$\alpha=grid$')
t.plot(features,lm.coef_,alpha=0.5,linestyle='none',marker='d',markersize=7,color='yellow',label='LinearRegression')
t.xticks(rotation=90)
t.title('Comparison between Rudge,Lasso and RidgeCV')
t.legend()
t.show()
```



In [18]:

```
# Linear CV model using Ridge
from sklearn.linear_model import RidgeCV
ridge_CV=RidgeCV(alphas=[0.1,0.4,1.1]).fit(x_train,y_train)
print('The Train score for ridge model is {}'.format(ridge_CV.score(x_train,y_train)))
print('The Test score for ridge model is {}'.format(ridge_CV.score(x_test,y_test)))
```

```
The Train score for ridge model is 0.8421187226355313
The Test score for ridge model is 0.8441837688019272
```

In [19]:

```
# Linear CV model using Lasso
from sklearn.linear_model import LassoCV
lasso_CV=LassoCV(alphas=[1,10,20]).fit(x_train,y_train)
print("The train score for lasso model is {}".format(lasso_CV.score(x_train,y_train)))
print("The test score for lasso model is {}".format(lasso_CV.score(x_test,y_test)))
```

```
The train score for lasso model is 0.8420675579097567
The test score for lasso model is 0.8441521459676745
```

In [22]:

```
# Elstic Net
from sklearn.linear_model import ElasticNet
regr = ElasticNet()
regr.fit(x,y)
#print(regr,coef_)
print(regr.intercept_)
y_pred_elastic = regr.predict(x_train)
mean_squared_error = np.mean((y_pred_elastic-y_train)**2)
print('Mean squared error on test set',mean_squared_error)
```

```
9248.105972029809
Mean squared error on test set 593118.858587302
```