# 1)Advertising

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso,RidgeCV,Ridge
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

In [2]:
```
df=pd.read_csv(r"C:\Users\HP\Downloads\Advertising.csv")
df
```

Out[2]:

| | TV | Radio | Newspaper | Sales |
|---|---|---|---|---|
| **0** | 230.1 | 37.8 | 69.2 | 22.1 |
| **1** | 44.5 | 39.3 | 45.1 | 10.4 |
| **2** | 17.2 | 45.9 | 69.3 | 12.0 |
| **3** | 151.5 | 41.3 | 58.5 | 16.5 |
| **4** | 180.8 | 10.8 | 58.4 | 17.9 |
| **...** | ... | ... | ... | ... |
| **195** | 38.2 | 3.7 | 13.8 | 7.6 |
| **196** | 94.2 | 4.9 | 8.1 | 14.0 |
| **197** | 177.0 | 9.3 | 6.4 | 14.8 |
| **198** | 283.6 | 42.0 | 66.2 | 25.5 |
| **199** | 232.1 | 8.6 | 8.7 | 18.4 |

200 rows × 4 columns

In [3]:
```
df.head()
```

Out[3]:

| | TV | Radio | Newspaper | Sales |
|---|---|---|---|---|
| **0** | 230.1 | 37.8 | 69.2 | 22.1 |
| **1** | 44.5 | 39.3 | 45.1 | 10.4 |
| **2** | 17.2 | 45.9 | 69.3 | 12.0 |
| **3** | 151.5 | 41.3 | 58.5 | 16.5 |
| **4** | 180.8 | 10.8 | 58.4 | 17.9 |

In [4]:
```
df.tail()
```

Out[4]:

| | TV | Radio | Newspaper | Sales |
|---|---|---|---|---|
| **195** | 38.2 | 3.7 | 13.8 | 7.6 |
| **196** | 94.2 | 4.9 | 8.1 | 14.0 |
| **197** | 177.0 | 9.3 | 6.4 | 14.8 |
| **198** | 283.6 | 42.0 | 66.2 | 25.5 |
| **199** | 232.1 | 8.6 | 8.7 | 18.4 |

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

# Linear Regression

In [6]:
```
feature=df.columns[0:3]
target=df.columns[-1]
x=df[feature].values
y=df[target].values
```

In [7]:
```
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))
```

```
0.9094112577349186
```

# Ridge

In [8]:
```
ridgeReg=Lasso(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("\n Lasso Model \n")
print("train score for ridge model is {}".format(train_score_ridge))
print("test score for ridge model is {}".format(test_score_ridge))
```

```
 Lasso Model

train score for ridge model is 0.8809055630163277
test score for ridge model is 0.8750753072424735
```

# Lasso

In [9]:
```
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("\n Lasso 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 0.8809055630163277
test score for lasso model is 0.8750753072424735
```

# Lasso cross validation

In [10]:
```
from sklearn.linear_model import LassoCV
lasso_cv=LassoCV(alphas=[0.0001,0.001,0.01,0.1,1,10]).fit(x_train,y_train)
print("The train score for lasso model is {}".format(lasso_cv.score(x_train,y_train
print("The train score for lasso model is {}".format(lasso_cv.score(x_test,y_test))
```

```
The train score for lasso model is 0.8983224157890476
The train score for lasso model is 0.9094111361092692
```

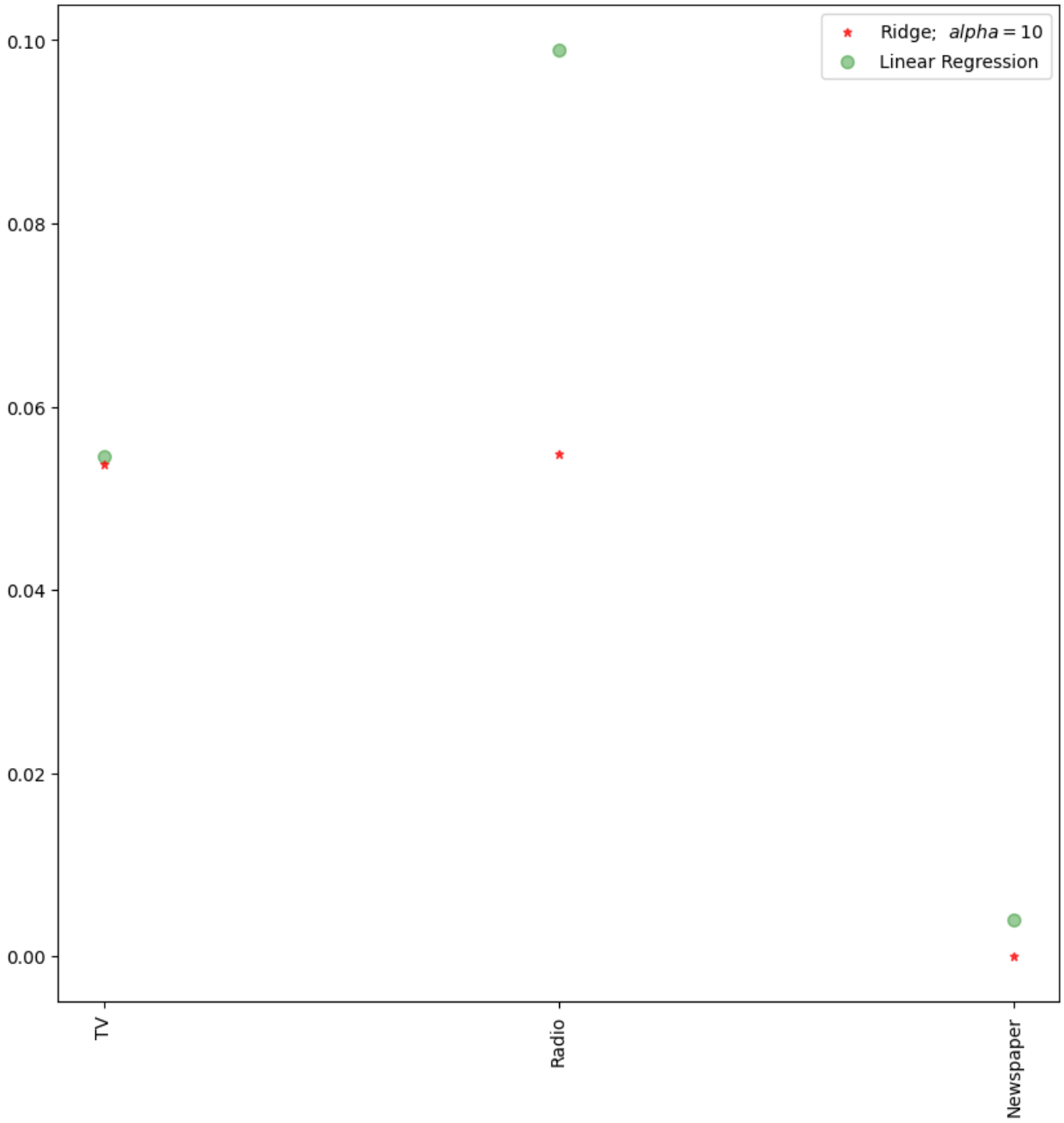
# Ridge cross validation

In [11]:
```
ridge_cv=RidgeCV(alphas=[0.0001,0.001,0.01,0.1,1,10]).fit(x_train,y_train)
print("The train score for ridge model is {}".format(ridge_cv.score(x_train,y_train
print("The train score for ridge model is {}".format(ridge_cv.score(x_test,y_test))
```

```
The train score for ridge model is 0.8983224075269427
The train score for ridge model is 0.9093961316810872
```

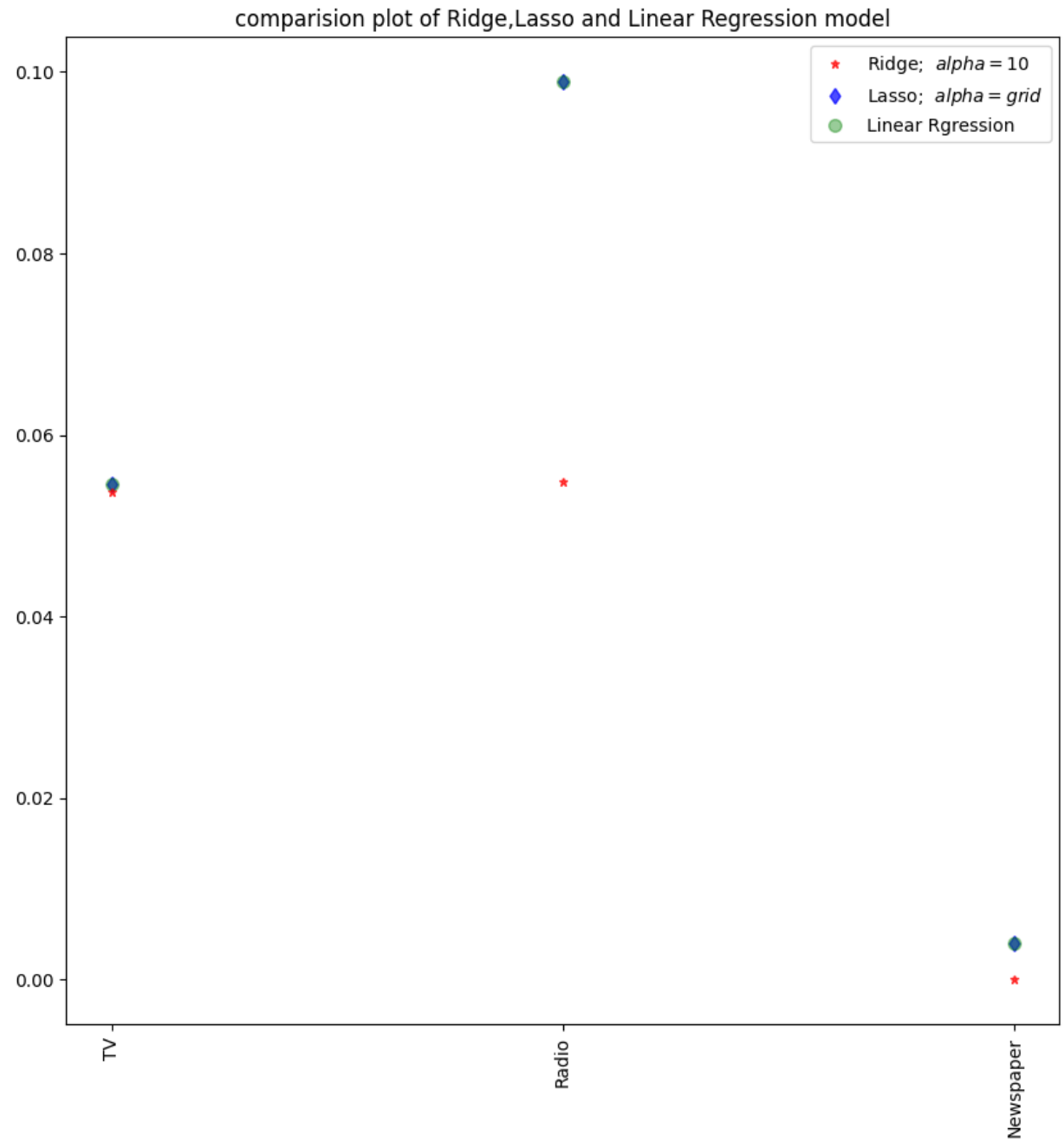
# Comparision b/w ridge & Linear Regression

In [12]:
```
plt.figure(figsize=(10,10))
plt.plot(feature,ridgeReg.coef_,alpha=0.7,linestyle='none',marker='*',markersize=5,
plt.plot(feature,regr.coef_,alpha=0.4,linestyle='none',marker='o',markersize=7,colo
plt.xticks(rotation=90)
plt.legend()
plt.show()
```




# Comparision between Ridge,Lasso & Linear Regression

In [13]:
```
plt.figure(figsize=(10,10))
plt.plot(feature,ridgeReg.coef_,alpha=0.7,linestyle='none',marker='*',markersize=5,
plt.plot(lasso_cv.coef_,alpha=0.7,linestyle='none',marker='d',markersize=6,color='b
plt.plot(feature,regr.coef_,alpha=0.4,linestyle='none',marker='o',markersize=7,colo
plt.xticks(rotation=90)
plt.legend()
plt.title("comparision plot of Ridge,Lasso and Linear Regression model")
plt.show()
```




## 2) vehicle selection

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso,RidgeCV,Ridge
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

In [2]:
```
df=pd.read_csv(r"C:\Users\HP\Downloads\fiat500_VehicleSelection_Dataset (5).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

# Linear Regression

In [5]:
```
feature=df.columns[3:6]
target=df.columns[-1]
x=df[feature].values
y=df[target].values
```

In [6]:
```
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))
```

```
0.8228774643074158
```

# Ridge

In [7]:
```
ridgeReg=Lasso(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("\n Lasso Model \n")
print("train score for ridge model is {}".format(train_score_ridge))
print("test score for ridge model is {}".format(test_score_ridge))
```

```
 Lasso Model

train score for ridge model is 0.8462884214607971
test score for ridge model is 0.8228443743079964
```

# Lasso

In [8]:
```
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("\n Lasso 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 0.8462884214607971
test score for lasso model is 0.8228443743079964
```

# Lasso cross validation

In [9]:
```
from sklearn.linear_model import LassoCV
lasso_cv=LassoCV(alphas=[0.0001,0.001,0.01,0.1,1,10]).fit(x_train,y_train)
print("The train score for lasso model is {}".format(lasso_cv.score(x_train,y_train
print("The train score for lasso model is {}".format(lasso_cv.score(x_test,y_test))
```

```
The train score for lasso model is 0.8462884214607971
The train score for lasso model is 0.8228443743079964
```

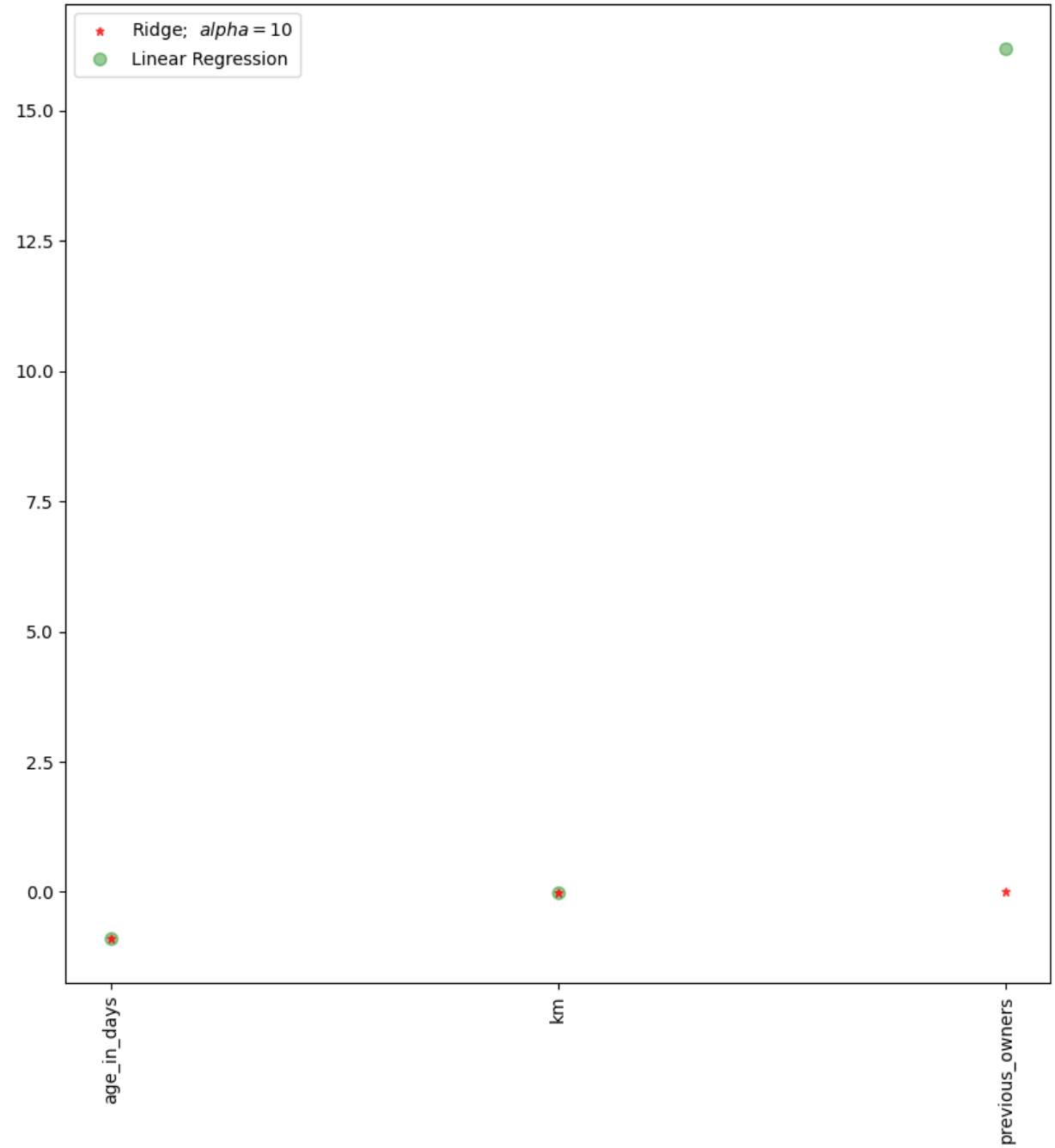
# Ridge cross validation

In [10]:
```
ridge_cv=RidgeCV(alphas=[0.0001,0.001,0.01,0.1,1,10]).fit(x_train,y_train)
print("The train score for ridge model is {}".format(ridge_cv.score(x_train,y_train
print("The train score for ridge model is {}".format(ridge_cv.score(x_test,y_test))
```

```
The train score for ridge model is 0.8463001992263106
The train score for ridge model is 0.8228741469736836
```

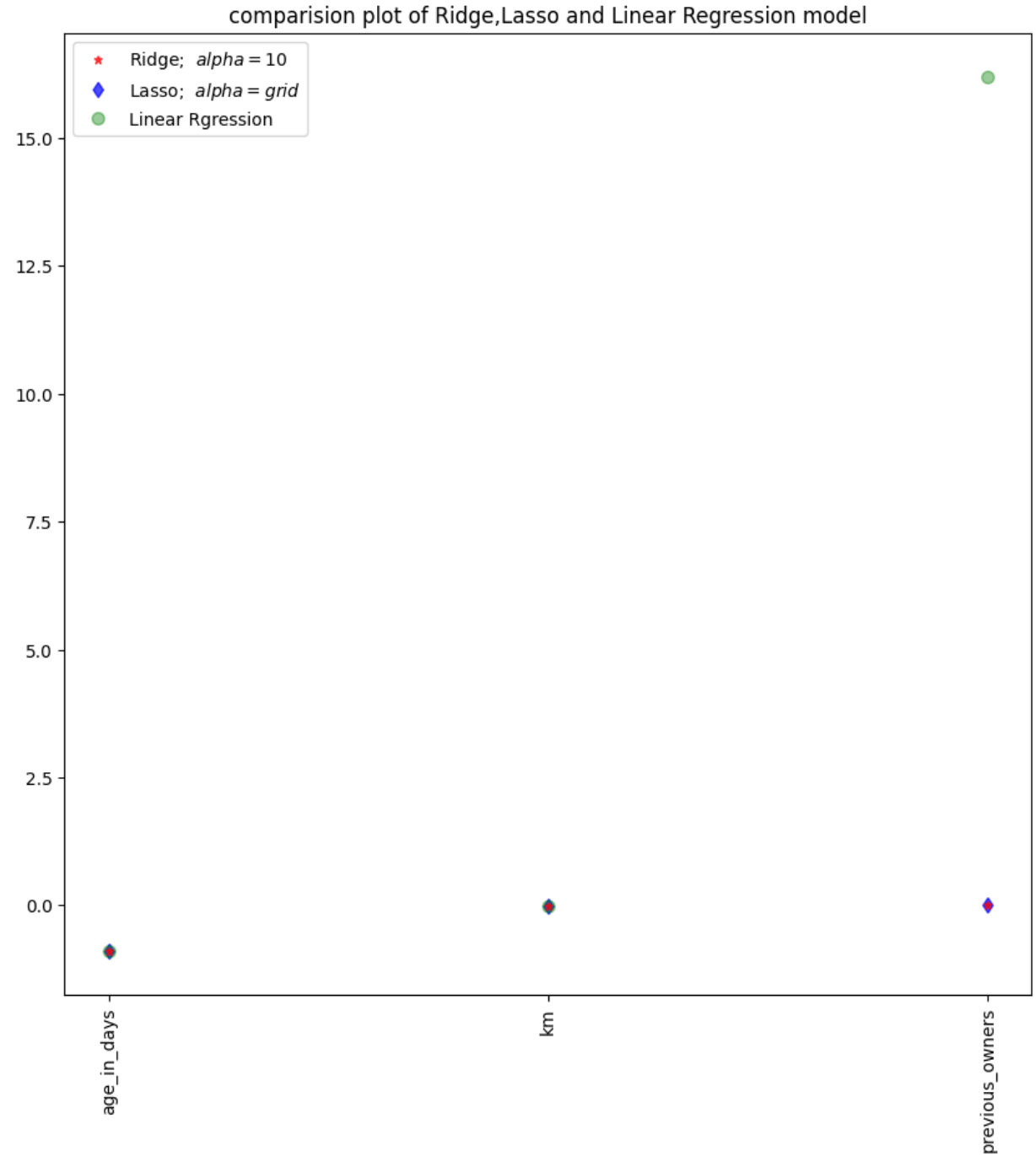
# comparision b/w Ridge & Linear Regression

In [11]:
```
plt.figure(figsize=(10,10))
plt.plot(feature,ridgeReg.coef_,alpha=0.7,linestyle='none',marker='*',markersize=5,
plt.plot(feature,regr.coef_,alpha=0.4,linestyle='none',marker='o',markersize=7,colo
plt.xticks(rotation=90)
plt.legend()
plt.show()
```




# comparision b/w Ridge,Lasso & Linear

In [12]:
```
plt.figure(figsize=(10,10))
plt.plot(feature,ridgeReg.coef_,alpha=0.7,linestyle='none',marker='*',markersize=5,
plt.plot(lasso_cv.coef_,alpha=0.7,linestyle='none',marker='d',markersize=6,color='b
plt.plot(feature,regr.coef_,alpha=0.4,linestyle='none',marker='o',markersize=7,colo
plt.xticks(rotation=90)
plt.legend()
plt.title("comparision plot of Ridge,Lasso and Linear Regression model")
plt.show()
```



In [ ]: