In [1]:
```
import pandas as pd
df = pd.read_csv('Housing.csv')
df
```

Out[1]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterh
0	13300000	7420	4	2	3	yes	no	no	
1	12250000	8960	4	4	4	yes	no	no	
2	12250000	9960	3	2	2	yes	no	yes	
3	12215000	7500	4	2	2	yes	no	yes	
4	11410000	7420	4	1	2	yes	yes	yes	
...	...	...	...	...	...	...	...	...	
540	1820000	3000	2	1	1	yes	no	yes	
541	1767150	2400	3	1	1	no	no	no	
542	1750000	3620	2	1	1	yes	no	no	
543	1750000	2910	3	1	1	no	no	no	
544	1750000	3850	3	1	2	yes	no	no	

545 rows × 13 columns

In [3]:
```
df['mainroad'] = df['mainroad'].astype('category')
df['mainroad'] = df['mainroad'].cat.codes
df
```

Out[3]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterh
0	13300000	7420	4	2	3	1	no	no	
1	12250000	8960	4	4	4	1	no	no	
2	12250000	9960	3	2	2	1	no	yes	
3	12215000	7500	4	2	2	1	no	yes	
4	11410000	7420	4	1	2	1	yes	yes	
...	...	...	...	...	...	...	...	...	
540	1820000	3000	2	1	1	1	no	yes	
541	1767150	2400	3	1	1	0	no	no	
542	1750000	3620	2	1	1	1	no	no	
543	1750000	2910	3	1	1	0	no	no	
544	1750000	3850	3	1	2	1	no	no	

545 rows × 13 columns

In [4]:
```
df['guestroom'] = df['guestroom'].astype('category')
df['guestroom'] = df['guestroom'].cat.codes

#basement
df['basement'] = df['basement'].astype('category')
df['basement'] = df['basement'].cat.codes

#hotwaterheating
df['hotwaterheating'] = df['hotwaterheating'].astype('category')
df['hotwaterheating'] = df['hotwaterheating'].cat.codes

#airconditioning
df['airconditioning'] = df['airconditioning'].astype('category')
df['airconditioning'] = df['airconditioning'].cat.codes

#prefarea
df['prefarea'] = df['prefarea'].astype('category')
df['prefarea'] = df['prefarea'].cat.codes

#furnishingstatus
df['furnishingstatus'] = df['furnishingstatus'].astype('category')
df['furnishingstatus'] = df['furnishingstatus'].cat.codes

df
```

Out[4]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterh
0	13300000	7420	4	2	3	1	0	0	
1	12250000	8960	4	4	4	1	0	0	
2	12250000	9960	3	2	2	1	0	1	
3	12215000	7500	4	2	2	1	0	1	
4	11410000	7420	4	1	2	1	1	1	
...	...	...	...	...	...	...	...	...	
540	1820000	3000	2	1	1	1	0	1	
541	1767150	2400	3	1	1	0	0	0	
542	1750000	3620	2	1	1	1	0	0	
543	1750000	2910	3	1	1	0	0	0	
544	1750000	3850	3	1	2	1	0	0	

545 rows × 13 columns

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
price             0
area              0
bedrooms          0
bathrooms         0
stories           0
mainroad          0
guestroom         0
basement          0
hotwaterheating   0
airconditioning   0
parking           0
prefarea          0
furnishingstatus  0
dtype: int64
```

Predicting Price from all

```
price is Dependent(Y) Feature, remaining are Independent (x1,x2,x3..)
y = m1x1 + m2x2 + m3x3 + ..... + c
```

In [13]:
```
x = df.drop(columns = 'price')
x
# training features
```

Out[13]:

	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	air
0	7420	4	2	3	1	0	0	0	
1	8960	4	4	4	1	0	0	0	
2	9960	3	2	2	1	0	1	0	
3	7500	4	2	2	1	0	1	0	
4	7420	4	1	2	1	1	1	0	
...	...	...	...	...	...	...	...	...	
540	3000	2	1	1	1	0	1	0	
541	2400	3	1	1	0	0	0	0	
542	3620	2	1	1	1	0	0	0	
543	2910	3	1	1	0	0	0	0	
544	3850	3	1	2	1	0	0	0	

545 rows × 12 columns

In [12]:
```
y = df['price']
y
```

Out[12]:
```
0      13300000
1      12250000
2      12250000
3      12215000
4      11410000
         ...
540     1820000
541     1767150
542     1750000
543     1750000
544     1750000
Name: price, Length: 545, dtype: int64
```

In [21]:
```
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_stat
```

In [22]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
```

In [23]:
```
lr.fit(x_train,y_train)
```

Out[23]:
```
LinearRegression
LinearRegression()
```

In [25]:
```
c = lr.intercept_
c
```

Out[25]:
```
107133.47334964667
```

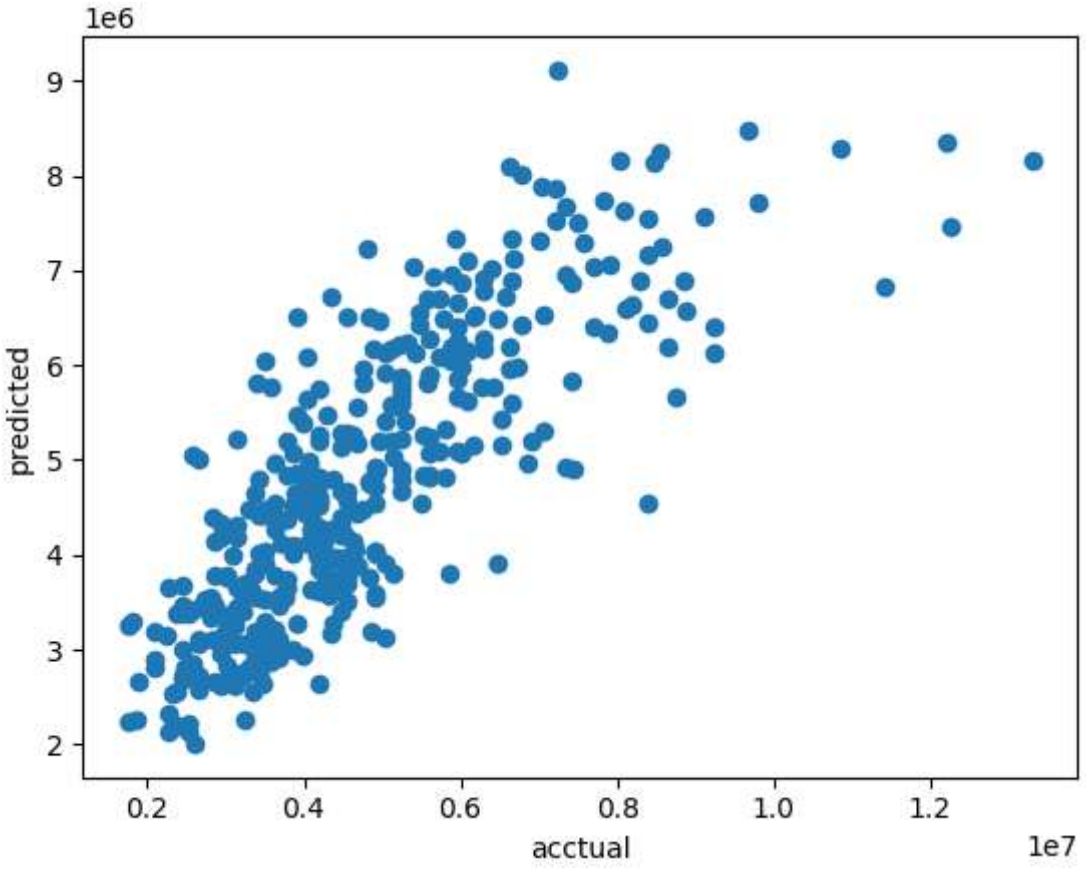
In [26]:
```
m = lr.coef_
m
```

Out[26]:
```
array([ 2.48857876e+02,  1.34994406e+05,  9.50583380e+05,  4.18321569e+05,
        4.66890751e+05,  3.68497644e+05,  3.59364424e+05,  1.24665331e+06,
        8.97037026e+05,  2.23301809e+05,  6.96754525e+05, -2.30222653e+05])
```

In [28]:
```
y_pred_train = lr.predict(x_train)
y_pred_train
```

```
6657156.22511678, 2660555.06766909, 3463986.85956226,
3653118.84497729, 3305527.03635915, 2865887.67529188,
5763928.85796843, 8087961.25015711, 2804763.6428699 ,
3983693.86582161, 6264431.61898825, 3545475.26518708,
4297067.4111761 , 9113766.77085733, 3811498.3627716 ,
3784615.58961515, 6083205.70458157, 4400989.86635754,
4960981.33359114, 6458300.22027978, 3540746.05272729,
7049254.85163629, 5159329.25927309, 3214119.9010676 ,
4865564.70787037, 4218834.55169852, 4153165.583687  ,
6948496.53063947, 2722348.11577067, 6042110.3257973 ,
7557402.30987993, 5023631.92081738, 2623475.24421273,
2137287.01629013, 5806375.5368455 , 5284551.79931901,
3949476.51538238, 4816433.79744079, 2820684.11944114,
5078614.8137583 , 4679398.83541737, 5776868.47673382,
3448371.95230411, 4488552.55526695, 6876809.98089264,
5985489.87956284, 4899366.98434045, 3909134.96918223,
3475842.58655745, 3629704.52899915, 3111839.80959965,
4244265.23137102, 6435182.16547809, 5480919.25827519,
7709538.96203166, 3006657.24744022, 5414990.96431643])
```

In [38]:
```
import matplotlib.pyplot as plt
plt.scatter(y_train,y_pred_train)
plt.xlabel("acctual")
plt.ylabel("predicted")
plt.show()
```



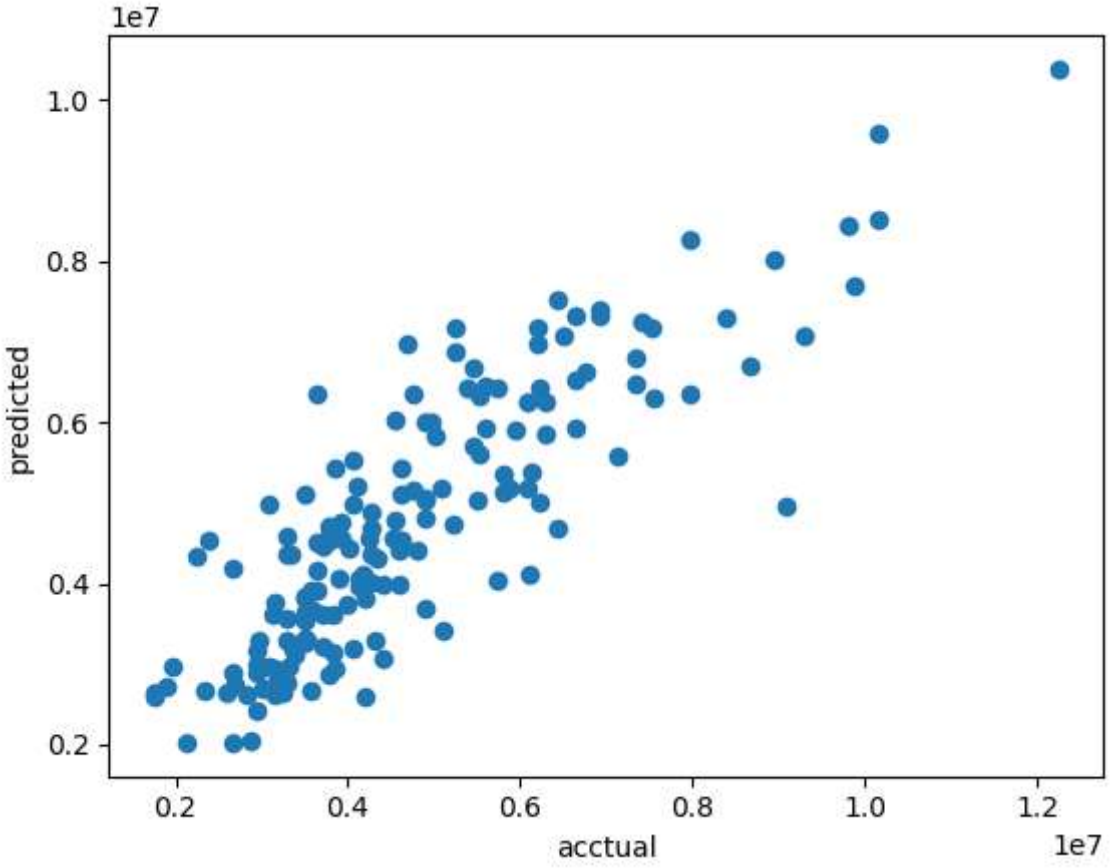
In [33]:
```
from sklearn.metrics import r2_score

r2_score(y_train,y_pred_train)
```

Out[33]: 0.6575703217254214

In [34]:
```
y_pred_test = lr.predict(x_test)
```

In [35]:
```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred_test)
plt.xlabel("acctual")
plt.ylabel("predicted")
plt.show()
```

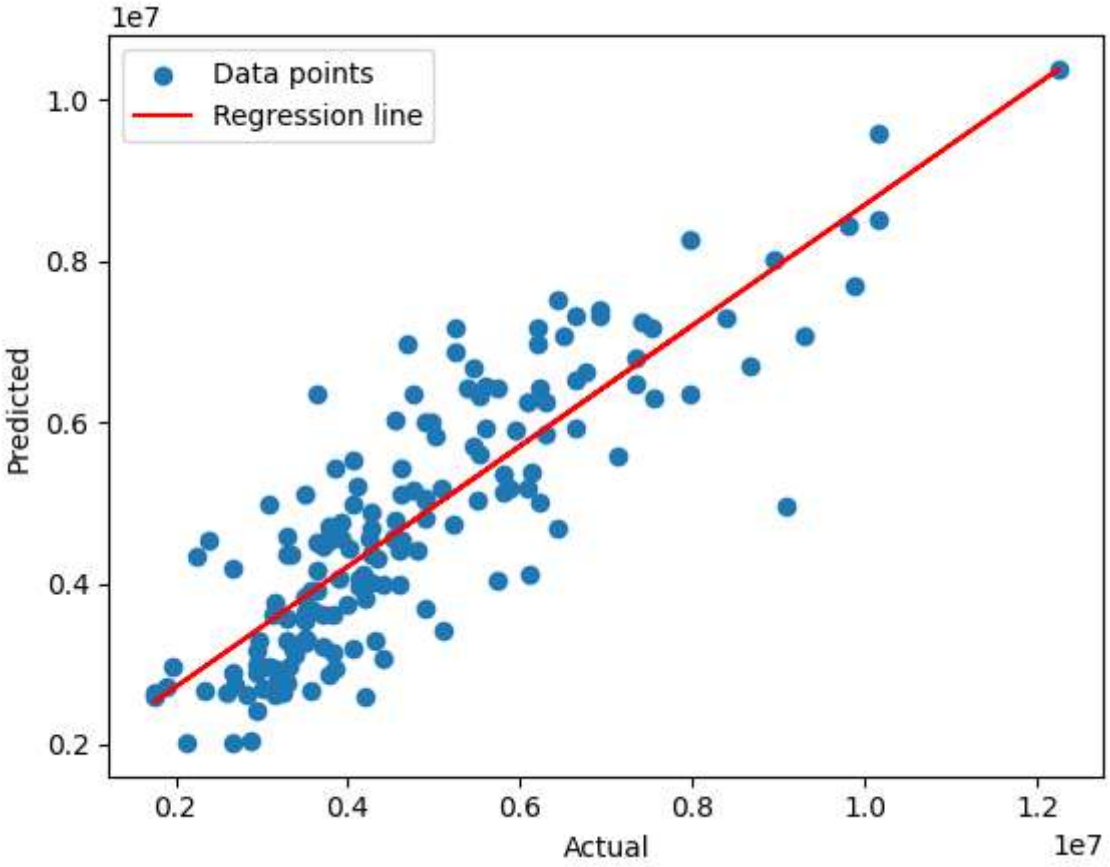


In [36]:
```
r2_score(y_test,y_pred_test)
```

Out[36]: 0.723501522320035

In [39]:
```
import matplotlib.pyplot as plt
import numpy as np
plt.scatter(y_test, y_pred_test, label='Data points')
coefficients = np.polyfit(y_test, y_pred_test, 1)
regression_line = np.polyval(coefficients, y_test)
plt.plot(y_test, regression_line, color='red', label='Regression line')

plt.xlabel("Actual")
plt.ylabel("Predicted")
plt.legend()
plt.show()
```



In [44]:
```
import numpy as np

# Take input for features
area = int(input("Enter the area :"))
bedrooms = int(input("Enter the number of bedrooms: "))
bathrooms = int(input("Enter the number of bathrooms: "))
stories = int(input("Enter the number of stories: "))
mainroad = input("Is the house on the main road? (yes/no): ").lower() == 'yes'
guestroom = input("Does the house have a guest room? (yes/no): ").lower() == '
basement = input("Does the house have a basement? (yes/no): ").lower() == 'yes
hotwaterheating = input("Does the house have hot water heating? (yes/no): ").l
airconditioning = input("Does the house have air conditioning? (yes/no): ").lo
parking = int(input("Enter the number of parking spaces: "))
prefarea = input("Is the house in a preferred area? (yes/no): ").lower() == 'y
furnishingstatus = input("Enter the furnishing status (furnished, semifurnishe

# Convert categorical features to numerical values
mainroad = 1 if mainroad else 0
guestroom = 1 if guestroom else 0
basement = 1 if basement else 0
hotwaterheating = 1 if hotwaterheating else 0
airconditioning = 1 if airconditioning else 0
prefarea = 1 if prefarea else 0

# Convert furnishingstatus to numerical value
furnishingstatus_map = {'furnished': 2, 'semifurnished': 1, 'unfurnished': 0}
furnishingstatus = furnishingstatus_map.get(furnishingstatus, -1)

# Make a prediction
input_data = np.array([[area,bedrooms, bathrooms, stories, mainroad, guestroom
predicted_price = lr.predict(input_data)

print(f"The predicted price for the house is: ${predicted_price[0]:,.2f}")
```

```
Enter the area :1000
Enter the number of bedrooms: 2
Enter the number of bathrooms: 2
Enter the number of stories: 1
Is the house on the main road? (yes/no): no
Does the house have a guest room? (yes/no): no
Does the house have a basement? (yes/no): no
Does the house have hot water heating? (yes/no): no
Does the house have air conditioning? (yes/no): no
Enter the number of parking spaces: 2
Is the house in a preferred area? (yes/no): yes
Enter the furnishing status (furnished, semifurnished, unfurnished): semifurn
ished
The predicted price for the house is: $3,858,603.98

C:\Users\peddi\AppData\Local\Programs\Python\Python311\Lib\site-packages\skle
arn\base.py:464: UserWarning: X does not have valid feature names, but Linear
Regression was fitted with feature names
  warnings.warn(
```