In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

In [2]:
```
df= pd.read_csv('insurance.csv')
```

In [3]:
```
df
```

Out[3]:

| | age | sex | bmi | children | smoker | region | charges |
|---|---|---|---|---|---|---|---|
| **0** | 19 | female | 27.900 | 0 | yes | southwest | 16884.92400 |
| **1** | 18 | male | 33.770 | 1 | no | southeast | 1725.55230 |
| **2** | 28 | male | 33.000 | 3 | no | southeast | 4449.46200 |
| **3** | 33 | male | 22.705 | 0 | no | northwest | 21984.47061 |
| **4** | 32 | male | 28.880 | 0 | no | northwest | 3866.85520 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **1333** | 50 | male | 30.970 | 3 | no | northwest | 10600.54830 |
| **1334** | 18 | female | 31.920 | 0 | no | northeast | 2205.98080 |
| **1335** | 18 | female | 36.850 | 0 | no | southeast | 1629.83350 |
| **1336** | 21 | female | 25.800 | 0 | no | southwest | 2007.94500 |
| **1337** | 61 | female | 29.070 | 0 | yes | northwest | 29141.36030 |

1338 rows × 7 columns

# converting categorical values to numeric values

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

In [5]:
```
df
```

Out[5]:

| | age | sex | bmi | children | smoker | region | charges |
|---|---|---|---|---|---|---|---|
| **0** | 19 | 0 | 27.900 | 0 | yes | southwest | 16884.92400 |
| **1** | 18 | 1 | 33.770 | 1 | no | southeast | 1725.55230 |
| **2** | 28 | 1 | 33.000 | 3 | no | southeast | 4449.46200 |
| **3** | 33 | 1 | 22.705 | 0 | no | northwest | 21984.47061 |
| **4** | 32 | 1 | 28.880 | 0 | no | northwest | 3866.85520 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **1333** | 50 | 1 | 30.970 | 3 | no | northwest | 10600.54830 |
| **1334** | 18 | 0 | 31.920 | 0 | no | northeast | 2205.98080 |
| **1335** | 18 | 0 | 36.850 | 0 | no | southeast | 1629.83350 |
| **1336** | 21 | 0 | 25.800 | 0 | no | southwest | 2007.94500 |
| **1337** | 61 | 0 | 29.070 | 0 | yes | northwest | 29141.36030 |

1338 rows × 7 columns

In [6]:
```
df['smoker'] = df['smoker'].astype('category')
df['smoker'] = df['smoker'].cat.codes
df['region'] = df['region'].astype('category')
df['region'] = df['region'].cat.codes
```

In [7]:
```
df
```

Out[7]:

| | age | sex | bmi | children | smoker | region | charges |
|---|---|---|---|---|---|---|---|
| **0** | 19 | 0 | 27.900 | 0 | 1 | 3 | 16884.92400 |
| **1** | 18 | 1 | 33.770 | 1 | 0 | 2 | 1725.55230 |
| **2** | 28 | 1 | 33.000 | 3 | 0 | 2 | 4449.46200 |
| **3** | 33 | 1 | 22.705 | 0 | 0 | 1 | 21984.47061 |
| **4** | 32 | 1 | 28.880 | 0 | 0 | 1 | 3866.85520 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **1333** | 50 | 1 | 30.970 | 3 | 0 | 1 | 10600.54830 |
| **1334** | 18 | 0 | 31.920 | 0 | 0 | 0 | 2205.98080 |
| **1335** | 18 | 0 | 36.850 | 0 | 0 | 2 | 1629.83350 |
| **1336** | 21 | 0 | 25.800 | 0 | 0 | 3 | 2007.94500 |
| **1337** | 61 | 0 | 29.070 | 0 | 1 | 1 | 29141.36030 |

1338 rows × 7 columns

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [14]:
```
x = df.drop(columns='charges')
```

In [15]:
```
x
```

Out[15]:

|  | age | sex | bmi | children | smoker | region |
|---|---|---|---|---|---|---|
| **0** | 19 | 0 | 27.900 | 0 | 1 | 3 |
| **1** | 18 | 1 | 33.770 | 1 | 0 | 2 |
| **2** | 28 | 1 | 33.000 | 3 | 0 | 2 |
| **3** | 33 | 1 | 22.705 | 0 | 0 | 1 |
| **4** | 32 | 1 | 28.880 | 0 | 0 | 1 |
| **...** | ... | ... | ... | ... | ... | ... |
| **1333** | 50 | 1 | 30.970 | 3 | 0 | 1 |
| **1334** | 18 | 0 | 31.920 | 0 | 0 | 0 |
| **1335** | 18 | 0 | 36.850 | 0 | 0 | 2 |
| **1336** | 21 | 0 | 25.800 | 0 | 0 | 3 |
| **1337** | 61 | 0 | 29.070 | 0 | 1 | 1 |

1338 rows × 6 columns

In [21]:
```
y = df['charges']
```

In [19]:
```
x
```

Out[19]:

| | age | sex | bmi | children | smoker | region |
|---|---|---|---|---|---|---|
| **0** | 19 | 0 | 27.900 | 0 | 1 | 3 |
| **1** | 18 | 1 | 33.770 | 1 | 0 | 2 |
| **2** | 28 | 1 | 33.000 | 3 | 0 | 2 |
| **3** | 33 | 1 | 22.705 | 0 | 0 | 1 |
| **4** | 32 | 1 | 28.880 | 0 | 0 | 1 |
| **...** | ... | ... | ... | ... | ... | ... |
| **1333** | 50 | 1 | 30.970 | 3 | 0 | 1 |
| **1334** | 18 | 0 | 31.920 | 0 | 0 | 0 |
| **1335** | 18 | 0 | 36.850 | 0 | 0 | 2 |
| **1336** | 21 | 0 | 25.800 | 0 | 0 | 3 |
| **1337** | 61 | 0 | 29.070 | 0 | 1 | 1 |

1338 rows × 6 columns

In [22]:
```
y
```

Out[22]:
```
0       16884.92400
1        1725.55230
2        4449.46200
3       21984.47061
4        3866.85520
           ...
1333    10600.54830
1334     2205.98080
1335     1629.83350
1336     2007.94500
1337    29141.36030
Name: charges, Length: 1338, dtype: float64
```

In [23]:
```
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.3, random_state = 
```

In [24]:
```
from sklearn.linear_model import LinearRegression
```

In [25]:
```
lr=  LinearRegression ()
```

In [26]:
```
lr.fit(xtrain,ytrain)
```

Out[26]:
```
▾ LinearRegression
LinearRegression()
```

In [27]:
```
c = lr.intercept_
```

```
In [28]: c
```

```
Out[28]: -11827.733141795678
```

```
In [29]: m = lr.coef_
```

```
In [30]: m
```

```
Out[30]: array([  256.5772619 ,   -49.39232379,   329.02381564,   479.08499828,
               23400.28378787,  -276.31576201])
```

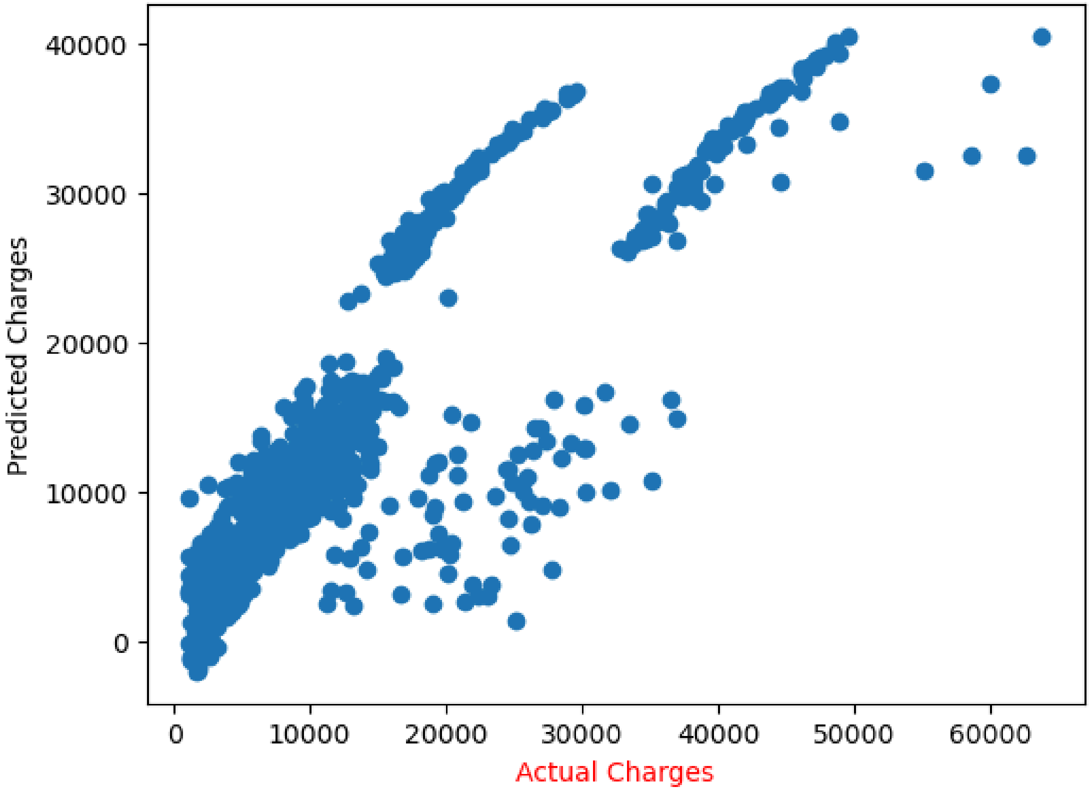
```
In [31]: # here in the above output , six coefficients for our six training features
```

```
In [32]: y_pred_train = lr.predict(xtrain)
```

```
In [33]: y_pred_train
```

```
         1080.53148314, 14633.39591662, 14243.30696952, 10577.01100155,
         3684.11562875, 29818.68446151, 32645.16083099,  6814.62074089,
        17990.69760509,  9501.45430502, 13096.05168663, 10131.77648372,
        10314.27461209,  3059.58352207, 10947.36144854,  9175.55641866,
         5794.96602322, 26970.5619884 , 10579.87748415,  5935.06772785,
        32505.66662638, 13113.41966313,  2006.18837709, 15066.19364927,
        10464.17148585,  7117.77806354,  2075.48695532,  5444.3979288 ,
        26309.98534611,   498.36583523,   -53.00017083, 31878.55790341,
          843.33151719, 13535.24091903,  6425.419617  ,  9657.95890588,
        36864.03879134,   937.11209112, 40474.11036389, 11641.54038524,
        14929.85130408, 33731.88094337, 10630.82488483,  4602.14460222,
         3285.00775314, 31519.53149568, 14184.46131254,  3881.66997242,
         2535.01446757, 13891.6193726 , 26794.30663013, 31193.2623816 ,
         7345.51499093,  6130.80303308, 13096.25777931, 12925.94625636,
        12765.64430125, 11624.73866694, 37091.3776533 , 10040.56284824,
        12742.06373425,  5025.10451419, 10661.09402373,  5348.7667013 ,
         7593.42274582, 28002.53724231,  5122.59866875, 13176.62524711,
        14542.75739319,   983.29805392, 23236.94762067,  3614.95052669,
        11139.64465512,  6085.25078246,  4411.74291809,  2376.39480234])
```

In [35]:
```
import matplotlib.pyplot as plt
plt.scatter(ytrain,y_pred_train)
#plt.xlabel('Actual Charges')
#plt.ylabel('Predicted Charges')
plt.xlabel('Actual Charges', color='red')
plt.ylabel('Predicted Charges', color='black')
plt.show()
```



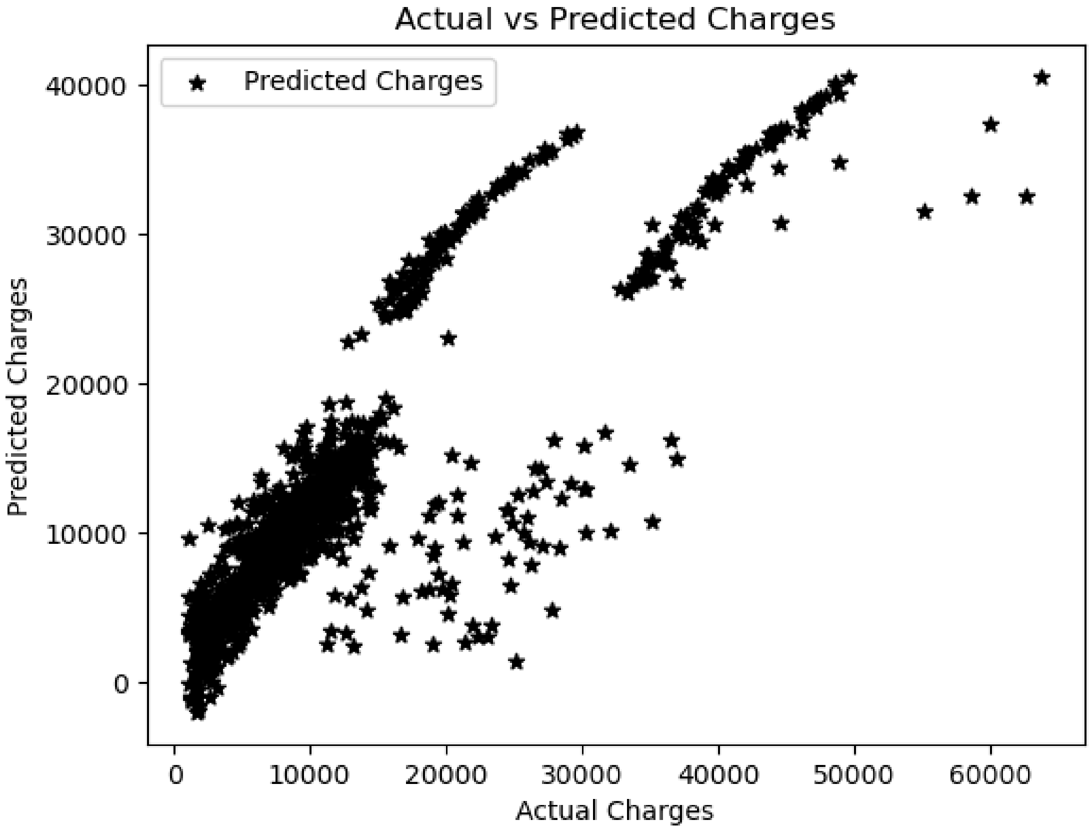
In [50]:
```
# Plot 'ytrain' using '*'
#plt.scatter(ytrain, ytrain, marker='*', label='Actual Charges', color='blue')

# Plot 'y_pred_train' using '+'
plt.scatter(ytrain, y_pred_train, marker='*', label='Predicted Charges', color='black'

plt.xlabel('Actual Charges')
plt.ylabel('Predicted Charges')
plt.title('Actual vs Predicted Charges')

# Add legend to differentiate between actual and predicted charges
plt.legend()

plt.show()
```



In [47]:
```
from sklearn.metrics import r2_score
r2_score (ytrain, y_pred_train)
```

Out[47]:
```
0.7306840408360218
```

In [48]:
```
y_pred_test = lr.predict(xtest)
```

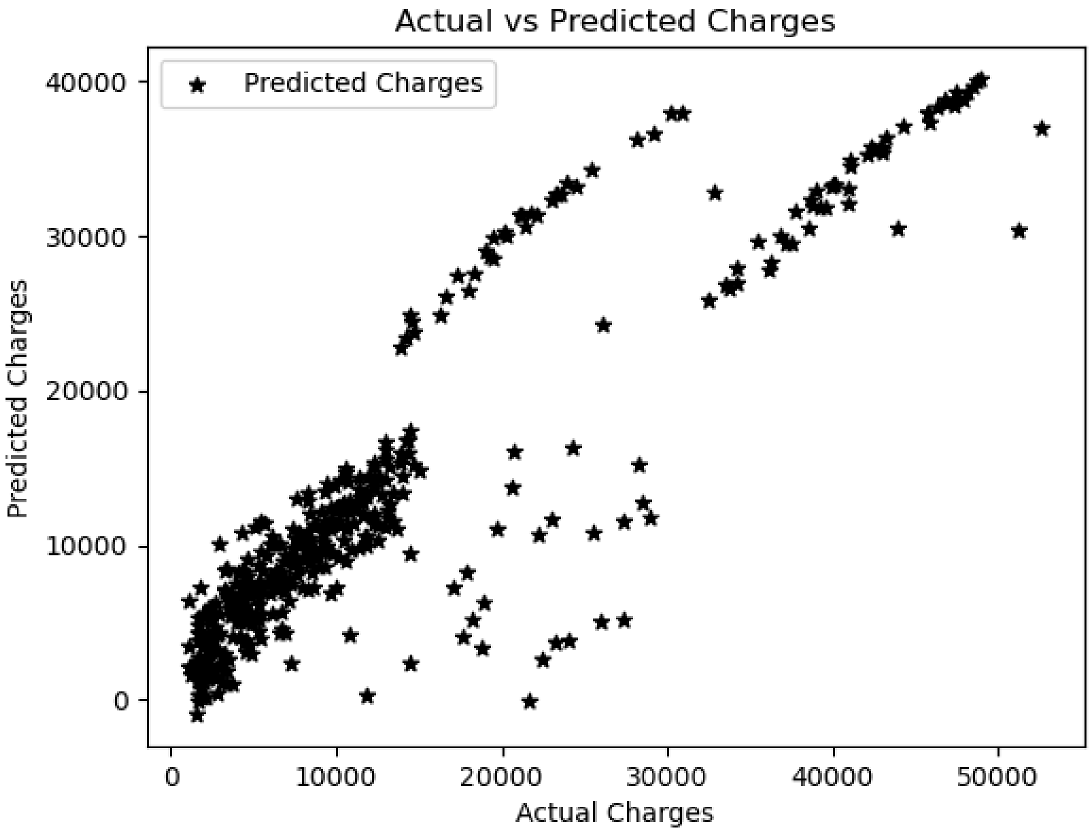
In [49]:
```
# Plot 'ytest' using '*'
#plt.scatter(ytest, ytest, marker='*', label='Actual Charges', color='blue')

# Plot 'y_pred_train' using '+'
plt.scatter(ytest, y_pred_test, marker='*', label='Predicted Charges', color='black')

plt.xlabel('Actual Charges')
plt.ylabel('Predicted Charges')
plt.title('Actual vs Predicted Charges')

# Add legend to differentiate between actual and predicted charges
plt.legend()

plt.show()
```



In [51]:
```
r2_score (ytest, y_pred_test)
```

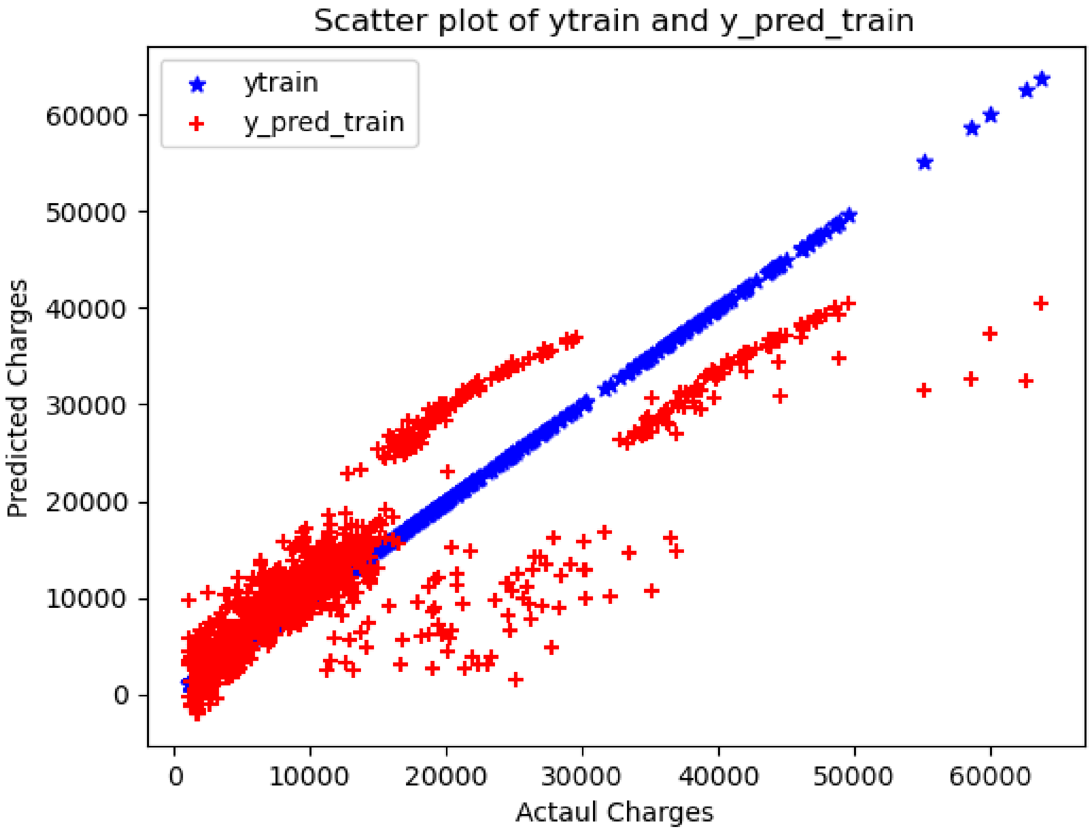
Out[51]:
```
0.7911113876316933
```

In [53]:

```
# Create scatter plot
plt.scatter(ytrain, ytrain, marker='*', label='ytrain', color='blue')       # ytrain w
plt.scatter(ytrain, y_pred_train, marker='+', label='y_pred_train', color='red')  # y_

# Add labels and legend
plt.xlabel('Actaul Charges')
plt.ylabel('Predicted Charges')
plt.title('Scatter plot of ytrain and y_pred_train')
plt.legend()

# Show plot
plt.show()
```

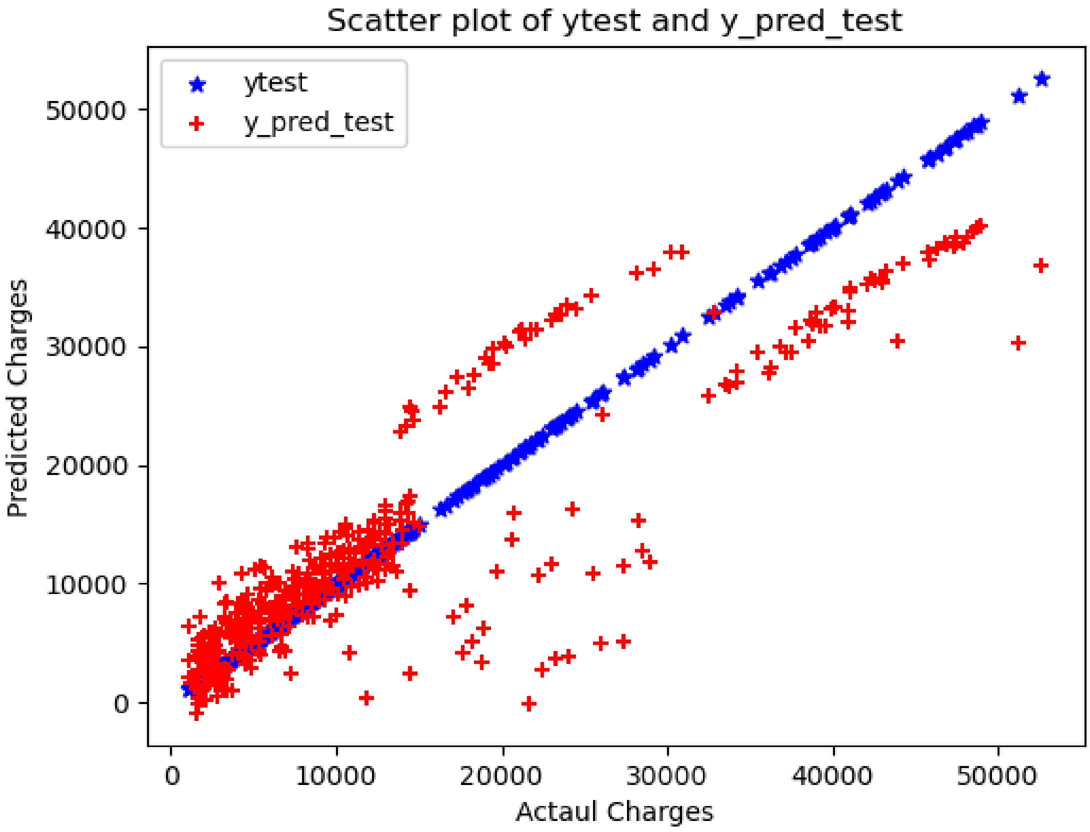


In [54]:

```
# Create scatter plot
plt.scatter(ytest, ytest, marker='*', label='ytest', color='blue')       # ytrain with
plt.scatter(ytest, y_pred_test, marker='+', label='y_pred_test', color='red')  # y_pre

# Add labels and legend
plt.xlabel('Actaul Charges')
plt.ylabel('Predicted Charges')
plt.title('Scatter plot of ytest and y_pred_test')
plt.legend()

# Show plot
plt.show()
```



In [ ]: