Load the dataset

In [1]:
```
import pandas as pd

# Load the dataset
df = pd.read_csv('dataset1.csv')
```

Understanding the Dataset

In [2]:
```
print(df.head())
```

```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

Checking Missig values

In [3]:
```
print(df.isnull().sum())
```

```
total_bill    0
tip           0
sex           0
smoker        0
day           0
time          0
size          0
dtype: int64
```

In [4]:
```
print(df.describe())
```

```
       total_bill         tip        size
count  244.000000  244.000000  244.000000
mean    19.785943    2.998279    2.569672
std      8.902412    1.383638    0.951100
min      3.070000    1.000000    1.000000
25%     13.347500    2.000000    2.000000
50%     17.795000    2.900000    2.000000
75%     24.127500    3.562500    3.000000
max     50.810000   10.000000    6.000000
```

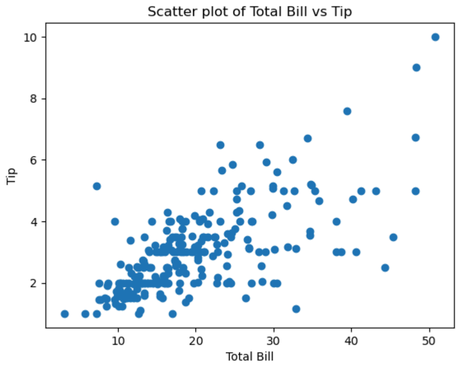
In [5]:
```
print("Columns in the DataFrame:", df.columns)
```

```
Columns in the DataFrame: Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'], dtype='object')
```

Data Visualization

In [6]:
```
import matplotlib.pyplot as plt

plt.scatter(df['total_bill'], df['tip'])
plt.xlabel('Total Bill')
plt.ylabel('Tip')
plt.title('Scatter plot of Total Bill vs Tip')
plt.show()
```



Model Building

In [7]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

X = df[['total_bill']]
Y = df['tip']


X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=0)

model = LinearRegression()
model.fit(X_train, Y_train)
```

Out[7]:
```
LinearRegression()
```

Model Evaluation

In [8]:
```
from sklearn.metrics import mean_squared_error, r2_score


Y_pred = model.predict(X_test)


mse = mean_squared_error(Y_test, Y_pred)
r2 = r2_score(Y_test, Y_pred)

print(f'Mean Squared Error: {mse}')
print(f'R-squared: {r2}')
```

```
Mean Squared Error: 0.821309064276629
R-squared: 0.5906895098589039
```

In [9]:
```
comparison_df = pd.DataFrame({'Actual': Y_test, 'Predicted': Y_pred})
print(comparison_df)
```

```
     Actual  Predicted
64     2.64   2.732195
63     3.76   2.799993
55     3.51   2.916217
111    1.00   1.730731
225    2.50   2.604349
92     1.00   1.585451
76     3.08   2.764157
181    5.65   3.288134
188    3.50   2.786433
180    3.68   4.384514
73     5.00   3.476998
107    4.29   3.470218
150    2.50   2.391271
198    2.00   2.287638
224    1.58   2.328317
44     5.60   3.972887
145    1.50   1.837270
110    3.00   2.384492
243    3.00   2.847451
189    4.00   3.265858
210    2.00   3.939957
104    4.08   3.054717
138    2.00   2.578198
8      1.96   2.485219
199    2.00   2.337033
203    2.50   2.616940
220    2.20   2.206281
125    4.20   3.914775
5      4.71   3.477966
22     2.23   2.555922
74     2.20   2.455195
124    2.52   2.237274
12     1.57   2.522023
168    1.61   2.054221
45     3.00   2.799993
158    2.61   2.325411
37     3.07   2.668272
136    2.00   2.029040
212    9.00   5.709469
223    3.00   2.576261
222    1.92   1.859546
118    1.80   2.232432
231    3.00   2.548174
155    5.14   3.919618
209    2.23   2.264393
18     3.50   2.672146
108    3.76   2.795150
15     3.92   3.118640
71     3.00   2.681832
```

In [ ]: