EXAMPLE

In [4]:
```python
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
```

In [6]:
```python
data=pd.read_csv('insurance.csv')
data
```

Out[6]:

	age	height	weight	premium
0	25	162.5	70	18000
1	30	172.7	95	38000
2	35	167.5	78	38000
3	40	NaN	110	60000
4	45	157.4	85	70000

In [11]:
```python
# Extract the features (independent variables) as a DataFrame
X = data[['age',  'weight']]  # Note the use of double square brackets [[]]

# Extract the target variable (dependent variable) as a Series
y = data['premium']

# Create a Linear Regression model
model = LinearRegression()

# Fit the model to the data
model.fit(X, y)
```

Out[11]:
```
▾ LinearRegression
LinearRegression()
```

In [15]:
```python
print("Coefficient (Slope):", model.coef_)
print("Intercept:", model.intercept_)
print("Predicted Value:", y_pred)
```

```
Coefficient (Slope): [2290.36842795  255.14619116]
Intercept: -57713.70132424285
Predicted Value: [5.8]
```

EXAMPLE

In [17]:
```python
import pandas as pd
import numpy as np

from sklearn.linear_model import LinearRegression
```

In [18]:
```python
data=pd.read_csv('insurance.csv')
data
```

Out[18]:

	age	height	weight	premium
0	25	162.5	70	18000
1	30	172.7	95	38000
2	35	167.5	78	38000
3	40	NaN	110	60000
4	45	157.4	85	70000

In [21]:
```python
mean_height = data['height'].mean()
data['height'] = data['height'].fillna(mean_height)
data['height']
```

Out[21]:
```
0    162.500
1    172.700
2    167.500
3    165.025
4    157.400
Name: height, dtype: float64
```

In [25]:
```python
X=data[['age','height','weight']]
y=data['premium']
```

In [26]:
```python
model = LinearRegression()

# Fit the model to the data
model.fit(X, y)
```

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

In [38]:
```python
y_pred = model.predict(X)
```

In [39]:
```python
y_pred
```

Out[39]:
```
array([18399.0846898 , 34480.9388677 , 41207.79358372, 62545.25747025,
       67366.92538853])
```

In [40]:
```python
print("Coefficient (Slope):", model.coef_)
print("Intercept:", model.intercept_)
```

```
Coefficient (Slope): [2152.80168625 -242.52223936  311.66290353]
Intercept: -17827.496816556588
```

EXAMPLE

In [70]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn .linear_model import LinearRegression
```

In [59]:
```python
data=pd.read_csv('Salary_Data[1].csv')
data
```

Out[59]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

In [60]:
```python
X=data[['YearsExperience']]
y=data['Salary']
```

In [61]:
```python
model = LinearRegression()

# Fit the model to the data
model.fit(X, y)
```

Out[61]:
```
▾ LinearRegression
LinearRegression()
```

In [62]:
```python
y_pred = model.predict(X)
```

In [66]:
```python
print("coffeficent(slope):", model.coef_)
```

```
coffeficent(slope): [9449.96232146]
```

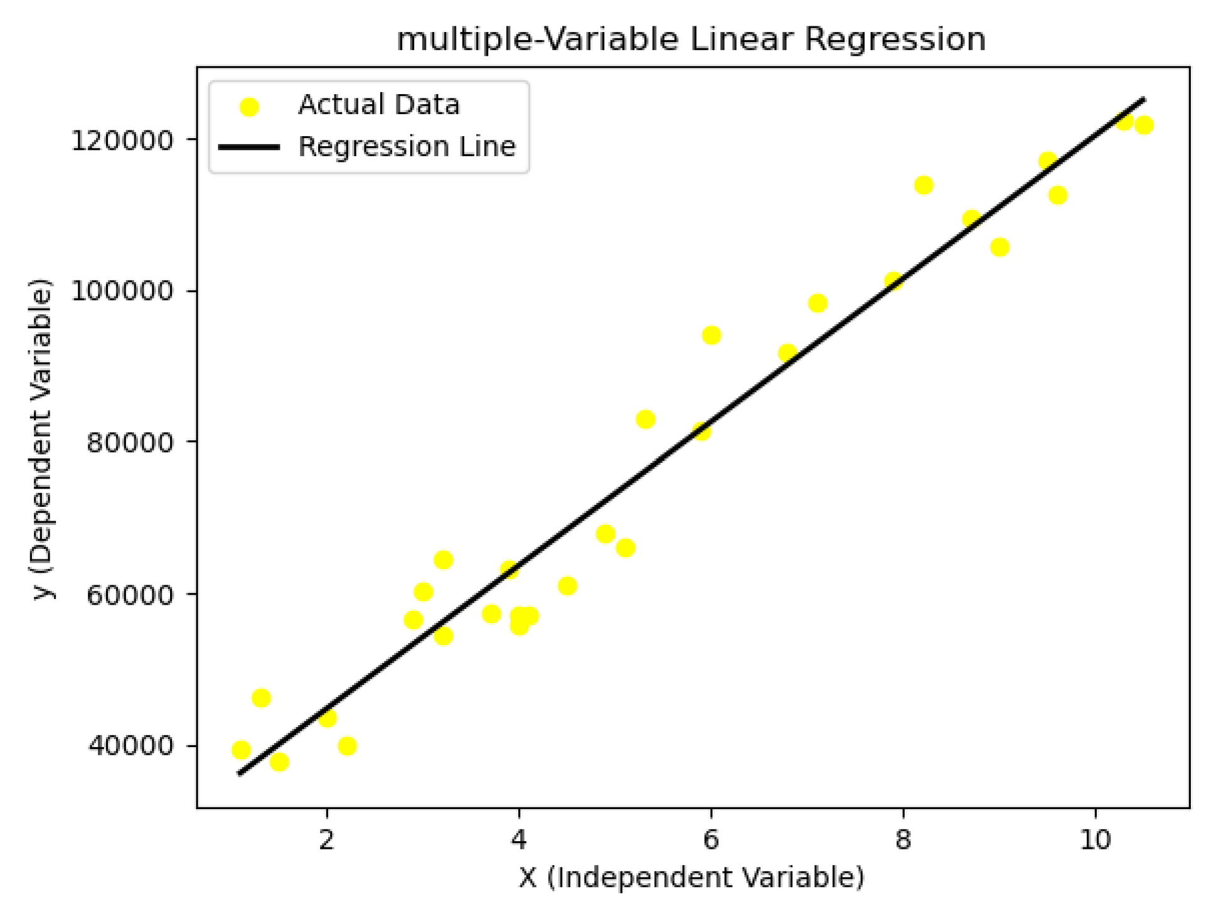
In [67]:
```python
print("Intercept:", model.intercept_)
```

```
Intercept: 25792.20019866871
```

In [68]:
```python
print("Predicted Value(s):", y_pred)
```

```
Predicted Value(s): [ 36187.15875227  38077.15121656  39967.14368085  44692.12484158
  46582.11730587  53197.09093089  54142.08716303  56032.07962732
  56032.07962732  60757.06078805  62647.05325234  63592.04948449
  63592.04948449  64537.04571663  68317.03064522  72097.0155738
  73987.00803809  75877.00050238  81546.97789525  82491.9741274
  90051.94398456  92886.932681   100446.90253816 103281.8912346
 108006.87239533 110841.86109176 115566.84225249 116511.83848464
 123126.81210966 125016.80457395]
```

In [78]:
```python
plt.scatter(X, y, color='yellow', label='Actual Data')
plt.plot(X, y_pred, color='black', linewidth=2, label='Regression Line')
plt.xlabel('X (Independent Variable)')
plt.ylabel('y (Dependent Variable)')
plt.title('multiple-Variable Linear Regression')
plt.legend()
plt.show()
```



EXAMPLE

In [106…
```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Step 1: Load the dataset
data = pd.read_csv('Ecommerce_Customers[1].csv')

# Step 2: Explore and preprocess the data (cleaning, feature selection, etc.)
# For this example, let's assume you're interested in 'Yearly Amount Spent' as the target variable and 'Length of Membership' as a feature.
X = data[['Length of Membership']]  # Feature(s)
y = data['Yearly Amount Spent']  # Target variable

# Step 3: Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 4: Create a Linear Regression model
model = LinearRegression()

# Step 5: Train the model on the training data
model.fit(X_train, y_train)

# Step 6: Make predictions on the testing data
y_pred = model.predict(X_test)

# Step 7: Evaluate the model's performance
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Step 8: Print model coefficients and evaluation metrics
print("Coefficient (Slope):", model.coef_[0])
print("Intercept:", model.intercept_)
print("Mean Squared Error (MSE):", mse)
print("R-squared (R2):", r2)
```

```
Coefficient (Slope): 64.64010065386711
Intercept: 271.35211280339297
Mean Squared Error (MSE): 2162.1116327764234
R-squared (R2): 0.5633624502240148
```

EXAMPLE

In [2]:
```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

#
```

In [3]:
```python
data = pd.read_csv('advertising[1].csv')
data
```

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
...	...	...	...	...
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 [4]:
```python
X = data[['TV']]  # Feature(s)
y = data['Sales']  # Target variable
```

In [5]:
```python
# Step 3: Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [6]:
```python
# Step 4: Create a Linear Regression model
model = LinearRegression()
```

In [7]:
```python
# Step 5: Train the model on the training data
model.fit(X_train, y_train)
```

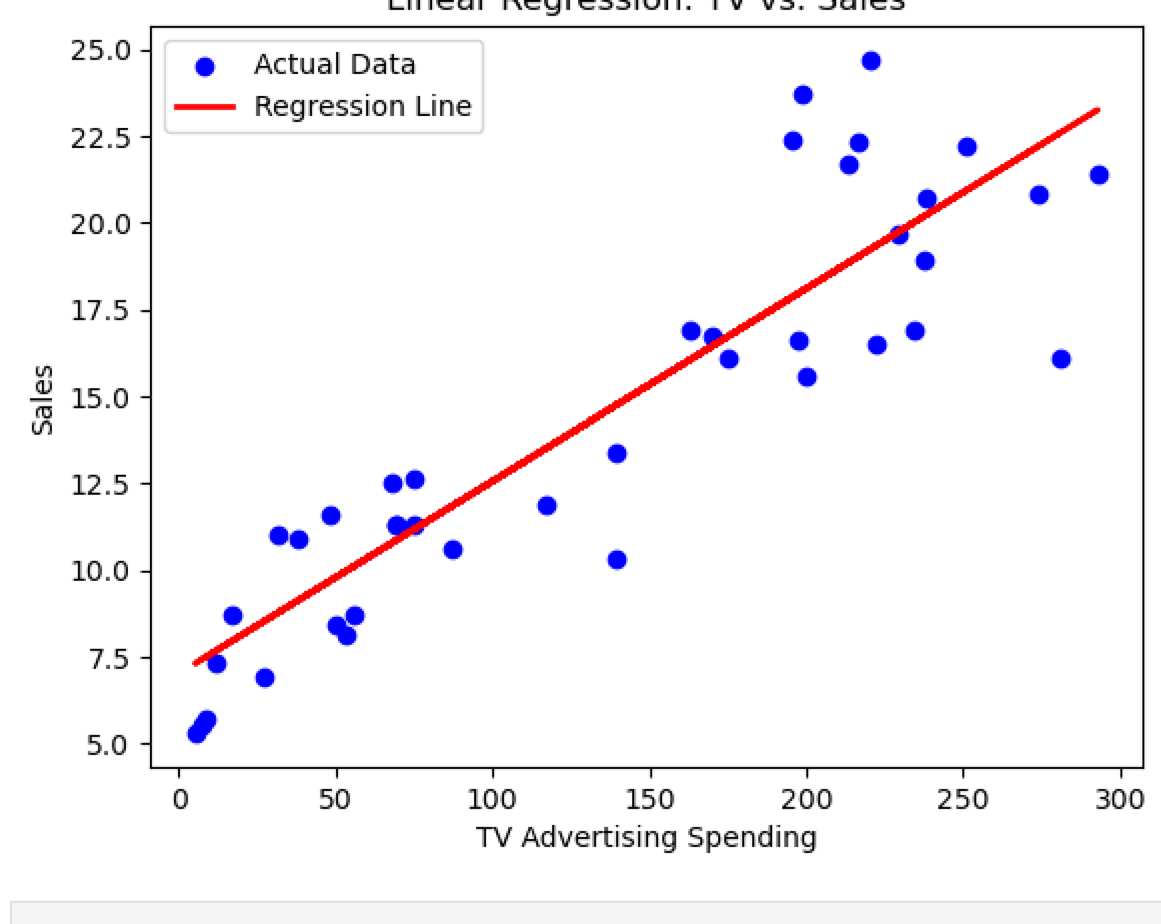
Out[7]:
```
▾ LinearRegression
LinearRegression()
```

In [8]:
```python
# Step 6: Make predictions on the testing data
y_pred = model.predict(X_test)
```

In [9]:
```python
# Step 8: Print model coefficients and evaluation metrics
print("Coefficient (Slope):", model.coef_[0])
print("Intercept:", model.intercept_)
```

```
Coefficient (Slope): 0.05548294393146318
Intercept: 7.007108428241851
```

In [10]:
```python
# Step 9: Visualize the regression line and data
plt.scatter(X_test, y_test, color='blue', label='Actual Data')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Regression Line')
plt.xlabel('TV Advertising Spending')
plt.ylabel('Sales')
plt.title('Linear Regression: TV vs. Sales')
plt.legend()
plt.show()
```



In [ ]: