Exercise 3

Given the dataset of 30 students' study hours and exam scores, how would you build a linear regression model to predict exam scores? Describe the steps you would take to diagnose the regression model, including checking assumptions, identifying outliers, and handling influential points. Finally, evaluate the model's performance and discuss any insights gained.

```python
In [71]: import matplotlib.pyplot as plt
         import pandas as pd
         from sklearn.metrics import r2_score, mean_squared_error
         from sklearn.model_selection import train_test_split
         from sklearn.linear_model import LinearRegression
```

```python
In [72]: df = pd.read_csv("student_data.csv")
         df.head()
```
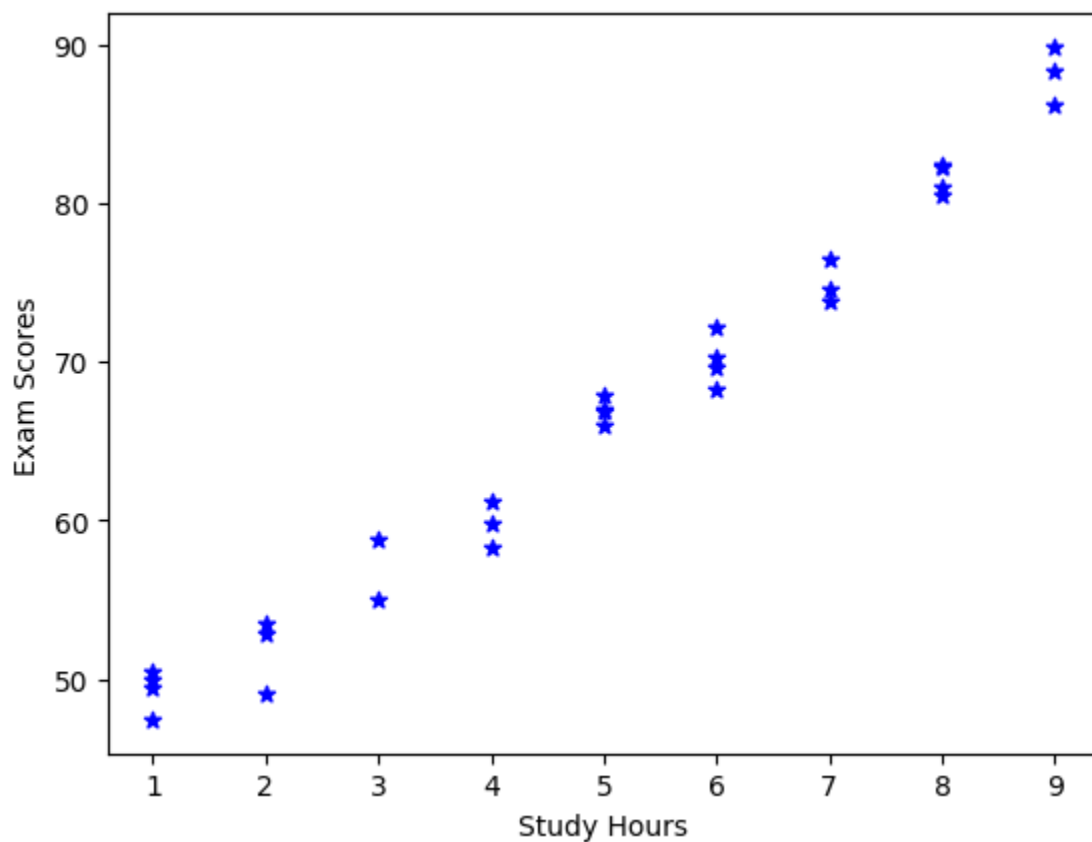
Out[72]:

	StudyHours	ExamScore
0	5	66.938936
1	3	58.791081
2	7	73.818557
3	4	59.844898
4	6	69.690213

```python
In [73]: X = df[['StudyHours']]
         y = df['ExamScore']

         plt.scatter(X, y, color="b", marker="*")
         plt.xlabel("Study Hours")
         plt.ylabel("Exam Scores")
```

Out[73]: Text(0, 0.5, 'Exam Scores')

In [74]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42, test_size=0.2
```

In [75]:
```python
model = LinearRegression()

model.fit(X_train, y_train)
```

Out[75]:
```
▾ LinearRegression    ⓘ ?

LinearRegression()
```

In [76]:
```python
y_pred = model.predict(X_test)
```

In [77]:
```python
plt.title("Comparing Actual and Predicted Linear Regression Data")
plt.xlabel("Study Hours")
plt.ylabel("Exam Scores")
plt.scatter(X_test, y_test, color="b", label="Actual Data")
plt.plot(X_test, y_pred, color="r", label="Predicted Data")
plt.legend()
```

Out[77]:   <matplotlib.legend.Legend at 0x7cb01a7ae0d0>

Comparing Actual and Predicted Linear Regression Data
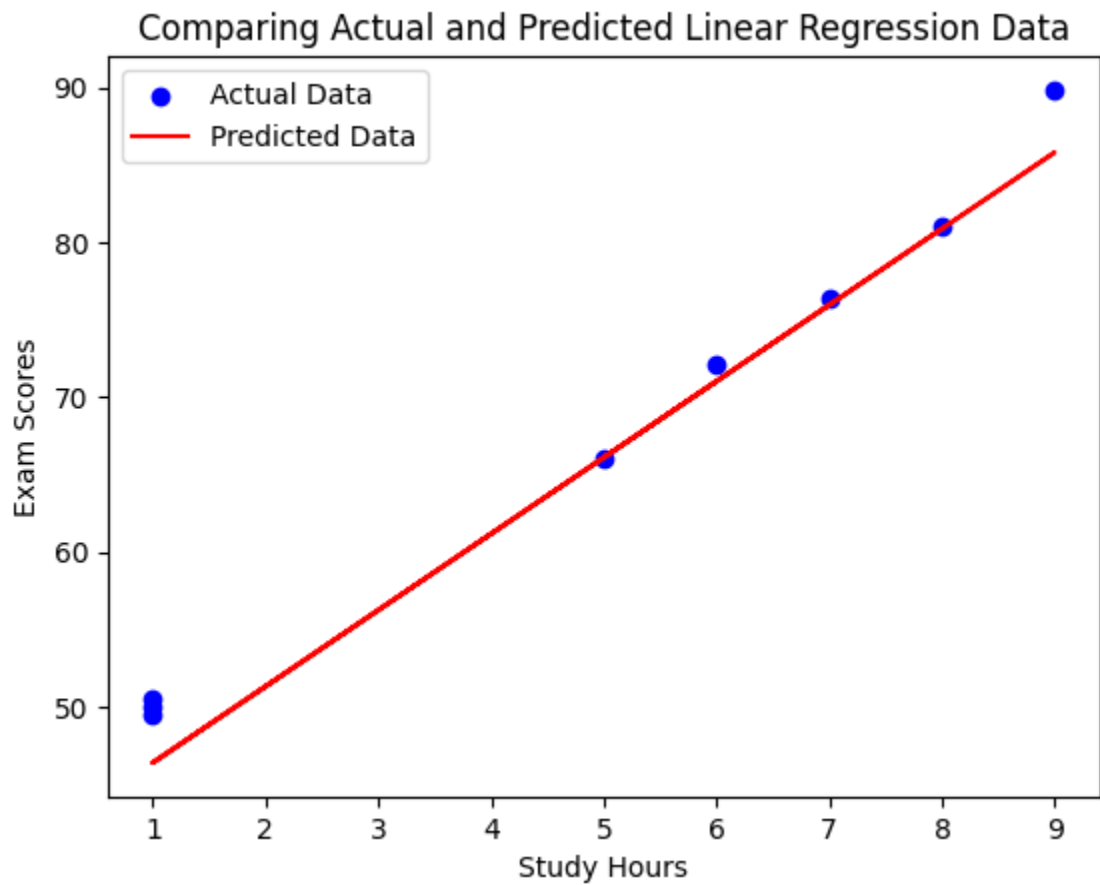

```
In [78]: r2 = r2_score(y_pred, y_test)
         mse = mean_squared_error(y_pred, y_test)

         print(f"Mean Squared Error = {mse}\nr^2 = {r2}")
```

```
Mean Squared Error = 7.148419001716639
r^2 = 0.9696208656224866
```