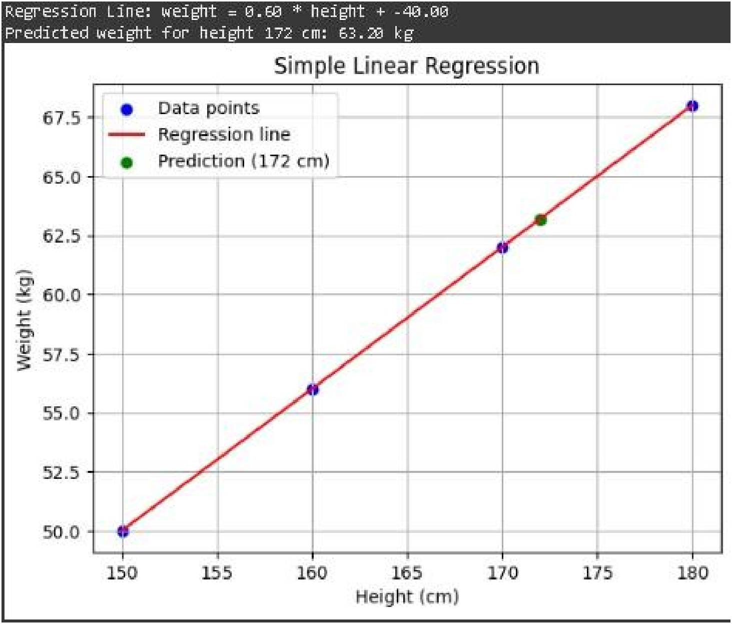
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
height = np.array([150, 160, 170, 180]).reshape(-1, 1)
weight = np.array([50, 56, 62, 68])  # Corresponding weights
model = LinearRegression()
model.fit(height, weight)
slope = model.coef_[0]
intercept = model.intercept_
print(f"Regression Line: weight = {slope:.2f} * height +
{intercept:.2f}")
height_to_predict = np.array([[172]])
predicted_weight = model.predict(height_to_predict)
print(f"Predicted weight for height 172 cm:
{predicted_weight[0]:.2f} kg")
plt.scatter(height, weight, color='blue', label='Data points')
plt.plot(height, model.predict(height), color='red',
label='Regression line')
plt.scatter(172, predicted_weight, color='green', label='Prediction
(172 cm)')
plt.xlabel("Height (cm)")
plt.ylabel("Weight (kg)")
plt.title("Simple Linear Regression")
plt.legend()
plt.grid(True)
plt.show()
```

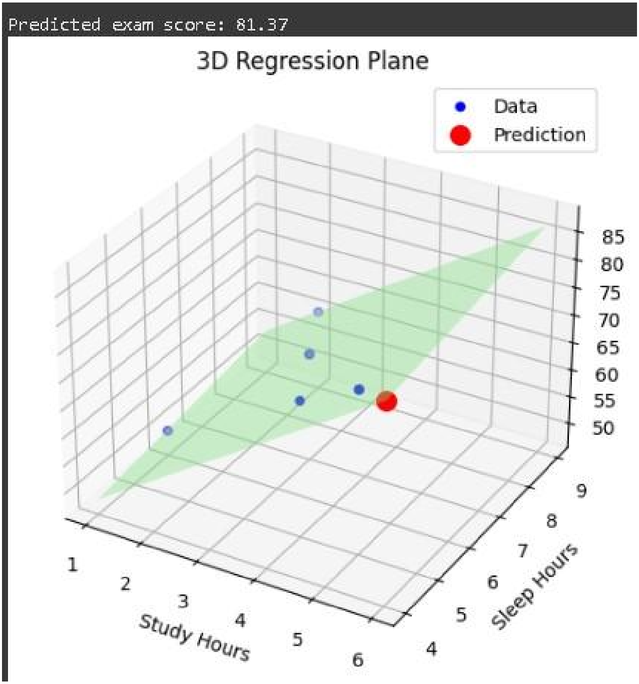


```
Regression Line: weight = 0.60 * height + -40.00
Predicted weight for height 172 cm: 63.20 kg
```

```
])

y = np.array([60, 50, 65, 70, 75])  # Exam scores
model = LinearRegression()
model.fit(X, y)
new_data = np.array([[6, 4]])
predicted_score = model.predict(new_data)
print(f"Predicted exam score: {predicted_score[0]:.2f}")
```

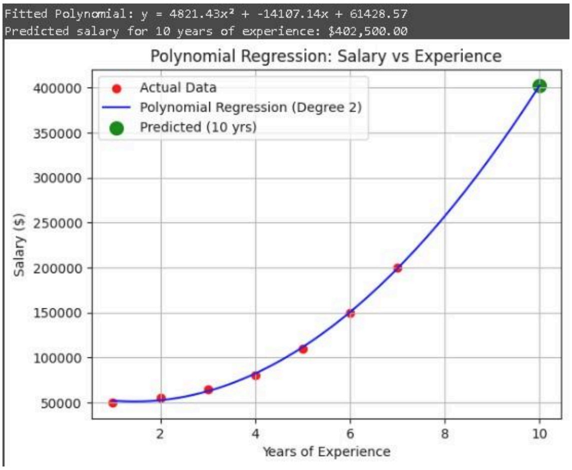
Output :



```
Predicted exam score: 81.37
```

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
X = np.array([1, 2, 3, 4, 5, 6, 7]).reshape(-1, 1)
y = np.array([50000, 55000, 65000, 80000, 110000, 150000, 200000])
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)
model = LinearRegression()
model.fit(X_poly, y)
a = model.coef_[2]
b = model.coef_[1]
c = model.intercept_
print(f"Fitted Polynomial: y = {a:.2f}x² + {b:.2f}x + {c:.2f}")
X_new = np.array([[10]])
X_new_poly = poly.transform(X_new)
predicted_salary = model.predict(X_new_poly)
print(f"Predicted salary for 10 years of experience:
${predicted_salary[0]:,.2f}")
x_range = np.linspace(1, 10, 100).reshape(-1, 1)
x_range_poly = poly.transform(x_range)
y_range = model.predict(x_range_poly)
plt.scatter(X, y, color='red', label='Actual Data')
plt.plot(x_range, y_range, color='blue', label='Polynomial
Regression (Degree 2)')
plt.scatter(10, predicted_salary, color='green', s=100,
label='Predicted (10 yrs)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary ($)')
plt.title('Polynomial Regression: Salary vs Experience')
plt.legend()
plt.grid(True)
plt.show()
```

Output :



```
Fitted Polynomial: y = 4821.43x² + -14107.14x + 61428.57
Predicted salary for 10 years of experience: $402,500.00
```

4. **Compute the MAE, MSE, RMSE and R^2 a model that performed below prediction as in the report.**

Observation	**Actual Value (y)**	**Predicted Value (ŷ)**
1	**3**	**2.5**
2	**5**	**5.1**
3	**7**	**6.8**
4	**9**	**9.3**

```
import numpy as np
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
y_true = np.array([3, 5, 7, 9])
y_pred = np.array([2.5, 5.1, 6.8, 9.3])
mae = mean_absolute_error(y_true, y_pred)
mse = mean_squared_error(y_true, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_true, y_pred)
print(f"MAE:  {mae:.3f}")
print(f"MSE:  {mse:.3f}")
print(f"RMSE: {rmse:.3f}")
print(f"R²:   {r2:.3f}")
```

Output :

```
MAE:  0.275

MSE:  0.098

RMSE: 0.312
```

```
R²:    0.980
```