#**1Q**

1Q

```
import numpy as np
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

```
Y_Actual = [20, 30, 40, 50, 60]
Y_pred = [20.5, 30.3, 40.2, 50.6, 60.7]

total_absolute_error = 0
total_squared_error = 0
```

```
for i in range(len(Y_Actual)):
    total_absolute_error += abs(Y_Actual[i] - Y_pred[i])
    total_squared_error += (Y_Actual[i] - Y_pred[i]) ** 2

MAE = total_absolute_error / len(Y_Actual)
MSE = total_squared_error / len(Y_Actual)
RMSE = (MSE) ** 0.5
```

```
print("Error Metrics (from scratch):")
print("Mean Absolute Error (MAE):", MAE)
print("Mean Squared Error (MSE):", MSE)
print("Root Mean Squared Error (RMSE):", RMSE)
```

```
Error Metrics (from scratch):
Mean Absolute Error (MAE): 0.4600000000000016
Mean Squared Error (MSE): 0.24600000000000147
Root Mean Squared Error (RMSE): 0.49598387070549127
```

The library error metrics

```
MAE_lib = mean_absolute_error(Y_Actual, Y_pred)
MSE_lib = mean_squared_error(Y_Actual, Y_pred)
RMSE_lib = np.sqrt(MSE_lib)

print("\nError Metrics (from sklearn):")
print("Mean Absolute Error (MAE) from sklearn:", MAE_lib)
print("Mean Squared Error (MSE) from sklearn:", MSE_lib)
print("Root Mean Squared Error (RMSE) from sklearn:", RMSE_lib)
```

```

Error Metrics (from sklearn):
Mean Absolute Error (MAE) from sklearn: 0.4600000000000016
Mean Squared Error (MSE) from sklearn: 0.24600000000000147
Root Mean Squared Error (RMSE) from sklearn: 0.49598387070549127
```

2Q

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
import numpy as np
```

```
Y_actual = [0, 0, 1, 1, 2, 0]
Y_pred = [0, 0, 1, 0, 2, 0]

TP = sum(1 for a, p in zip(Y_actual, Y_pred) if a == p and a != 0)
TN = sum(1 for a, p in zip(Y_actual, Y_pred) if a == p and a == 0)
FP = sum(1 for a, p in zip(Y_actual, Y_pred) if a != p and p != 0)
FN = sum(1 for a, p in zip(Y_actual, Y_pred) if a != p and a == 0)
```

```
precision = TP / (TP + FP) if (TP + FP) != 0 else 0
print(f"Precision: {precision}")
```

```
Precision: 1.0
```

```
recall = TP / (TP + FN) if (TP + FN) != 0 else 0
print(f"Recall: {recall}")
```

```
Recall: 1.0
```

```
f1_score = 2 * (precision * recall) / (precision + recall) if (precision + recall) != 0 else 0
print(f"F1 Score: {f1_score}")
```

```
F1 Score: 1.0
```