



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 3.1

Student Name: Sumant Rao

UID:21BCS7561

Branch: CSE

Section/Group: 911 “B”

Semester: 5th

Date of Performance: 17/10/23

Subject Name: AIML

Subject Code:21CSH-316

1. Aim:

Write a python program to compute Mean, Median, Mode, Variance and Standard Deviation using Datasets

2. Objective:

The objective of this experiment is to evaluate the performance of an algorithm using two error metrics: Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

3. Algorithm:

Initialize squared_errors as an empty list.

For each pair of actual and predicted values in the actual_values and predicted_values lists, do the following:

Calculate the squared error by subtracting the actual value from the predicted value and squaring the result. Append the squared error to the squared_errors list. Calculate the MSE (Mean Squared Error) by taking the mean of the squared errors in the squared_errors list.

Calculate the RMSE (Root Mean Squared Error) by taking the square root of the MSE.

Return the calculated MSE and RMSE.

4. Code:

```
import numpy as np
def calculate_mse(actual, predicted):
    actual = np.array(actual)
    predicted = np.array(predicted)
```

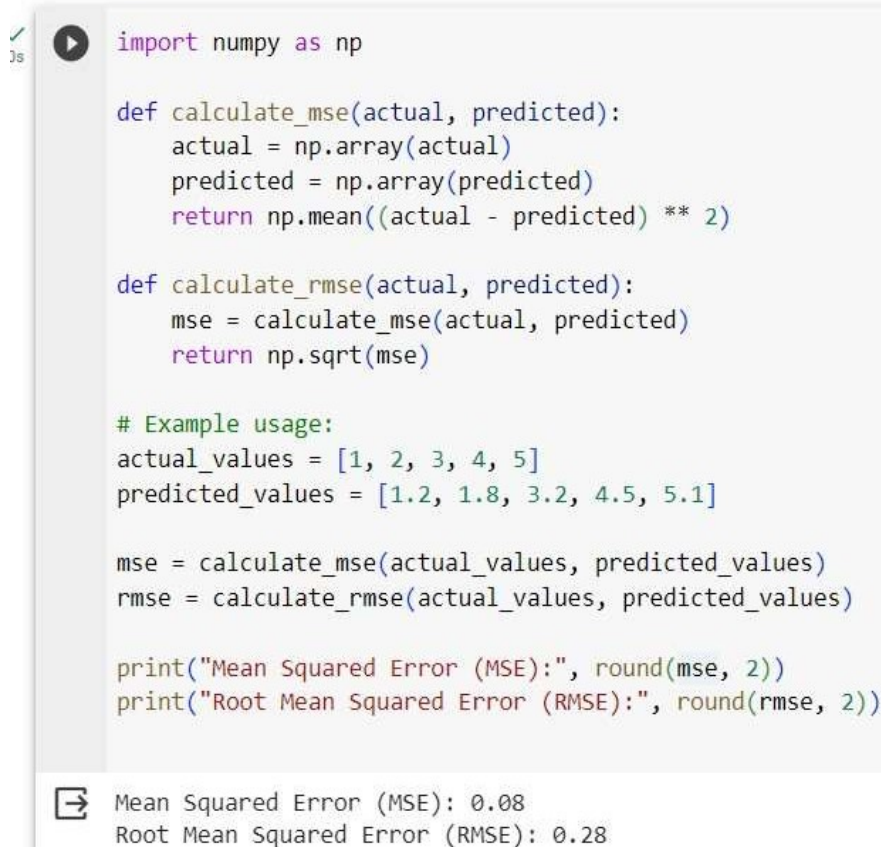
```
return np.mean((actual - predicted) ** 2)

def calculate_rmse(actual, predicted):
    mse = calculate_mse(actual, predicted)
    return np.sqrt(mse)

# Example usage: actual_values = [1, 2, 3, 4, 5]
predicted_values = [1.2, 1.8, 3.2, 4.5, 5.1]

mse = calculate_mse(actual_values, predicted_values)
rmse = calculate_rmse(actual_values, predicted_values)
print("Mean Squared Error (MSE):", round(mse, 2))
print("Root Mean Squared Error (RMSE):", round(rmse, 2))
```

5. Output



```
import numpy as np

def calculate_mse(actual, predicted):
    actual = np.array(actual)
    predicted = np.array(predicted)
    return np.mean((actual - predicted) ** 2)

def calculate_rmse(actual, predicted):
    mse = calculate_mse(actual, predicted)
    return np.sqrt(mse)

# Example usage:
actual_values = [1, 2, 3, 4, 5]
predicted_values = [1.2, 1.8, 3.2, 4.5, 5.1]

mse = calculate_mse(actual_values, predicted_values)
rmse = calculate_rmse(actual_values, predicted_values)

print("Mean Squared Error (MSE):", round(mse, 2))
print("Root Mean Squared Error (RMSE):", round(rmse, 2))
```

Mean Squared Error (MSE): 0.08
Root Mean Squared Error (RMSE): 0.28