Experiment: 3.1

Student Name: Nishant Kumar Mehta **UID:** 21BCS-*3402*
Branch: CSE **Section/Group:** IOT-602/B
Semester: 5th **Date:** 5/10/2023
Subject Name: AIML Lab **Subject Code:** 21CSH-316

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

1. Tools/Resource Used:

Goggle CoLab (Online Compiler)

2. Algorithm:

Root Mean Square Error is the measure of how well a regression line fits the data points. RMSE can also be construed as Standard Deviation in the residuals.

The **Mean Squared Error** or Mean Squared Deviation (MSD) of an estimator measures the average of error squares i.e. the average squared difference between the estimated values and true value.

Variance is the sum of squares of differences between all numbers and means.

Standard Deviation is the square root of variance. It is a measure of the extent to which data varies from the mean.

3. Program Code:

```
import numpy as np

def calculate_mse(actual, predicted):
    actual = np.array(actual)  # Convert the input lists to NumPy arrays
    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):", mse)
print("Root Mean Squared Error (RMSE):", rmse)

#Mean
l = [1, 3, 8, 15]
print(statistics.mean(l))

#Median
l = [3, 1, 8]
print(statistics.median(l))
print(statistics.median_low(l))
print(statistics.median_high(l))

#Mode
l = [3, 2, 3, 2, 1, 2]
print(statistics.mode(l))
print(statistics.multimode(l))

#Variance
list = [212, 231, 234, 564, 235]
print(np.var(list))

#Standard Deviation
list = [290, 124, 127, 899]
print(np.std(list))
```

4. Output/Result:

```
#Mean
l = [1, 3, 8, 15]
print(statistics.mean(l))

6.75

#Median
l = [3, 1, 8]
print(statistics.median(l))
print(statistics.median_low(l))
print(statistics.median_high(l))

3
3
3

[4] #Mode
l = [3, 2, 3, 2, 1, 2]
print(statistics.mode(l))
print(statistics.multimode(l))

2
[2]

[5] #Variance
list = [212, 231, 234, 564, 235]
print(np.var(list))

18133.359999999997

[6] #Standard Deviation
list = [290, 124, 127, 899]
print(np.std(list))

318.35750344541907
```

```
import numpy as np

def calculate_mse(actual, predicted):
    actual = np.array(actual)  # Convert the input lists t
    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):", mse)
print("Root Mean Squared Error (RMSE):", rmse)

Mean Squared Error (MSE): 0.07599999999999998
Root Mean Squared Error (RMSE): 0.2756809750418044
```

5. Learning Outcomes:

- To Learn about MSE and RMSE
- To learn About MSE limitations and advantages.
- To Learn about RMSE limitations and advantages.