Experiment 3.1

Student Name: Utkarsh Joshi **UID: 21BCS9158**
Branch: BE CSE **Section/Group: 802-A**
Semester: 5 Sem **Date of Performance: 25/10/2023**
Subject Name: AI &ML with Lab **Subject Code: 21CSH-316**

1. **Aim:** Write a python program to compute Mean, Median, Mode, Variance and Standard Deviation using Dataset.

2. **Objective:** The objective is to assess how well the Mean, Median, Mode, Variance and Standard Deviation using Dataset and to analyze its effectiveness in comparison to other algorithms or approaches.

3. **Input/Apparatus Used:** Google collab and python libraries.

4. **Hardwire Requirements:** Computer/Laptop minimum 4GB, windows and Power Supply

5. **Code:**

Task 1:

```
import numpy as np

from sklearn.metrics import mean_squared_error import

matplotlib.pyplot as plt

# Example data

names = ["Alice", "Bob", "Charlie", "David", "Emma", "Frank", "Grace",
"Hannah", "Ivy", "Jack"]
```

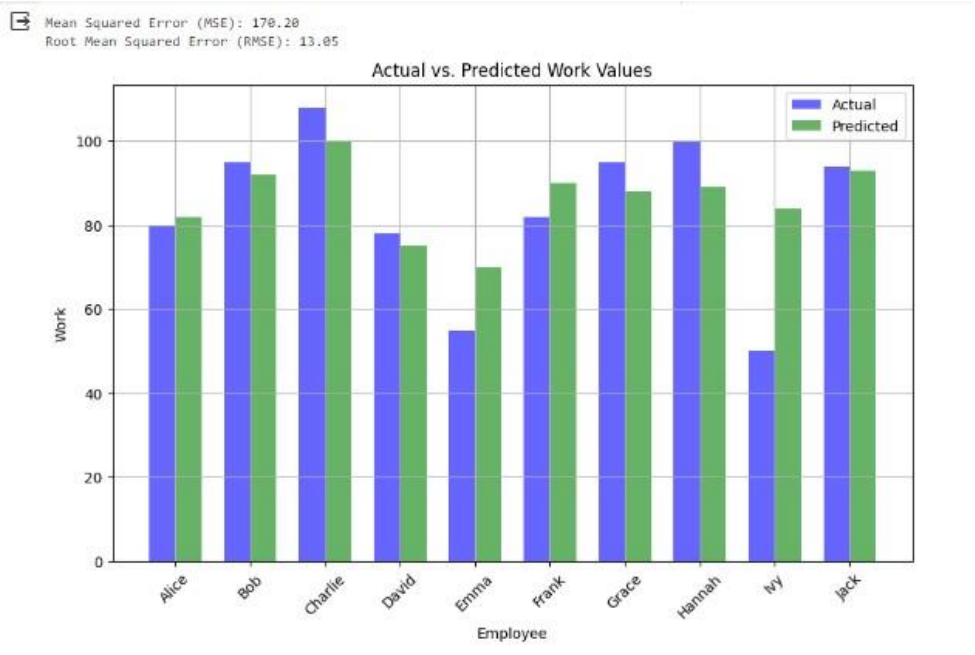
```
actual = np.array([80, 95, 108, 78, 55, 82, 95, 100, 50, 94]) # Actual work values
predicted = np.array([82, 92, 100, 75, 70, 90, 88, 89, 84, 93]) # Predicted work
values

# Calculate Mean Squared Error (MSE)

mse = mean_squared_error(actual, predicted) #

Calculate Root Mean Squared Error (RMSE)

rmse = np.sqrt(mse)

# Print the MSE and RMSE

print(f"Mean Squared Error (MSE): {mse:.2f}") print(f"Root

Mean Squared Error (RMSE): {rmse:.2f}")

# Create a plot to visualize the actual vs. predicted work values side by side

x = np.arange(len(names)) # x-coordinates for bars width = 0.35 # Width of

the bars plt.figure(figsize=(10, 6)) plt.bar(x - width/2, actual, width,

label='Actual', alpha=0.6, color='b') plt.bar(x + width/2, predicted, width,

label='Predicted', alpha=0.6, color='g') plt.legend() plt.title('Actual vs.

Predicted Work Values') plt.xlabel('Employee')

plt.ylabel('Work')

plt.xticks(x, names, rotation=45) plt.grid(True)

plt.show()
```

Task 2:

```
import numpy as np

from sklearn.metrics import mean_squared_error from

math import sqrt

import matplotlib.pyplot as plt

# Sample predicted and actual target values

predicted_values = [1.6, 2.4, 3.7, 4.1, 5.2] actual_values =

[1.0, 2.5, 3.4, 4.5, 5.2] # Calculate Mean Squared Error

(MSE) mse = mean_squared_error(actual_values,

predicted_values) # Calculate Root Mean Squared Error

(RMSE) rmse = sqrt(mse) # Print the results

print(f"Mean Squared Error (MSE): {mse}") print(f"Root

Mean Squared Error (RMSE): {rmse}")


# Create a scatter plot to visualize the actual vs. predicted values

plt.figure(figsize=(8, 6))

plt.scatter(actual_values, predicted_values, color='b', label='Actual vs.
Predicted')
```

```
plt.plot([min(actual_values), max(actual_values)], [min(actual_values),
max(actual_values)], color='r', linestyle='--')

plt.xlabel('Actual Values')

plt.ylabel('Predicted Values')

plt.title('Actual vs. Predicted Values')

plt.legend() # Show the plot

plt.show()
```

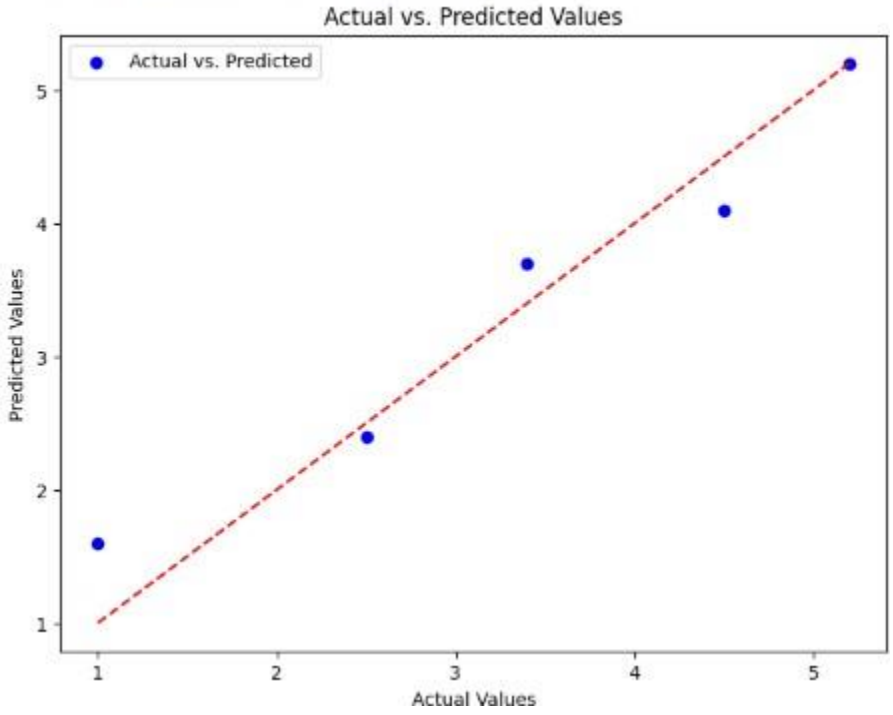
6. Output:



Task: 1

Task: 2

```
Mean Squared Error (MSE): 0.12400000000000011
Root Mean Squared Error (RMSE): 0.35213633723318033
```



7. Learning Outcomes:

1. Implement how to use mean, median, mode, variance and standard deviation.
2. Understanding of various functions and mathematical expressions.
3. Understanding of graphs and there relations