

# Data Analytics

Subject Code: BCSDS0651 | Even Semester 2025-26

---

**Name** Yashvi Dixit  
**Roll No.** 2301330100233  
**Branch** Computer Science and Engineering  
**Semester** 6  
**Section** D  
**Submitted to** Ms. Ayushi Gupta

---

# 1. Write a Python program to compute Central Tendency and Dispersion Measures

## Mathematical Formulas

Mean:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Sample Variance:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

Standard Deviation:

$$s = \sqrt{s^2}$$

## Python Program

```
1 import statistics
2
3 data = [10, 20, 20, 30, 40, 50]
4
5 # Central Tendency
6 mean_val = statistics.mean(data)
7 median_val = statistics.median(data)
8 mode_val = statistics.multimode(data)
9
10 # Dispersion (Sample)
11 variance_val = statistics.variance(data)
12 std_dev_val = statistics.stdev(data)
13
14 print("Mean:", mean_val)
15 print("Median:", median_val)
16 print("Mode:", mode_val)
17 print("Variance:", variance_val)
18 print("Standard Deviation:", std_dev_val)
```

# 2. Study of Python Basic Libraries: Statistics, Math, NumPy and SciPy

## Statistics Library

Provides built-in functions for mean, median, variance, and standard deviation.

## Math Library

Used for mathematical operations such as:

- `math.sqrt()`
- `math.log()`
- `math.factorial()`

## NumPy

Efficient numerical computations using arrays.

```
1 import numpy as np
2
3 arr = np.array([10, 20, 20, 30, 40, 50])
4
5 print("NumPy Mean:", np.mean(arr))
6 print("NumPy Variance:", np.var(arr))
```

## SciPy

Used for advanced statistical functions.

```
1 from scipy import stats
2
3 print("SciPy Mode:", stats.mode(arr, keepdims=True))
```

# 3. Study of Python Libraries for ML Applications: Pandas and Matplotlib

## Pandas

Used for structured data manipulation.

```
1 import pandas as pd
2
3 df = pd.DataFrame({"Values": arr})
4 print(df.describe())
```

## Matplotlib

Used for visualization.

```
1 import matplotlib.pyplot as plt
2
3 plt.hist(arr)
4 plt.title("Histogram of Data")
5 plt.xlabel("Values")
```

```
6 plt.ylabel("Frequency")
7 plt.show()
```

## 4. Write a Python program to implement Simple Linear Regression

### Mathematical Model

$$y = \beta_0 + \beta_1 x$$

$$\beta_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

### Python Program

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.array([1, 2, 3, 4, 5])
5 y = np.array([2, 4, 5, 4, 5])
6
7 x_mean = np.mean(x)
8 y_mean = np.mean(y)
9
10 beta_1 = np.sum((x - x_mean) * (y - y_mean)) \
11     / np.sum((x - x_mean)**2)
12
13 beta_0 = y_mean - beta_1 * x_mean
14
15 print("Intercept:", beta_0)
16 print("Slope:", beta_1)
17
18 y_pred = beta_0 + beta_1 * x
19
20 plt.scatter(x, y)
21 plt.plot(x, y_pred)
22 plt.title("Simple Linear Regression")
23 plt.xlabel("X")
24 plt.ylabel("Y")
25 plt.show()
```

## 5. Load a Sample Dataset to Build a Predictive Model in Python

### Problem Statement

1. Import Python Libraries
2. Read the Dataset

### Solution

```
1 # 1. Import Required Libraries
2 import pandas as pd
3 import numpy as np
4 from sklearn.datasets import load_iris
5
6 # 2. Load Sample Dataset (Iris Dataset)
7 iris = load_iris()
8
9 # Convert to DataFrame
10 df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
11 df[‘target’] = iris.target
12
13 print(df.head())
```

## 6. Explore the Dataset and Perform Feature Selection

### Problem Statement

1. Explore the Dataset
2. Perform Feature Selection

### Solution

```
1 # Dataset Exploration
2 print("Dataset Shape:", df.shape)
3 print(df.info())
4 print(df.describe())
5
6 # Feature Selection
7 X = df.drop("target", axis=1)
8 y = df[“target”]
9
10 print("Selected Features:")
11 print(X.columns)
```

## 7. Build the Model and Evaluate Performance

### Problem Statement

1. Build the Model
2. Evaluate the Model's Performance

### Solution

```
1 from sklearn.model_selection import train_test_split
2 from sklearn.linear_model import LogisticRegression
3 from sklearn.metrics import accuracy_score, classification_report
4
5 # Split Data
6 X_train, X_test, y_train, y_test = train_test_split(
7     X, y, test_size=0.3, random_state=42
8 )
9
10 # Build Model
11 model = LogisticRegression(max_iter=200)
12 model.fit(X_train, y_train)
13
14 # Predictions
15 y_pred = model.predict(X_test)
16
17 # Evaluation
18 print("Accuracy:", accuracy_score(y_test, y_pred))
19 print("Classification Report:\n", classification_report(y_test,
y_pred))
```

## 8. Installing and Setting Up Python Environment

### Problem Statement

1. Installing Pandas and Other Dependent Python Modules
2. Setting Up and Using Jupyter Notebooks

### Solution

#### Step 1: Install Python

Download Python from the official website and verify installation:

```
1 python --version
```

#### Step 2: Install Required Libraries

```
1 pip install numpy pandas matplotlib scipy scikit-learn
```

### **Step 3: Install Jupyter Notebook**

```
1 pip install notebook
```

### **Step 4: Launch Jupyter Notebook**

```
1 jupyter notebook
```

Jupyter Notebook opens in a web browser where Python code can be executed interactively.