# Data Analytics

## Assignment 1

Subject Code: BCSDS0651 | Even Semester 2025-26

---


| | |
|---|---|
| **Name** | Suryansh Singh |
| **Roll No.** | 2301330100208 |
| **Branch** | Computer Science and Engineering |
| **Semester** | 6 |
| **Section** | C |
| **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

```python
import statistics

data = [10, 20, 20, 30, 40, 50]

# Central Tendency
mean_val = statistics.mean(data)
median_val = statistics.median(data)
mode_val = statistics.multimode(data)

# Dispersion (Sample)
variance_val = statistics.variance(data)
std_dev_val = statistics.stdev(data)

print("Mean:", mean_val)
print("Median:", median_val)
print("Mode:", mode_val)
print("Variance:", variance_val)
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.

```
import numpy as np

arr = np.array([10, 20, 20, 30, 40, 50])

print("NumPy Mean:", np.mean(arr))
print("NumPy Variance:", np.var(arr))
```

### SciPy

Used for advanced statistical functions.

```
from scipy import stats

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.

```
import pandas as pd

df = pd.DataFrame({"Values": arr})
print(df.describe())
```

### Matplotlib

Used for visualization.

```
import matplotlib.pyplot as plt

plt.hist(arr)
plt.title("Histogram of Data")
plt.xlabel("Values")
```

```
plt.ylabel("Frequency")
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

```
import numpy as np
import matplotlib.pyplot as plt

x = np.array([1, 2, 3, 4, 5])
y = np.array([2, 4, 5, 4, 5])

x_mean = np.mean(x)
y_mean = np.mean(y)

beta_1 = np.sum((x - x_mean) * (y - y_mean)) \
         / np.sum((x - x_mean)**2)

beta_0 = y_mean - beta_1 * x_mean

print("Intercept:", beta_0)
print("Slope:", beta_1)

y_pred = beta_0 + beta_1 * x

plt.scatter(x, y)
plt.plot(x, y_pred)
plt.title("Simple Linear Regression")
plt.xlabel("X")
plt.ylabel("Y")
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

```python
# 1. Import Required Libraries
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris

# 2. Load Sample Dataset (Iris Dataset)
iris = load_iris()

# Convert to DataFrame
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df['target'] = iris.target

print(df.head())
```

# 6. Explore the Dataset and Perform Feature Selection

## Problem Statement

1. Explore the Dataset
2. Perform Feature Selection

## Solution

```python
# Dataset Exploration
print("Dataset Shape:", df.shape)
print(df.info())
print(df.describe())

# Feature Selection
X = df.drop("target", axis=1)
y = df["target"]

print("Selected Features:")
print(X.columns)
```

# 7. Build the Model and Evaluate Performance

## Problem Statement

**1.** Build the Model

**2.** Evaluate the Model's Performance

## Solution

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

# Split Data
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)

# Build Model
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
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:

```
python --version
```

### Step 2: Install Required Libraries

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

**Step 3: Install Jupyter Notebook**

```
pip install notebook
```

**Step 4: Launch Jupyter Notebook**

```
jupyter notebook
```

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