Experiment 1: Exploration of Python Libraries

Sri Sivasubramaniya Nadar College of Engineering, Chennai
(An Autonomous Institution Affiliated to Anna University)

Sreenethi G S

July 2025

Degree & Branch	B.E. Computer Science & Engineering	**Semester**	V
Subject Code & Name	ICS1512 & Machine Learning Algorithms Laboratory		
Academic Year	2025-2026 (Odd)	Batch: 2023–2028	**Due Date:**

Aim: To explore and understand the core functions and methods of Python libraries NumPy, Pandas, SciPy, Scikit-learn, and Matplotlib by performing array manipulations, data preprocessing, mathematical computing, machine learning workflows, and data visualization. To study public datasets, identify appropriate machine learning tasks, and apply the ML workflow steps including feature selection and model evaluation.

Libraries Used: NumPy, Pandas, SciPy, Scikit-learn, Matplotlib, Public Datasets.

1. Exploration of Python Libraries

— NumPy —

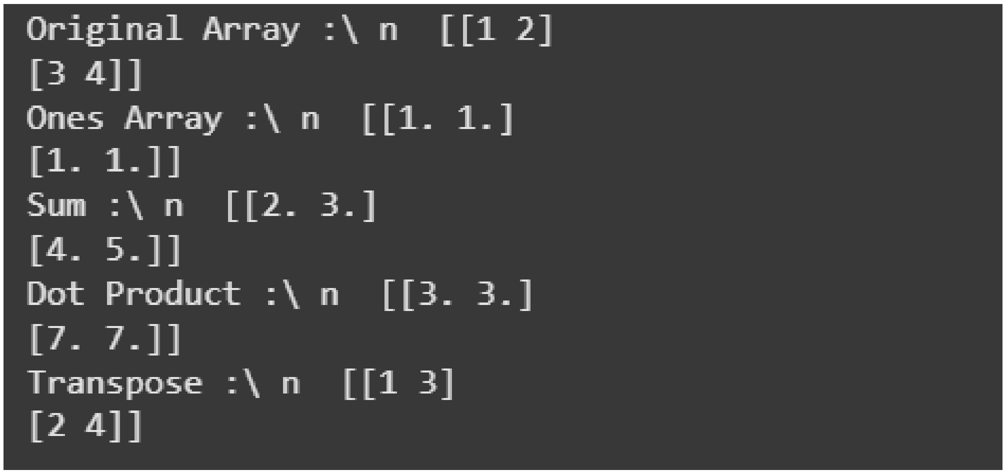
Used for: Efficient numerical computations, multi-dimensional arrays.

```python
import numpy as np

a = np.array([[1, 2], [3, 4]])
b = np.ones((2, 2))

sum_ab = a + b
product = np.dot(a, b)
transpose = a.T

print("Original Array:\n", a)
print("Ones Array:\n", b)
print("Sum:\n", sum_ab)
print("Dot Product:\n", product)
print("Transpose:\n", transpose)
```



```
Original Array :\ n  [[1 2]
[3 4]]
Ones Array :\ n  [[1. 1.]
[1. 1.]]
Sum :\ n  [[2. 3.]
[4. 5.]]
Dot Product :\ n  [[3. 3.]
[7. 7.]]
Transpose :\ n  [[1 3]
[2 4]]
```

Figure 1: Output Example 1

— Pandas —

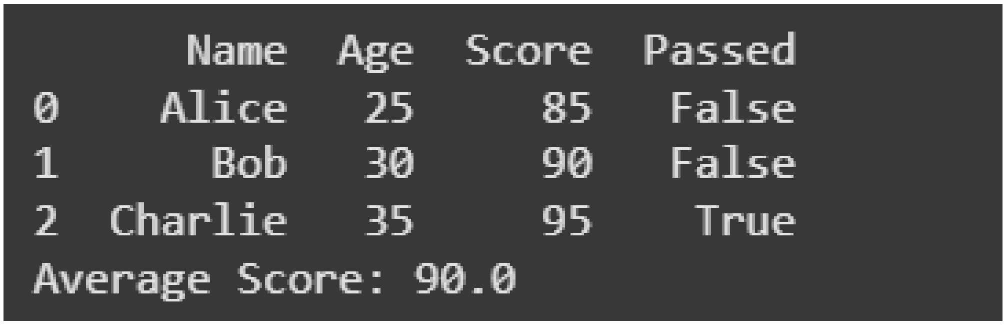
Used for: Data manipulation and analysis, working with tabular data.

```
import pandas as pd

data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35],
        'Score': [85, 90, 95]}
df = pd.DataFrame(data)

df['Passed'] = df['Score'] > 90
mean_score = df['Score'].mean()

print(df)
print("Average Score:", mean_score)
```



```
      Name  Age  Score  Passed
0    Alice   25     85   False
1      Bob   30     90   False
2  Charlie   35     95    True
Average Score: 90.0
```

Figure 2: Output Example 1

— SciPy —

Used for: Scientific and technical computing (integration, optimization, statistics).

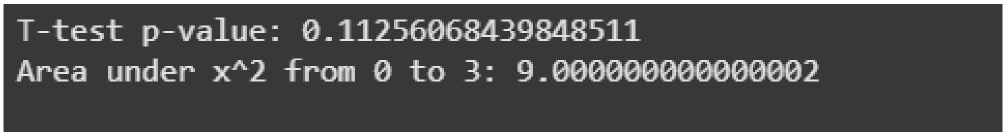
```
from scipy import stats, integrate

group1 = [22, 21, 23, 25, 30]
```

```
group2 = [25, 26, 27, 29, 32]
t_stat, p_val = stats.ttest_ind(group1, group2)

area = integrate.quad(lambda x: x**2, 0, 3)[0]

print("T-test p-value:", p_val)
print("Area under x^2 from 0 to 3:", area)
```



```
T-test p-value: 0.11256068439848511
Area under x^2 from 0 to 3: 9.000000000000002
```

Figure 3: Output Example 1

— Scikit-learn —

Used for: Machine learning workflows like preprocessing, training, and evaluation.

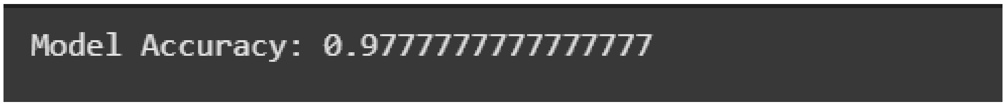
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

iris = load_iris()
X_train, X_test, y_train, y_test = train_test_split(
    iris.data, iris.target, test_size=0.3)

model = RandomForestClassifier()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)

print("Model Accuracy:", accuracy)
```



```
Model Accuracy: 0.9777777777777777
```

Figure 4: Output Example 1

— Matplotlib —

Used for: Data visualization.

```
import matplotlib.pyplot as plt

x = [0, 1, 2, 3, 4]
y = [i**2 for i in x]

plt.plot(x, y, label='y = x^2', color='blue', marker='o')
plt.title("Simple Line Plot")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.legend()
plt.grid(True)
plt.show()
```

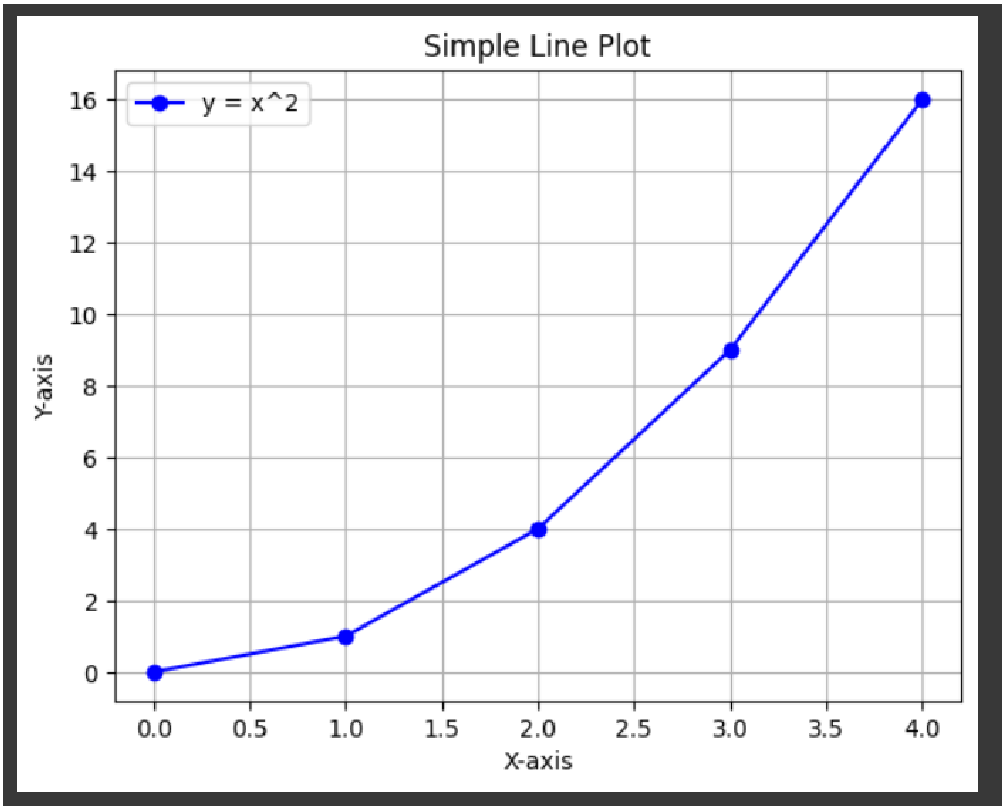


Figure 5: Output Example 1

2. Exploration of Public Datasets and ML Model Type Identification

- **i) Loan Amount Prediction** *Source:* Kaggle *ML Task:* Supervised – Regression / Classification
- **ii) Handwritten Character Recognition** *Source:* UCI / Kaggle *ML Task:* Supervised – Multi-class Classification
- **iii) Email Spam and MNIST Classification** *Source:* UCI (Spam), Kaggle (MNIST) *ML Task:* Supervised – Binary / Multi-class Classification
- **iv) Predicting Diabetes** *Source:* UCI (Pima) *ML Task:* Supervised – Binary Classification
- **v) Iris Dataset** *Source:* UCI *ML Task:* Supervised – Multi-class Classification

Dataset Example	ML Task Type	Description
House Price Prediction	Supervised – Regression	Predicts continuous value like price.
Email Spam Detection	Supervised – Classification	Classifies emails as spam or not spam.
Customer Segmentation	Unsupervised – Clustering	Groups similar users based on features.
Movie Recommendation	Unsupervised – Association	Finds items that are bought/watched together.
Stock Forecasting	Supervised – Time Series	Predicts future stock values using past data.

3. Machine Learning Workflow and Task Identification

Types of Machine Learning Tasks

Machine Learning Workflow Steps

i. **Loading the Dataset:** Using libraries like `pandas` or `NumPy`.

ii. **Exploratory Data Analysis (EDA):**
- Summary: `.describe()`, `.info()`
- Plots: Histogram, Boxplot, Heatmap

iii. **Data Preprocessing:**
- Handle missing values, encode categories, normalize features

iv. **Feature Selection:**
- Use `SelectKBest`, Chi-square, or ANOVA F-test

v. **Data Splitting:**
- Train/Test split, e.g., 70% train and 30% test

vi. **Performance Evaluation:**
- Classification: Accuracy, F1-score; Regression: RMSE, R^2
- Clustering: Silhouette Score

4. Example Dataset-based Tasks

Dataset	Type of ML Task	Feature Selection	Suggested Algorithm
Iris	Classification	All features	Logistic Regression
Loan Prediction	Regression / Classification	Income, Credit, Education	Decision Tree
Diabetes Prediction	Classification	Glucose, BMI, Age	Random Forest
Email Spam Detection	Classification	Word frequencies	Naive Bayes
MNIST Digits	Classification	Pixel Intensities	SVM / CNN

5. Results and Discussions

- Successfully explored core Python libraries (NumPy, Pandas, SciPy, Scikit-learn, Matplotlib) for data handling, statistical computing, and machine learning.
- Implemented machine learning models like Random Forest on the Iris dataset and evaluated performance using accuracy metrics.
- Visualized trends using plots such as line charts and histograms to enhance understanding of data distribution.
- Identified suitable ML task types (classification, regression, clustering) for various public datasets.
- Applied all stages of a machine learning workflow: loading, EDA, preprocessing, feature selection, model building, and evaluation.

6. Learning Practices

I learned how to:

- Use Python libraries (NumPy, Pandas, SciPy, Scikit-learn, Matplotlib) effectively.
- Load and preprocess real-world datasets.
- Apply supervised learning algorithms for classification and regression tasks.
- Visualize data and model performance.