Experiment 1: Working with Python Packages

Machine Learning Algorithms Laboratory
Subject Code: ICS1512

Name: Naveenraj J   Roll No: 3122237001030
Sri Sivasubramaniya Nadar College of Engineering, Chennai
Batch: 2023–2028

Date of Experiment: 23-07-2025

Aim

To explore Python packages such as NumPy, SciPy, Pandas, Scikit-learn, and Matplotlib, and apply machine learning workflows on datasets from UCI and Kaggle repositories.

Objective

- Understand the core operations of libraries like NumPy, Pandas, SciPy, Scikit-learn, and Matplotlib.
- Perform array manipulations, scientific computing, model building, and visualization.
- Apply machine learning workflows to real-world datasets.
- Identify suitable machine learning tasks and algorithms for various datasets.
- Execute data loading, EDA, preprocessing, feature selection, model training, and evaluation.

Exploring Python Libraries

NumPy

- **Key Functions:** array, reshape, arange, linspace, zeros, ones, sum, mean, std, dot, random
- **Operations:** Vectorized computation, matrix algebra, broadcasting

Pandas

- **Functions:** DataFrame, read_csv, head(), info(), describe(), groupby(), merge(), pivot_table()
- **Data Cleaning:** dropna(), fillna(), astype(), map(), replace()

SciPy

- **Purpose:** Scientific computation and mathematical operations
- **Modules:** scipy.stats, scipy.optimize, scipy.integrate, scipy.signal, scipy.spatial

Scikit-learn

- **Tasks:** Classification, regression, clustering, model evaluation
- **Modules:** datasets, model_selection, preprocessing, metrics, linear_model, tree, svm

Matplotlib

- **Purpose:** Data visualization and graphical representation
- **Functions:** plot(), scatter(), bar(), hist(), imshow(), subplot(), title(), xlabel(), ylabel()

Machine Learning Workflow Steps

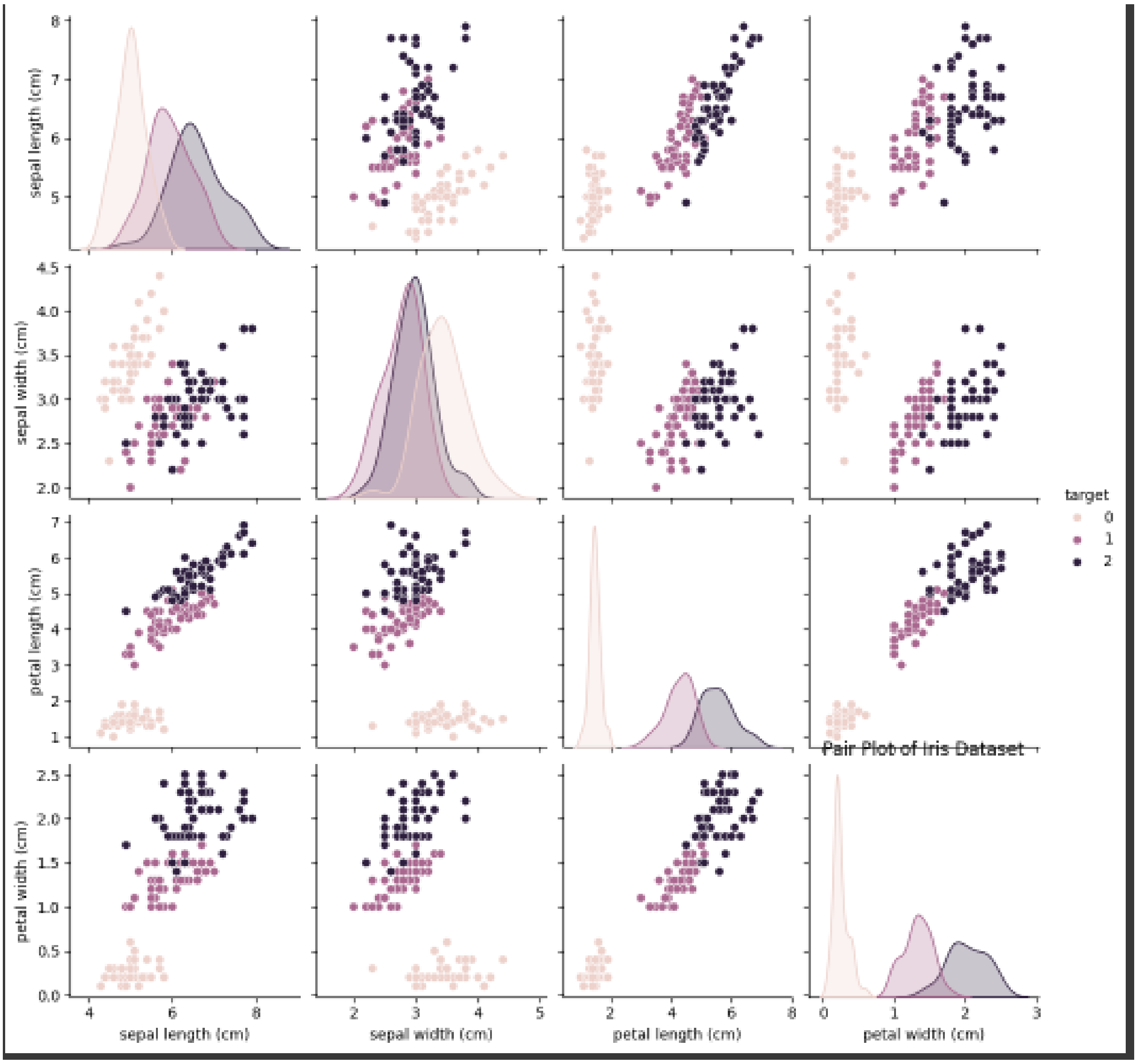
- Load dataset using Pandas (read_csv)
- EDA using describe(), info(), value_counts(), plotting
- Preprocessing: LabelEncoder, StandardScaler, handling NaN
- Feature Selection: SelectKBest, f_classif, chi2
- Split Data: train_test_split()
- Model Selection Training
- Performance Evaluation: Accuracy, confusion matrix, classification report

Sample code 1: Iris Dataset

ML Task: Supervised Classification

Python Code

```
from sklearn.datasets import load_iris
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

# Load dataset
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

# EDA
sns.pairplot(df, hue='target')
plt.title("Pair Plot of Iris Dataset")
plt.show()

# Feature Scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df.drop('target', axis=1))
y = df['target']

# Data Split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, rando
```



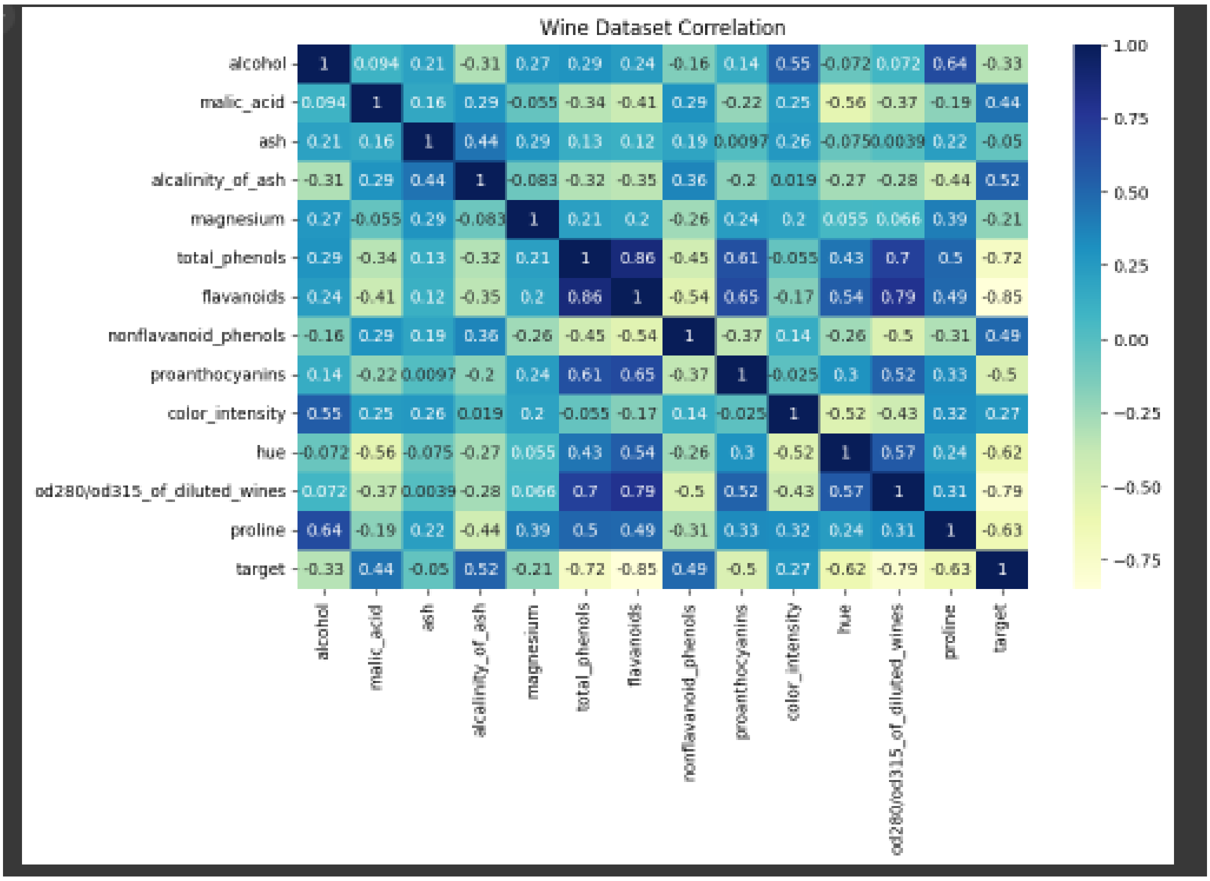
Sample code 2: Wine Dataset

ML Task: Supervised Classification

Python Code

```
from sklearn.datasets import load_wine
import pandas as pd
from sklearn.preprocessing import StandardScaler
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

# Load dataset
wine = load_wine()
df = pd.DataFrame(wine.data, columns=wine.feature_names)
df['target'] = wine.target
```

```
# Correlation Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap='YlGnBu')
plt.title("Wine Dataset Correlation")
plt.show()

# Preprocessing
X = df.drop('target', axis=1)
y = df['target']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, rando
```



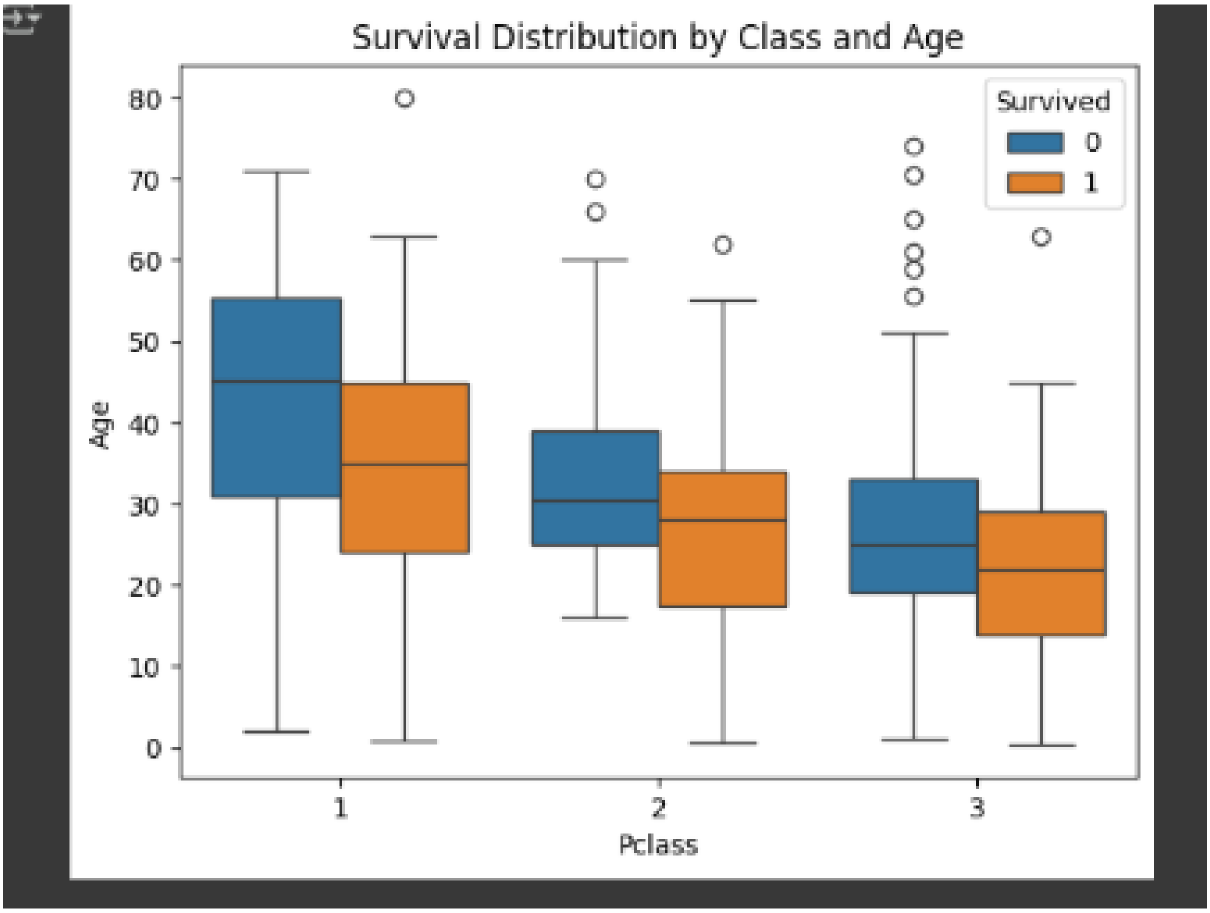
Sample code 3: Titanic Dataset

ML Task: Supervised Classification (Survival Prediction)

Python Code

```
import seaborn as sns
```

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

# Load dataset
df = sns.load_dataset('titanic')
df = df[['survived', 'pclass', 'sex', 'age', 'fare', 'embarked']]

# Handle missing values
df.dropna(inplace=True)

# Encode categorical variables
le = LabelEncoder()
df['sex'] = le.fit_transform(df['sex'])
df['embarked'] = le.fit_transform(df['embarked'])

# EDA
sns.boxplot(data=df, x='pclass', y='age', hue='survived')
plt.title("Survival by Class and Age")
plt.show()

# Preprocessing
X = df.drop('survived', axis=1)
y = df['survived']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.25, rand
```



Results and Discussions

Dataset	ML Task	ML Type and Source
Loan Amount Prediction	Regression	Supervised – Kaggle / UCI
Handwritten Digit Recognition	Classification	Supervised – MNIST
Email Spam Classification	Classification	Supervised – UCI SpamBase
MNIST Digits Classification	Classification	Supervised – MNIST
Predicting Diabetes	Classification	Supervised – PIMA Indian Dataset
Iris Dataset	Classification	Supervised – UCI Iris Dataset

Feature Selection and Algorithm Mapping

Dataset	**ML Task**	**Feature Selection**	**Suitable Algorithms**
Iris Dataset	Classification	ANOVA (f_classif), SelectKBest	Logistic Regression, KNN
Loan Prediction	Regression	SelectKBest (f_regression)	Linear Regression, Random Forest
Diabetes Prediction	Classification	Chi2, SelectKBest	SVM, Random Forest, XGBoost
Email Spam Classification	Classification	Chi2, Mutual Info	Naive Bayes, Decision Tree
MNIST Handwriting	Classification	PCA, CNN-based FS	KNN, SVM, CNN

Learning Outcomes

- Acquired practical experience using NumPy, Pandas, SciPy, Scikit-learn, and Matplotlib.
- Learned to load and analyze real-world datasets from UCI, Kaggle, and Seaborn repositories.
- Performed EDA using visual tools like pair plots, heatmaps, and boxplots.
- Applied data preprocessing techniques such as encoding and scaling.
- Used feature selection methods like SelectKBest (f_classif and chi2) to improve models.
- Understood how to build and evaluate machine learning models for different problem types.