Experiment 1: Working with Python packages - Numpy, Scipy, Scikit-Learn, Matplotlib

Sudharshan Vijayaragavan Reg No: 3122237001054

Academic Year: 2025-2026 (Odd)

1 Aim of the Experiment

The primary objective of this experiment is to gain practical experience with essential Python libraries for machine learning, including Numpy, Pandas, Scipy, Scikit-learn, and Matplotlib. The experiment involves exploring various functions, understanding their applications in data manipulation, preprocessing, and visualization, and applying these skills to different real-world datasets. The core tasks include identifying the appropriate machine learning models for specific problems and analyzing their performance.

2 Summary of Tasks

The following table summarizes the datasets, the type of machine learning tasks performed, the feature selection techniques used, and the suitable algorithms employed.

Dataset	**Type of ML Task**	**Feature Selection Technique**	**Suitable ML Algorithm**
hlineIris Dataset	Classification	SelectKBest	Logistic Regression
hlineLoan Amount Prediction	Regression	Not explicitly used in code	Linear Regression
hlinePredicting Diabetes	Classification	Feature Importance (Random Forest)	Random Forest Classifier
hlineClassification of Email Spam	Classification	Not explicitly used in code	Gaussian Naive Bayes
hlineHandwritten Character Recognition / MNIST hline	Classification	Not explicitly used in code	Logistic Regression

Table 1: Summary of ML tasks and models.

3 Python Code and Output Screenshots

The Python code provided was used to perform data preprocessing, model training, and evaluation for each dataset. The following sections include the key code snippets and corresponding output screenshots.

3.1 Iris Dataset

The Iris dataset is a classic example for a classification problem. The goal is to classify the species of an iris flower based on its sepal and petal measurements.

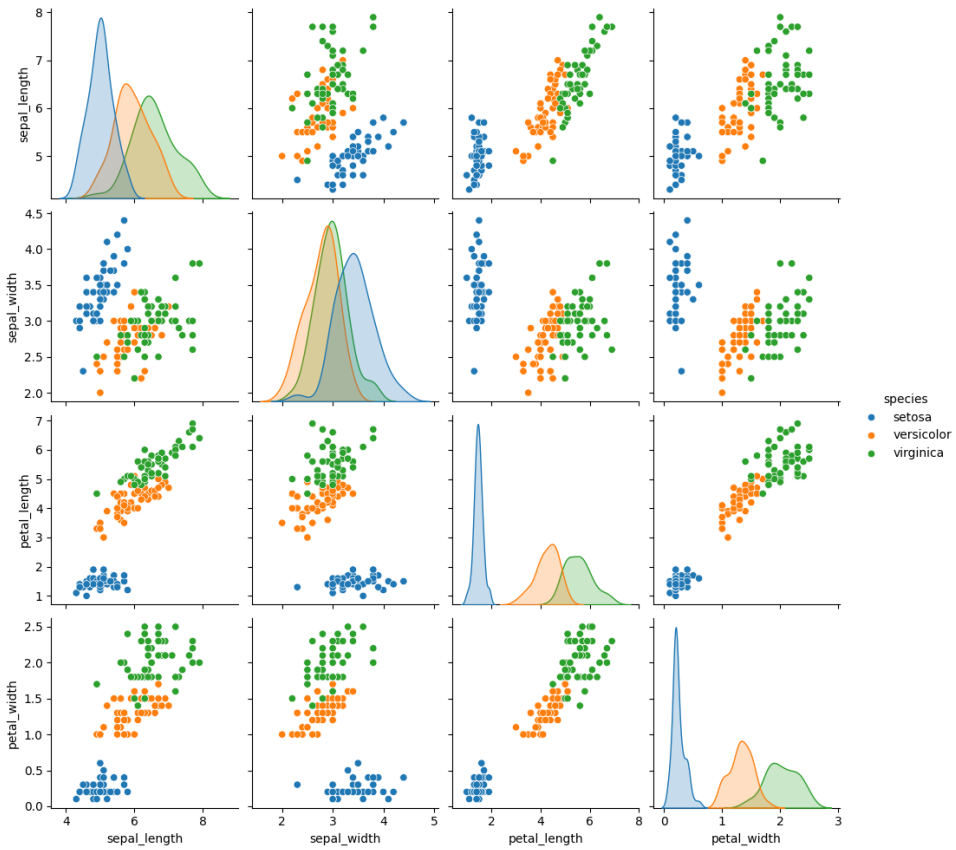

Figure 1: Pair plot visualization for the Iris dataset showing relationships and species separation.

```
import seaborn as snsimport matplotlib.pyplot as pltfrom sklearn.model_selection impo
```

```
--- IRIS DATASET ---
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

Figure 2: Confusion matrix and classification report for the Iris dataset.

3.2 Loan Amount Prediction Dataset

This task involves predicting a continuous value (loan amount), which is a regression problem. Linear Regression is a suitable model for this task.

Figure 3: Actual vs. Predicted Loan Amounts, with the red dashed line representing a perfect prediction.

```
import pandas as pdimport numpy as npimport seaborn as snsimport matplotlib.pyplot as
```

3.3 Diabetes Dataset

Predicting diabetes is a binary classification problem. A Random Forest Classifier is used to predict the outcome.

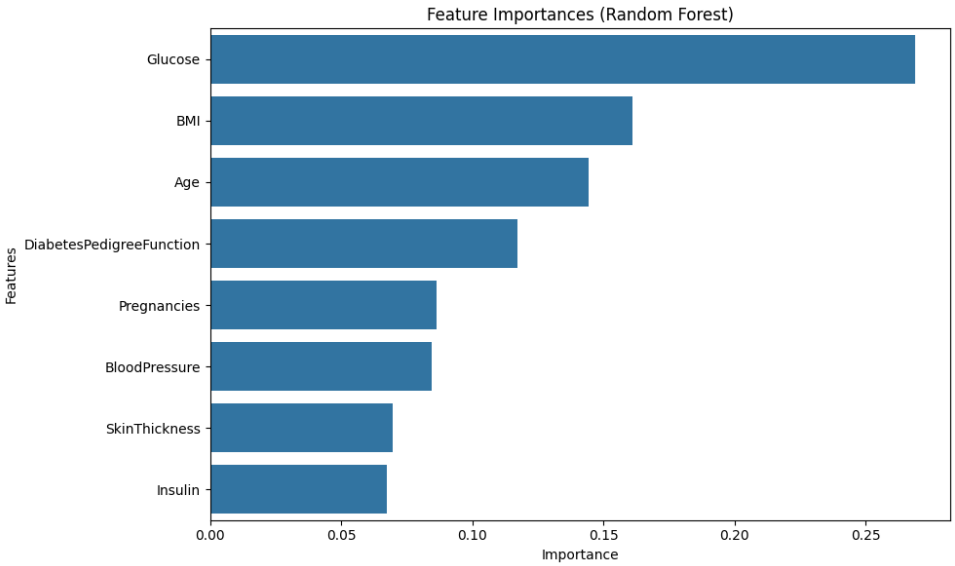

Figure 4: Feature importances from the Random Forest Classifier for the Diabetes dataset.

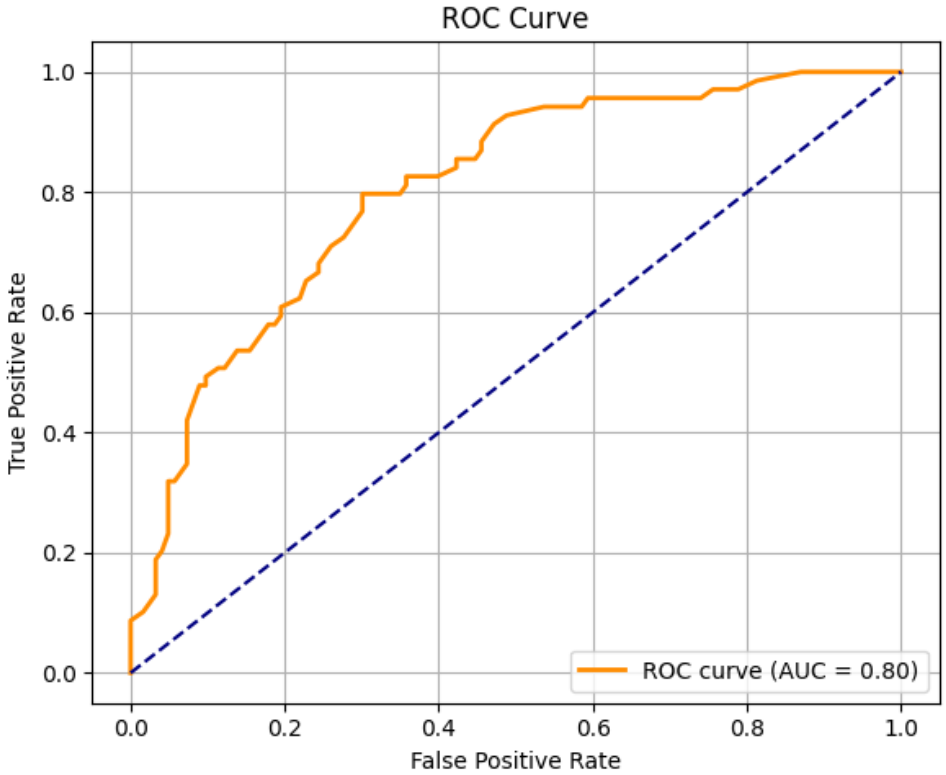

Figure 5: Receiver Operating Characteristic (ROC) curve for the Diabetes dataset.

```
from sklearn.ensemble import RandomForestClassifierfrom sklearn.metrics import accura
```

3.4 Email Spam Classification Dataset

This is another binary classification task, where the goal is to distinguish between spam and non-spam emails. A Naive Bayes classifier is a common choice for text-based classi-

fication.

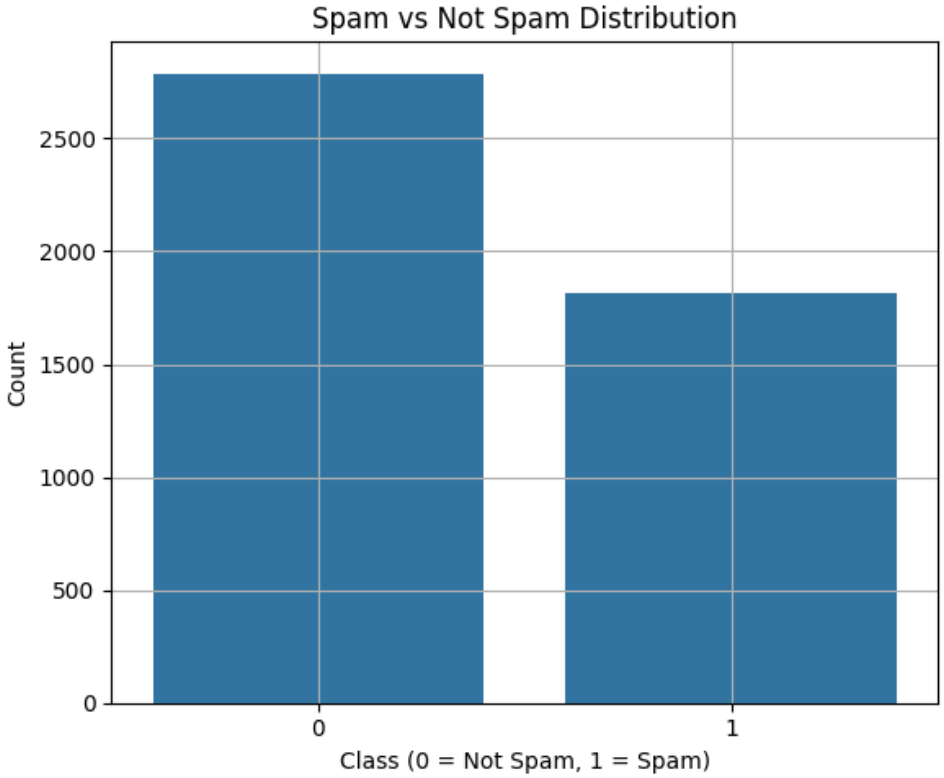

Figure 6: Distribution of spam vs. non-spam emails in the dataset.

```
from sklearn.naive_bayes import GaussianNBdf = pd.read_csv("/content/drive/MyDrive/sp
```

3.5 Handwritten Character Recognition / MNIST Dataset

Recognizing handwritten digits is a multi-class classification problem. The code uses Logistic Regression to classify the digits.

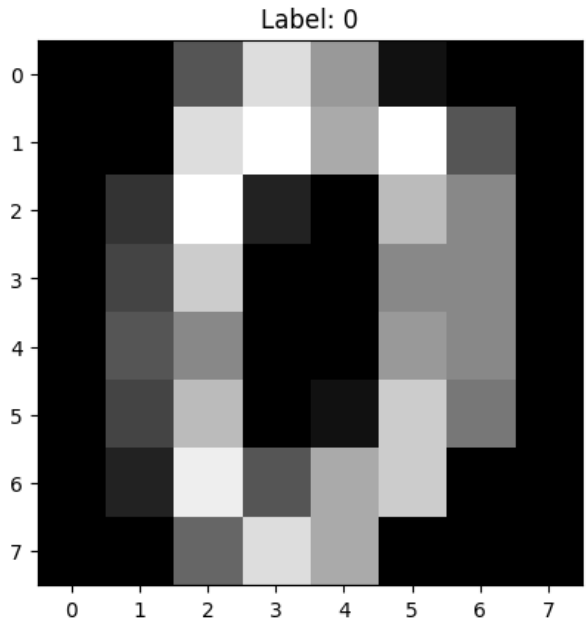

Figure 7: A sample handwritten digit from the MNIST dataset.

```
from sklearn.datasets import load_digitsdigits = load_digits()X = digits.datay = digi
```

4 Reflection on Learning Outcomes

This experiment provided a hands-on introduction to the machine learning workflow using powerful Python libraries. Key learning outcomes include:

- **Data Exploration:** Using Matplotlib and Seaborn, I learned to visualize datasets to gain initial insights into their structure, distributions, and relationships between features. The pair plot for the Iris dataset, for example, clearly showed how different species can be separated based on their features.
- **Data Preprocessing:** The experiment demonstrated the importance of data preprocessing steps like feature scaling with StandardScaler and handling categorical variables using pd.get$_dummies.Thisensuresthatalgorithms, especiallythosesensitivetofeaturem$
- **Performance Evaluation:** The experiment highlighted the significance of evaluating model performance using appropriate metrics. For classification, I used classification$_reportandconfusion$ $offs$.Overall, this experiment provided a solid foundation for understanding the practical aspects of machine learning development, from data loading to model evaluation.