Sri Sivasubramaniya Nadar College of Engineering, Chennai
(An autonomous Institution affiliated to Anna University)

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:**

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

1 Aim:

To understand and explore essential Python libraries used in machine learning, such as pandas, numpy, matplotlib, seaborn, and scikit-learn. This includes learning how to handle, visualize, and preprocess datasets effectively for model building.

Libraries Used

In this project, several Python libraries were used to facilitate data analysis, model building, and visualization. Below is a brief description of each library and its relevance to the experiment.

- **NumPy**
  NumPy (Numerical Python) is a fundamental package for scientific computing with Python. It provides support for arrays, matrices, and a large collection of mathematical functions. In this experiment, NumPy is used for internal mathematical operations, vectorized array computations, and support during model calculations.

- **Pandas**
  Pandas is a fast, powerful, and flexible open-source data analysis and manipulation library built on top of Python. It provides high-performance, easy-to-use data structures such as DataFrames. In this project, Pandas is used to load and inspect the dataset, handle missing values, encode categorical variables, and manipulate feature columns for training and evaluation.

- **Matplotlib**
  Matplotlib is a 2D plotting library for Python that enables the creation of static, interactive, and animated visualizations. In this assignment, Matplotlib is used to plot histograms, scatter plots, and predicted versus actual values to assess model performance and feature relationships.

- **Scikit-learn**
  Scikit-learn is a widely used machine learning library that offers simple and efficient tools for data mining and analysis. It includes modules for classification, regression, clustering, dimensionality reduction, and model evaluation. For this experiment, Scikit-learn is used to

build and train the Linear Regression model, split the dataset, evaluate model performance using metrics such as MAE, MSE, RMSE, and R^2 score.

- **Seaborn**
  Seaborn is a data visualization library based on Matplotlib that provides a high-level interface for creating attractive and informative statistical graphics. It is used in this project to generate correlation heatmaps, boxplots, and distribution plots for better visual interpretation of the dataset and model results.

2 Mathetical/theoritical description of the algorithm/objective performed:

2.1 Handling Missing Values

Missing values can negatively impact the performance of machine learning models by:

- Distorting statistical summaries
- Causing errors in training algorithms
- Leading to biased predictions

So, it's crucial to detect and properly handle them before modeling.

There are several ways to handle missing values:

- Missing values in a dataset can be handled using imputation techniques such as replacing them with the mean, median, or mode of the respective feature by using fillna() method in pandas
- If a column contains a large number of missing values and does not contribute significantly to the prediction task, it can be dropped to simplify the dataset. Removing such irrelevant or incomplete features helps reduce noise and improve model efficiency.

2.2 Label encoding:

To train machine learning models, all input features must be in numeric format. Hence, categorical variables (like "Yes"/"No", "Graduate"/"Not Graduate") need to be converted into numbers. This process is essential for enabling algorithms to interpret and process the data correctly.

- Categorical values can be directly replaced with numeric codes, such as mapping "Yes" to 1 and "No" to 0. This is useful when the categories are binary or have no specific order. It ensures compatibility with machine learning models that require numerical input.
- If a categorical feature has more than two values, simple replacement may introduce unintended ordinal relationships. In such cases, **one-hot encoding** is preferred, where each category becomes a separate binary column. This prevents the model from assuming any order or ranking between the categories.

2.3 Plotting:

To better understand the patterns and relationships within the dataset, various data visualization techniques are used. These plots help in identifying correlations, distributions, and outliers effectively.

- The **heatmap()** function visualizes the correlation between numerical features using a colored matrix. Darker shades typically represent stronger correlations, helping identify redundant or related features. It's a useful tool for understanding feature interdependencies before model building.
- A **histogram** displays the frequency distribution of a numeric variable, showing how data is spread across ranges. It helps detect skewness, modality, and presence of outliers or missing value gaps. This is often used as a first step in understanding individual feature behavior.
- A **box plot** (or whisker plot) shows the spread and central tendency of a feature using quartiles. It clearly identifies the median, interquartile range (IQR), and outliers. This makes it an excellent tool for spotting extreme values and data symmetry.

2.4 Removal of Outliers:

After detection of outliers using boxplot, we can remove the outliers by using any one of the below mentoined ways:

- This method involves **dropping the rows that contain outlier** values beyond a certain threshold (usually outside 1.5×IQR).It helps in reducing noise from the dataset, especially when outliers are errors or irrelevant. However, **excessive removal may lead to loss of valuable information** if not done carefully.
- Instead of deleting, outlier values can be **replaced with the column's mean or median to preserve data size**. Median is preferred when data is skewed, as it minimizes distortion. This method smooths the dataset without losing records, maintaining balance in feature distributions.

2.5 Standardization:

- Standardization is a feature scaling technique that **transforms data to have a mean of 0 and a standard deviation of 1**. It is especially useful when features have different units or scales, ensuring all variables contribute equally to the model. Many machine learning algorithms, like logistic regression and KNN, perform better when data is standardized.
- The formula used for standardization is:

$$z = \frac{x-\mu}{\sigma}$$

- where x is the original value, μ is the mean, and σ is the standard deviation. This process centers the data around zero and makes it easier for models to converge efficiently.

The preprocessing steps involved handling missing values, encoding categorical variables, visualizing data using heatmaps, histograms, and boxplots, and addressing outliers. Additionally, feature standardization was applied to bring all variables to a common scale, ensuring better model performance and stability.

Dataset	Type of ML Task	Suitable ML Algorithm
Iris Dataset	Multi-class Classification	KNN, SVM
Loan Amount Prediction	Regression	Linear Regression
Predicting Diabetes	Binary Classification	SVM, XGBoost
Classification of Email Spam	Binary Classification	Logistic Regression, SVM
Handwritten Character Recognition	Multi-class Classification	CNN, SVM

Table 1: ML Task and Suitable Algorithms for Different Datasets

3 Results and Discussions:

3.1 Handwritten Digit Recognition (MNIST)

This task involves identifying digits (0–9) from grayscale image data. Due to its image-based nature, Convolutional Neural Networks (CNNs) are the most suitable choice. SVM can also be considered when using pixel-intensity features after preprocessing.

3.2 Loan Amount Estimation

This is a regression problem aimed at predicting the loan amount based on applicant details. Linear Regression is a recommended algorithm due to the continuous nature of the output. Feature scaling and handling outliers are essential before modeling.

3.3 Iris Flower Classification

The Iris dataset is a classic example of multi-class classification with three flower types. K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) are well-suited for this problem. These algorithms handle class boundaries effectively with proper scaling.

3.4 Diabetes Prediction

This is a binary classification task to detect the likelihood of diabetes based on medical attributes. Support Vector Machines (SVM) and Logistic Regression are the most effective for such structured data. Normalization and feature selection improve model performance.

3.5 Email Spam Classification

The goal is to categorize emails as spam or not based on word frequency features. Logistic Regression and SVM work best for this high-dimensional, text-based dataset. TF-IDF vectorization is often used to convert text into meaningful numerical inputs.

4 Learning Practices:

- **Handling Incomplete Data:** Missing values were addressed by replacing them with statistical measures such as mean or median, or by removing irrelevant features to maintain dataset integrity.

- **Encoding Categorical Variables:** Categorical attributes were converted into numerical form using techniques like label encoding and one-hot encoding to ensure compatibility with machine learning algorithms.
- **Understanding Feature Impact:** Statistical correlation and visualization methods like heatmaps and boxplots were used to evaluate feature significance and detect redundancy or outliers.
- **Model Selection Awareness:** Based on the nature of each problem (classification or regression), appropriate machine learning models were selected, considering factors like data type, dimensionality, and target variable.