Sri Sivasubramaniya Nadar College of Engineering, Chennai

(An Autonomous Institution Affiliated to Anna University)

Degree & Branch: Integrated M.Tech. Computer Science & Engineering
Semester: V

Course Code & Title: ICS1512 - Machine Learning Algorithms Laboratory

Academic Year: 2025-2026 (Odd) **Batch:** 2023-2028

Name: Vishwajith L K **Reg. No.:** 3122237001061

Aim

To familiarize students with essential Python libraries for machine learning, perform exploratory data analysis on diverse datasets, and implement complete ML pipelines including preprocessing, training, and evaluation.

1. Overview of Python Libraries

NumPy

- Common functions: `np.array()`, `np.linspace()`, `np.sum()`, `np.dot()`.
- Key Uses: Multi-dimensional array handling, mathematical computations, and matrix operations.

Pandas

- Common functions: `pd.read_csv()`, `DataFrame.fillna()`, `groupby()`.
- Key Uses: Dataset manipulation, cleaning, merging, and time-series analysis.

SciPy

- Common functions: `scipy.stats.describe()`, `scipy.optimize.minimize()`.
- Key Uses: Scientific computing, numerical integration, hypothesis testing.

Scikit-learn

- Common functions: `train_test_split()`, `StandardScaler`, `GridSearchCV`.
- Key Uses: Preprocessing, model training, hyperparameter tuning, pipelines.

Matplotlib / Seaborn

- Common functions: `plt.plot()`, `plt.subplots()`, `sns.heatmap()`.
- Key Uses: Visualizations like histograms, boxplots, correlation maps.

2. Datasets and Machine Learning Tasks

Dataset	ML Task Type	Recommended Algorithm
Iris Flower Data	Classification	K-Nearest Neighbors (KNN)
Loan Status Dataset	Classification	Decision Trees / Random Forest
Diabetes Health Data	Regression	Linear Regression
Spam Email Dataset	Classification	Support Vector Machines (SVM)
MNIST Digit Images	Classification	Convolutional Neural Networks (CNN)

3. Machine Learning Workflow

1. **Data Acquisition:** Load datasets using `pandas.read_csv()` or inbuilt Scikit-learn datasets like `load_iris()`.

2. **Exploratory Data Analysis (EDA):**

    - Generate summary statistics using `df.describe()`, `df.info()`, and `df.isnull().sum()` to identify data quality issues.
    - Visualize feature distributions using histograms, KDE plots, and boxplots to detect outliers and skewed data.
    - Use correlation heatmaps (`sns.heatmap()`) and scatterplots (`sns.pairplot()`) to identify relationships between variables.
    - Check class balance for classification problems using bar plots; unbalanced classes can lead to biased models.

3. **Data Preprocessing:** Handle missing values, encode categorical variables, normalize and scale features.

4. **Feature Engineering:** Create new features using domain knowledge or transformations (e.g., logarithmic scaling).

5. **Data Splitting:** Divide into training, validation, and test sets (e.g., 70-20-10 split) using `train_test_split()`.

6. **Model Training:** Fit algorithms such as KNN, SVM, Random Forest, or CNN using Scikit-learn or Keras.

7. **Model Evaluation:** Use metrics such as Accuracy, F1-score, MSE, and visualize ROC/AUC curves and confusion matrices.

Learning Outcomes

- Developed practical understanding of major ML libraries (NumPy, Pandas, SciPy, Scikit-learn, Matplotlib, Seaborn).

- Learned to map datasets to suitable machine learning algorithms based on problem type.

- Implemented end-to-end workflows involving EDA, preprocessing, feature selection, and model building.
- Gained experience with hyperparameter tuning and interpreting evaluation metrics.