<table>
<tr><td colspan="6">Devi Ahilya Vishwavidyalaya, Indore, India
Institute of Engineering & Technology</td><td>III-Year
B.Tech.
(Electronics and Instrumentation Engineering)</td></tr>
<tr><td>Course Code & Name</td><td colspan="5">Instructions Hours per Semester and Credits</td></tr>
<tr><td rowspan="2">5REPC4

SOFTWARE WOKSHOP-II</td><td colspan="2">Classroom Instruction (CI)</td><td>Lab Instruction (LI)</td><td>Term Work (TW) and Self Learning (SL)</td><td>Total no. of Hours Per semester</td><td>Total Credits (Total Hours/30)</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>TW+SL</td><td rowspan="2">60</td><td rowspan="2">2</td></tr>
<tr><td>0</td><td>0</td><td>40</td><td>20</td></tr>
</table>

Course Learning Objectives:

1. To implement foundational scripting structures in Python using scientific libraries like NumPy, Pandas, Scikit-Learn, and Matplotlib for engineering datasets.
2. To apply informed and uninformed search techniques alongside core supervised and unsupervised algorithms to resolve real-world engineering problems.
3. To engineer standalone AI/ML applications incorporating hardware-software co-design concepts specifically suited for signalling, telemetry, and smart electronic automation.

Prerequisites:

Basic knowledge of c++, linear algebra, data structure, probability.

COURSE CONTENTS

Unit –I

Python environment setup for AI & ML programming: install and configure Python, Jupyter Notebook, and the essential libraries including NumPy, Pandas, Matplotlib, Seaborn and SciPy. Introduction to python programming for statistical modules, central tendency measure, measure of dispersion, covariance, correlation coefficient, Gaussian distribution, CDF and PDF, data visualization techniques using bar chart, pie chart, box plot, scatter plot and histogram.

Unit-II

Implementation of search strategies using Python programming: Uninformed Search - BFS, DFS and Informed Search- A* navigation, hill climbing variations. Introduction to Exploratory Data Analysis (EDA), including data loading and inspection, duplicate records, summary statistics, outlier detection, and univariate and bivariate data analysis. Data preprocessing: data cleaning, handling missing values, encoding categorical variables: label encoding & one hot encoding techniques, feature scaling, and feature transformation.

Unit-III

Introduction to supervised learning concepts, types of supervised learning (regression and classification), and the machine learning workflow including data preparation, model training, testing, and validation. Implementation of supervised learning algorithms using Python and the Scikit-learn library, including Linear Regression, Multiple Linear Regression, Logistic Regression, L1 (Lasso) and L2 (Ridge) Regularization, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Tree, and Random Forest. Feature selection, feature scaling, train-test split, and hyperparameter tuning.

Performance evaluation of ML models using appropriate metrics. Comparative analysis of supervised learning algorithms and their applications through case studies and real-world datasets using Python.

Unit-IV

Introduction to unsupervised learning concepts and clustering techniques. Data preprocessing for clustering, feature scaling, and dimensionality reduction basics. Implementation of unsupervised learning algorithms using Python and the Scikit-learn library, including K-Means Clustering and DBSCAN, with applications on real-world datasets. Cluster analysis and visualization, selection of optimal clusters using the Elbow Method and Silhouette Score, and comparative analysis of clustering algorithms. Introduction to deep learning concepts, artificial neural networks (ANN), perceptron, activation functions, forward and backward propagation, and implementation of basic deep learning models using Python. Performance evaluation of clustering and deep learning models using appropriate metrics. Practical implementation and application of unsupervised learning and deep learning techniques using real-world datasets.

Unit-V

Project based on the machine learning.

Course Outcomes:

CO. No.	CO
CO1	Demonstrate proficiency in Python programming for AI/ML by performing basic statistical analysis and data visualization using essential Python libraries.
CO2	Apply Python to implement search algorithms and perform exploratory data analysis and data preprocessing techniques for AI and Machine Learning applications.
CO3	Develop and assess supervised machine learning models for regression and classification using Python
CO4	Design and implement unsupervised learning and basic deep learning models using Python, and evaluate their performance on real-world datasets.
CO5	Build a complete working machine learning project application.

Books Recommended:

[1] Aurélien Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 3rd Edition, 2022
[2] Sebastian Raschka, Yuxi Liu, and Vahid Mirjalili, Python Machine Learning, Packt Publishing, 3rd Edition, 2019
[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 1st Edition, 2016
[4] Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter, O'Reilly Media, 3rd Edition, 2022

CO-PO-PSO Relationship

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO 1**	**PSO 2**	**PSO 3**
5REPC4.CO1	3	2	-		3	-	-	-	-	-	-	2	1	2	1
5REPC4.CO2	3	3	2	3	3	-	-	-	-	-	-	2	1	3	2
5REPC4.CO3	3	3	3	3	3	-	-	-	-	-	-	2	1	3	2
5REPC4.CO4	3	3	3	3	3	-	-	-	-	-	-	2	1	3	3

5REPC4.CO5	3	3	3	3	3	2	-	-	3	2	2	3	1	3	3

List of Practical Assignment

1. Installation and configuration of Python, Jupyter Notebook, and AI/ML libraries (NumPy, Pandas, Matplotlib, Seaborn, SciPy, and Scikit-learn).
2. Write Python programs to perform basic statistical operations, including measures of central tendency, measures of dispersion, covariance, and correlation.
3. Perform Exploratory Data Analysis (EDA) on a real-world dataset using Pandas, including data inspection, summary statistics, handling missing values, duplicate removal, and outlier detection.
4. Visualize datasets using Matplotlib and Seaborn by plotting bar charts, pie charts, histograms, box plots, scatter plots, and correlation heatmaps.
5. Implement data preprocessing techniques, including categorical data encoding, feature scaling, and train-test data splitting.
6. Implement Simple Linear Regression using Scikit-learn and evaluate the model using MAE, MSE and RMSE.
7. Implement Multiple Linear Regression and analyse the effect of multiple independent variables on prediction accuracy.
8. Implement Logistic Regression for a classification problem and evaluate the model using Confusion Matrix, Accuracy, Precision, Recall, and F1-Score.
9. Implement L1 (Lasso) and L2 (Ridge) Regularization and compare their effects on model performance.
10. Implement the K-Nearest Neighbours (KNN) algorithm for classification and compare its performance with Logistic Regression.
11. Implement Support Vector Machine (SVM) for classification and analyse the effect of different kernel functions.
12. Implement Decision Tree and Random Forest algorithms and compare their classification performance.
13. Perform comparative analysis of supervised learning algorithms using appropriate performance evaluation metrics.
14. Implement K-Means Clustering and determine the optimal number of clusters using the Elbow Method and Silhouette Score.
15. Implement DBSCAN clustering and compare its performance with K-Means on different datasets.
16. Implement a basic Artificial Neural Network (ANN) for classification or regression problems.
17. Evaluate the performance of the deep learning model using accuracy, loss curves, confusion matrix, and other relevant metrics.
18. Develop an application using any machine learning algorithm.