COURSE PLAN

Course Name & Code	DATA SCIENCE & COMPUTER APPLICATIONS
Course Name & code	Fundamentals of Machine Learning Lab & DSE 2242
Semester & branch	4th Semester, BTech
Name of the faculty	Padmashree G & Shavantrevva S B
No. of contact hours/week	3 hours/week

Continuous Evaluation	**60%**
Evaluations	Quiz - 14 Marks Execution - 14 Marks Record - 12 Marks Mid-Sem - 20 Marks **Max.Marks:60**
Lab Examination	**40%**
	Examination of 3 hours duration that includes questions based on: Python Basics, Machine Learning Algorithms and ensemble approach **Max.Marks:40**

Instructions to the Students

1. Students should be regular and come prepared for the lab practice.
2. In case a student misses a class, it is his/her responsibility to complete that missed exercise(s).
3. They should implement the given program individually.
4. Students should listen to the instructions given by the faculty.
5. Questions for lab tests and exams need not necessarily be limited to the questions in the manual but could involve some variations and/or combinations of the questions.

CONTENTS

Week	Topics to be covered
1	Basics of Python, Data Visualizations
2 -3	Regression - Linear, Multiple Linear Regression, Polynomial
4	Supervised Learning - Decision Trees, Random Forest, SVM, Naïve Bayes
5	Unsupervised Learning - KNN, K-Means
6 - 7	Dimensionality Reduction techniques - PCA, LDA , ICA
8 - 9	Ensemble Machine Learning
10 - 11	Mini- Project Implementation and Demo
12	End Sem Practical Examination

Course Outcomes (COs)

At the end of this course, the student should be able to:

CO1	Understand the basics of Machine learning using Python.
CO2	Design, evaluate, and analyze supervised machine learning algorithms.
CO3	Design, evaluate, and analyze unsupervised machine learning algorithms.
CO4	Design and analyze the ensembled approach of various machine learning classifiers.

Week 1

1. Find the mean, median, mode, variance, and standard deviation of a list. (import statistics)
2. Create a list of numbers and use list comprehension to generate a new list with the squares of each number.
3. Write a function that takes a tuple of numbers and returns the product of all the elements.
4. Write a program that counts the frequency of each letter in each string and stores the results in a dictionary.
5. Explore two different datasets provided in the shared folders with the following steps:
    a. Import the required libraries.
    b. Load the dataset.
    c. Examine data information.
    d. Identify and analyze null values using df. isnull().
    e. Conduct statistical analysis. (count, mean, SD, min, max, Quartile)
    f. Generate boxplots for each column to detect outliers.
    g. Determine quartiles and remove outliers.
    h. Calculate correlations and visualize them with a heatmap.
    i. Normalize the various features of the given datasets. (MinMaxScaler)
    j. Arrange correlation values for each attribute in ascending order.
6. Conduct a visual exploration of the given dataset by employing four distinct Python plotting libraries: Matplotlib, Seaborn, Bokeh, and Plotly. Perform the following visualizations:
    a. Scatter plots with coloured points and adjustable sizes using columns (1) col1, col2 and (2) col3, col4 (3) col5, col6.
    b. Line charts for columns (1) col1, col2 (2) col3, col4 (3) col5, col6.
    c. Bar charts representing columns (1) col1, col2 (2) col3, col4 (3) col5, col6.
    d. Histograms to depict the dataset's distribution.
    e. Employ Bokeh for interactive data visualization.
    f. Utilize Bokeh's GUI features, including buttons, sliders, checkboxes, and radio buttons.

g. Employ Plotly to create a dropdown menu.

h. Develop custom action buttons using Plotly.

Week 2

Linear Regression

(a) Develop a custom function for Simple Linear Regression with one predictor attribute and one response attribute. The function should return the coefficients of the linear regression line.

(b) Utilize the provided sample dataset with attributes 'week' and 'sales' (in thousands) to construct a linear regression model for sales prediction(manual evaluation). Print the regression equation.

x_i (Week)	y_j (Sales in Thousands)
1	1.2
2	1.8
3	2.6
4	3.2
5	3.8

(c) Plot the regression lines for visual representation.

(d) What are the sales predictions for the 7th and 12th week?

(e) Calculate and display accuracy measures, including RMSE and R2.

(f) Repeat the process using built-in functions from the sklearn package.

(g) Download any accessible dataset for linear regression and repeat steps (a) to (f).

Multiple Linear Regression

(a) Create a custom function for Multiple Linear Regression with two predictor attributes (x1, x2) and one response attribute (y). The function should return the coefficients of the linear regression plane.

(b) Employ the provided sample dataset with attributes (x1, x2) and y to establish a multiple linear regression model.

SUBJECT	Y	X_1	X_2
1	-3.7	3	8
2	3.5	4	5
3	2.5	5	7
4	11.5	6	3
5	5.7	2	1
6	?	3	2

(c) What will be the prediction when x1 = 3 and x2 = 2?

(d) Calculate and display accuracy measures, including RMSE and R2.

(e) Repeat the process using built-in functions from the sklearn package.

(f) Download any accessible dataset for multiple linear regression and repeat steps (a) to (e).

Random Data Generation

(a) Utilize the random package to generate random numbers.

(b) Build linear and multiple linear regression models with the generated random numbers.

(c) Plot the regression lines for visual representation.

Week 3

Implement Logistic Regression in Python

a) Import the dataset.

b) Begin by pre-processing the dataset. Detail the steps involved in cleaning, handling missing values, and encoding categorical variables.

c) Write a Python function or script to fit a Logistic Regression model to the training set. Specify the necessary parameters and configurations for the model training.

d) After training the Logistic Regression model, implement code to predict the test results using the trained model.

e) Evaluate the accuracy of the model on the test set. Create a confusion matrix to further analyze the classification results.

f) Develop code to visualize the results of the Logistic Regression model on the test set.

Implement Linear, multiple linear, logistic, and polynomial regression models using standard machine learning datasets by partitioning the data into training, validation, and test datasets.

Week 4

Implement Decision Tree Algorithm in Python

a) Import the dataset.
b) Initiate the implementation by performing data pre-processing steps.. Detail the steps involved in cleaning, handling missing values, and encoding categorical variables.
c) Develop a Python function or script to fit a Decision Tree algorithm to the training set. Specify the key hyperparameters and configurations involved in training the Decision Tree model.
d) Implement code to predict the test results using the trained Decision Tree model.
e) Evaluate the accuracy of the decision tree model on the test set. Generate a confusion matrix to analyze the model's performance.
f) Create visualizations to represent the results of the Decision Tree model on the test set. Create visualizations to represent the results of the Decision Tree model on the test set.

Implement Random Forest Algorithm in Python

g) Import the dataset.
h) Begin by pre-processing the dataset. Detail the steps involved in cleaning, handling missing values, and encoding categorical variables.
i) Write a Python function or script to fit a Logistic Regression model to the training set. Specify the necessary parameters and configurations for the model training.
j) After training the Logistic Regression model, implement code to predict the test results using the trained model.
k) Evaluate the accuracy of the model on the test set. Create a confusion matrix to further analyze the classification results.

l) Develop code to visualize the results of the Logistic Regression model on the test set.

Week 5

Supervised Learning

Implement support vector machine algorithm in Python.

a) Import the dataset.
b) Begin by pre-processing the dataset. Detail the steps involved in cleaning, handling missing values, and encoding categorical variables.
c) Write a Python function or script to fit the SVM classifier to the training set. Specify the necessary parameters and configurations for the model training. Use different kernels and evaluate
d) After training the SVM classifier, implement code to predict the test results using the trained model.
e) Create the confusion matrix.
f) Visualize the training data results.
g) Visualize the test data results.

Implement Naïve Bayes algorithm in Python

a) Import the dataset.
b) Begin by pre-processing the dataset. Detail the steps involved in cleaning, handling missing values, and encoding categorical variables.
c) Write a Python function or script to fit the Naïve Bayes classifier to the training set. Specify the necessary parameters and configurations for the model training. Use different kernels and evaluate
d) After training the Naïve Bayes classifier, implement code to predict the test results using the trained model.
e) Create the confusion matrix.
f) Visualize the training data results.
g) Visualize the test data results.

Week 6

Unsupervised Learning

Implement K-Means algorithm in Python

a) Import the dataset.
b) Begin by pre-processing the dataset. Detail the steps involved in cleaning, handling missing values, and encoding categorical variables.
c) Extract independent variables
d) Finding the optimal number of clusters using the elbow method
e) Write a Python function to train the K-Means algorithm on the training set. Specify the necessary parameters and configurations for the model training.
f) After training the K-Means algorithm, visualize the clusters.

Week 7

Dimensionality Reduction Techniques

Implement Principal Component Analysis (PCA) in Python. Perform the following

a) Import all the libraries.
b) Load Data
c) Apply PCA
    a. Standardize the dataset prior to PCA.
    b. Import PCA from sklearn.decomposition.
    c. Choose the number of principal components.
d) Check Components
e) Plot the components (Visualization)
f) Calculate variance ratio

Week 8

Dimensionality Reduction Techniques

Implement Linear Discriminant Analysis(LDA) in Python. Perform the following.

a) Install and import relevant libraries
b) Read and load the data
c) Preprocess the data
d) Perform exploratory data analysis
    a. Pair Plots
    b. Histograms
    c. Correlation heatmaps
e) Split the data set
f) Implement LDA
g) Visualize the data
h) Classify the data with random forest
i) Evaluate the LDA model

Week 9

Ensemble approach

Considering various regression and classification models, explore the following ensemble approaches

a) Averaging method
b) Max voting
c) Stacking
d) Blending
e) Bagging
f) Boosting

Implementation Steps:

Importing Necessary Libraries: Begin the implementation by importing the required Python libraries for machine learning.

Creating Dataset: Generate or import a suitable dataset for regression or classification tasks. Explain the characteristics of the dataset chosen and its relevance to the ensemble methods.

Initializing the Models: Initialize different regression and classification models, considering a diverse set of algorithms for ensemble techniques.

Fitting Training Data: Train each model on the training dataset. Specify the hyperparameters and configurations used for each model during the training phase.

Testing the Model: Evaluate the performance of each ensemble method on the test dataset. Discuss the metrics used for assessment and compare the results obtained from the different approaches.

Week 10 & Week 11

Project work

Machine Learning Lab Project: Implementing and Evaluating Multiple Algorithms

Problem Statement: Select a real-world problem or scenario suitable for machine learning applications. Clearly articulate the problem you aim to solve or analyze using machine learning techniques.

Dataset Selection: Choose a relevant dataset that aligns with your selected problem. Justify why this dataset is appropriate for your machine learning project.

Data Pre-processing: Implement data pre-processing techniques on the chosen dataset. Include steps such as handling missing values, encoding categorical variables, and scaling features. Explain how these steps contribute to preparing the data for modeling.

Algorithm Selection: Select at least three different machine learning algorithms studied in the course (e.g., Decision Trees, Logistic Regression, Support Vector Machines, K-nearest neighbors, etc.). Justify the choice of each algorithm based on the problem and dataset characteristics.

Model Training: Train each selected machine learning algorithm on the pre-processed training data. Specify the hyperparameters chosen for each algorithm and explain the reasoning behind these choices.

Model Evaluation: Evaluate the performance of each trained model on a separate test dataset. Use appropriate evaluation metrics (e.g., accuracy, precision, recall, F1-score) to assess the strengths and weaknesses of each algorithm.

Comparative Analysis: Conduct a comparative analysis of the results obtained from different algorithms. Discuss the advantages and limitations of each algorithm in the context of your problem.

Hyperparameter Tuning: Choose one of the algorithms and perform hyperparameter tuning to optimize its performance. Discuss the impact of hyperparameter tuning on the model's results.

Visualization: Create visualizations that help interpret and communicate the results. This can include confusion matrices, ROC curves, or other relevant plots to showcase the models' performance.

Conclusion and Recommendations: Summarize your findings and provide recommendations based on the performance of the implemented machine learning algorithms. Discuss potential areas for improvement or further exploration.

Week 12

END SEM LAB EXAM

Submitted by: Padmashree G & Shavantrevva S B

(Signature of the faculty)

Date: 3/01/2024

Approved by: Dr. Radhika M Pai

(Signature of HOD)
Date: 3/01/2024