IARE

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad, Telangana

COURSE CONTENT

DATA MINING AND WAREHOUSING LABORATORY								
V Semester: CSE (DS)								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
ACDD14	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		0	0	2	1	40	60	100
Contact Classes: Nil	**Tutorial Classes: Nil**	**Practical Classes: 45**				**Total Classes: 45**		
Prerequisites: Database Management Systems								

I. COURSE OVERVIEW:

The Data Mining and Warehousing lab provides practical experience in working with data using Python. Students will learn how to use libraries like NumPy, Pandas, SciPy, and Matplotlib for tasks such as building data models, performing data cleaning, and analyzing data. The course covers topics like data mining techniques (classification, clustering, association), time series analysis, and creating visualizations. By the end of the course, students will be equipped to solve real-world data problems and apply their skills in data mining and warehousing.

II. COURSE OBJECTIVES

The students will try to learn:

I. Create and visualize data models like Star and Snowflake using Python.

II. Apply data mining tasks and machine learning algorithms to real-world datasets.

III. Conduct time-series forecasting, develop interactive dashboards, and extract web data for analysis.

III. COURSE OUTCOMES

At the end of the course students should be able to:

CO1: Design and **visualize** Star and Snowflake schemas, and distinguish OLAP vs. OLTP queries

CO2: Construct multi-dimensional data models and perform OLAP operations using Python.

CO3: Implement data cleaning and integration techniques by removing errors, merging datasets, and reducing dimensionality.

CO4: Apply data mining methods like classification, clustering, and association rule mining.

CO5: Build and **evaluate** machine learning models for classification, regression, and clustering.

CO6: Develop visualizations, handle outliers, and perform data scaling and transformation for analysis.

IV. COURSE CONTENT:

WEEK 1: Data Warehousing Basics

1. Write a Python script to implement and visualize a Star Schema.
2. Write a Python program to implement a Snowflake Schema.
3. Simulate and differentiate OLAP vs. OLTP queries using Python.
4. Simulate a three-tier data warehouse architecture in Python, including a data source, a staging area, and a presentation layer. Populate the layers with sample data and visualize the flow.

WEEK 2: Multi-dimensional Data Models

1. Create and analyze a multi-dimensional array using NumPy.

2. Perform OLAP operations (roll-up, drill-down, slice, dice, and pivot) using Pandas.
3. Visualize a Fact Constellation schema with Matplotlib.
4. Design and implement an ETL (Extract, Transform, Load) pipeline in Python to populate a Star Schema and visualize OLAP operations using a real-world dataset.

WEEK 3: Data Preprocessing

1. Perform data cleaning (removing null values, duplicates) on a sample dataset using Pandas.
2. Apply data integration techniques to merge datasets.
3. Use Python to reduce data dimensionality using Principal Component Analysis (PCA).
4. Write a Python script to preprocess a dataset for a machine learning pipeline, including advanced transformations such as polynomial feature generation and custom feature engineering**.**

WEEK 4: Data Mining Concepts

1. Write a Python program to classify data mining tasks (classification, clustering, association).
2. Create and process a dataset to explore data mining functionalities.
3. Implement basic concept hierarchies on a dataset.
4. Create a Python program to implement a rule-based data mining system. Use real-world data to extract and evaluate insights using custom mining rules.

WEEK 5: Association Rule Mining

1. Implement the Apriori algorithm using Python to generate frequent itemsets.
2. Write a Python program to calculate support and confidence for association rules.
3. Use the FP-Growth algorithm to find frequent itemsets in a dataset.
4. Write a Python program to compare the performance of the Apriori and FP-Growth algorithms on a large dataset (e.g., a transactional dataset from Kaggle).

WEEK 6: Classification

1. Implement a decision tree classifier using scikit-learn.
2. Use Python to train and test a Naive Bayes classifier.
3. Build a neural network model using TensorFlow or PyTorch for classification tasks.
4. Build an ensemble classifier using Python by combining decision trees, Naive Bayes, and logistic regression. Evaluate its performance using k-fold cross-validation on a real-world dataset.

WEEK 7: Prediction

1. Write a Python script to predict numerical data using linear regression.
2. Implement random forest regression and compare its performance with linear regression.
3. Evaluate model accuracy using metrics like MAE, RMSE, and R2.
4. Develop a Python application for time-series forecasting using LSTM (Long Short-Term Memory) networks. Train the model on financial or weather data and evaluate its performance.

WEEK 8: Clustering

1. Implement the K-means clustering algorithm using scikit-learn.
2. Perform hierarchical clustering and visualize the dendrogram using scipy.
3. Use DBSCAN to detect clusters in spatial data.
4. Implement a custom clustering algorithm, such as Agglomerative Clustering, with Python and compare its results with K-means and DBSCAN on a high-dimensional dataset**.**

WEEK 9: Dataframe Operations

1. Perform attribute filtering using conditions, slicing, and queries on a dataset.
2. Compute and interpret ranking and summary statistics.
3. Handle missing values using statistical and interpolation techniques.
4. Write a Python program to create a dynamic filtering dashboard using Streamlit or Dash. Allow users to filter attributes interactively and visualize the results.

WEEK 10: Time Series Analysis

1. Parse and analyze time-series data using pandas.
2. Compute rolling mean and standard deviations for a time-series dataset.
3. Plot trends and seasonality of time-series data.
4. Perform seasonal decomposition of a time series using Python (statsmodels). Analyze trends, seasonal, and residual components, and apply SARIMA modeling for forecasting**.**

WEEK 11: Data Visualization

1. Plot categorical data using bar charts, grouped bar charts, and stacked bar charts.
2. Visualize numerical data distributions using histograms and density plots.
3. Use box-and-whisker plots to identify data distribution and outliers.
4. Develop an interactive data visualization tool using Plotly or Bokeh to visualize categorical and numerical data dynamically, with filters for real-time analysis.

WEEK 12: Handling Outliers

1. Identify outliers in a dataset using the quartile and standard deviation methods.
2. Remove outliers and visualize the clean dataset.
3. Compare the impact of outlier removal on dataset analysis.
4. Implement advanced outlier detection using Isolation Forest or One-Class SVM in Python. Compare these methods with traditional quartile-based and standard deviation methods.

WEEK 13: Data Scaling and Transformation

1. Scale data using Min-Max scaling and StandardScaler from scikit-learn.
2. Normalize and transform data using logarithmic and square root transformations.
3. Compare the results of different scaling methods and interpret.
4. Use advanced data transformation techniques like Quantile Transformation and Power Transformation in Python. Compare their effects on the performance of a machine learning model.

WEEK 14: Web Scraping

1. Scrape product data (name, price, rating) from an e-commerce website using BeautifulSoup.
2. Extract tabular data from a webpage and convert it to a DataFrame.
3. Scrape images from a website and save them locally.
4. Build a Python scraper using Selenium to scrape dynamic content from a JavaScript-heavy website. Save the data in a structured format and visualize insights using Pandas and Matplotlib.

V. TEXTBOOKS

1. **Dushyant Singh Sengar, Vikash Chandra**, *"Modern Data Mining with Python"*, BPB Publications, 2023.

2. **Dr. Jugnesh Kumar**, *“Data Warehouse and Data Mining”*, BPB Publications, 2023.

VI.REFERENCE BOOKS:

1. **Meta S. Brown**, *“Data Mining For Dummies”*, Wiley, 2023.
2. **Jake VanderPlas**, *“Python Data Science Handbook: Essential Tools for Working with Data”*, O’Reilly Media, 2017.
3. **Aurélien Géron**, *“Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”*, O’Reilly Media, 2nd Edition, 2019.
4. **Charu C. Aggarwal**, *“Data Mining: The Textbook”*, Springer, 2015.

VII. ELECTRONIC RESOURCES

1. https://www.python.org/
2. https://realpython.com/
3. https://www.khanacademy.org/computing/computer-programming
4. https://jakevdp.github.io/PythonDataScienceHandbook/
5. https://scikit-learn.org/stable/
6. https://seaborn.pydata.org/
7. https://github.com/jakevdp/PythonDataScienceHandbook

VIII. MATERIALS ONLINE

1. Course Content
2. Lab Manual