Register Number														

SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing
SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamil Nadu
Academic Year: 2024-25 (EVEN)

Set -

Test: FT4 Date: 29-04-2025
Course Code & Title: 21CSS303T-Data Science Duration: Two periods
Year& Sem: III Year /VI Sem Max.Marks:50

Course Articulation Matrix:

Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO3	-	-	-	-	1	-	-	-	-	-	-	-
CO4	-	-	-	-	1	-	-	-	-	-	-	-
CO5	-	-	-	-	1	-	-	-	-	-	-	-

Note: CO3 – To identify data manipulation and cleaning techniques using pandas
CO4 – To constructs the Graphs and plots to represent the data using python packages
CO5 – To apply the principles of the data science techniques to predict and forecast the outcome of real-world problem

Part – A (10 x 1 = 10 Marks)

Instructions:
1) Answer **ALL** questions.
2) The duration for answering Part A is **15 minutes** (this sheet will be collected after 15 minutes).
3**) Encircle the correct answer**.

S.No	Question	Marks	BL	CO	PO	PI Code
1	State the data wrangling operation that handles errors, missing data and inconsistencies a. Validation b. Data enrichment c. Cleaning d. Organization	1	1	3	5	5.4.1
2	Name the pandas method that can be used to combine DataFrames using one or more keys, as in database join operations a. pandas.concat b. pandas.merge c. DataFrame.combine_first d. DataFrame.join	1	1	3	5	5.4.1
3	Define the objective of imputation process a. Remove entire rows or columns containing missing values b. Remove pairs of observations where at least one value is missing c. Replacing missing data with estimated values d. Remove noise from the dataset using some algorithms	1	1	3	5	5.4.1
4	Identify the reshape process among the following that turns unique values from one column into new column headers, effectively transforming long-form data to wide -form a. Melting b. Stacking c. Pivoting d. Unstacking	1	2	3	5	5.4.1
5	Which among the following is a common measure of dispersion of data a. median b. standard deviation	1	2	3	5	5.4.1

	c. histogram d. skewness					
6	In Matplotlib, which of the following correctly creates a subplot at position 5 in a 4-row by 3-column grid? a. plt.subplot(3, 4, 5) b. plt.subplot(5, 3, 4) c. plt.subplot(4, 3, 5) d. plt.subplot(5, 4, 3)	1	1	4	5	5.5.2
7	From the below list, recall the construct used to add text or markers to specific locations on a plot to highlight particular features a. Legends b. Labels c. Annotations d. Ticks	1	1	4	5	5.4.1
8	Among the following statements, recognize the **correct** statement about Python's matplotlib.pyplot package a. pyplot is used only for 3D plotting in Python. b. pyplot automatically displays plots without the need to call show(). c. pyplot provides a MATLAB-like interface for creating static, interactive, and animated plots. d. pyplot cannot save plots in pdf format.	1	1	4	5	5.5.1
9	Identify the Seaborn package feature that allows you to visualize relationship between all pairs of numeric columns in DataFrames a. FacetGrid b. Pairplot c. Scatterplot d. subplot	1	2	5	5	5.4.1
10	Identify the **incorrect** statement regarding seaborn package a. Seaborn is a data visualization library built on top of Matplotlib b. Seaborn allow us to represent data points in three-dimensional space c. Seaborn can be imported using import matplotlib.seaborn as sns d. Seaborn can be used to visualize textual data by creating wordcloud	1	2	5	5	5.5.1

Register Number														

SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing
SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamil Nadu
Academic Year: 2024-25 (EVEN SEM)

Set -

Test: FT4 Date:29-04-2025
Course Code & Title: 21CSS303T-Data Science Duration: Two periods
Year& Sem: III Year /VI Sem Max.Marks:50

Part – B (4 x 5 = 20 Marks)
Instructions: Answer **ANY FOUR** Questions

Q. No	Question	Marks	BL	CO	PO	PI Code
11	Discuss different data structures that help optimize memory and computation while handling large data volumes. Briefly review their strengths and weaknesses.	5	2	3	5	5.6.1

Ans:
Data structures have different storage requirements, but also influence the performance of CRUD (create, read, update, and delete) and other operations on the data set

• **Tree** is a hierarchical data structure where each node has a parent and may have child nodes, used for searching and sorting. Trees are a class of data structure that allows you to retrieve information much faster than scanning through a table

• **Hash** is a key-value data structure that provides fast lookups using a hash function. A key for every value in your data and put the keys in a bucket. This way you can quickly retrieve the information by looking in the right bucket when you encounter the data.

Dictionaries in Python are a hash table implementation, and they're a close relative of key-value stores

• **Sparse data** refers to datasets with mostly zero or missing values, stored efficiently to save memory.

12	Given the following scenario, perform appropriate data cleaning, transformation, and merging steps: Dataset A contains employee records with columns: ***EmpID, Name, Age***, and ***Department***. Some age values are missing, and department names have inconsistent casing (e.g., "HR", "hr", "Hr"). Dataset B contains salary details with columns: ***EmpID, MonthlySalary***. Write Python code (using pandas) to: 1. Clean the ***Age*** using suitable imputation 2. Clean the ***Name*** by removing unnecessary spaces 3. Apply standardize capitalization on the column ***Department***. 4. Merge the two datasets on EmpID. 5. Display the total salary aggregated on the ***Department*** column (You may assume dummy data for illustration.) **Ans:** 1. Convert datasets to DataFrames df_a = pd.DataFrame(data_a) df_b = pd.DataFrame(data_b) 2. Clean the Age column using suitable imputation df_a['Age'].fillna(df_a['Age'].mean(), inplace=True) 3. Clean the Name column by removing unnecessary spaces df_a['Name'] = df_a['Name'].str.strip() 4. Standardize capitalization of the Department column df_a['Department'] = df_a['Department'].str.capitalize() 5. Merge the two datasets on EmpID merged_df = pd.merge(df_a, df_b, on='EmpID') 6. Display the total salary aggregated by the Department total_salary_by_dept = merged_df.groupby('Department')['MonthlySalary'].sum().reset_index() 7. Display the result print(total_salary_by_dept)	5	3	3	5	5.5.2
13	Distinguish between Z-score normalization and Min-max normalization. Under what data conditions would each method be more appropriate? **Ans:** **Z-score normalization** is a data preprocessing technique that transforms numerical data to have a mean of 0 and a standard deviation of 1. This is particularly useful when dealing with features that have different scales or units, as it ensures that all features contribute equally to the model. z = (x - mean) / standard_deviation ; $z = \frac{x - \mu}{\sigma}$ Advantages: 1. Handles different Scales 2. Improves Machine Learning Models 3. Reduce Bias 4. Helps with outliers	5	2	3	5	5.6.1

	Min-max normalization is a data preprocessing technique that scales numerical data to a specific range, typically between 0 and 1. It's useful when you want to preserve the relative distances between data points while ensuring that all features have a similar scale The formula used is: x_scaled = (x - min(x)) / (max(x) - min(x))					
14	Write the python code for creating s 2 X 2 grid of plots with the following subplots using matplotlib.pyplot 1. Grid 1 – line plot 2. Grid 2 – Scatter plot 3. Grid 3 – Bar 4. Gid 4 – histogram (You may assume dummy data (Qno:12) for illustration.) **Ans:** import matplotlib.pyplot as plt import numpy as np **#Data** x = np.arange(1, 6) y = x ** 2 categories = ['A', 'B', 'C', 'D', 'E'] values = [5, 7, 3, 8, 6] hist_data = np.random.randn(1000) **#Plotting** plt.figure(figsize=(10, 8)) plt.subplot(2, 2, 1) plt.plot(x, y, marker='o') plt.title('Line Plot') plt.subplot(2, 2, 2) plt.scatter(x, y, color='green') plt.title('Scatter Plot') plt.subplot(2, 2, 3) plt.bar(categories, values, color='orange') plt.title('Bar Plot') plt.subplot(2, 2, 4) plt.hist(hist_data, bins=20, color='purple') plt.title('Histogram') plt.tight_layout() plt.show()	5	3	4	5	5.5.2
15	You are given a dataset that contains the daily temperature (Temp), humidity (Humidity), and air quality index (AQI) recorded over 5 days . Days = [1,2,3,4,5] Temperature = [23,25,28,32,35] AQI = [3,5,4,2,5]	5	3	5	5	5.5.2

	Write Python code using **Seaborn and Matplotlib** to visualize the relationship among these three variables using a **3D line plot**, where: • X-axis → Day (as a sequence) • Y-axis → Temperature • Z-axis → AQI **Ans:** import matplotlib.pyplot as plt import seaborn as sns **# Data** Days = [1, 2, 3, 4, 5] Temperature = [23, 25, 28, 32, 35] AQI = [3, 5, 4, 2, 5] **# Create 3D plot** sns.set(style="whitegrid") fig = plt.figure(figsize=(8, 6)) ax = fig.add_subplot(111, projection='3d') **# Plotting the 3D line** ax.plot(Days, Temperature, AQI, marker='o', color='blue', label='Temp vs AQI') **# Label axes** ax.set_xlabel('Day') ax.set_ylabel('Temperature (°C)') ax.set_zlabel('AQI') ax.set_title('3D Line Plot of Day vs Temperature vs AQI') **# Show plot** plt.legend() plt.show()					

Part – C (2 x 10 = 20 Marks)
Instructions: Answer ALL questions.

Q. No	Question	Marks	BL	CO	PO	PI Code
16 a	How missing values are represented in a dataset? With examples, describe the various imputation techniques used for handling of missing values so that there is minimum loss of information. **Ans:** Imputation is the process of replacing missing data with estimated values to maintain dataset integrity. **Mean/Median/Mode Imputation**: Replace missing values with the mean, median, or mode of the respective column. This is a simple approach but can introduce bias if the distribution is skewed When to Use: •Mean: Best for normally distributed data. •Median: Preferred when data is skewed or has outliers. •Mode: Used for categorical data.	10	2	3	5	5.5.1

	K-Nearest Neighbors (KNN) Imputation: Impute missing values using the average values of the k nearest neighbors. This method can be effective for numerical data. **Regression Imputation:** Use regression models to predict missing values based on other features. This is suitable for numerical data with strong relationships between features. **Multiple Imputation**: Create multiple imputed datasets by filling in missing values with different plausible values. This method can help to account for uncertainty in the imputation process **Choosing the right approach** The best approach for handling missing values depends on the nature of your data, the amount of missing data, and the specific requirements of your analysis. Consider the following factors: • **Amount of missing data**: If there are many missing values, imputation might be preferable to deletion. • **Distribution of missing data**: If missingness is random, imputation might be suitable. If missingness is related to other variables, more sophisticated techniques might be necessary. • **Impact of missing data on the analysis:** If missing values are likely to bias your results, it's important to address them. Give a simple example.					
	(OR)					

16 b	You are given a Pandas DataFrame containing a column Customer_Info with inconsistent entries like:
	
	" Mr. Ramesh K , Chennai - 600001 "
	"Ms. PRIYA D,COIMBATORE-641002"
	"Dr. Arjun,Madurai - 625001"
	"Mrs. Leela S , Chennai - 6251 "
	
	Perform the following tasks using Pandas string manipulation methods:
	1. Strip leading and trailing whitespaces from the entire Customer_Info column.
	2. Replace all hyphens - with a single space and convert multiple spaces to a single space.
	3. Extract the following components into new columns:
	o Title (Mr., Ms., Dr., etc.)
	o Name (in uppercase)
	o City (in title case)
	4. Pad the PIN code column (if needed) so that all valid entries have 6 digits (e.g., "6251" becomes "006251").
	
	Ans:
	import pandas as pd
	
	# Create dataframe
	data = {
	'Customer_Info': [
	" Mr. Ramesh K , Chennai - 600001 ",
		10	3	3	5	5.5.2

```
        "Ms. PRIYA D,COIMBATORE-641002",
        "Dr. Arjun,Madurai - 625001",
        "Mrs. Leela S , Chennai -  6251 "
    ]
}

df = pd.DataFrame(data)
```

1. Strip leading and trailing whitespaces

```
df['Customer_Info'] = df['Customer_Info'].str.strip()
```

2. Replace hyphens with space and normalize multiple spaces

```
df['Customer_Info'] = df['Customer_Info'].str.replace('-', ' ', regex=False)
df['Customer_Info'] = df['Customer_Info'].str.replace(r'\s+', ' ', regex=True)
```

3. Extract Title, Name, City, and PIN using regex

```
df[['Title', 'Name', 'City', 'PIN']] = df['Customer_Info'].str.extract(
    r'(Mr\.|Mrs\.|Ms\.|Dr\.)\s+([A-Za-z\s]+),?\s*([A-Za-z]+)\s+(\d+)', expand=True
)
```

4. Format extracted fields

```
df['Name'] = df['Name'].str.upper().str.strip()
df['City'] = df['City'].str.title().str.strip()
```

5. pad PIN with zeros if less than 6 digits

```
df['PIN'] = df['PIN'].str.zfill(6)


print(df[['Title', 'Name', 'City', 'PIN']])
```

17 a	Explain the features of Seaborn library. Also describe the importance of Facet Grid, joint plot and pair plot with example implementation.	10	2	4	5	5.5.1

Ans:

- **Seaborn** is a library mostly used for statistical plotting in Python.
- It is built on top of Matplotlib and provides beautiful default styles and color palettes to make statistical plots more attractive.

Features of Seaborn

Statistical Graphics: Seaborn is specifically designed for creating statistical graphics, providing built-in functions for common visualizations like scatter plots, line plots, histograms, and more. This makes it easier to create visually appealing and informative plots for data analysis.

Data Visualization Themes: Seaborn offers pre-defined styles and themes that can quickly change the overall appearance of your

plots. This helps create consistent and aesthetically pleasing visualizations without requiring extensive customization.

Integration with Pandas and NumPy: Seaborn seamlessly integrates with Pandas and NumPy, making it easy to work with dataframes and arrays directly. This simplifies the workflow and reduces the amount of code needed for data analysis and visualization.

FacetGrid and Pair Plots: Seaborn provides FacetGrid for grouping data and creating subplots based on categorical variables. This is useful for comparing distributions or relationships across different groups. Pair plots allow you to visualize the relationships between all pairs of numeric columns in a DataFrame, helping you identify correlations and patterns.

Customization and Flexibility: While Seaborn provides a high-level interface, it's built on top of Matplotlib, giving you access to its extensive customization options. This allows you to fine-tune your plots to meet your specific needs.

Ease of Use: Seaborn's API is designed to be user-friendly and intuitive, making it easier to learn and use compared to Matplotlib. Its documentation is also well-written and provides clear examples.

3D Plots

FacetGrid: Group data by a categorical variable and plot individual subplots for each category.

g = sns.FacetGrid(df, col="hue", height=4)

Jointplot: Visualize the relationship between two variables and their distributions.

sns.jointplot(x='x', y='y', kind="scatter", data =data)

Pairplot: Visualize the relationships between all pairs of numeric columns in a DataFrame.

sns.pairplot(df)

(OR)

17 b | You are provided with a sample dataset of product sales in a CSV file named product_sales.csv. The dataset contains the following columns: | 10 | 3 | 5 | 5 | 5.5.2

Product_ID	Category	Region	Units_Sold	Sale_Price
P001	Electronics	South	120	14500
P002	Furniture	North	75	9800
P003	Electronics	East	10	13200
P004	Clothing	West	160	3200
P005	Furniture	South	90	8900
P006	Electronics	East	110	15000
P007	Clothing	North	140	3000

Using Seaborn, generate:

Ans:
Import Libraries

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the CSV file
df = pd.read_csv('product_sales.csv')

# Set Seaborn style
sns.set(style='whitegrid')

#1.  Histogram of Units_Sold
plt.figure(figsize=(8, 5))
sns.histplot(df['Units_Sold'], bins=10, kde=True, color='skyblue')
plt.title('Distribution of Units Sold')
plt.xlabel('Units Sold')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()

#2.  Box plot of Sale_Price by Category
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='Category', y='Sale_Price', palette='Set2')
plt.title('Sale Price by Product Category')
plt.xlabel('Product Category')
plt.ylabel('Sale Price')
plt.tight_layout()
plt.show()
```

- A histogram showing the distribution of Units_Sold for all products.
- A box plot comparing Sale_Price across different Category values.

Course Outcome (CO) and Bloom's level (BL) Coverage in Questions