|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RegisterNumber** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**SRM Institute of Science and TechnologyCollegeofEngineeringandTechnology**

Set- A

**SchoolofComputing**

SRMNagar,Kattankulathur–603203,ChengalpattuDistrict,TamilNadu

# AcademicYear:2024-25(Even)

Test: FT1 Date:25-02-2025

CourseCode&Title:21CSS303T-Data Science Duration:50 Minutes

Year& Sem: IIIYear /VISem Max.Marks:25

CourseArticulationMatrix:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course  Outcome | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CO1 | - | - | - | - | 1 | - | - | - | - | - | - | - |
| CO2 | - | - | - | - | 1 | - | - | - | - | - | - | - |

**Note:** CO1 - To understand the relationship between data

CO2 - Identify the different data structures to represent data

**Part– A**

(5x2= 10 Marks)

Answer ALL the questions

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Q.No | Question | Marks | BL | CO | PO | PI.Code |
| 1 | How do you concatenate two Numpy arrays along a specified axis?  Use numpy.concatenate() to concatenate two NumPy arrays along a specified axis.  *Ex code:*  *import numpy as np*  *arr1 = np.array([[1, 2], [3, 4]])*  *arr2 = np.array([[5, 6]])*  *result = np.concatenate((arr1, arr2), axis=0)*  *# Concatenates along rows*  *print(result*) | 2 | 2 | 1 | 5 | 5.4.1 |
| 2 | How can you filter rows of a Pandas DataFrame based on a condition?  use the .loc[] or boolean indexing method in Pandas.  *Ex Code:*  *import pandas as pd*  *data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35]}*  *df = pd.DataFrame(data)*  *filtered\_df = df[df['Age'] > 28] # Selects rows where Age > 28*  *print(filtered\_df)* | 2 | 3 | 1 | 5 | 5.4.1 |
| 3 | Write a python program to get the positions of items of ser2 in ser1 as a list.  Input:  ser1 = pd.Series([10, 9, 6, 5, 3, 1, 12, 8, 13])  ser2 = pd.Series([1, 3, 10, 13])  Code: *import pandas as pd*  *ser1 = pd.Series([10, 9, 6, 5, 3, 1, 12, 8, 13])*  *ser2 = pd.Series([1, 3, 10, 13])*  *positions = [ser1[ser1 == val].index[0] for val in ser2]*  *print(positions) # Output: [5, 4, 0, 8]* | 2 | 3 | 2 | 5 | 5.4.2 |
| 4 | What is the difference between a Pandas Series and a DataFrame?   **Pandas Series**: A one-dimensional labeled array that can hold any data type (like a column in a table).   **Pandas DataFrame**: A two-dimensional table-like structure with labeled rows and columns (like a spreadsheet). | 2 | 1 | 2 | 5 | 5.4.1 |
| 5 | What is Web Scraping? Explain the steps involved with an example.  **Web Scraping** is the process of extracting data from websites using automated scripts. **Steps:**   1. Send an HTTP request to the website. 2. Parse the HTML content. 3. Extract the required information. 4. Store the data in a structured format (CSV, database, etc.).   Ex Code:  *import requests*  *from bs4 import BeautifulSoup*  *url = "https://example.com"*  *response = requests.get(url)*  *soup = BeautifulSoup(response.content, "html.parser")*  *print(soup.title.text) # Extracts page title* | 2 | 1 | 1 | 5 | 5.6.1 |

**Part– B**

(3x5= 15 Marks)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Q.No | Question | Marks | BL | CO | PO | PI.Code |
| 1. | Imagine you are working as a Data Scientist for an e-commerce company that wants to improve customer satisfaction by analyzing user behavior on their platform. Your task is to collect and analyze data to identify patterns that impact customer experience and purchase decisions. Brief the different phases involved in your assignment.  **1. Phases involved in data analysis for customer behavior in an e-commerce platform**  When working as a Data Scientist for an e-commerce company, analyzing user behavior involves multiple phases:  **1. Data Collection:**   * Gather data from various sources, such as user clicks, product views, purchase history, and customer reviews. * Data can be obtained from databases, web logs, APIs, or third-party sources.   **2. Data Cleaning and Preprocessing:**   * Handle missing values, duplicate records, and incorrect data. * Standardize formats (e.g., date formats, categorical values). * Remove irrelevant or noisy data (e.g., bot-generated interactions).   **3. Exploratory Data Analysis (EDA):**   * Use statistical methods and visualizations to identify trends and patterns. * Example: Finding which products are frequently bought together. * Tools: Pandas, Matplotlib, Seaborn for data exploration.   **4. Feature Engineering and Data Transformation:**   * Extract meaningful features from raw data. * Example: Creating a "customer lifetime value" feature based on past purchases. * Convert categorical data into numerical format for machine learning models.   **5. Model Building and Analysis:**   * Apply machine learning algorithms (e.g., clustering for customer segmentation, recommendation systems for personalized shopping). * Example: Predicting which users are likely to abandon their cart. * Use Scikit-Learn, TensorFlow, or PyTorch for modeling.   **6. Visualization and Reporting:**   * Present insights using dashboards, reports, and visualizations. * Example: Using Tableau or Power BI to display sales trends. * Helps stakeholders make data-driven decisions.   By following these phases, an e-commerce company can improve customer satisfaction and increase sales through better user experience. | 5 | 3 | 1 | 5 | 5.4.1 |
| 2. | Explain the following Numpy operations with an example   * Indexing of array * Slicing of array * Reshaping of array   Joining and splitting of arrays NumPy Operations Explained with Examples**1. Indexing of Array** Indexing in NumPy allows accessing specific elements within an array. NumPy arrays are zero-indexed, meaning the first element is at index 0.  **Example:**  python  CopyEdit  import numpy as np  arr = np.array([10, 20, 30, 40, 50])  print(arr[0]) # Accessing the first element  print(arr[2]) # Accessing the third element  **Output:**  CopyEdit  10  30  For multi-dimensional arrays:  python  CopyEdit  arr2D = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])  print(arr2D[1, 2]) # Accessing element at row index 1, column index 2  **Output:**  CopyEdit  6 **2. Slicing of Array** Slicing extracts a portion of an array using the syntax [start:stop:step].  **Example:**  python  CopyEdit  arr = np.array([10, 20, 30, 40, 50])  print(arr[1:4]) # Extract elements from index 1 to 3  print(arr[:3]) # Extract first three elements  print(arr[::2]) # Extract every second element  **Output:**  csharp  CopyEdit  [20 30 40]  [10 20 30]  [10 30 50]  For multi-dimensional arrays:  python  CopyEdit  arr2D = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])  print(arr2D[:2, 1:]) # Extract first two rows, last two columns  **Output:**  lua  CopyEdit  [[2 3]  [5 6]] **3. Reshaping of Array** Reshaping changes the structure of an array without modifying the data.  **Example:**  python  CopyEdit  arr = np.array([1, 2, 3, 4, 5, 6])  reshaped = arr.reshape(2, 3) # Convert 1D array into 2x3 2D array  print(reshaped)  **Output:**  lua  CopyEdit  [[1 2 3]  [4 5 6]] **4. Joining and Splitting of Arrays****Joining Arrays** Combining multiple arrays using functions like np.concatenate().  **Example:**  python  CopyEdit  arr1 = np.array([1, 2, 3])  arr2 = np.array([4, 5, 6])  joined = np.concatenate((arr1, arr2))  print(joined)  **Output:**  csharp  CopyEdit  [1 2 3 4 5 6]  For multi-dimensional arrays:  python  CopyEdit  arr1 = np.array([[1, 2], [3, 4]])  arr2 = np.array([[5, 6], [7, 8]])  joined = np.concatenate((arr1, arr2), axis=1) # Join along columns  print(joined)  **Output:**  lua  CopyEdit  [[1 2 5 6]  [3 4 7 8]] **Splitting Arrays** Splitting an array into multiple parts using np.split().  **Example:**  python  CopyEdit  arr = np.array([10, 20, 30, 40, 50, 60])  split\_arrays = np.split(arr, 3) # Split into 3 equal parts  print(split\_arrays)  **Output:**  scss  CopyEdit  [array([10, 20]), array([30, 40]), array([50, 60])]  For multi-dimensional arrays:  python  CopyEdit  arr2D = np.array([[1, 2, 3], [4, 5, 6]])  split\_arrays = np.hsplit(arr2D, 3) # Split columns into 3 parts  print(split\_arrays)  **Output:**  lua  CopyEdit  [array([[1],  [4]]), array([[2],  [5]]), array([[3],  [6]])] | 5 | 2 | 2 | 5 | 5.4.2 |
| 3. | Describe various ways of data acquisition. Discuss the significance of Web APIs, Open Data Sources, and Web Scraping with practical examples.  **1. Manual Entry:**   * Data is manually collected from surveys, reports, or research papers. * Suitable for small datasets but time-consuming for large-scale analysis.   **2. Database Queries:**   * Extracting data from relational databases like MySQL, PostgreSQL. * Example SQL query: * Used for structured and historical data analysis.   **3. Web APIs (Application Programming Interfaces):**   * APIs provide programmatic access to data from various platforms. * Example: Fetching weather data using an API. * Used in automation, machine learning applications, and real-time data analysis.   **4. Open Data Sources:**   * Government and research institutions provide free datasets. * Example sources:   + **Kaggle** (https://www.kaggle.com/datasets)   + **UCI Machine Learning Repository**   + **Google Dataset Search** * Used in academic research, public policy analysis, and training machine learning models.   **5. Web Scraping:**   * Extracts data from websites automatically. * Steps involved:   1. Send an HTTP request to the website.   2. Parse the HTML content using BeautifulSoup.   3. Extract relevant information.   4. Store the data in CSV, database, etc. | 5 | 2 | 2 | 5 | 5.4.1 |

**Course Outcome (CO)andBloom’s level (BL)Coverage in Questions**