INSTITUTE OF INNOVATION

Semester - 1 Evaluation Assignment

Python

1. **Exploratory Data Analysis of Customer Travel Preferences and Behaviours**

Objective:- The primary objective of this Exploratory Data Analysis (EDA) is to uncover insights into customer demographics, behaviours, and preferences within the context of travel product offerings. By investigating various aspects of the data, such as age distribution, contact type, city tier, occupation, and purchasing behaviour, the analysis aims to:

A. What is the distribution of customerages, and how does age relate to product purchase (ProdTaken)?

B. How does the type of contact(type of contact)influence the likelihood of purchasing a travel product?

C. Is there a correlation between the city tier(citytier)and monthly income (monthly income) of customers?

D. What is the average duration of a pitch (duration of pitch)for customers who purchased a product versus those who did not?

E. How do occupation types(occupation)distribute among customers,and is there an occupation type that is more likely to purchase a travel product?

F. Are there any gender-based preferences for the type of product pitched (ProductPitched)?

G. What is the relationship between the number of trips(number of trips)taken by a customer and their likelihood to purchase a new travel product?

H. Does having a passport(passport)correlate with a higher number of trips taken or a higher likelihood of purchasing a travel product?

I. How satisfied are customers with the pitch (pitch satisfaction score),and does this satisfaction influence product purchase?

J. Among customers with children (number of children visiting), how does the number of children impact travel product purchases?

K. Gender preferences for product types: how does gender influence the preference for different types of travel products pitched (product pitched)?

L. Impact of marital status on travel preferences: does marital status affect the choice of travel product, and if so, how do different marital statuses correlate with product purchases (prod taken)?

M. Relationship between number of children and travel product interest: how does the number of children visiting (number of children visiting) impact the likelihood of purchasing a travel product?

N. Influence of owning a car on travel decisions: is there a correlation between owning a car (own car) and the number of trips taken (number of trips) or the type of travel product purchased?

O. Income level and product choice: how does the monthly income (monthly income) of customers influence their choice of travel products, and is there a preferred product for different income levels?

2. Container 2 Question

2.1 Building a Secure Authentication System

Objective: Writing a Python script for setting a password involves incorporating various parameters to ensure the password meets specific security criteria. Below are questions that outline different requirements for such a script, each introducing new conditions to make the password more secure:

Minimum Length: The password must be at least 10 characters long. Character Variety: It must contain at least:

- Two uppercase letters.
- Two lowercase letters.
- Two digits.
- Two special characters (e.g., @, #, $, %, &, *, !).

Sequence and Repetition Restrictions:

- The password should not contain any sequence of three or more consecutive characters from the username (if provided).
- No character should repeat more than three times in a row (e.g., "aaa" is allowed, but "aaaa" is not).

Historical Password Check:

- The new password must not be the same as the last three passwords used by the user.

Solution Approach

1. Define a function to validate the password, taking the password, username, and a list of the last three passwords as parameters.

2. Use regular expressions and other string operations to perform the various checks.

3. Implement logic to ensure that the password does not contain sequences from the username and does not match any of the last three passwords.

4. Provide clear feedback to the user about which criteria the password fails to meet.

5. Loop until the user provides a valid password.

2.2 Automated Library Management System

You've been tasked with designing a Python script for a small library to automate their book checkout and return process, including overdue book tracking and fine calculations.

Requirements:

Catalog Management:

- The system should have a predefined list (catalog) of books, each with a unique ID, title, author, and quantity available.
- Implement functionality to display the current catalog to users, including the availability status of each book.

User Registration:

- Implement a simple user registration system that captures and stores user information, including name and a unique user ID.

Book Checkout Process:

- Allow users to check out books by book ID, with a limit of 3 books per user at any time.
- Record the checkout date for each book. Books are due back 14 days after checkout. Book Return Process:
- When a book is returned, update the catalog to reflect the increased availability.

Task:

- If a book is returned after its due date, calculate the overdue fine ($1 per day overdue).

Overdue Books Tracking:

- Provide functionality to list all overdue books for a particular user, along with the total fine due.

Develop a Python script that implements the above functionalities. Use data structures like dictionaries and lists to manage users, books, and transactions. Ensure that your script can handle basic edge cases, such as checking out an unavailable book or returning a book that wasn't checked out.

Advanced Features (Optional):

- Implement a simple command-line interface (CLI) for users to interact with the system.
- Add functionality to add or remove books from the catalog.
- Include a feature for users to extend their due date by 7 days for a book, with a limit of one extension per checkout.

Competitive Programming

Attempt all the problems, try various approaches and compare their performance in terms of space and time complexities. Your submission will be judged based on the correctness of the code first and then on the time and space complexity. You are free to use any programming language of your choice.

1. Given an empty list and a stream of N numbers. Print min, max, sum, average and mode (optional and if there are multiple modes then print any) after insertion of each element from the stream to the list.

   Example Input:
   5
   2 4 3 2 -3

   Example output

   - min, max, sum, average and mode after addition of 2 is 2, 2, 2, 2, 2.
   - min, max, sum, average and mode after addition of 4 is 2, 4, 6, 3, 4.
   - min, max, sum, average and mode after addition of 3 is 2, 4, 9, 3, 4.
   - min, max, sum, average and mode after addition of 2 is 2, 4, 11, 2.75, 2.
   - min, max, sum, average and mode after addition of -3 is -3, 3, 8, 1.6, 2.

2. Given a number N, Print first N prime number starting from 2 and skipping every alternate prime number

   Example Input:
   5

   Example Output:
   2 5 11 17 23

   Explanation :

   - First few prime numbers are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31 …
   - First five alternate prime numbers will be 2, 5, 11, 17, and 23

3. Given two sorted arrays, merge them to get another sorted array
   Example Input:
   5 #size of first array
   2 4 11 17 91 #first array

   4 #size of second array
   3 7 9 100 #second array

   Example Output :
   2 3 4 7 9 11 17 91 100

4. Given three numbers a, b and m. Calculate (a^b % m)
   Example input :
   2 5 3
   Example output :
   2
   Explanation :
   2^5 % 3 = 32 % 3 = 2

5. Given a sorted array. Write functions using binary search to calculate following things:

. lower_bound(x) : index of first element which is greater or equal to x in the given array
. upper_bound(x): index of first element which is greater than x in the given array
. is_present(x): return true if x is present in the array else return false

Core Web Development

1. Create your own Portfolio Website using HTML + CSS + JavaScript

   Pointers to keep in mind:

   a. Design of your website
   b. Details like LinkedIn, GitHub, Experience, Qualification, Projects, Contact information etc. should be properly present in your web page.
   c. Your Portfolio website should be hosted on Vercel or Netlify and Domain Name of your Website should be "your-name.service-provider.domain"
      Example: anuragtiwari.vercel.app
   d. GitHub Account should be properly setup (Example: GitHub Home Page, Project Repository)

2. Create a clone of website you like the most and also make sure not to use anything extra than what is thought inside the class [Single Page Clone].

Data Science

1. Student Registration Form

Objective:
Develop a web-based student registration system using Flask for the backend, HTML for the frontend, and MySQL as the database. The system should allow students to register by filling out a form. Upon submission, the user's information should be stored in a MySQL database.

Requirements:

Database Setup:
A. Create a MySQL database named registration_db.

B. Insideregistration_db,create a table named users with the following columns:
id: An auto-incrementing primary key.
Student name: A string to store the student's name.
Father name: A string to store the student's father's name. mother name: A string to store the student's mother's name. Phone number: A string to store a student's phone number. email: A string to store the student's email address.
Date of birth: A date to store the student's email address Address: A string to store the student's address
Blood Group: A string to store the student's blood group address Department: A string to store the student's Department address Course: A string to store the student's Course address password: A string to store the student's password.

2. Flask Application

Set up a Flask application with the following endpoints:
A. A root(/) end point that renders the registration form.
B. A/register endpoint that handles the form submission via POST request,saves the data to the MySQL users table, and then redirects to the root endpoint.

3. HTML Form:
A. Create an HTML form in a file named register.html with the following fields:

- Student name: A string to store the student's name.
- Father name: A string to store the student's father's name.
- mother name: A string to store the student's mother's name.
- Phone number: A string to store a student's phone number.
- email: A string to store the student's email address.
- Date of birth: A date to store the student's email address
- Address: A string to store the student's address
- Blood Group: A string to store the student's blood group address
- Department: A string to store the student's Department address
- Course: A string to store the student's Course address
- password: A string to store the student's password.

The form should submit the data to the register endpoint of your Flask application.

1. Write the SQL commands to create the required database and table(s).
2. Implement the Flask application with the specified endpoints and database connection logic.
3. Design the HTML form for user registration.
4. Ensure that the Flask application inserts the form data into the MySQL database upon **submission**.
1. Add form validation to ensure that all fields are filled out before submission.
2. Implement password hashing to store passwords securely in the database.
3. Create a user login page and functionality to authenticate users against the database.

Submission:
Provide the following in your submission:

1. The SQL commands for database and table creation.
2. The complete Flask application code, including route handlers and database connection logic.
3. The HTML code for the registration form.
4. Instructions on how to run your application and any additional setup required.

2. Dataset: Download here
"How have national sports programs influenced the evolution of Olympic success, and what patterns of improvement and dominance can be observed over time across different countries and sports disciplines?"
Solve Below Questions to perform analysis on Olympics (ANY 15)

A. Retrieve all records from the Olympics table.
B. Retrieve only the Athlete names and the Medals they won
C. Count the total number of athletes in the dataset.
D. Find all records of athletes who won a Gold medal
E. List all athletes who won Silver, ordered by the Year they won it.
F. Count how many Gold, Silver, and Bronze medals each country has won.
G. Identify countries that have won more than 50 Gold medals.
H. Olympics data with country information to show the total number of medals won by countries with a population over 10 million.
I. Find the athlete who has won the most medals
J. List all events that include the term 'Freestyle' in their name
K. Find the top 3 athletes by the total number of medals won in each sport.
L. List athletes who won more than one medal in a single Olympic year.
M. Identify countries that have won gold medals in both Summer and Winter Olympics (assuming a 'Season' column exists).
N. Show the year difference between the first and last medal won by each country.
O. Calculate the average number of medals won per athlete for each country.
P. Identify athletes who have won a silver medal and then later won a gold in the same event.
Q. Find the country that showed the largest improvement in total medals won from one Olympics to the next.
R. Assuming there's a coaches table with Athlete and Year, find athletes who won medals and later served as coaches.
S. List countries that have won medals in more than 10 different sports.
T. Find athletes who won medals in three consecutive Olympic games in the same event.

Linear Algebra

Q.1. Find the rank of the matrix A by reducing in Row reduced echelon form

$$A = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 2 & 4 & 3 & 2 \\ 3 & 2 & 1 & 3 \\ 6 & 8 & 7 & 5 \end{bmatrix}$$

Q.2. Let W be the vector space of all symmetric 2 x 2 matrices and let $T: W \to P_2$ be the linear transformation defined by $T\begin{bmatrix} a & b \\ c & d \end{bmatrix} = (a-b)+(b-c)x+(c-a)x^2$. Find the rank and nullity of T.

Q.3. Let A = $\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$. Find the eigenvalues and eigenvectors of A^{-1} and A + 4 ***I***.

Q.4. Solve by Gauss-Seidel Method (Take three iteration)

$$3x-0.1y-0.2z = 7.85$$
$$0.1x+7y-0.3z = -19.3$$
$$0.3x-0.2y+10z = 71.4$$

with initial values x(0) =0, y(0) =0, z(0) =0.

Q5. Define consistent and inconsistent system of equations. Hence solve the following system of equations if consistent $x + 3y + 2z = 0, 2x - y + 3z = 0, 3x - 5y + 4z = 0, x + 17y + 4z = 0$

Q.6: Determine whether the function T: $P_2 \to P_2$ is linear transformation or not.
Where $T(a+bx+cx^2) = (a+1)+(b+1)x+(c+1)x^2$

Q7: Determine the whether the set $S = \{(1,2,3), (3,1,0), (-2,1,3)\}$ is a basis of $V_3(R)$. In case S is not a basis determine the dimension and the basis of the sub space spanned by S.

Q8: Using Jacobi's method (perform 3 iterations), solve
3x-6y+2z = 23, - 4x+y-z = -15, x-3y+7z = 16 , with initial values $x_0 =1$, $y_0 =1$, $z_0 =1$.

Q9: Explain one application of matrix operations in image processing with example.

Q10: Give a brief description of Linear transformations for computer Vision for rotating 2D image.