Introduction to Data Science (DS 2001)

Fall 2024

Total marks : **60**

Due date: 09th Sept, 2024, 11:59 PM

Assignment 1

Instructions:

- All questions must be answered within a single notebook or .py file.
- Follow the file naming conventions: Name your submission file as RollNo.ipynb or RollNo.py
- (e.g., i23_xxxx.ipynb, where xxxx is your Roll Number).
- Use headings to distinguish each question in the notebook. (A notebook is provided as a sample)
- Late submissions will not be accepted and will be given a zero.
- No external libraries are allowed for this assignment unless instructed to do so.
- Any form of plagiarism will result in a zero for both parties involved.
- AI-generated content is prohibited. Best believe detection of such content will lead to a zero score.
- For Question 2 and 4, You may use the Random library. (Import random)

Question 1 [20 marks] :

20 years ago, FAST Islamabad campus used to store student's CNIC numbers only, to uniquely identify them. The administration decided that they should expand its information keeping facilities, and store other information about the students as well, such as their gender, the province, division, tehsil and union council they belong to, and their family number.

Hint: Each digit or group of digits in a Pakistani CNIC represents a specific category of information. For the last digit, an odd number represents MALE, while an even digit represents FEMALE.

Your task is to write a python program that takes a student's name and CNIC number as input and extracts the relevant information. Use tuples and dictionaries to store the mappings between the CNIC number segments and the respective details (Gender, Province, Division, etc.). Make sure your code handles edge cases as well.

Your Program will print the information in the following manner.

```
Enter Name of Student:  Hamza
Enter ID of Student:  6110166903075


Gender of Hamza is: Male
Province of Hamza is: Islamabad
Division of Hamza is: Islamabad-1
District of  Hamza is: Islamabad-1
Tehsil of  Hamza is: Islamabad-1
Union Council of Hamza is: Islamabad-1
Family Tree of Hamza is: 6690307
```

Question 2 [10 marks] : Hi-Lo Game

Your task is to implement a simple "Hi-Lo" game in Python. The goal of the game is to guess a randomly chosen number between 1 and 100.

Your Program should randomly select an integer between 1 and 100, and allow the user to guess the number within 5 attempts. Provide feedback after each guess:

1. If the guess is too high, tell the user "Too High!"
2. If the guess is too low, tell the user "Too Low!"
3. If the guess is correct, congratulate the user and end the game.

Handle invalid input:

Ignore any non-numeric characters in the guess (e.g., "69x" should be interpreted as "69").

Disregard guesses that are out of bounds (e.g., 0 or 101).

If the user does not guess the correct number after 5 tries, reveal the correct number and end the game.

Question 3 [10 marks] : (No external libraries allowed)

Stop words are common, frequently used words (like "the" and "is") that are often filtered out in text processing because they carry less meaning and do not contribute significantly to the analysis.

Write a program and code the following functions along with a main function :

1. ***removePunctuationMarks(input text):*** this function would take an input string and remove all the punctuation marks and special characters from the text.

2. ***removeStopWords(input text):*** this function would take an input string and remove all the stop words and return the filtered string.

For example:

Original String:

Hello, this is an example sentence to show stop words and punctuations.

Stop words:

['Hello', ',', 'this', 'is', 'an', 'example', 'sentence', 'to', 'show', 'stop', 'words', 'and', 'punctuations', '.']

Punctuations:

[',', 'this', 'is', 'an', 'to', 'show', 'stop', 'and', '.']

Filtered Output (No Stop Words and Punctuations):

Hello example sentence words punctuations

Instead of using any external libraries, you'll define your own stop-words and punctuations. This exercise is designed to help you understand the fundamentals of Natural Language Processing (NLP) by manually handling basic tasks like identifying and removing common words, which are typically filtered out to focus on more meaningful content in text analysis 😊

Question 4 [20 marks] : Dungeons & Dragons

Scenario: Dungeon Escape

You are going to create a simple text-based game called "Dungeon Escape." The goal of the game is to navigate & move through a dungeon, avoid traps, and collect health potions to survive.

Part 1: Class Definitions

1. ***Class: DungeonEscape***

Attributes:

health (int): The player's health points. Start with 100.

position (tuple of two integers): The player's current position on a 5x5 grid. Start at position (0, 0).

grid (list of lists): A 5x5 grid representing the dungeon. Each cell in the grid can contain:

'T' for a trap that reduces health by 20 points.

'H' for a health potion that increases health by 10 points.

'E' for an empty cell with no effect.

2. Class: Player

Attributes:

name (string): The player's name.

score (int): The player's score. Start with 0.

3. Class: DungeonGrid

Attributes:

grid (list of lists): The 5x5 grid for the dungeon.

Part 2: Function Definitions

1. Function: initialize_grid

Description: This function should randomly place traps ('T') and health potions ('H') on the 5x5 grid. The remaining cells should contain 'E' for empty.

Parameters: None.

Returns: A 5x5 grid.

2. Function: move_player

Description: This function should update the player's position on the grid based on the direction they choose (up, down, left, right).

Parameters: direction (string) - The direction in which the player wants to move.

Returns: The new position of the player.

3. Function: check_cell

Description: This function should check the cell at the player's current position to see if it contains a trap, health potion, or is empty, and then update the player's health and score accordingly.

Parameters: position (tuple of two integers) - The current position of the player.

Returns: None.

4. **Function: play_turn**

Description: This function should manage a single turn of the game, including moving the player and checking the cell.

Parameters: None.

Returns: None.

You can **ADD** more functions as per your ease.

A structure has been provided to you for your feasibility.

```
import random

class DungeonEscape:
    def init(self):        #constructor
        self.health = 100
        self.position = (0, 0)
        self.grid = DungeonGrid().initialize_grid()   #initialize function
defined in DungeonGrid class

    def move_player(self, direction):
     #interpretation here

    def check_cell(self):
    #interpretation here

    def play_turn(self):
    #interpretation here
```

Happy Coding :)