Introduction to Data Science

Fall 2023
Total marks 60
Due date: 9th November, 2023, 11:59 PM

Assignment 3

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., i22_xxxx.ipynb, where xxxx is your Roll Number).
- Use headings to distinguish each question in the notebook.
- Late submissions will not be accepted and will be given a zero.
- AI-generated content is prohibited. Detection of such content will lead to a zero score.

1. **[30 Marks - English Poetry Generation with N-grams]**

   The goal of this assignment is to explore natural language processing (NLP) techniques to generate creative and expressive poetry. Students will leverage these language models to construct three stanzas, each comprising four verses with 7 to 10 words in each verse, resulting in a cohesive and meaningful poem that not only exhibit technical proficiency but also evoke emotions and imagery. The provided stanza serves as an example, and the generation model will be trained on a Poetry Corpus containing works by renowned poets such as William Shakespeare and Maya Angelou.

   Below mentioned poem is a computer-generated poem in William Shakespeare style

   Upon yon stage, where moonlight gently plays,
   A soliloquy of stars in cosmic ballet.
   Whispers of fate in the evening's embrace,
   Shall we dance with destiny, in this sacred space?

   In fair Verona, where love blooms like spring,
   Romeo's heart, a sonnet that sweetly sings.
   Juliet, a rose by any other name,
   In the garden of passion, love's eternal flame.

   Morn breaks, as the sun's golden chariot rides,
   A symphony of dawn, where darkness hides.
   Oh, the world's a stage, and we but players,
   In life's grand drama, love the sweetest layers.

   Task:

   Your task is to print three stanzas with an empty line between them, generated using unigram, bigram and trigram models. Follow these steps to accomplish the assignment:

(a) Load the Poetry Corpus:

   i. Import the provided Poetry Corpus containing works by William Shakespeare and Maya Angelou.

   ii. Tokenize each corpus separately to create a list of words in each poet style.

(b) Generate N-gram Models:

   i. Develop unigram, bigram, and trigram models for each poet.

   ii. Select the first word randomly from the starting words in the vocabulary.

   iii. Use the unigram, bigram and trigram model to predict the next word, continuing until the verse is complete.

   iv. Apply the same approach to generate the next four lines.

(c) Algorithm for Poetry Generation:

```
For each stanza:
For each verse:
Generate a random number in the range [7...10].
Select the first word.
Choose subsequent words until the end of the verse.
Print the verse.
Print an empty line after the stanza.
```

Implementation Challenges:

(a) Word Prediction:
Selecting subsequent words after choosing the first word requires predicting the most probable next word. Utilize Conditional Frequency Distribution (CFD) to determine the set of words that occur most frequently after the selected word.

(b) Rhyming:
Bonus - Rhyming the generated verses poses an additional challenge. Consider building a rhyming dictionary to enhance the artistic quality of the poem.

(c) Standard N-gram Models:
We can develop our model using the Conditional Frequency Distribution method. First, develop a unigram model (Unigram Model), then the bigram model (Bigram Model), and then the trigram model. Select the first word of each line randomly from the starting words in the vocabulary and then use the bigram model to generate the next word until the verse is complete. Generate the next three lines similarly. Follow the same steps for the trigram model and compare the results of the two n-gram models.

Desired output:
Ask the user in which style they want to generate the poem. Give him two options -> William Shakespeare and Maya Angelou. Then display the poem in that generated style.

2. **[30 Marks - Puzzle Reconstruction]**

   You are tasked with solving a puzzle game using Python. Your goal is to reconstruct a complete image from a folder containing random sub-images that are pieces of the original picture. To achieve this, you need to perform the following steps:

(a) **Data Retrieval:**
Gather a collection of random sub-images from a specified folder. Each sub-image represents a portion of the complete image.

(b) **Preprocessing:**
Prepare the sub-images for stitching by applying necessary image preprocessing techniques, such as resizing, alignment, and enhancement.

(c) **Stitching:**
Utilize the provided image stitching helper code to combine the preprocessed sub-images into a single, coherent image. Ensure that the sub-images are correctly positioned and seamlessly stitched together.

(d) **Output:**
The final output of your Python code should be the original complete image, reconstructed from the sub-images.

Requirements:

You must implement a Python program to automate this process. Ensure that the sub-images are correctly aligned and stitched together to form the original image. Utilize the provided image stitching helper code or create your own if necessary. The resulting complete image should closely resemble the original picture. The program should be capable of processing sub-images from the specified folder.

Input:

A folder containing random sub-images of a complete picture.

Output:

The reconstructed complete image, with all sub-images correctly positioned and stitched together.

Your Python code should effectively solve this puzzle game by combining the sub-images to produce the original complete image.

HINT - You need to figure out how many images are present in a folder as well.

Happy Coding :)