Homework 1:
Python Practice and Linear Algebra
CPE 486/586

Instructor: Rahul Bhadani

Due: Sept 14, 2025, 11:59 PM
110 points

You are allowed to use a generative model-based AI tool for your assignment. However, you must submit an accompanying reflection report detailing how you used the AI tool, the specific query you made, and how it improved your understanding of the subject. You are also required to submit screenshots of your conversation with any large language model (LLM) or equivalent conversational AI, clearly showing the prompts and your login avatar. Some conversational AIs provide a way to share a conversation link, and such a link is desirable for authenticity. Failure to do so may result in actions taken in compliance with the plagiarism policy.

Additionally, you must include your thoughts on how you would approach the assignment if such a tool were not available. Failure to provide a reflection report for every assignment where an AI tool is used may result in a penalty, and subsequent actions will be taken in line with the plagiarism policy.

Submission instruction:

You may either complete your entire homework on Python Notebook and submit a Google Colab link or you may choose a combination of a pdf and Google Colab Notebook. However, you must provide **publicly accessible** Google Colab URL on Canvas.

For .pdf on Canvas, follow the format `CPE486586-LastFirst-HW-XX`. For example, if your name is Sam Wells, your file name should be `CPE486586-WellsSam-HW-XX.pdf`.

Your submission must contain your name and UAH Charger ID or your UAH email address. Please number your pages as well.

Please write down your unique keyword corresponding to your package.

Note: Python community has adopted following import convention that we will follow for the rest of the course:

```
    import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd
    import seaborn as sns
    import statsmodels as sm
```

For this portion of the Homework, complete your implementation entirely in the notebook and for a portion, you will contribute to your package and demonstrate its usage in your notebook.

1 The Longest Substring (10 Points)

Write a function `LongestSubstring` in Python that takes a Python string as an argument. Your function should perform the following task:

1. If the string is empty print ‘empty string’ and return.
2. If the string length exceeds 20 characters, print ‘maximum length exceeded’ and return.
3. Otherwise, find the long substring in the given string that contains same characters. For example, if the input string is ‘2242777232888823’, then the output should be ’8888’.
4. Write code to use the function `LongestSubstring` as well.

The implementation doesn’t need to be the most efficient implementation, and you may choose to use nested `for` loops as well.

2 Data and Time in Python (5 Points)

The built-in Python `datetime` module provides `datetime`, `date`, and `time` types. The `datetime` type combines the information stored in date and time.

Write a Python program that creates an empty file with the file name in the format ML-HW01_%Y-%m-%d-%H-%M.txt, where %Y is substitute for current year, %m is the substitute for current month, %d is substitute for current day, %H is the substitute for current hour, and %M is the substitute for current minute. One possibility can be a file name **ML-HW01-2011-10-29-20-30.txt**.

3 Working with Data Types in Python (5 Points)

In Python, there are some special data types such as table, and lists. A tuple is a fixed-length, immutable sequence of Python objects which, once assigned, cannot be changed. An example is

(2, 4, 5). On the other hand, lists are variable length and their contents can be modified in place. An example is [1, 4, 5]. Write a Python function that takes an argument as a list or a tuple, and returns a tuple containing the number of element the list or a tuple has, the maximum value, and a minimum value.

4 Slicing (5 Points)

You can select sections of most sequence types by using slice notation, which in its basic form consists of `start:stop` passed to the indexing operator `[]`:

In this part of the homework, you will create a 4×4 tensor using PyTorch package. Initialize your tensor randomly with only 0s and 1s. Tensor is a generalized name of a multi-dimensional array or matrix in machine learning domain. Note that this concept of tensor differs from one seen in Mathematics textbooks. Write a Python program to slice the last column of a 4×4 tensor.

5 Dictionary (5 Points)

Dictionary is a key-value pair data-types in Python. Create a dictionary in Python consisting of two letter state code of the neighboring states of Alabama as the key and their capital as the value.

6 Vector and Matrix Operations (10 Points)

Let $\mathbf{u} = [4, -1, 3]^T$, $\mathbf{v} = [2, 0, 5]^T$.

1. Compute the inner product $\mathbf{u} \cdot \mathbf{v}$.
2. Compute the angle between the vectors in degrees.
3. Implement the calculation in NumPy and verify the result.

7 Matrix Multiplication and Rank (10 Points)

Given matrices:

$$A = \begin{bmatrix} 2 & 4 \\ 0 & 3 \\ 1 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 1 \end{bmatrix}$$

1. Compute AB.
2. Determine the rank of matrix A.
3. Use NumPy to compute the rank.

8 Eigenvalue Decomposition with PyTorch (10 Points)

Let $M = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$.

1. Use PyTorch to compute the eigenvalues and eigenvectors.
2. Verify that $Mv = \lambda v$ for one eigenpair.

9 Solving System of Linear Equations(10 Points)

Consider a system of linear equations:

$$\begin{aligned} 4x + 3y + 2z &= 25 \\ -2x + 2y + 3z &= -10 \\ 3x - 5y + 2z &= -4 \end{aligned} \tag{1}$$

In a matrix form, they can be written $AX = B$. Identify A, and B.

Write a Python function `linearsolve` that takes two arguments: (a) matrix A; (b) vector B and return the solution vector X. You may implement your solution using Numpy or PyTorch.

Through the next few questions, you will make a contribution to your Python package with your unique keyword name.

10 Elementary Operations on Matrices (40 Points)

Create a subpackage called `matrix` in your package. Inside the subpackage, you will be creating a module `elementary.py` with the following requirements usin PyTorch based arrays and matrices:

1. Write a function `rowswap` that takes a matrix as input, along with the index of source row and the target row, and swap the content of source row with the target row.
2. Write a function `rowscale` that takes a matrix as an input, along with the index of source row and scaling factor, and perform row-scaling and return the new matrix. The scaling factor can be any real number, positive or negative.
3. Write a function `rowreplacement` to perform $jR_i + kR_j$ that takes a matrix as an input, first row, the second row, the scaling factors j and k. You may use function `rowscale` for scaling operation.

4. Write a function called `rref` that takes a matrix as an input. The task is to (i) Make the pivot element in the every row 1, (ii) make all the element below pivot element as 0. Here, you need to use if-else or other programming construct to determine which rows to swap, and what should be the scaling factor. The function will be return the reduced row echelon form of the matrix. You can use three functions developed earlier to assisting you in implementing `rref`.

5. Explain your idea of implementing `rref` in paragraphs.

6. Build your package, commit to GitHub and publish to PYPI after properly incrementing the version number, install the package in your Google Colab notebook using `pip install` command.

7. To test your program consider the input matrix as

$$\begin{bmatrix} 1 & 3 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 9 \\ 0 & 0 & 0 & 1 & -4 \end{bmatrix} \tag{2}$$

   - Perform elementary operation $R_1 \leftrightarrow R_2$ using `rowswap` function.
   - Perform elementary operation $\frac{1}{3}R_1$ using `rowscale` function on the resulting matrix from the previous step.
   - Perform elementary operation $-3R_1 + R_3$ using `rowreplacement` function on the resulting matrix from the previous step.

Tip: Use `__main__` function in your module elementary.py to directly test your code first before publishing your package Github and PYPI. You can write test code from Step 6 in `__main__` and execute `python elementary.py` in package's UV environment in your local machine.

Do not forget to commit your package code to GitHub. That will be a part of the assessment.