Math 446, Fall 2024, USC Instructor: Steven Heilman

Name: ______________________ USC ID: ______________ Date: ___________

Signature: ______________________. Discussion Section: ______________________

(By signing here, I certify that I have taken this test while refraining from cheating.)

Exam 1

This exam contains 8 pages (including this cover page) and 5 problems. Enter all requested information on the top of this page.

You may *not* use your books, notes, or any calculator on this exam.

You are required to show your work on each problem on this exam. The following rules apply:

- You have 50 minutes to complete the exam, starting at the beginning of class.
- **Organize your work**, in a reasonably neat and coherent way, in the space provided. Work scattered all over the page without a clear ordering will receive very little credit.
- **Mysterious or unsupported answers will not receive full credit**. A correct answer, unsupported by calculations, explanation, or algebraic work will receive no credit; an incorrect answer supported by substantially correct calculations and explanations might still receive partial credit.
- If you need more space, use the back of the pages; clearly indicate when you have done this. Scratch paper appears at the end of the document.

Problem	Points	Score
1	8	
2	10	
3	10	
4	10	
5	10	
Total:	48	

Do not write in the table to the right. Good luck![a]

[a] October 2, 2024,

1. Label the following statements as TRUE or FALSE. If the statement is true, **EXPLAIN YOUR REASONING**. If the statement is false, **PROVIDE A COUNTEREXAMPLE OR EXPLAIN YOUR REASONING**.

   (a) (2 points) Python raises an exception (i.e. gives an error) when given the command `{[1, 2], 3}`

   TRUE FALSE (circle one)

   [this was discussed in class]

   (b) (2 points) If we enter the following command into the Python

   ```
   ((2 < 4) and (4 < 3)) or not(2 < 7)
   ```

   Python outputs `True`.

   TRUE FALSE (circle one)

   [this was a repeated homework question]

(c) (2 points) Python's implementation of k-means clustering is deterministic. That is, if I use a dataset and ask Python to perform k-means clustering on that dataset, the output of the `KMeans` function from `sklearn.cluster` will be the same, regardless of how many different times I ask for an output, and regardless of any random seed that is provided to Python.

TRUE FALSE (circle one)

[this was repeated from the practice exam]

(d) (2 points) Python always finds the exact minimum of the k-means clustering objective function

$$\sum_{i=1}^{k} \sum_{j \in S_i} \left\| w^{(j)} - \frac{1}{|S_i|} \sum_{\ell \in S_i} w^{(\ell)} \right\|_2^2, \qquad (*)$$

That is, if k, m, q are positive integers with $k \leq m$, and if $w^{(1)}, \ldots, w^{(m)} \in \mathbf{R}^q$, then Python's `KMeans` function (from `sklearn`) is able to find a partition $S_1, \ldots, S_k$ of $\{1, \ldots, m\}$ minimizing the quantity $(*)$ over all partitions $S_1, \ldots, S_k$ of $\{1, \ldots, m\}$. (As usual, we define $|S|$ to be the number of elements of $S \subseteq \{1, \ldots, m\}$, and we define $||w||_2^2 := \sum_{i=1}^{q} w_i^2$ for any $w = (w_1, \ldots, w_q) \in \mathbf{R}^q$. Also, in case $S_i = \emptyset$, we define $\frac{1}{|S_i|} \sum_{\ell \in S_i} w^{(\ell)}$ to be zero.)

TRUE FALSE (circle one)

[this was a discussed in class]

2. (10 points) Give an example showing that the singular value decomposition is not unique.

   That is, find positive integers m, n, p and find a real $m \times n$ matrix A, $m \times m$ orthogonal matrices $U, \widetilde{U}$, $n \times n$ orthogonal matrices $V, \widetilde{V}$ and $p \times p$ diagonal matrices $D, \widetilde{D}$ (with $p \leq \min(m, n)$ and with nonzero diagonal entries) such that

   $$A = U \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} V = \widetilde{U} \begin{pmatrix} \widetilde{D} & 0 \\ 0 & 0 \end{pmatrix} \widetilde{V},$$

   and such that either: $U \neq \widetilde{U}$, or $V \neq \widetilde{V}$, or $D \neq \widetilde{D}$.

   (Recall that an orthogonal $n \times n$ matrix U satisfies $U^T U = U U^T = I$, where I denotes the $n \times n$ identity matrix.)

   (Recall also that $\begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix}$ is an $m \times n$ matrix, i.e. it is D with zero entries added to its right and bottom sides if necessary in order to make $\begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix}$ an $m \times n$ matrix.)

   [this was repeated and modified from the practice exam]

3. (10 points)

- Describe the output of the following Python program.

  ```
  x = 1
  for i in range(2000):
      x = 2 * x
      print(x)
  ```

  Describe in detail what the program does, and how many iterations the for loop performs.

- Describe the output of the following Python program.

  ```
  x = 1.0
  for i in range(2000):
      x = 2 * x
      print(x)
  ```

  Describe in detail what the program does, and how many iterations the for loop performs.

[this was a modified homework question]

4. (10 points) Write a program in Python that estimates the integral

$$\int_3^7 (1 + e^x) dx.$$

by averaging 1000 i.i.d. uniform random variables in the interval $[3, 7]$.

Hint: you can use the following Numpy built-in functions: `np.mean`, `np.exp`. Also `np.random.rand(1000)` outputs 1000 i.i.d. uniform random variables in $[0, 1]$.

(You can and should assume we already ran the command `import numpy as np` .)

[this was a repeated and modified homework question]

5. (10 points) Suppose we have a Pandas DataFrame named `df` with the following entries

```
            product_name    units_sold  unit_price  sale_date   region
product_id
4           widget_a        150         2.5         2023-01-10  east
3           widget_b        200         3.0         2023-01-12  east
2           widget_c        250         1.5         2023-01-14  west
1           widget_d        300         4.0         2023-01-10  south
0           widget_e        100         5.0         2023-01-15  east
```

That is, the `index` of `df` is named `product_id`, so the command `df.index` returns `Index([4, 3, 2, 1, 0], dtype='int64', name='product_id')`

Answer the following questions.

- What is the output of `df.loc[1]` ?
- What is the output of `df.iloc[1]` ?
- What is the output of `df[2]["units_sold"]` ?
- Write a single line of Python code that returns a DataFrame containing only the rows of `df` where sales occurred in the `east` region.
- Write a single line of Python code to compute the total sales for each row of `df` (i.e. compute `units_sold` multiplied by `unit_price`) and create a new column of `df` called `total_sales` that contains the total sales of each row of `df`.

[this was mostly discussed in class]

(Scratch paper)