תרגיל בית 9

NUMPY, Pandas & image processing

<u>הנחיות כלליות:</u>

- קראו היטב את השאלות והקפידו שהתכניות שלכם פועלות בהתאם לנדרש.
- את התרגיל יש לפתור לבד!
- אין לשנות את שמות הפונקציות והמשתנים שכבר מופיעים בקובץ השלד של התרגיל.
- אין למחוק את ההערות שמופיעות בשלד.
- הקפידו על כללי ההגשה המפורסמים באתר. בפרט, יש להגיש את כל השאלות יחד בקובץ ex9_012345678.py המצורף לתרגיל, לאחר החלפת הספרות 012345678 במספר ת.ז שלכם, כל 9 הספרות כולל ספרת הביקורת.
- <u>אופן ביצוע התרגיל:</u> בתרגיל זה עליכם לממש את הפונקציות הנתונות ניתן להוסיף פונקציות עזר.
- מועד אחרון להגשה: כמפורסם באתר.
- היות ובדיקת התרגילים עשויה להיות אוטומטית, יש להקפיד על פלטים מדויקים על פי הדוגמאות (עד לרמת הרווח).
- <u>בדיקה עצמית:</u> כדי לוודא את נכונותן ואת עמידותן של התוכניות לקלטים שגויים, בכל שאלה, הריצו את תוכניתכם עם מגוון קלטים שונים, אלה שהופיעו כדוגמאות בתרגיל וקלטים נוספים עליהם חשבתם (וודאו כי הפלט נכון וכי התוכנית אינה קורסת)
- בכל השאלות ניתן להניח את תקינות הקלט על פי המפורט בשאלה.

שאלה 1

משרד הבריאות רצה לבדוק את ההישגים של תכנית אימונים חדשה. במשך כמה חודשים רצופים נאספו נתונים על המועמדים. הנתונים נאספים בטבלאות csv, כך שבעמודה הראשונה מופיעים שמות המשתתפים, בשורה הראשונה שמות החודשים, וכל שורה הינה איסוף של הנתונים בסוף כל חודש. כלומר, בעמודה השנייה יופיעו הנתונים של כל מועמד בסוף החודש הראשון, בעמודה השנייה בסוף החודש השני וכן הלאה. (ראו את קובץ הדוגמא המצורף לתרגיל weight_input.csv, בו נמדדו משקליהם של ארבעה מתאמנים בקילוגרמים. חודש הדגימה הראשון הוא אוגוסט).

א. כתוב פונקציה load_training_data שמקבלת את הנתיב לקובץ ה-csv ומחזירה שלושה אובייקטים:

- data: הנתונים שבטבלה במטריצת numpy (ללא שורת החודשים או עמודת שמות המתאמנים).
- column_names: רשימת שמות החודשים לפי סדר העמודות (מהשורה הראשונה ללא התא הראשון), מטיפוס numpy.array של מחרוזות.
- row_names: רשימת המועמדים לפי סדר השורות (מהעמודה הראשונה ללא התא הראשון), מטיפוס numpy.array של מחרוזות.

דוגמת הרצה על הקובץ המצורף לתרגיל:

	August	September	October	November	December	January
Orit	84	81.3	82.8	80.1	77.4	75.2
Miki	79.6	75.2	75	74.3	72.8	71.4
Roni	67.5	66.5	65.3	65.9	65.6	64
Assaf	110.7	108.2	104.1	101	98.3	95.5

```
>>> data, column_names, row_names = load_training_data("weight_input.csv")
>>> print(data)
[[  84.    81.3   82.8   80.1   77.4   75.2]
 [  79.6   75.2   75.    74.3   72.8   71.4]
 [  67.5   66.5   65.3   65.9   65.6   64. ]
 [ 110.7  108.2  104.1  101.    98.3   95.5]]
>>> print(column_names)
['August' 'September' 'October' 'November' 'December' 'January']
>>> print(row_names)
['Orit' 'Miki' 'Roni' 'Assaf']
```

בסעיפים 2-5 להלן:

- **הפונקציות מקבלות את שלושת משתני הפלט של הפונקציה משאלה 1 על קובץ המשקלים:** `data, column_names, row_names`.
- **יש לממשן <u>ללא שימוש בלולאות</u>.**

2. כתוב פונקציה get_highest_weight_loss_trainee שמחזירה את שם המועמד שירידתו במשקל הייתה הכי גדולה מתחילת התכנית ועד סופה (ניתן להניח שיש אחד כזה). כלומר, שההפרש בין משקל ההתחלה למשקל הסיום הגדול ביותר.

ניתן (אבל לא חובה) להשתמש בפונקציה numpy.argmax שמחזירה את האינדקס שבו נמצא הערך המקסימלי.

דוגמת הרצה על טבלת הקלט הנתונה:

```
>>> get_highest_weight_loss_trainee(data, column_names, row_names)
'Assaf'
```

3. נגדיר את ה-'הפרש החודשי' כהפרש עבור מועמד מסויים בין חודש אחד לחודש הקודם לו. כתוב פונקציה get_diff_data שמחזירה את מטריצת ההפרשים. מימדי מטריצה זו זהים לאלו של מטריצת ה-data, כך שבכל עמודה יש את ההפרש בין החודש של עמודה זו לבין החודש הקודם לו. את העמודה הראשונה מלאו באפסים. שימו לב שמטריצת הקלט לא משתנה.

דוגמת הרצה על טבלת הקלט הנתונה:

```
>>> get_diff_data(data, column_names, row_names)
array([[ 0. , -2.7,  1.5, -2.7, -2.7, -2.2],
       [ 0. , -4.4, -0.2, -0.7, -1.5, -1.4],
       [ 0. , -1. , -1.2,  0.6, -0.3, -1.6],
       [ 0. , -2.5, -4.1, -3.1, -2.7, -2.8]])
```

4. כתוב פונקציה get_highest_loss_month שמחזירה את שם החודש שבו סכום ההפרשים החודשיים על פני כל המועמדים הוא הגדול ביותר (כלומר, **שההורדה** במשקל הייתה מקסימלית). שימו לב שהפונקציה מקבלת את data, columns_names, row_names (ולא את מטריצת הפלט מסעיף 3). השתמשו בפונקציה מסעיף 3.

דוגמת הרצה: במטריצת הדוגמא, בחודש ספטמבר המתאמנים הורידו ביחד 10.6 ק"ג, סכום שיותר גדול מאשר בשאר החודשים.

```
>>> get_highest_loss_month(data, column_names, row_names)
'September'
```

(ניתן להניח שיש חודש אחד כזה)

5. ידוע ששינוי במשקל הוא יחסי למשקל ההתחלתי. כתוב פונקציה get_relative_diff_table שתחזיר את טבלת השינוי במשקל, כך שלכל משתתף בכל חודש יופיע השינוי היחסי למשקלו בחודש הקודם, כלומר, ה'הפרש החודשי' לחלק למשקל בחודש הקודם. שימו לב שהפונקציה מקבלת את data, columns_names, row_names (ולא את מטריצת הפלט מסעיף 3). השתמשו בפונקציה מסעיף 3.

דוגמת הרצה:

```
>>> get_relative_diff_table(data,column_names,row_names)
array([[ 0., -0.03214286,  0.01845018, -0.0326087 , -0.03370787, -0.02842377],
       [ 0., -0.05527638, -0.00265957, -0.00933333, -0.02018843, -0.01923077],
       [ 0., -0.01481481, -0.01804511,  0.00918836, -0.00455235, -0.02439024],
       [ 0., -0.02258356, -0.03789279, -0.02977906, -0.02673267, -0.02848423]])
```

למשל, הנתון המסומן בפלט לעיל, הוא תוצאת החישוב:

$$\frac{november-october}{october} = \frac{74.3-75}{75} = -0.0093333$$

שאלה 2: נושא זה יילמד בתאריכים 12-16.1

גראלט מריוויה ("Geralt of Rivia") מסתובב בכל רחבי היבשת במצוד אחר מפלצות המטרידות את תושבי הממלכות. כוויצ'ר ("witcher") חסר רגשות, גראלט בוחר את יעדו לפי חישוב קר של עלות מול תועלת.

https://en.wikipedia.org/wiki/The_Witcher_(TV_series)

במהלך השאלה, נעזר בקובץ המשימות המכיל את כלל היעדים והמפלצות אותם יש לחסל. הקובץ בפורמט CSV ומכיל את המידע הבא:

- בשורה הראשונה מופיעים שמות עמודות המידע בסדר הבא:
  - Kingdom - העמודה הראשונה, מכילה את שם הממלכה במצוקה.
  - Bounty - העמודה השנייה, מכילה את הגמול על חיסול המפלצת.
  - Expenses - העמודה השלישית, מכילה את עלות המסע של גראלט לאותה ממלכה.
  - Duration - העמודה הרביעית מכילה את מספר הימים שייקח לגראלט להשלים את המשימה.
- שאר השורות מכילות את המידע הרלוונטי בהתאם לעמודות כמצוין לעיל (ראו את טבלת הדוגמא למטה)
- לנוחיותכם הטבלה הבאה נמצאת כקובץ בשם "missions.csv" בפורמט csv בין קבצי התרגיל שקיבלתם:

Kingdom	**Bounty**	**Expenses**	**Duration**
Temeria	1000	250	5
Redania	1500	500	3
Kaedwen	500	100	7
Cintra	2500	2000	3

א. ממשו את הפונקציה:
read_missions_file(file_name)

- הפונקציה מקבלת את שמו של קובץ המידע כפי שהוגדר לעיל (מחרוזת)
- הפונקציה תבנה dataframe של החבילה **pandas** כך ש:
  - הטבלה בעלת 3 עמודות: Bounty, Expenses, Duration.
  - שורות הטבלה בעלות שמות הממלכות, כלומר הindex של הטבלה הוא לפי Kingdom.
- במקרה של **שגיאת IO**, יש להעלות את השגיאה עם הכיתוב "An IO error occurred" (ראו דוגמת הרצה בעמוד הבא).
- ניתן להניח כי אם ניתן שם קובץ קיים, ערכיו יהיו תקינים ומכילים לפחות שורת משימות אחת (ושורת כותרות לעמודות).
- רמז: ניתן (אך לא חייבים) להיעזר בפקודה pd.read_csv על מנת לטעון את הקובץ לטבלה של pandas. הסתכלו בתיעוד הפונקציה באינטרנט בכדי להעביר את הארגומנטים המתאימים להשלמת השאלה.

ב. ממשו את הפונקציה:
sum_rewards(bounties)

- קלט הפונקציה הינו טבלת pandas המייצג את טבלת המשימות כפי שיוחזר בסעיף א'.
- על הפונקציה לחשב ולהחזיר את סכום הכסף שגראלט יקבל אם יבצע את **כלל** המשימות **בניקוי** הוצאות המסע.
- ניתן להניח כי הגמול ממשימה גדול מהוצאות המסע למשימה.
- לדוגמא, אם יבצע משימה בממלכת Kaedwen, גראלט ירוויח 500, אך יאבד 100 על הוצאות מסע. בסה"כ ירוויח 400.
- **אין להשתמש בלולאות ויש לכתוב את גוף הפונקציה בשורה אחת.**

ג. ממשו את הפונקציה:
find_best_kingdom(bounties)

- קלט הפונקציה הינו טבלת pandas המייצג את טבלת המשימות כפי שיוחזר בסעיף א'.
- על הפונקציה למצוא ולהחזיר את **שם הממלכה** הכדאית ביותר לגראלט. כדאיות משימה נמדדת לפי תגמול בניקוי ההוצאות חלקי מספר הימים שלוקח לבצעה.
- **לדוגמא:** בהינתן מערך כפי שמופיעה בטבלה לעיל, כדאיות המשימה בממלכת Temeria הינה 150 מכיוון ש-$\frac{1000-250}{5} = 150$
- ניתן להניח כי כל ממלכה מופיעה פעם אחת בלבד בקובץ וכי יש ממלכה כדאית אחת ויחידה.
- **אין להשתמש בלולאות, רצוי אך לא חובה לכתוב את גוף הפונקציה בשורה אחת.**

דוגמת הרצה:

בדוגמא הבאה נשתמש את הקובץ המצורף בתרגיל - "missions.csv" ונפעיל עליו את סעיפי השאלה.
בנוסף, סעיף א' הורץ עם שם קובץ שלא קיים במערכת הקבצים וכתוצאה מכך נזרקה שגיאה מתאימה עם ההודעה הנדרשת בסעיף א'.

```
In[29]: import pandas as pd
In[30]: file_name = "missions.csv"
In[31]: bounties = read_missions_file(file_name)
In[32]: print(bounties)
         Bounty  Expenses  Duration
Kingdom
Temeria    1000       250         5
Redania    1500       500         3
Kaedwen     500       100         7
Cintra     2500      2000         3
In[33]: read_missions_file("not_real_file.csv")
Traceback (most recent call last):
  File "<ipython-input-28-e708d74fcfa7>", line 5, in read_missions_file
    bounties = pd.read_csv(file_name, index_col="Kingdom")
  File "D:\python\lib\site-packages\pandas\io\parsers.py", line 685, in parser_f
    return _read(filepath_or_buffer, kwds)
  File "D:\python\lib\site-packages\pandas\io\parsers.py", line 457, in _read
    parser = TextFileReader(fp_or_buf, **kwds)
  File "D:\python\lib\site-packages\pandas\io\parsers.py", line 895, in __init__
    self._make_engine(self.engine)
  File "D:\python\lib\site-packages\pandas\io\parsers.py", line 1135, in _make_engine
    self._engine = CParserWrapper(self.f, **self.options)
  File "D:\python\lib\site-packages\pandas\io\parsers.py", line 1917, in __init__
    self._reader = parsers.TextReader(src, **kwds)
  File "pandas/_libs/parsers.pyx", line 382, in pandas._libs.parsers.TextReader.__cinit__
  File "pandas/_libs/parsers.pyx", line 689, in pandas._libs.parsers.TextReader._setup_parser_source
FileNotFoundError: [Errno 2] File b'not_real_file.csv' does not exist: b'not_real_file.csv'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "D:\python\lib\site-packages\IPython\core\interactiveshell.py", line 3319, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)
  File "<ipython-input-33-e5abd32b322c>", line 1, in <module>
    read_missions_file("not_real_file.csv")
  File "<ipython-input-28-e708d74fcfa7>", line 7, in read_missions_file
    raise IOError("An IO error occurred")
OSError: An IO error occurred
In[34]: print(sum_rewards(bounties))
2650
In[35]: print(find_best_kingdom(bounties))
Redania
```

שאלה 3 - עיבוד תמונה

ניתן לייצג תמונות בגווני אפור כמערך ndarray דו מימדי בו כל איבר הינו מספר בטווח 0-255 כאשר 0 מייצג שחור ו -255 מייצג לבן.

השתמשו בחבילה imageio בכדי לטעון את התמונות.

א. אחד הגדלים החשובים בתחום התקשורת היא האנטרופיה. האנטרופיה היא דרך לכמת את חוסר הסדר שמכילה מילה, תמונה או אירוע. חישבו על תמונה בעלת גוון אחד בלבד, למשל שחור. תמונה זו היא מאוד "מסודרת" (חישבו כמה מילים צריך כדי לתאר את התמונה הזאת) ואכן ערך האנטרופיה שלה הוא 0 (למה?). לעומת זאת חישבו על תמונה בעלת מנעד רחב של גווני אפור – תמונה זו מכילה המון אינפורמציה – והיא פחות "מסודרת" וערך האנטרופיה שלה גדול.

בסעיף זה נרצה לחשב את האנטרופיה של תמונה בגווני אפור ובכך לכמת את אי הסדר שבתמונה. הנוסחא לחישוב אנטרופיה היא:

$$S = \sum_{i=0}^{N} -P_i \cdot \log_2 P_i$$

כאשר P_i היא השכיחות לקבל גוון אפור כלשהו בין 0 ל 255. במילים אחרות P_i הם הערכים של ההיסטוגרמה המנורמלת של התמונה. N הוא מספר גווני האפור.

לדוגמא: נחשב את האנטרופיה של תמונה בעלת 4 פיקסלים. 2 פיקסלים בצבע שחור. ו2 פיקסלים בצבע לבן.

$$S = \sum_{i=0}^{N} -P_i \cdot \log_2 P_i = -\frac{1}{2} \cdot \log_2\left(\frac{1}{2}\right) - \frac{1}{2} \cdot \log_2\left(\frac{1}{2}\right) = 1$$

במקרה הספציפי הזה, השכיחות של פיקסל לבן או שחור היא חצי.

- כתבו פונקציה def compute_entropy(img) אשר מקבלת שם של תמונה img (string) אשר מחזירה את האנטרופיה (float) של תמונת גווני אפור.

- שימו לב להתעלם מערכי P_i שהם אפס.
- ניתן להניח כי הקלט תקין – תמונה בגודל כלשהו עם גווני אפור בין 0 ל 255.
- ניתן להניח שהתמונה מכילה יותר מגוון אפור אחד.
- רמז: השתמשו ב-np.bincount ו-np.log2

דוגמת הרצה:

```
>>>print(compute_entropy('cameraman.tif'))
>>>7.009716283345514
```

ב. כאשר אנו מגדילים תמונה, אנחנו צריכים למלא בצורה כלשהי את הפיקסלים החדשים בעזרת הפיקסלים הישנים. אחת השיטות הבסיסיות לעשות זאת היא באמצעות עיקרון "השכן הקרוב". ערך הפיקסל בתמונה המוגדלת יקבל את ערך הפיקסל הקרוב ביותר לפיקסל הנ"ל בתמונה המקורית. הנוסחא לפיקסל בתמונה המוגדלת תהיה:

$$Bigpixelvalue[i,j] = smallpixelvalue[floor(i * \frac{ySmallSize}{yBigSize}), floor(j * \frac{xSmallSize}{xBigSize}))]$$

כאשר xBigSize הוא הגודל בציר x (מספר עמודות) של התמונה המוגדלת וכן הלאה לגבי שאר הערכים.
j ו i הם האינדקסים של הפיקסלים בתמונה המוגדלת ו$floor(\mathrm{x})$ מחזירה את החלק השלם של x.

כתבו פונקציה nearest_enlarge(img,a) אשר:

- מקבלת שם של תמונה img (string)
- ערך הגדלה a גדול מ1 (int)
- מחזירה תמונה (מערך ndarray דו מימדי) מוגדלת כאשר מספר הפיקסלים בכל אחד ממימדי התמונה מוגדל פי a.
- רמז: השתמשו ב-np.floor

דוגמת הרצה:

```
>>>I=nearest_enlarge('cameraman.tif',2)
>>>plt.figure()
>>>plt.imshow(I,cmap = plt.cm.gray)
>>>plt.show()
```