

מבחן בקורס Big Data, למידת מכונה ומולטימדיה בסביבת Python

הוראות:

1. המבחן נעשה במחשב.
2. אין שימוש בחומר עזר.
3. מותר להשתמש במחשבון, דף ריק מתוכן וכלי כתיבה.
4. משך הבחינה - 3 שעות
5. במבחן שלוש חלקים:
 - חלק ראשון חובה - יש לענות על כל השאלות.
 - חלק שני בחירה - יש לענות על שתי שאלות מתוך שלוש.
 - חלק שלישי בונים - לא חובה לענות על חלק זה.
6. אין לעזוב את המבחן לפני שהוגש כנדרש (אחרת הציון 0 אוטומטית)
7. שימו לב! לא ליצור את הפרויקט בקונן C (אם קיים קונן נוסף) אחרת יש סיכוי לעובדן מידע.
8. שימו לב! תוודאו שאתם עובדים בגרסת פייתון נכונה.
9. יש ליצור קובץ פונקציות אחד בלבד ובו לממש את כל הפונקציות הנדרשות.
10. כל פונקציה שאתם מתבקשים לכתוב חייבת להיות בעלת שם זהה לזה שבשאלה! כולל התייחסות לאותיות קטנות/גדולות. פונקציה שהשם שלה לא יהיה רשום במדויק, לא תקבל ניקוד.
11. אין להגיש שום סקריפט שלא התבקשתם לכתוב וזה כולל סקריפט הבדיקות שלכם.
12. כל קטע קוד שלא קשור לאחת הפונקציות שהתבקשתם לכתוב יגרור הורדה בציון.
13. שימו לב! יש להתחשב במקרי הקצה שעלולים להיות. לדוגמה קלט לא תקין או כתובת שלא קיימת.
14. אתם יכולים להשתמש בכל פונקציה שתמצאו לפתרון המבחן.
15. במהלך המבחן אין גישה לאינטרנט.
16. מי שמסיים את המבחן מחכה במקומו עד להגעת המרצה בכדי להגיש את הקוד.
17. יש לרשום את מספרי ת.ז. על גבי טופס זה ולהגיש אותו בסוף המבחן.

חלק א' (חובה): 50%

שאלה 1 (5 נק.)

1. הגדירו פונקציה בשם: **myFullTensor**
הפונקציה צריכה לקבל משתנה יחיד מסוג tuple שמכיל מספרים טבעיים (כמות לא מוגבלת).
הפונקציה צריכה להחזיר מערך n-מימדי שכל איבר בו יהיה שווה ל 7.5.

דוגמה:

```
myDim = (3,5)
myMat = myFullTensor(myDim)
```

תוצר הפונקציה תהיה מטריצה 3x5:

$$\text{myMat} = \begin{bmatrix} 7.5 & 7.5 & 7.5 & 7.5 & 7.5 \\ 7.5 & 7.5 & 7.5 & 7.5 & 7.5 \\ 7.5 & 7.5 & 7.5 & 7.5 & 7.5 \end{bmatrix}$$

שאלה 2 (15 נק.)

2. הגדירו פונקציה בשם: **myFilesList**
הפונקציה צריכה לקבל לכל היותר משתנה אחד שהוא הכתובת של תיקיה כלשהי במחשב. משתנה זה חייב להיות מסוג string.
הפונקציה צריכה להחזיר את רשימת הקבצים (תתי תיקיות לא נחשבות כקבצים) שנמצאים בתיקיה שכתובתה קיבלה הפונקציה.
במידה והפונקציה לא קיבלה אף משתנה או קיבלה כתובת שלא קיימת במחשב, הפונקציה תחזיר את רשימת הקבצים בתיקית העבודה הנוכחית.

דוגמה:

```
myFolder = 'C:\Users'
noFolder = 'Hello World'
list1 = myFilesList(myFolder)
list2 = myFilesList()
list3 = myFilesList(noFolder)
```

לקלט זה, רשימות 2 ו 3 יהיו זהות והתוכן שלהן יהיה רשימת הקבצים בתיקית העבודה הנוכחית. רשימה 1 תכיל את רשימת הקבצים הנמצאים בתובת הנתונה.

פונקציות עזר:

```
os.getcwd()    os.path.exists(myPath)    os.listdir(myPath)
os.path.isfile(myFilePath)    os.path.join(myPath, fileName)
```

שאלה 3 (7 נק.)

3. הגדירו פונקציה בשם: **myImageMean**
הפונקציה צריכה לקבל תמונה. תמונה זו חייבת להיות מסוג ndarray.
הפונקציה צריכה להחזיר את הממוצע של התמונה כ float64. (ערך הממוצע של כלל האיברים הנמצאים בתמונה ללא חשיבות למימדיה).

דוגמה:

```
myMean = myImageMean(img)
```

ספריית עזר: numpy

שאלה 4 (15 נק.)

4. הגדירו פונקציה בשם: **myCommonTone**
הפונקציה צריכה לקבל תמונה בגוויי אפור. תמונה זו חייבת להיות מסוג ndarray.
הפונקציה צריכה לחשב את הגוון (ערך) הנפוץ ביותר בתמונה (uint8 בעלת ערכים בתווח: 0-255) ולהחזיר את הערך שלו.

דוגמה:

```
myTone = myCommonTone(img)
```

שאלה 5 (8 נק.)

5. הגדירו פונקציה בשם: **myLine**
להלן הפונקציה:

```
def myLine(img):
```

```
    myImg = img.copy()
    center = myImg.shape[0] // 2
```

```
    # -----
    for i in range(center - 30, center + 40):
        for j in range(myImg.shape[1]):
            myImg[i, j, 2] = 0
    # -----
```

```
    return myImg
```

תחליפו את קטע הקוד שבין הקווים השבורים בשורת קוד אחת בלבד.

חלק ב' (בחירה): 50%

שאלה 6 (25 נק.)

6. הגדירו פונקציה בשם: **myPhaseRound**
הפונקציה צריכה לקבל מטריצה (מערך דו מימדי) מסוג ndarray בעלת ערכים בין 0 ל 180.
הפונקציה צריכה להחזיר מטריצה בה כל איבר ממטריצה שנקלטה מעוגל לערך הקרוב ביותר מבין: 0, 45, 90, 135, 180. כל ערך שמעוגל ל 180 צריך להחליף ב 0.
לפי כך, הפונקציה צריכה להחזיר מטריצה בעלת ערכים {0, 45, 90, 135} בלבד.
דוגמה:

PhaseMat = **myPhaseRound**(myMat)

פונקציות עזר:

np.logical_and(BoolMat1, BoolMat2), np.logical_or(BoolMat1, BoolMat2)
np.random.uniform(lowVal, highVal, ShapeTuple)

שאלה 7 (25 נק.)

7. הגדירו פונקציה בשם: **myDataSplit**
הפונקציה צריכה לקבל מטריצה (מערך דו מימדי) מסוג ndarray בעלת 100 שורות.
הפונקציה צריכה להחזיר שלוש מטריצות בגדלים 50, 30 ו- 20 אשר מכילות שורות מהמטריצה המקורית (כל שורה ממטריצה המקורית מופיעה אך ורק באחת מהמטריצות המוחזרות).
יש לבנות את המטריצות באופן הבא:
במידה והיינו מחלקים את המטריצה המקורית ל 10 מטריצות בעלות 10 שורות כל אחת (ללא חפיפה בין מטריצות) אז 5 שורות הראשונות מכל מטריצה כזו מרכיבות את המטריצה בעלת 50 שורות, כל שתי שורות אחרונות במטריצות אלה מרכיבות את המטריצה בעלת 20 שורות והשורות הנותרות מרכיבות את המטריצה בעלת 30 שורות.

דוגמה:

TrData, TeData, ValData = **myDataSplit**(FullData)

פונקציות עזר:

np.random.uniform(lowVal, highVal, ShapeTuple),
np.concatenate((mat1,mat2), axis=AxisVal)

שאלה 8 (25 נק.)

8. הגדירו פונקציה בשם: **myFunctionPlotter**
הפונקציה צריכה לקבל אחת מהמחרוזות הבאות: {'sin', 'cos', 'con', 'All'}
הפונקציה צריכה לפלוט גרף של הפונקציה המתאימה בתחום בין 0 ל- 2π בעלת 1000 נקודות מחוברות בקו רציף.
'sin' - פולטת את פונקציית סינוס בצבע כחול
'cos' - פולטת את פונקציית קוסינוס בצבע ירוק
'con' - פולטת פונקציה של קו ישר וקבוע ששווה ל 0 בצבע אדום
'All' - פולטת את שלושת פונקציות יחד באותו הגרף ובצבעים שהוגדרו ערום.
עבור כל מחרוזת אחרת יש להחזיר None.

פונקציות עזר:

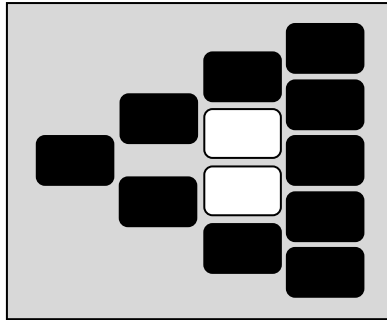
np.sin(Mat), np.cos(Mat), np.pi, np.linspace(0, 2π , 1000),
plt.plot(x,y, strLineSettings)

חלק ג' (בונוס): 15%

שאלה 9 (15 נק.)

9. הגדירו פונקציה בשם: `myPlotShape`
הפונקציה צריכה לקבל תמונה בגוויני אפור. תמונה חייבת להיות מסוג `ndarray`.
הפונקציה צריכה לפלוט גרף של ספריית `matplotlib` שנראה כך:

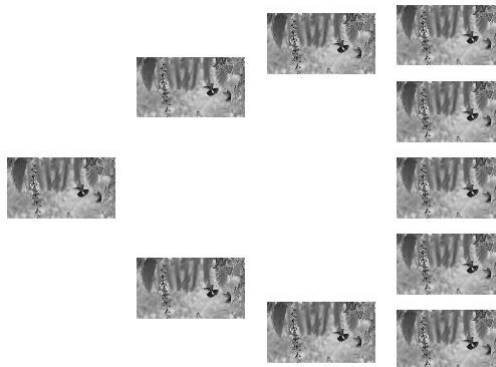
סקיצה:



דוגמה א:



דוגמה ב:



שימו לב ששתי התמונות לדוגמה בעצם אותה התמונה. יש הבדלים בתצוגה בגלל גודל החלון ולכן תצוגה יכולה להיות שונה במסכים או מחשבים שונים.
סקיצה מתארת לכם איך התמונות צריכות להיות ממוקמות בתוך החלון, כאשר שחור זו תמונה ולבן זה מקום ריק מתמונה.

בהצלחה!