תרגיל בית מספר 1 – MapReduce, Spark, תכן מסדי נתונים מבוזרים

הנחיות להגשת התרגיל:

1. **תאריך הגשה - 5/5/22 בשעה 23:55.**
2. הגשה בזוגות. יש להגיש את הפתרון דרך אתר הקורס במקום המתאים ב-moodle על ידי **אחד** מבני הזוג.
3. יש להגיש שני קבצי פייתון (.py) ומסמך pdf. קובץ הפייתון עבור השאלה הראשונה צריך להיות בפורמט ID1_ID2_mapreduce.py, קובץ הפייתון עבור השאלה השנייה צריך להיות בפורמט ID1_ID2_spark.py ומסמך ה-PDF צריך להיות בפורמט ID1_ID2.pdf, כאשר ID1 ו-ID2 הם מספרי הזהות של הסטודנטים המגישים. אי עמידה בהנחיות יגרור הורדת ניקוד.
4. **הפתרון חייב להיות מוקלד באמצעות מעבד תמלילים (כגון word, latex).**
5. איחור בהגשת התרגיל יגרור קנס בגובה 20% מהציון עבור כל יום איחור (פרט למקרים חריגים כגון מילואים. במקרים אלה יש לפנות לסגל הקורס טרם הגשת התרגיל).

הקדמה

כחלק מפרויקט בניה ירוקה הוחלט לאסוף מידע על מצבי פתיחה של וילונות בבניין משרדים בתל אביב על מנת לאפשר ניטור יעיל של כמות הקרינה הנכנסת לכל משרד. המידע מכיל את הנתונים הבאים:

- StartedOn: זמן תחילת המדידה (תאריך ושעה).
- Value: מדד מ-0 עד 1 המתאר עד כמה הווילון פתוח (1 אומר שהוילון סגור באופן מלא, 0 אומר שהוילון פתוח באופן מלא).
- face: חזית החלון של הווילון.
- curtain: מספר הווילון/החלון (ייחודי לכל וילון).
- azimuth: המפנה של החזית בזווית.
- min_range: הזווית המינימלית של השעון הסולארי לסגירת הווילון.
- max_range: הזווית המקסימלית של השעון הסולארי לסגירת הווילון.
- f_hour: ייצוג של השעה ב-float.
- auto_sched: משתנה שמציין האם לפי השעון הסולארי הוילון אמור להיות במצב פתוח או סגור.

הניחו ש-StartedOn ו-curtain מהווים יחדיו מפתח ראשי.

קובץ נתונים לדוגמה בפורמט CSV מצורף לקבצי התרגיל. הנתונים מכילים מידע שנאסף בחודשים אוגוסט, ספטמבר ואוקטובר של שנת 2021.

חלק א'- MapReduce (25 נקודות):

על מנת לסייע בתכנון מערכת אוטומטית של פתיחה וסגירה של הווילונות, אתם מתבקשים לכתוב שאילתה המבוססת על המבנה שהוצג בהקדמה .

רמת פתיחות ממוצעת/ערך Value ממוצע של ווילון ביום מסוים נקבעת על ידי סכימה של רמת הפתיחות (Value) של הווילון באותו יום חלקי מספר המדידות שנעשו באותו יום. לא ניתן להניח מידע אודות מספר המדידות שמבוצעות בכל יום.

עבור כל אחד מהחודשים וכל ווילון עליכם לחשב חסם תחתון וחסם עליון על רמת הפתיחות הממוצעת. החסמים מוגדרים בצורה הבאה:

חסם תחתון- ערך Value יומי ממוצע הכי נמוך של ווילון מבין ערכי ה-Value היומיים בחודש.

חסם עליון- ערך Value יומי ממוצע הכי גבוה של ווילון מבין ערכי ה-Value היומיים בחודש.

שאלה 1

עליכם לממש את השאילתה תוך שימוש במחלקה יחידה, ספריית MRJob ו-Python 3.6+. קובץ נתונים לדוגמה בפורמט CSV מצורף לקבצי התרגיל. עליכם להחזיר כפלט צמדים של:

Array(<Month>, <Curtain>): <Upper_Bound>

Array(<Month>, <Curtain>): <Lower_Bound>

כאשר השורה הראשונה מחזירה חסם תחתון עבור כל חודש ווילון, והשורה השנייה מחזירה חסם עליון עבור החודש והווילון. <> מסמן משתנה שיתקבל כפלט מהתוכנית שתיצרו.

עליכם לוודא כי הקובץ רץ על ידי הרצת הפקודה הבאה ב-console:

python "./[ID1_ID2]_mapreduce.py" "./data_HW1.csv"

חלק ב'- Spark (25 נקודות):

על מנת לאסוף נתונים על רמת הקרינה בבניין, אתם מתבקשים לכתוב שאילתה המבוססת על המבנה שהוצג בהקדמה .

רמת פתיחות ממוצעת/ערך Value ממוצע של ווילון ביום מסוים נקבעת על ידי סכימה של רמת הפתיחות (Value) של הווילון באותו יום חלקי מספר המדידות שנעשו באותו יום. לא ניתן להניח מידע אודות מספר המדידות שמבוצעות בכל יום.

עבור כל אחד מהחודשים עליכם להחזיר את החזיות (face) שבהן אין אף ווילון שממוצע ה-Value היומי שלו קטן מ-0.05.

שאלה 2

עליכם לממש את השאילתה תוך שימוש ב-pyspark RDDs ו-Python 3.6+. קובץ נתונים לדוגמה בפורמט CSV מצורף לקבצי התרגיל. עליכם להחזיר כפלט צמדים של:

Month: Array(faces)

כאשר Month מציין את החודש ו-Array(faces) מכיל את רשימת ה-faces המקיימים את התנאי.

עליכם לוודא כי הקובץ רץ על ידי הרצת הפקודה הבאה ב-console:

python "./[ID1_ID2]_spark.py"

ניתן להניח שקובץ הנתונים נמצא בתיקייה של קובץ הקוד.

חלק ג' – תכן מסדי נתונים מבוזרים (50 נקודות):

נתונה הרלציה שמתארת את הנתונים שהוצגו בהקדמה כולל שדה Data שמציין את תאריך הדגימה (תאריך שמחולץ מ-StartedOn):

Curtains(StartedOn, curtain, Value, face, Date, azimuth, min_range, max_range, f_hour, auto_sched)

קיימים שלושה סוגים של שאילתות וארבעה אתרים כאשר תדירות הגישה של שאילתה לכל אתר זהה. השאילתות מתוארות באופן הבא:

1. שאילתות מסוג 1 (20% מהשאילתות)

```
SELECT face, COUNT(auto_sched)
FROM Curtains
WHERE auto_sched = <value>
GROUP BY face
```

כאשר <value> הוא ערך 0 או 1.

2. שאילתות מסוג 2 (60% מהשאילתות)

```
SELECT curtain, Date, AVG(Value)
FROM Curtains
GROUP BY curtain, Date
```

3. שאילתות מסוג 3 (20% מהשאילתות)

```
SELECT curtain, face, AVG(Value), AVG(auto_sched)
FROM Curtains
GROUP BY curtain, face
```

שאלה 3 (35 נקודות)

תוך שימוש בשיטות אשר נלמדו בכיתה, הציעו הִתְרוֹסְסוּת (fragmentation) של הנתונים. לצורך פתרון השאלה, הניחו שסט הערכים ידוע מראש ונתון. למשל, ניתן להניח שערכי ה-Date האפשריים הם ערכים שנעים בין ה-1.8.2021 עד 31.10.2021.

ציינו את ההנחות עליהן התבססתם בפתרון השאלה והסבירו את אופן יצירת הקְטָעים (fragments).

שאלה 4 (15 נקודות)

נתונה רלציה נוספת שמתעדת את טמפרטורת החדר בה הוילון נמצא:

CurtainsTemp(StartedOn, curtain, temp)

בנוסף, נתונה השאילתה הבאה:

```
SELECT face, AVG(temp)
FROM Curtains, CurtainsTemp
WHERE Curtains.StartedOn = CurtainsTemp.StartedOn AND Curtains.curtain =
        CurtainsTemp.curtain
GROUP BY face
```

האם המידע הנוסף שניתן בשאלה משנה את ההִתְרוֹסְסוּת המוצעת בשאלה 3? נמקו. במקרה שנדרש שינוי, הציגו את ההִתְרוֹסְסוּת לאחר השינוי ופרטו מהו השינוי שביצעתם לעומת ההִתְרוֹסְסוּת שהצעתם בשאלה 3.

בנוסף, הציעו הִתְרוֹסְסוּת (fragmentation) לרלציה CurtainsTemp. הסבירו את תשובתכם.

בהצלחה!