הטכניון – מכון טכנולוגי לישראל

הפקולטה למדעי הנתונים וההחלטות

096224: ניהול מידע מבוזר

סמסטר אביב תשפ"ד

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

הנחיות להגשת התרגיל:

1. תאריך ההגשה - 11/07/2024 בשעה 23:55.
2. הגשה **בזוגות בלבד**. יש להגיש את הפתרון דרך אתר הקורס במקום המתאים, ע"י אחד מבני הזוג.
   על הפתרון להיות **מוקלד** במעבד תמלילים (Word, LaTex... etc) בלבד.
3. **אין להגיש תיקיית zip.** אתם נדרשים להגיש 6 קבצים בסה"כ. בכל מקום בו מצוין id1,id2, הכוונה היא למספרי תעודות הזהות של המגישים.

   a. עבור **חלקים א' ו-ב' יחד (קוד) יש להגיש קובץ מחברת ipynb יחיד**, כאשר שני החלקים מופרדים זה מזה באופן ברור בעזרת כותרות Markdown. על הקוד להיות מתועד וקריא.
   שם הקובץ צריך להיות "hw1_code_id1_id2.ipynb"

      i. במידה והקוד שלכם מייצר קבצי פייתון py נוסף, **חובה** לצרף גם אותו, בהתאם להנחיות ההגשה בחלק א' והסעיף המתאים.
      שם הקובץ הזה צריך להיות "hw1_mr[A/B]_id1_id2.py"

   b. עבור חלקים א' ו-ב' יש להגיש **גם גרסת PDF וגם גרסת HTML של המחברת** שכוללת את תוצאות ההרצות - ובמיוחד את הפלטים הדרושים בשאלות.
   הקבצים ייקראו "hw1_code_id1_id2.pdf" , "hw1_code_id1_id2.html".

   c. עבור **חלק ג'** (תכן מסדי נתונים מבוזרים) יש להגיש **קובץ PDF נפרד**.
   הקובץ ייקרא "hw1_dry_id1_id2.pdf"

הקדמה – חלקים א' ו-ב'

במרחב הלמידה בספריה יושבים מסביב לשולחן קבוצת חברים - ג'ף, בריטה, עאבד וטרוי. במסגרת אחד הקורסים, שעוסק בין היתר בנתוני עתק במערכות טלוויזיה, מיזוג אוויר ושרברבות, ניתנה להם גישה לקובץ נתונים באדיבות FourthWallMedia שמכיל את המידע הבא על כל תכניות הטלוויזיה ששודרו ב-2015:

- prog_code - מזהה ייחודי לכל אחת מהתכניות. אם מדובר בסדרה - מזהה פרק ספציפי.
- title - כותרת התכנית. במידה ומדובר בסדרה - שם הסדרה כולה.
- genre - רשימת הז'אנרים אליה משויכת התכנית, מופרדים בפסיקים.
- air_date - התאריך בו שודרה התכנית, בפורמט "YYYYMMDD"
- air_time - השעה בה שודרה התכנית, בפורמט "HHMMSS"
- Duration - אורך התכנית ששודרה (בדקות)

חלק א' – MapReduce (20 נקודות):

טרוי ועאבד מפיקים יחד תכנית בוקר מפורסמת בטלוויזיה. לכן בתרגיל זה נגדיר Rise&Shine-Time בתור טווח השעות שבין 7 ל-9 בבוקר (כולל 7:00:00 אך **לא** כולל 9:00:00)

הם מעוניינים למצוא את תכניות הבוקר המתחרות המוצלחות ביותר כדי לדעת כיצד לשפר את התכנית שלהם. לפיכך הם יצרו יחד את השיטה הבאה לאיתור תכניות בוקר מצטיינות:

- נסתכל באופן בלעדי על שידורים שהחלו בתחום Rise&Shine-Time .
- מתוכם ניקח רק את השידורים שכוללים לפחות אחד מהז'אנרים הרלוונטיים -
  ['Sitcom', 'Talk', 'Politics', 'Spanish', 'Community', 'Martial arts']
  וגם כותרתם מכילה לפחות אחת מהאותיות הבאות- (**Case insensitive!**)
  ['j', 'q', 'z']
- עבור כל תכנית בוקר מצטיינת נחזיר את -
  1. כותרת התכנית
  2. פירוט רשימת הז'אנרים של התכנית
  3. כמות התאריכים השונים שהיא שודרה בהם (לפי כותרת - לא לפי פרק ספציפי)
  4. כמות הז'אנרים שמשויכים לה

שאלה 1 - סעיף א' (15 נקודות)

עליכם לממש את השאילתה תוך שימוש במחלקה **יחידה**, ספריית MRJob ו-Python3.6+. קובץ נתונים לדוגמה בפורמט CSV מצורף לקבצי התרגיל. עליכם להחזיר כפלט את כל התכניות העונות על התנאים לעיל, ואת מאפייניהן בפורמט הבא:

(title, genres), (sum_dates, amount_genres)
=== Example ===
["La Biblioteca", "Talk"], [116, 1]

במידה והמחברת שלכם מייצרת קובץ py נוסף - חובה לצרף גם אותו להגשה, ולוודא שהקובץ רץ ומתקבל הפלט התקין בעזרת הרצת הפקודה הבאה ב-terminal:

python "./hw1_mr**A**_id1_id2.py" "./420k_daily_prog_data.csv"

שאלה 1 - סעיף ב' (5 נקודות)

עליכם להוסיף קוד ל-MRJob שכתבתם בסעיף הקודם, כך שתוחזר רק התכנית המצטיינת הטובה ביותר, בעלת הניקוד הגבוה ביותר. נדרג את התכניות המצטיינות לפי סכום כמות התאריכים השונים שהיא שודרה בהם + כמות הז'אנרים שמשוייכים לה (3+4 מסעיף א').

שימו לב: סעיף ב' הינו הרחבה (העתק+הדבק ותוספת קוד) של סעיף א', כלומר הקוד שתגישו בסעיף ב' חייב להכיל בתוכו את מה שכתבתם בסעיף הקודם - השלבים החדשים ב-Job מתבססים על הפלט של סעיף א'. לכל סעיף MRJob יחיד משלו.

על הפלט להיות בפורמט הבא:

(title, genres), (total_score)

=== Example ===

["La Biblioteca", "Talk"], 117

כאשר total_score = sum_dates+amount_genres.
תודפס רק תכנית יחידה בעלת הניקוד הגבוה ביותר.

במידה והמחברת שלכם מייצרת קובץ py נוסף - חובה לצרף גם אותו להגשה, ולוודא שהקובץ רץ ומתקבל הפלט התקין בעזרת הרצת הפקודה הבאה ב:terminal-

python "./hw1_mr**B**_id1_id2.py" "./420k_daily_prog_data.csv"

חלק ב׳ - Spark (30 נקודות):

עאבד בדיוק סיים לצפות בעונה האחרונה בתכנית הטלוויזיה האהובה עליו ומחפש כעת סדרה חדשה שתמלא את החלל שנפער בליבו. לשם כך הוא אסף דגימה מנתוני התכניות שקיבל ויצר שיטת ניקוד מפורטת.

מצאו עבורו את 20 הסרטים/סדרות המדורגות הכי גבוה בשיטת הניקוד הבאה:

- עאבד אוהב תכניות טלוויזיה במבנה קבוע -
  לכן תכניות שהז׳אנר שלהן הוא "Sitcom" בלבד יקבלו +7 נקודות.
- תכניות שהז׳אנר שלהן הוא "Action" או "Documentary" בלבד יקבלו +90 נקודות.
- כל תכנית תקבל מספר נקודות נוסף לפי אורכה בדקות חלקי 15.
- עבור כל פעם שתוכנית שודרה ביום אי זוגי בשבוע (ראשון, שלישי, חמישי, שבת) היא תקבל בונוס +3 נקודות.
- עאבד לא מוכן לצפות בתכנית שכותרתה כוללת את אחת המילים הבאות (**Case insensitive!**)
  - Big, the, bang, theory, almanac, met, mother, your, city, anatomy, game, thrones, guy, family, friends, senate, two

כך למשל רשומה בקובץ עם הערכים הבאים:

prog_code	title	genre	air_date	air_time	Duration
EP300S03EP08	Abed	Documentary	20151220	153000	60.0

תקבל ניקוד של 3+60/15+90 = 97 = דוקומנטרי באורך 60 דק' ששודר ביום ראשון.

כדי להגיע לניקוד הסופי שניתן לסרט/סדרה נסכום את כל הנקודות שניתנו לכל הרשומות שנושאות את אותה הכותרת ואותו ז׳אנר. זה יהיה הציון הסופי של התכנית. נרצה להציג את הכותרת והז׳אנר בתוצאה הסופית.

עבור כל רשומה של צמד של כותרת וז'אנר - סיכמו את סך כל הנקודות על פני כל הרשומות בנתונים. החזירו את 20 הצמדים בעלי הניקוד הגבוה ביותר. שימו לב - בתוצאה הסופית כל זוג שכזה יכול להופיע פעם אחת בלבד.

שאלה 2 (30 נקודות)

עליכם להדפיס כפלט את 20 הצמדים בעלי הניקוד הגבוה ביותר, מסודרים בסדר ניקוד יורד בפורמט הבא:

{כותרת, ז'אנרים} | ניקוד ===לדוגמה=== 987654 | {Abed, Documentary}

גם פלט שמופרד לעמודות (לא עמודה יחידה שמכילה צמד) מקובל.

יש לוודא שתוכן התאים לא ייחתך בפלט הסופי - יש להשתמש ב Truncate=False-במידה וזה קורה.

לתרגיל מצורפים שני קבצי - CSV קטן (50k) וגדול (420k)
על מנת להקל על עבודת הפיתוח, הוכן עבורכם קובץ (50k) tiny_sample של הנתונים.
שימו לב! עליכם להגיש את **הפלט** של התוכנית שלכם על **קובץ הנתונים הגדול מבין השניים!**

- עליכם לממש את השאילתה תוך שימוש ב- RDDs/DataFrames in PySpark
- אין להשתמש בספריית Pandas או כל ספריה שאינה PySpark לעיבוד הנתונים.
  - ניתן להשתמש ב-Pandas אך ורק למטרות data exploration/visualization על דגימה קטנה מתוך הנתונים.

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

בתרגיל הבית נעשה שימוש ברלציות ממערכת מידע אשר משמשת לניהול מידע של ממירים בבתי אב, ומיועדת למעקב אחר צפייה בטלוויזיה. הסכימה היא הסכימה הבאה:

- **הרלציה הבאה מתארת מידע של משפחות המשתמשות בממירי טלוויזיה:**

MediaData(HHID, deviceID, DMA, zipCode, housholdSize, NumOfAdults, NetWorth, GreenLiving)

HHID - מזהה משפחה (Household ID)

DeviceID - מזהה מכשיר ממיר של משפחה

DMA - מזהה איזור מגורים של משפחה (Designated Market Area)

zipCode - מיקוד איזור מגורים של משפחה

householdSize - מספר הנפשות במשפחה

NumOfAdults - מספר המבוגרים במשפחה

NetWorth - רמת העושר של המשפחה (יש 6 רמות עושר סך הכל)

GreenLiving - אינדיקטור המציין האם המגורים מתנהלים באופן ידידותי לסביבה

- **הרלציה הבאה מתארת אירועי צפייה לפי מזהי ממירים:**

ViewingData (HHID, deviceID, Prog_code, genre, eventDate, eventTime)

HHID - מזהה משפחה (Household ID)

DeviceID מזהה מכשיר ממיר של משפחה

Prog_code - מזהה ייחודי לתכנית טלוויזיה (למשל, מזהה סדרה)

Genre - קטגוריה של תוכנית הטלויזיה (למשל, חדשות). ניתן להניח שלכל תכנית ז'אנר יחיד.

eventDate, **eventTime** - מועד תחילת הצפייה בתוכנית במכשיר

- **הרלציה הבאה נותנת מידע על שידורי תכניות:**

DailyProgramData (Prog_code, Air_date, Air_time, Title, Genre, Duration)

Prog_code - מזהה ייחודי לתכנית טלוויזיה (למשל, מזהה פרק ספציפי)

Air_date - תאריך שידור התכנית בפורמט YYYY-MM-DD

Air_time - השעה של שידור התכנית באותו יום בפורמט HH:MM:SS

Title - כותרת התכנית.

Genre - קטגוריה של תוכנית הטלויזיה (למשל, חדשות). ניתן להניח שלכל תכנית ז'אנר יחיד.

Duration - אורך התכנית ששודרה (בדקות)

עם כל שינוי במצב ממיר, נוספת רשומה חדשה לרלציית .ViewingData **בנוסף**, קיימות שלוש השאילתות הבאות במסד הנתונים:

שאילתה 1: מחזירה את מספר הממירים עבור כל רמת עושר. שאילתה זו נשאלת אחת לחודש. מאזורי מיקוד במדינת קליפורניה בחוף המערבי.

שאילתה 2: מספר הממירים אשר משדרים תכניות מסוג "חדשות". השאילתה נשאלת מדי יום בשעה 20:00. השאילתה נשאלת מאזור מיקוד במדינת טקסס אשר במרכז ארצות הברית.

שאילתה 3: משך זמן צפייה ממוצע בתוכניות מסוג "חדשות", מחושב בנפרד למגורים המתנהלים באופן ידידותי לסביבה וכאלו שלא. השאילתה נשאלת אחת לשבוע מאזור מיקוד אשר בוושינגטון בחוף המזרחי.

שאלה 3 (35 נקודות)

1) עבור המשימה נעם פינק אתכם עם הקצאה של 6 שרתים כאשר בכל שרת מעבד יחיד. השרתים ממוקמים במדינות ניו ג'רזי, שכנתה ניו יורק, טקסס ושכנתה אוקלהומה, קליפורניה ושכנתה אורגון .

a. (25 נקודות) הציעו הִתְרוֹסְסוּת (fragmentation) של הנתונים כך שניתן יהיה לעבד את השאילתות באופן המהיר ביותר. יש צורך להראות חישובים שבצעתם. ציינו את ההנחות עליהן התבססתם בפתרון והסבירו את אופן יצירת הקְטָעים (fragments) ובפרט **כיצד השתמשתם באלגוריתמים שנלמדו בכיתה** להגעה לפתרון .

b. (10 נקודות) הסבירו כיצד תחלקו את הקטעים שנוצרו בין השרתים.

שאלה 4 (15 נקודות)

2) נעם נבהל כשהבין שעם 6 שרתים, הפתרון שנתתם יקר, והחליט להוריד את ההקצאה לשני שרתים בלבד .

a. הסבירו האם וכיצד ההִתְרוֹסְסוּת שהצעתם תשתנה תחת ההקצאה החדשה .

b. כיצד תחלקו את הקטעים כעת? נמקו

c. אילו שניים מבין ששת השרתים שהוקצו לכם בסעיף הקודם תבחרו להשאיר פעילים? נמקו

בהצלחה!