תרגיל בית 4

חלק א' – רטוב (60%):

סט הנתונים שנתון לנו הוא salary_data.csv והוא זמין להורדה במודל. סט הנתונים מתאר נתונים שנאספו ב1994 ע"י לשכת מפקד האוכלוסין של ארצות הברית. סט הנתונים מתאר מדגם בגודל של כ32500 תושבים עבורם תועדו התכוניות הבאות:

age: continuous.

workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.

education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.

education-num: continuous.

marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

sex: Female, Male.

capital-gain: continuous.

capital-loss: continuous.

hours-per-week: continuous.

native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

בנוסף, לכל תושב מתוארת התכונית **salary,** שערכה הוא ">50K" או "<=50K" בהתאם להאם התושב מרוויח יותר או פחות מ$50,000 בשנה או לא, ומטרתנו בתרגיל זה תהיה לחזות את ערך התכונית **salary.** נגדיר שכאשר ">50K" = salary התושב משויך לקטגוריה 0, וכאשר "<=50"=salary התושב משויך לקטגוריה 1.

למידע נוסף על סט הנתונים: https://www.kaggle.com/uciml/adult-census-income .

דרך נפוצה להתמודדות עם משתנים קטגוריאליים שנתנסה בה בתרגיל זה הינה המרתם לקבוצת אינדיקטורים, הנקראים dummy variables.

למשל, נחליף את התכונית **occupation** בk dummy variables: $\{occ_1, \dots, occ_k\}$. עבור כל רשומה, $k-1$ מתוך k התכוניות הנ"ל יקבלו את הערך 0, והתכונית הנותרת תקבל את הערך 1, כאשר תכונית זו מתאימה ל**occupation** שתועד עבור רשומה זו.

במקרים מסוימים, ייתכן ועבור רשומה מסוימת, שני dummy variables או יותר יקבלו את הערך 1 (לדוגמה, בנתונים על סרטים: סרט שמתאים לשתי קטגוריות – קומדיה ורומנטיקה).

אחת המשמעויות של שיטה זו, היא שמספר התכוניות שעליו אנחנו מפעילים את האלגוריתם שלנו גדל משמעותית.

תיאור המשימות

סעיף 1

תחילה, עליכם להמיר את המשתנים הקטגוריאליים בדאטא לdummy variables :
workclass ,**education** ,**martial-status** ,**occupation** ,**relationship**, **race** ,**native-country**, **sex**

שימו לב שעליכם להמיר את העמודה **salary** כפי שהוגדר מעלה. כמו כן יהיה עליכם לבצע נרמול למשתנים הרציפים כפי שיוסבר בהמשך.

לאחר מכן, עליכם לבנות מעטפת גנרית ללמידה והסקה של שני אלגוריתמי סיווג שלמדתם בהרצאה, כאשר מטרת הסיווג היא לחזות את ערך התכונית **salary**:

- K – nearest neighbors (KNN) classifier – כאשר נבחר בK=15.
- Rocchio classifier.

לכל מסווג יש להציג:
- Precision
- Recall
- Accuracy

עליכם לאמן את האלגוריתמים ולדווח את הביצועים בשיטת cross validation, כאשר מספר הfolds הינו 5. שימו לב שיש לבצע חלוקה לtrain and test בלבד, ללא validation.

(50%)

סעיף 2 - תחרות

בשאלה זו עליכם למקסם את מדד הaccuracy, כאשר ניתן לכם חופש פעולה (כמעט) מלא:

- בחירת התכוניות
- Preprocessing
- בחירת הפרמטרים החופשיים של האלגוריתמים

המגבלה היחידה הינה שגם כאן עליכם לאמן את האלגוריתמים ולדווח את הביצועים בשיטת cross validation, כאשר מספר הfolds הינו 5. שימו לב שיש לבצע חלוקה לtrain and test בלבד, ללא validation.

<u>כחלק מהסעיף עליכם לגלות יצירתיות ולהסביר במפורט את תהליך ביצוע המשימה (צרפו הסבר זה כחלק מהחלק היבש של התרגיל).</u>

לצורך המשימה באפשרותכם להשתמש בשלב הלמידה וההסקה בחבילה sklearn בלבד. בכל שלב אחר בתוכנית שלכם, כל פונקציה כשירה לשימוש לרבות פונקציות שלא הועברו בתרגולים ובמעבדות ובתנאי שסיפקתם עבורן תיעוד מתאים או התייחסתם אליהם בקובץ ה- word או ה- pdf אליהן בהסבר לסעיף זה.

<u>משימה שלא תספק הסבר מתאים תקבל ניקוד מופחת.</u>

הקבוצות ימויינו לפי ערכי הaccuracy שהשיגו (ערך הaccuracy הממוצע בין שני האלגוריתמים), כך ש10% העליונים של ההגשות יקבלו 10/10, 10% הבאים יקבלו 9/10 וכן הלאה. שימו לב שפתרון לסעיף 1 מהווה פתרון לסעיף 2 – לדוגמה, אם לא הצלחתם לשפר מעבר לסעיף 1.
(10%)

פלט התוכנית צריך להיות מהצורה הבאה:

Question 1:
KNN classifier: value of precision, value of recall, value of accuracy
Rocchio classifier: value of precision, value of recall, value of accuracy

Question 2:
KNN classifier: value of accuracy
Rocchio classifier: value of accuracy

בין התוצאות של Question 1 ו Question 2 -יש רווח של שורה אחת.

דרישות למימוש:

המימוש **חייב** להכיל לפחות את המודולים, המחלקות והמתודות הבאות:

- ניתן להוסיף data members למחלקות המתוארות.
- ניתן לשנות טיפוסים של data members.
- ניתן לשנות חתימות של מתודות, **אך לא את ייעודן.**

לצורך המימוש עליכם להשתמש בחבילה sklearn. ניתן להשתמש בספרית Numpy, pandas לטובת מימוש הפוקנציות.

1. **main.py** – ממשק ראשי לריצת התוכנית, כפי שהיא מתוארת בפרק "תיאור המשימות". בין היתר, בקובץ זה יופיעו השורות:

```
def main(argv):
    pass


if __name__ == '__main__':
    main(sys.argv)
```

כאשר במקום "pass" יופיע קטע הקוד. עליכם לייבא את הספריה sys בראש הקובץ, ע"י import sys. ע"י המבנה הנ"ל תוכלו להריץ את התוכנית שלכם באמצעות:

python **your_path**/main.py arguments

בתרגיל בית זה:

python /home/student/**your_path**/main.py /home/student/**your_path**/salary_data.csv

שימו לב:

argv[0] = /home/student/**your_path**/main.py,

argv[1] = /home/student/**your_path**/salary_data.csv

כאשר **your_path** הוא הנתיב בו שמור המודול main.py, שאר המודולים, וקובץ הנתונים salary_data.csv.

2. data.py – ממשק לאיסוף הנתונים.

מחלקה Data:

<u>data members:</u>

data – הדאטא לאחר שלבי הPreprocess.

<u>function members:</u>

a. **preprocess** – מתודה לביצוע הנקודות:

1. עליכם לקרוא את כל תכוניות הדאטא למעט **fnlwgt**.
2. אין לקלוט רשומות בהן מופיע לפחות תכונית אחת עם ערך "?".
3. טיפול בתכונות הקטגוריאליות כפי שצוין בפרק "תיאור משימות".
4. טיפול בתכונה salary כפי שהוגדר.
5. נבצע נרמול MinMax של סט הנתונים לכל התכוניות הרציפות (**age, education-num, capital-gain, capital-loss, hours-per-week**).

<u>נרמול MinMax</u> **:** תהיי תכונית i. יהיו M, m הערכים המקסימליים והמינימליים שמקבלת התכונית i בהתאמה. תהיי x תצפית מהתכונית i. הערך המנורמל של x הינו $\frac{x-m}{M-m}$.

b. **split_to_k_folds** – תפקידה להחזיר אובייקט מסוג *Kfold*. יש להשתמש בקריאה:

sklearn.model_selection.KFold(n_splits=k, shuffle=True, random_state=10)

כאשר *k* הינו מספר הfolds.

יש להשתמש בקריאה זו הן בחלק 1 והן בחלק 2 (תחרותי).

3. algorithm_runner.py – ממש להפעלת אלגוריתמי סיווג

מחלקה AlgorithmRunner:

<u>data members:</u>

algorithm – אובייקט של המודול sklearn דרכו נבצע את פעולת האלגוריתם (כל פו' היא אובייקט בפייתון). שימו לב שכל מופע של המחלקה AlgorithmRunner פועל מסווג מסוג אחד בלבד ולא שניהם.

עבור KNN:

```
from sklearn.neighbors import KNeighborsClassifier
```

עבור Rocchio:

```
from sklearn.neighbors import NearestCentroid
```

ניתן להעביר ל__init__ ארגומנט המציין אילו מבין השניים יש לאתחל, וו עבור KNN.

:function members

run(self, Data, folds=5)

המתודה מקבלת אובייקט מסוג Data ומדווחת

- Precision
- Recall
- Accuracy

לשני האלגורתמים, בשיטת CV. ניתן להשתמש ב

```
from sklearn.model_selection import cross_validate
```

בקריאה לפונקציה יש להשתמש ב:

cv=data.split_to_k_folds(folds)

חלק ב' – יבש (40%):

1. בהינתן 3 תצפיות בלבד, Agglomerative Clustering בשיטות Complete, Single and Average יניב תוצאות זהות (לשלושת הגישות). הוכיחו או הפריכו את הטענה.

(8%)

2. Agglomerative clustering בגישת ward ואלגוריתם k-means בהכרח יתנו את אותו פלט לכל קלט. הוכיחו או הסבירו מדוע הטענה אינה נכונה.

(8%)

3. הוכיחו את הטענה הבאה:

נתונה שאילתה q, ומאגר מסמכים C. נניח שכל המסמכים במאגר C, וגם השאילתא q, מיוצגים ע"י וקטורי משקולות tf. אזי דירוג המסמכים בסדר יורד לפי cosine similarity שקול לדירוגם בסדר עולה לפי Euclidian distance. כלומר לכל שני מסמכים $d, e \in C$:

$$\cos(q,d) < \cos(q,e) \Leftrightarrow \|d-q\|_2 > \|e-q\|_2$$

(12%)

4. להלן שאלות נכון/לא נכון. נמקו את תשובתכם.
   a. ככל השה*recall* גדל אזי בהכרח גם ה*precision* גדל, ולהיפך.

(4%)

   b. במידה וקיבלנו *accuracy* של 100% במדגם האימון, בסבירות גבוהה שה*accuracy* במדגם המבחן יהיה גם כן גבוה.

(4%)

   c.
ערך ה*tf*idf* של מילה יהיה גבוה יותר ככל שמספר הפעמים שמילה הופיעה במסמכים שמכילים אותה יורד, וככל שהשכיחות שלה במאגר עולה.

(4%)

דגשים נוספים:

1. עליכם לכתוב את הקוד בהתאם לדגשים והסטנדרטים לפי pep8. לשימושכם המסמך Code Quality " Requirements " באתר ה moodle- של הקורס. קוד אשר לא יעמוד בסטנדרטים הנדרשים, יקבל ניקוד מופחת.
2. ניתן להוסיף מתודות נוספות, במידה ותמצאו לנכון. יש להימנע מכפילויות קוד.
3. ניתן להשתמש במתודות שהן in-built בשפה. קרי, מתודות אשר **לא** דורשות ייבוא של ספריות.
4. יש לתת שמות בעלי משמעות לכל משתנה.
5. חובה לתעד את הקוד באנגלית. בפרט עליכם לכתוב עבור כל מתודה docstring

הוראות הגשה:

- התרגיל להגשה בזוגות בלבד.
- לפני ההגשה, חובה לוודא שהתוכנית עובדת במעבדת ההוראה ולא בסביבה אחרת.

- ההגשה חייבת להכיל קובץ אחד (קובץ zip) :
  - שם הקובץ חייב להיות hw4 _xxxxxxxx_yyyyyyyy.zip כאשר xxxxxxxx ו-yyyyyyyy הם מספרי תעודות הזהות של המגישים, כולל ספרת ביקורת.
  - הקובץ מכיל את כל קבצי הקוד. אין להכיל תיקייה ובתוכה קבצי הקוד, אלא את קבצי הקוד עצמם.
  - **הערה :** עליכם לוודא שהתוכנית מתחילה לפעול מקובץ "main.py" בלבד.
  - תשובות לחלקים יבשים יש להקליד במעבד תמלילים. אין להגיש תשובות בכתב יד.

- ההגשה היא אלקטרונית בלבד, דרך אתר ה-moodle של הקורס. תרגילים שיוגשו בכל דרך אחרת לא ייבדקו.
- אין להגיש את אותו הקובץ פעמיים. התרגיל יוגש ע"י אחד מבני הזוג.
- שימו לב שההגשה תיחסם בדיוק בשעה 23:55 ביום ההגשה. מומלץ להגיש לפחות שעה לפני המועד האחרון.
- ניתן להגיש כמה פעמים. רק ההגשה האחרונה תישמר.
- תרגיל בית שלא יוגש לפי הוראות ההגשה – לא ייבדק (כלומר יקבל ציון 0).
- לצורך תרגיל הבית ייפתח פורום. ניהול שאלות ומתן תשובות בנושא התרגיל יתבצע דרך הפורום בלבד.

בהצלחה!