בס"ד

בינה מלאכותית - תרגיל 2 - למידה

<u>הבעיה</u> – נתונים הקבצים הבאים:

- קובץ dataset.txt בו נתונות דוגמאות המתארות מאפיינים שונים.
- קובץ Attribute Information.docx בו נתונים המאפיינים השונים וערכיהם.

בתרגיל זה נבצע חיזוי באמצעות האלגוריתמים Decision Tree, KNN ו- naïve base.

יש להעריך את הדיוק על פי K-FOLD CROSS VALIDATION, עם K=5

כתוב תוכנית הקוראת מקובץ dataset.txt את סט הדוגמאות כאשר, **השורה הראשונה** של קובץ זה תכלול את שמות השדות (כלומר, את המאפיינים של הנתונים)
הערכים האפשריים של כל מאפיין הם הערכים שמופיעים בעמודת המאפיין בקובץ ה-dataset (לא יהיו ערכי מאפיינים שלא יפיעו בקובץ).
העמודה האחרונה בכל שורה הינה הסיווג (ה-class).
כל הערכים מופרדים ב <tab>.

שמות המאפיינים תמיד יורכבו מתוים ללא רווחים.

כל המאפיינים והערכים האפשריים השונים של ה dataset ניתן לראות בקובץ Attribute Information המצורף לכם.

<u>Decision Tree</u>

כתבו פונקציה שמיישמת את אלגוריתם ID3.
את העץ שנבנה מהאלגוריתם יש להדפיס לקובץ בשם tree.txt בפורמט הבא:

```
<attribute_name>=<attribute_value>
<tab>|<attribute_name>=<attribute_value>:class
```

מבחינת ערכי ה – values של ה - attribute - ההדפסה צריכה להיות בסדר אלפביתי

לדוגמה:

```
age = child
 |pclass = crew: yes
 |pclass = 1st: yes
 |pclass = 2nd: yes
 |pclass = 3rd: no
```

שים לב שדוגמה זו רק מדגימה את הרעיון הכללי של פורמט הפלט.

(בנוסף, מצוף לכם קובץ דוגמא לעץ – **אין** זה העץ עבור פיתרון התרגיל)

KNN

כתבו פונקציה הממשת את אלגוריתם KNN כאשר K=5. חישוב המרחק יעשה באמצעות מרחק hamming.

(מידע על מרחק hemming ניתן למצוא כאן – בהקשר שלכם, תתייחסו לכל feature כתו)

במידה ויש לכם יותר מ K אובייקטים במרחק הקטן ביותר, יש לקחת את ה K הראשונים לפי סדר טעינת הנתונים)

Naïve Base

כתבו פונקציה הממשת את חיזוי naïve base.

Accuracy

לאחר בנית המודל לכל אחד מהאלגוריתמים, הדפס לקובץ accuracy.txt את דיוק החיזוי שיצא לכם בפורמט הבא:

<DT_accuracy>**tab**<KNN_accuracy>**tab**<naiveBase_accuracy>

בדיוק של 2 ספרות אחרי הנקודה. (סטיה של ספרה למעלה\למטה לא תוריד ניקוד)

יש להגיש:

- קובץ details.txt בו יש לכתוב את שם המגיש באותיות אנגליות קטנות בשורה הראשונה ובשורה השניה את מספר ת.ז.
- קובץ py_ex2 אשר יכיל את הקוד. (יש לתעד את הקוד)
- קובץ tree.txt ו- accuracy.txt עם התשובות שלכם

Test

עבור הבדיקה הסופית התוכנית תקבל קובץ train.txt, test.txt (שמות קבצים אלו הם hard-code) בונה את שלושת המסווגים לעי"ל מסט הנתונים train.txt ומחזירה קובץ output.txt – כשירשור של הקובץ tree.txt <שורה רווח> ולאחריו ה accuracy.txt כאשר הדיוק הוא עבור הקובץ test.txt שהתקבל.

(**הערה!!!:** בעת בדיקת הקוד לא בהכרח נשתמש בנתונים שקיבלתם; לכן, אל תשתמש ב hard-code לנתונים ספציפים.

במקרה של שוויון שיש לסווג לפי הסיווג השכיח יותר.
במקרה של ששני הclass-ים שכיחים באותה מידה, אין עדיפות לסיווג ספציפי.

אתם כן יכולים להניח שגם בבדיקות עתידיות הסיווג הסופי תמיד יהיה בינארי בשימוש בערכים yes\no)

בהצלחה!

הנחיות כלליות:

- ההגשה ביחידים בלבד. תתבצע בדיקת העתקות.
- ניתן לכתוב את התוכנית ב- python בלבד.
- בתרגיל זה אין להשתמש בשום סיפריה של python (כולל numpy).
- יש לוודא שהתוכנית מתקפלת ורצה על שרת המחלקה planet.
- גרסת ה python שרצה בשרת היא 3.6
- לכל פונקציה יש להקדיש לפחות שורה אחת של תיעוד. לכל מחלקה יש להקדיש לפחות 2 שורות של תיעוד.
- ההגשה מעשית (קוד) דרך מערכת submit (עזרה בנושא נמצאת ב- http://help.cs.biu.ac.il/submit.htm)
- במידה ולא הגיעה הודעת דוא"ל המאשרת את שליחת התרגיל- התרגיל לא הוגש.
- שם התרגיל למערכת submit יהיה py_ex2.
- יש להגיש קבצי מקור בלבד (source code).
- במידה ויינתן קלט לדוגמא, ודאו שתוכניתכם עובדת איתו, אך זהו לא הקלט איתו תיבדק התוכנית.
- לכל תרגיל יש לצרף קובץ טקסט שייקרא details.txt. הקובץ יכלול תמיד בהתחלתו את פרטי המגיש בפורמט הבא:

  <ID> <first name> <last name>

  (another requirements to the specific assignment…)

לדוגמא:

876543210 Shimon Peres

(another requirements to the specific assignment…)

שימו לב כי אין בתעודת הזהות סימן להפרדת ספרת הביקורת. יש רווח בודד בין רשומה לרשומה.