בס"ד

דוח למידת מכונה – תרגיל 2:

הסבר האלגוריתמים ונכונותם:

1. **KNN–** נרצה לממש את האלגוריתם שנלמד בתרגול.
עבור כל ערך בtest, נמצא את k הדוגמאות הכי קרובות אליו (באמצעות נורמה) מתוך ה-train.
נבדוק מה הסיווג שמופיע הכי הרבה פעמים בk הדוגמאות שמצאנו – וזה יהיה הסיווג של אותו ערך שבדקנו בtest.
2. **Perceptron–** מחולק לאימון וחיזוי. בפונקציית האימון נרצה למצוא את וקטורי המשקולות הכי טובים.
נממש כפי שנלמד בתרגול.
באימון - נאתחל שלושה וקטורי משקולות (כיוון שיש לנו 3 מחלקות – 0,1,2). עבור כל ערך בtrain, נבצע מכפלה בינו לבין w. נשמור את הווקטור מתוך w שהחזיר את המקסימום.
אם החיזוי אינו צלח, כלומר בחרנו בערך הלא נכון- נרצה לעדכן את וקטורי המשקולות להיות טובים יותר.
נבצע את העדכון כפי שהוגדר בקובץ בתרגיל.
בפונקציית החיזוי, נחשב את כל הy_hat (עבור כל ערך בtest), ונחזיר את רשימת החיזויים.
3. **PA–** ביצוע דומה לאלגוריתם הקודם. גם פה נחלק לאימון וחיזוי. החלק השונה הוא – עדכון וקטורי המשקולות (כפי שהוגדר בתרגיל).
4. **SVM–** גם פה נקבל אלגוריתם דומה. נעדכן את וקטורי המשקולות באופן שונה.
בנוסף, בביצוע argmax לא נרצה לקבל את y_i, כלומר נרצה לכאורה להסיר את הערך במיקום הנ"ל במכפלה שהתקבלה. פתרון לכך- נשנה את הערך במיקום הזה להיות מינוס אינסוף, ובכך נבטיח שargmax לעולם לא תחזיר את הערך הזה.

נרמולים:

לפי הנלמד בתרגול, נממש את שני הנרמולים הבאים: 1. zscore 2. minmax.
נממש לפי הנוסחאות שנלמדו בתרגול, באופן הבא (מימוש בקובץ התרגיל שלי):

- Min-Max Normalization -> $v \in [min_A, max_A], \rightarrow v' \in [new\_min_A, new\_max_A]$

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- Z-Score Normalization

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

בחירת הנרמול המתאים עבור כל אלגוריתם – הרצת האלגוריתם עם נרמול מסוים ובדיקת התוצאה מול הרצת האלגוריתם עם הנרמול השני או ללא נרמול כלל. נבחר את הנרמול שיוביל אותנו לתוצאה טובה יותר. בסופו של דבר, הנרמולים לא סייעו לKNN ולכן לא ביצעתי שם נרמול. באלגוריתמים האחרים, התוצאות היו טובות לאחר נרמול zscore ולכן נרמלתי את המידע שם לפי נרמול זה.

Bias:

נבצע לערכי הtrain וה-test. כלומר, נוסיף עמודה חדשה (הוספתי עמודת אחדות).
ובאופן דומה, נצטרך להתאים את גדלי וקטורי המשקולות – להוסיף גם להם עמודה (שם הוספתי עמודה שאותחלה לאפסים). ביצוע הbias הוביל לשיפור התוצאות בכל האלגוריתמים (פרט לKNN שבו לא היה צורך).

בס"ד

בדיקת אחוזי דיוק עבור כל הרצה:

נבנה פונקציה שבודקת את אחוזי ההצלחה – כלומר, נבדוק כמה תוצאות שגויות קיבלנו, ביחס לסך כל הדוגמאות.
נרצה להדפיס לנו את אחוז הסיווגים הנכונים, כדי לקבל אינדיקציה אודות נכונות האלגוריתם.

מימוש – נחלק את הtrain לk חלקים, כאשר באיטרציה מסוימת, למשל באיטרציה ה-i, החלק ה-i של הtrain יהווה את הvalidation, ושאר הtrain יהווה את האימון. על מנת לקבל הערכת מצב, נחזיר את ממוצע כל הריצות – וזה יהיה אחוז הדיוק המשוער שלנו, אותו נרצה למקסם.

בחירת מספר האיטרציות – epoch:

נציב מספרים וננסה לקבל אינטואיציה. נבדוק מה המספר שנותן לנו אחוזי הצלחה גבוהים, וזה מספר האיטרציות שנרצה שהאלגוריתם ירוץ.
מבחינת מימוש (קוד), נוכל לבנות פונקציה באופן הבא: ניצור לולאה שתבדוק את אחוזי הדיוק עבור כל מיני ערכי epoch. עבור כל ערך נבדוק את אחוזי ההצלחה, ונעדכן אם התקבלו אחוזים גבוהים יותר.
בסוף, נחזיר את הערך עבורו התקבלו אחוזי ההצלחה הגבוהים ביותר.

בחירת k אופטימלי לאלגוריתם KNN:

נרצה למצוא את הk שעבורו נקבל את אחוז השגיאות הקטן ביותר.
נממש פונקציה שמחלקת את הtrain לשני חלקים – 70 אחוז אימון, ו30 אחוז validation.
נרוץ על כל הk האפשריים בטווח מסוים (למשל עד שורש גודל הtrain), ונמצא את הk שעבורו מתרחשות הכי פחות שגיאות. זה הk שנבחר.
לאחר הרצת הפונקציה קיבלתי כי k=9.

מציאת ערכי η, λ:

נתחיל להציב מספרים באופן ידני, נריץ את הפונקציה שבודקת את אחוזי הדיוק עבור הערכים השונים שנציב.
נרצה להישאר עם הערכים שיביאו אותנו לתוצאות יציבות ואחוזי דיוק גבוהים.
הערכים שהביאו לי תוצאות טובות מצוינים בקוד שלי בקובץ התרגיל.

בדיקה נוספת שנבצע – **shuffle** על המידע, כלומר ערבוב הtrain על מנת לוודא שלאלגוריתמים אין תלות בסדר הנתונים המתקבלים, ושעדיין מתקבלות תוצאות טובות גם לסדר אחר של המידע.