מבוא לבינה מלאכותית 236501
תרגיל בית 3

יוני בן־צבי 203668900 דני פריימק 307003434

מבנה הקוד

פונקציית ה־ main נמצאת בקובץ classifier.py. ה־ main מכיל את הפונקציות שמריצות את הקוד של שאלות $3, 5, 7$. כמו־כן, הפונקציה האחרונה מייצרת את קובץ ה־ results.data של חלק ג'. לבסוף, בפונקציית ה־ main ובשאר הקובץ ניתן לראות פונקציות ששומשו לבדיקות וניסויים. הסבר מפורט יותר מופיע בחלק ג'.

חלק ב'

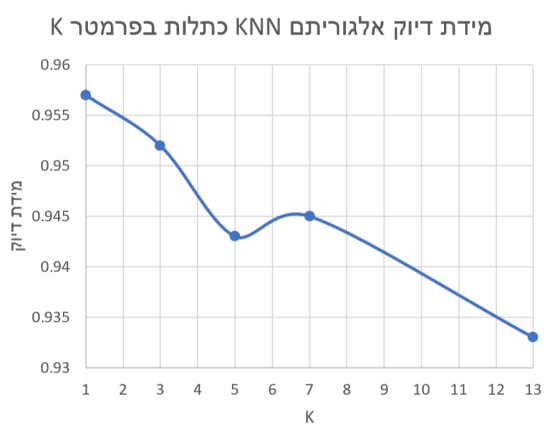
שאלה 3

סעיף 2

חשוב לשמור על עקביות זו משום שהחלוקה אקראית. אמנם מספר הדוגמאות המסווגות כחיוביות ושליליות קבוע בכל קבוצה עבור כל הרצה של הפונקציה (עם אותו num_folds) אבל הדוגמאות מחולקות בין הקבוצות באופן אקראי. לכן, על־מנת לשמור על אותו מסווג יש להשתמש באותה החלוקה, כלומר לקרוא לפונקציה פעם אחת בלבד.

שאלה 5

סעיף 2



איור 1: דיוק אלגוריתם $k$nn על קבוצת המבחן עבור num_folds=2 כפונקציה של k

סעיפים 3 ו־ 4

כפי שניתן לראות מהגרף בסעיף 2, הביצועים הטובים ביותר התקבלו עבור $k = 1$ והביצועים הגרועים ביותר התקבלו עבור $k = 7$. ערכי הדיוק הממוצע הם 0.953 ו־0.937 בהתאמה. ראשית, נציין שהבדלי הדיוק יחסית קטנים וכן החלוקה ל־ folds

אקראית, לכן לא ניתן להסיק מסקנות נחרצות מדי מהתוצאות. אם בכל־זאת מנתחים את התוצאות ניתן לשער שהביצועים היו מיטביים עבור $k = 1$ וגרועים ביותר עבור $k = 7$ משום שהגיוני שככל שאדם דומה יותר לאדם אחר (מבחינת נתונים גופניים), כך סביר יותר שיהיו לו בעיות רפואיות דומות לאותו אדם. לכן, במקרה שלנו, סביר שלאדם יהיו בעיות לב אם לאדם הדומה לו ביותר מבחינת נתונים גופניים יש בעיות לב, והתחשבות באנשים דומים פחות יכולה אף לפגוע בדיוק המסווג כפי שניתן לראות בתוצאותינו. בתוצאותינו ניתן כמו־כן להבחין במגמת ירידה בטיב הביצועים ככל שגדל ערכו של k, דבר שמחזק את הנימוק לעיל (הדיוקים הממוצעים עבור $k = 3, 5$ כמעט זהים וכנ"ל עבור $k = 7, 13$).

שאלה 7

סעיף 4

הניסוי שבו התקבלו התוצאות הטובות ביותר הוא הניסוי משאלה 3 סעיף 5 ($k$nn עם $k = 1$).

חלק ג'

נסיונות השיפור שבחרנו לנסות על־מנת למקסם את אחוזי הדיוק של המסווג היו:

1. שימוש במס' אי־זוגי של מסווגים שונים על אותו אובייקט מבחן, והכרעה עפ"י רוב קולות בהצבעה. המסווגים שבחרנו הם ID3, מסווג Perceptron, ומסווג $k$nn עבור $k = 1$ (שהניב את התוצאות המדוייקות ביותר בתהליך הפיתוח). נעיר כי את רמת הדיוק של המסווג לאורך כל התהליך, בחנו בעזרת שיטת Stratified k-fold cross validation עבור ערכי k שונים, כדי לקבל תוצאות ממוצעות מדוייקות ככל הניתן מתוך קבוצת המבחן.

2. מסווג נוסף ששקלנו להכניס להצבעה היה מסווג בייסיאני נאיבי מתוך ספריית sklearn (שתי המחלקות שמימשנו בעצמנו MultinomialNB_classifier ו־ MultinomialNB_factory עדיין תחת הערה בקוד). המסווג מסוג MAP שקיים בספריה (MultinomialNB) עובד עם תכונות שערכיהן נעים בטווח $[0, 1]$. מאחר והתכונות שקיבלנו בתרגיל אינן בטווח זה, לא רצינו לשנות או לנרמל אותן מתוך חשש שיפגע בדיוק הסיווג (כי לדוגמא, יתכן שחלק מהתכונות הן ב־scale לוגריתמי, ונרמול ליניארי ישנה את ה־ data באופן מהותי ולא מדויק). מסווגים בייסיאנים נוספים בספריה שמצאנו הניחו התפלגויות ידועות של הסיווגים שהיו שונות מההתפלגות האמיתית שהתקבלה בקבוצת האימון, ולכן השימוש בהם לא היה נכון מבחינה עקרונית.

3. הסרת תכונות החשודות כפחות רלוונטיות לסיווג. כדי למצוא את התכונות שהורידו את אחוזי הדיוק של המסווג, נקטנו בתהליך אלימינציה שבו הסרנו בכל צעד תכונה יחידה מתוך סט התכונות, ובדקנו את תוצאת הסיווג של מסווגי ה־ Perceptron וה־ ID3 (ללא $k$nn בשל זמן סיווג איטי יותר) לאחר הסרת תכונה זו (הפונקציות checking_bad_features, evaluate_without_bad_features ו־ evaluate_without_known_bad_features שימשו למטרה זו). התבוננו בקבוצת החיתוך של שתי קבוצות התכונות הבעייתיות (אחת לכל מסווג), והסרנו תכונות אלו מקבוצות האימון והמבחן. לבסוף הסרנו בסה"כ 5 תכונות שאכן הניבו שיפור קטן בדיוק הסיווג.

4. דרך נוספת ששקלנו לשיפור הסיווג הייתה זיהוי והסרת דוגמאות רועשות. הדרך שבה חשבנו לעשות זאת, היא להתבונן בדוגמאות שסיווגן נכשל עבור כל המסווגים באופן גורף. דוגמאות אלו הינן מועמדים טובים להיות דוגמאות רועשות. אך, משום שמהסווגים אינם מניבים תוצאות מושלמות, יתכן מאד שדוגמא שאינה רועשת תיחשב רועשת באופן שגוי. במילים אחרות, כדי למצוא דוגמאות רועשות אנו צריכים מסווג טוב, וכדי לבנות מסווג טוב אנו צריכים לדעת להיפטר מדוגמאות רועשות, וזו כמובן בעיה מעגלית. לכן, לבסוף זנחנו את גישת השיפור הזו (למרות שיתכן וטעינו).