חלק ב'

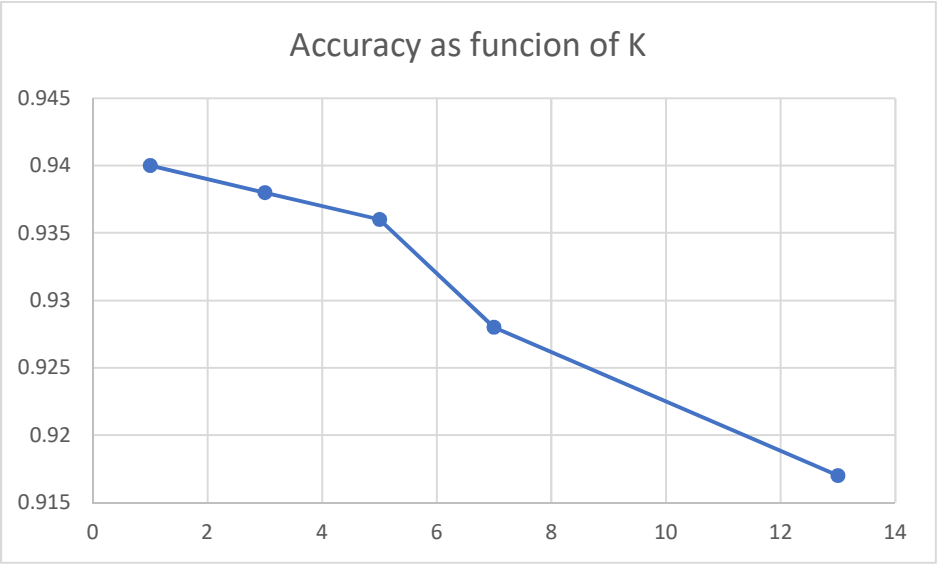
שאלה 3:

1. שמנו לב כי בעת חלוקה לקבוצות תוך שמירה על יחס קבוע בין הדוגמאות החיוביות לשליליות נוצר לעיתים (בעת עבודה עם שברים) trade-off בין שמירה על מספר אחיד של דוגמאות מתויגות בקבוצה לבין שמירה על היחס המדויק בין הדוגמאות בתוך הקבוצה.
החלטנו, לאחר מחשבה רבה, שהחשיבות הגדולה ביותר עבורנו היא גודל הקבוצה האחיד ככל הניתן. כך אנו מוודאים שאנחנו מנצלים את כל הדוגמאות שניתנו לנו לאימון ולמבחן כיוון שהן אלו שמוסיפות אינפורמציה עבורנו בעת בנייה ובדיקה של מסווג. וויתור על מספר דוגמאות משמעותו ויתור על אינפורמציה שיכולנו לעשות בה שימוש. מנגד, איזון היחס בין הדוגמאות הוא אופטימיזציה לבדיקה בלבד. כלומר, אנחנו מעדיפים להתפשר על יחס הדוגמאות שגם ככה משתנה בצורה מאוד זניחה עם עיגול השברים ולהבטיח כי אנחנו משתמשים בכל הדוגמאות שניתנו לנו ושומרים על גודל קבוצה אחיד.
כמו כן נציין כי יש חשיבות גדולה לכך שלא תהיה חפיפה בין הקבוצות השונות.
יוצא דופן יחיד הוא במקרה שגם גודל הקבוצה המבוקש הוא שבר. במקרה זה נחלק את מספר הדוגמאות שחלוקתו תוביל למנה שלמה בעוד שאת יתר הדוגמאות נפזר בין כמה קבוצות שנוכל. (כך למשל עבור חלוקת 1000 דוגמאות ל6 קבוצות נקבל 4 קבוצות בגודל 167 ו-2 בגודל 166.)

2. הסיבה לשמירה על עקביות באמצעות שימוש באותם קבצי חלוקה לאורך כל התרגיל הינה שאנחנו בונים מסווג על בסיס סט דוגמאות. לכן, כאשר נרצה לבדוק התנהגות של מסווגים שונים חשוב לנו שמסווגים אלו יאומנו תמיד עם אותה קבוצת דוגמאות. בצורה כזו, נבודד את המשתנים שאותם אנחנו בודקים (=אלגוריתם המסווג) ולא נכניס תלויות נוספות לבדיקה (=קבוצת דוגמאות לאימון). כך נבטיח שהמסקנות שאליהן נגיע בניסויים יהיו תלויות בסוג המסווג ואלגוריתם הלמידה שלו בלבד ולא בקבוצת הדוגמאות שאיתן אומן.

שאלה 5:

2. הגרף שהתקבל הינו:



3. ערך ה-K שעבורו התקבלו הביצועים הטובים ביותר הינו k=1.

4. ננתח את הגרף.

- <u>ערכי המקסימום</u> של הגרף מתקבלים עבור K=1. ניתן להסביר זאת מכיוון שהמסווג בוחר רק את הדוגמא הקרובה ביותר ומסווג לפיה. נשים בל כי עבור K=1 מובטח כי שגיאת האימון תהיה אפס (תחת ההנחה שאין שתי דוגמאות בעלות אותם מאפיינים וסיווג שונה) שכן בחיפוש אחר הדוגמא הקרובה ביותר נמצא את הדוגמה אותה אנו בודקים. בדקנו שאכן בסט הדוגמאות שניתן אין סתירות מסוג זה. נצפה כי שגיאת המבחן עבור K=1 תהיה גדולה יותר משגיאת האימון וכן שתהיה רגישה יותר ל"רעשים" שהם דוגמאות שקריות במערכת. אכן קיבלנו שגיאת אימון גדולה יותר (דיוק של 94%) אך עדיין זו השגיאה המינימלית שהתקבלה לעומת ערכי K האחרים. נוכל להסביר זאת ע"י כך שקבוצת הדוגמאות שלנו מכילה בעיקר דוגמאות המתויגות בערך "True", וביחס שגדול מ-1:3 ולכן עבור כל K גדול מ-1 יש סבירות גדולה שנקבל שגיאת False Positive ונתייג דוגמא כ-True.
- <u>ערכי המינימום</u> – ערך המינימום לגרף מתקבל עבור K=13. ניתן להסביר זאת נוכח העובדה שהמסווג יתחשב במספר גדול מדי של דוגמאות, חלקן עלולות להיות רחוקות מאוד מהדוגמה שאותה אנו בוחנים. לא נרצה לקחת קבוצה גדולה מדי של שכנים, שכן אנו עלולים לבסס את הסיווג שלנו עבור הדוגמא הרלוונטית על דוגמאות רחוקות מדי. סיווג באמצעות דוגמאות רחוקות יוסיף "רעש" בהסתברות גבוהה ויוריד את אחוז ההצלחה.
- <u>הגרף הינו מונוטוני יורד</u> – כפי שערכי הקיצון השונים של הגרף מעידים ככל שאנו מוסיפים יותר דוגמאות לקבוצת השכנים הקרובים ביותר שלנו, אנו פוגעים באחוז ההצלחה שלנו. ניתן להסביר זאת באמצעות העובדה שככל שאנו מוסיפים יותר דוגמאות אשר אנו מבססים את הסיווג שלנו עליהם, אנו מוסיפים גם דוגמאות אשר מוסיפות רעש ופוגעות באחוז ההצלחה. לכן ככל ש"נתרחק" ונגדיל את מספר השכנים, כך גם אחוז ההצלחה שלנו יירד.
כפי שלמדנו בהרצאה, הגדלת K מחליקה את גבולות ההחלטה בין הקבוצות. כיוון שקבוצת הדוגמאות שלנו מכילה בעיקר דוגמאות המתויגות בערך "True" נקבל כי ככל ש-K גדל, דוגמאות ה-True "בולעות" את דוגמאות ה-False המעטות במספר. מסיבה זו, קיבלנו אחוזי הצלחה גבוהים יחסית עבור כל ערכי K וזאת משום שדוגמאות ה-True מהוות 88.1% מכלל הדוגמאות ולכן גם עבור ה-K הגרוע ביותר (K ששווה למספר הדוגמאות ומחזיר ערך אחיד לכל סיווג) נבטיח 88.1% הצלחה.

שאלה 6:

מבין כל הניסויים שהורצו, התוצאה הטובה ביותר התקבל עם מסווג KNN עבור K=1.

חלק ג' - תחרות

נתאר ראשית את אופן פעולת המסווג שבנינו לתחרות ולאחר מכן כיצד ביצענו את הבדיקות עליו לאורך בנייתו.

ראשית כל, שמנו לב כי רשימת התכונות של כל דוגמא מכילה 187 מאפיינים. הנחנו שלא כל המאפיינים הללו נחוצים לשם סיווג הדוגמה ובנוסף למדנו כי מסווג KNN רגיש לתכונות מיותרות אשר עלולות "לעוות" את המרחק האוקלידי המחושב בעת חיפוש דוגמאות קרובות. בשביל לפתור בעיה זו החלטנו ראשית כל לעבד את המידע הגולמי שקיבלנו ולבצע feature selection באמצעות מימוש אלגוריתם בחירה מקומית לפנים (SFS) כפי שנלמד בהרצאה. באלגוריתם זה אנו מוסיפים כל פעם בצורה חמדנית את התכונה שמניבה עבורנו את ה"רווח" הגדול ביותר. את הרווח שמניבה לנו תכונה הגדרנו באמצעות בדיקת cross_val_score שמסופקת בספריה sklearn עבור CV בגודל 4. בצורה כזו, חילקנו את קבוצת המבחן ל-4 קבוצות כאשר בכל פעם בודדנו קבוצה יחידה והשתמשנו בה כקבוצת מבחן. את המבחן ערכנו באמצעות מסווג KNN עם K=3, וה"רווח" היה הממוצע על 4 התוצאות של דיוק המסווג. הערכים לחלק זה (CV של 4 וk=3) נבחרו לאחר ניסוי וטעייה, הם נבחרו באופן שרירותי ובדקנו ידנית את הערכים העוקבים לכל כיוון וראינו כי הערכים שנבחרו הניבו לנו את הדיוק הגדול ביותר. בנוסף, גילינו בצורה דומה שמספר התכונות האופטימלי עבורנו הינו 10 תכונות. (בדקנו גם את אלגוריתם חיפוש מקומי לאחור SBS אך הוא הניב תוצאות פחות טובות וגם היה איטי בצורה משמעותית.)

נציין שאת ה-feature selection ביצענו על כלל הדוגמאות המסופקות ולאחר בחירת התכונות "סיננו" הן את הדוגמאות המתויגות והן את דוגמאות המבחן. כך קיבלנו שרשימת המאפיינים של כל דוגמא מכילה כ-10 תכונות.

כעת, כשהמידע "מעובד", החלטנו לבצע feature generation ולייצר תכונות נוספות שמעניינות אותנו. תכונות אלו הן תוצאות הסיווג של מסווגים שונים. המסווגים שבחרנו לעבוד איתם הינם:

1. ID3
2. עץ עם גיזום: ערך מינימום של 3 דוגמאות בעלים
3. עץ ID3 עם גיזום: ערך מינימום של 19 דוגמאות בעלים ועומק מקסימלי של 3
4. KNN עם K=1 שלוש פעמים
5. KNN עם K=2 פעמיים
6. KNN עם K=3 פעם אחת
7. Random Forest

כלומר, כעת יש בידינו תכונה חדשה עבור כל מסווג: "תוצאת הסיווג של המסווג על הקלט"

מוטיבציה לבחירת המסווגים הנ"ל:

בחרנו לשים KNN עם K=1 3 פעמים ולתת לו משקל גבוה בחישוב, שכן הוא היה המדויק ביותר ביחס למסווגים האחרים (ראינו זאת בחלק ב' ובמספר הרצות ידניות). באופן דומה רצינו לתת משקל גבוה גם ל-KNN עם ערך K=2 ולכן הכנסנו אותו פעמיים. כל שאר המסווגים קיבלו משקל זהה.

רצינו להשתמש במגוון רחב של מסווגים כיוון שלכל אחד מהם היתרונות והחסרונות שלו והאמנו ששקלול של התוצאות שלהם ("ועדה") יאפשר לנו לזכות בטוב מכל העולמות. מסווג Random Forest מממש אף הוא וועדה כפי שלמדנו בהרצאה אך לא רצינו להסתמך עליו בלבד. כמו כן השתמשנו במספר וריאציות של עצי החלטה. מסווג ID3 רגיל כפי שנלמד בכיתה (אלגוריתם חמדני שבוחר תכונות לפי IG), מסווג ID3 עם הגבלה על מספר דוגמאות בעלים ועומק מקסימלי ומסווג עץ החלטה עם ערך מינימום של 3 דוגמאות בעלים. האחרונים מאפשרים גיזום מוקדם של העץ מתוך תקווה שהגדלת שגיאת האימון תקטין את שגיאת המבחן ע"י החלשת אפקט התאמת היתר. החלטנו לשנות את הפרמטרים הללו שנלמדו בכיתה וראינו שאכן הם משפרים את אחוז הדיוק על קבוצות המבחן שלנו.

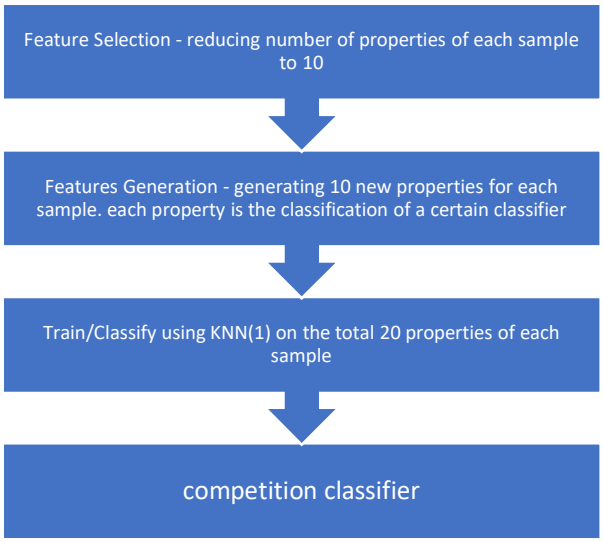
בתחילת הפעולה של המסווג שלנו, אנו מפעילים את המסווגים הנ"ל על הקלט ועבור כל דוגמה עם סט features אנו משרשרים לרשימת התכונות שלה את הפלטים של המסווגים. כך למעשה יצרנו 10 תכונות חדשות וכעת כל דוגמה מכיל 20 מאפיינים, 10 שנותרו לאחר עיבוד המידע ו-10 מאפיינים חדשים שהוספנו. פעולה זו נקראת

feature generation. נציין כי תחילה החלפנו לחלוטין את רשימת התכונות ב-10 התכונות החדשות אך ניסויים הראו כי שיפור הדיוק לא היה משמעותית במקרה זה אך השתפר כאשר בחרנו להוסיף את התכונות על המידע הקיים. באמצעות feature generation אנו חושבים שנוכל לשפר את אחוז דיוק התוצאות, שכן אנו בעצם מוסיפים תכונות נוספות עבור כל וקטור תכונות שיש לנו. בצורה זו, במקום לקבוע מראש את המשקל של כל מסווג ולחשב מה היא החלטת הרוב המשוקללת, אנחנו בוחרים לאמן וללמד מסווג חדש על בסיס התכונות הללו מתוך אמונה שהוא יבצע החלטות חכמות יותר מאיתנו ובעצם יקבע את המשקל הנכון בכוחות עצמו. עם זאת, כן שמנו מספר עותקים של מסווגים מסויימים כדי "לכפות" את הדעה שלהם יותר משל השאר.

לבסוף את הדוגמאות הסופיות (בעלות 20 תכונות) מכניסים כקלט לאימון של מסווג KNN עם K=1. כפי שנוכחנו לדעת בסעיפים הקודמים מסווג זה הוא בעל אחוז הדיוק הגבוה ביותר ולכן החלטנו להמשיך איתו.

נסכם בקצרה את אופן פעולת האימון ואופן פעולת הסיווג של המסווג שבנינו:

נצייר את סכמת המסווג שלנו:



אימון:

1. עיבוד מידע – צמצום 187 תכונות של כל דוגמא ל-10 תכונות באמצעות אלגוריתם SFS.
2. עבור כל דוגמא בסט האימון המעובד:
   a. סיווג עפ"י כל 1 מ-10 המסווגים המצוינים לעיל.
   b. שרשור 10 הסיווגים כ-10 תכונות נוספות לרשימת התכונות של כל דוגמא.
3. אימון מסווג KNN עם K=1 על קבוצת הדוגמאות לאחר הוספת התכוות החדשות.

סיווג:

1. עיבוד מידע - צמצום 187 תכונות של כל דוגמא ל-10 התכונות כפי שנקבעו בשלב באימון.
2. עבור כל דוגמא בסט המבחן המעובד:
   a. סיווג עפ"י כל 1 מ-10 המסווגים המצוינים לעיל.
   b. שרשור 10 הסיווגים כ-10 תכונות נוספות לרשימת התכונות של כל דוגמא.
3. סיווג באמצעות מסווג ה-KNN המאומן על תכונות אלו.

את המסווג שלנו בדקנו בצורה הבאה (הרצנו את הבדיקה כל פעם שרצינו לכוונן פרמטר אחר כך ששינינו פרמטר יחיד בכל הרצה כדי לבודד את התלויות):

עבור $k \in \{2,4,6,8,10\}$ פיצלנו כל פעם את קבוצת הדוגמאות שלנו באמצעות הפונקציה train_test_split שמסופקת עם sklearn. עבור כל k ביצענו k איטרציות שבהן חילקנו את הקבוצה כך שגודל קבוצת המבחן יהיה 1/k. הקבוצה חולקה בצורה רנדומלית, בכל פעם לפי ערך seed אחר כך שלא נחזור על אותה בדיקה יותר מפעם אחת.

עבור כל חלוקה שכזו: אימנו את המסווג לפי $\frac{k-1}{k}$ הדוגמאות הנותרות ובדקנו את התוצאות שהתקבלנו על קבוצת המבחן בגודל $\frac{1}{k}$ כאשר הבדיקה נעשתה באמצעות פונקציית score של מסווג ה-KNN שמסופק בספריה sklearn.

עבור כל k חישבנו את שגיאת המבחן הממוצעת ובנוסף אליה חישבנו את שגיאת המבחן הממוצעת הממוצעת (על טווח ערכי k).

לשם יצירת קובץ התיוגים אימנו לבסוף את המסווג שלנו על סמך כל 100 הדוגמאות המתויגות שסופקו עם התרגיל.

נספח - הסבר על הקבצים המוגשים:

הסבר על הקובץ	שם הקובץ
קובץ הדוגמאות שלנו לאחר סינון הדוגמאות עם אלגוריתם sfs	updated_features.data
קובץ הבדיקה עבור KNN כפי שנדרש בתרגיל.	KNN_test.py
המסווג עבור התחרות (חלק ג').	CompeteitionClassifier.py
קבצי הdata כפי שדרוש בתרגיל.	ecg_fold_1.data
קבצי הdata כפי שדרוש בתרגיל.	ecg_fold_2.data
הדוגמאות שלנו לאחר הרצת המסווג של התחרות.	updated_features.data
קובץ results לתחרות כפי שדרוש בתרגיל. אלו הסיווגים משלנו לדוגמאות המבחן.	results.data
תוצאות עבור שאלה 5 חלק ב'.	experiments6.csv
תוצאות עבור שאלה 6 חלק ב'.	experiments12.csv
הטסטים שמריצים את experiment 1 ו2 בחלק ב'	Additional_tests.py
קובץ אשר בו מימשנו את אלגוריתם SFS לצורך המסווג של התחרות. בנוסף מימשנו בקובץ זה את המתודה feature_selection אשר משתמשת באלגוריתם SFS.	our_hw3_utils.py
קובץ הרצה כפי שנדרש בתרגיל. מריץ את הטסטים ואת המסווג לתחרות	main.py
הפונקציות ומימוש ה-KNN כפי שנדרש בחלק ב'	classifier.py
דרישות הגשה	readme.txt
דרישות הגשה	requirements.txt