

תרגיל בית 3 – מבוא ללמידה

עברו על כלל ההנחיות לפני תחילת התרגיל.

הנחיות כלליות:

- תאריך ההגשה: עד לסוף מועדי א' - 17/05/2024 ב 23:59
- את המטלה יש להגיש **בזוגות בלבד**.
- יש להגיש מטלות מוקלדות בלבד. פתרונות בכתב יד לא ייבדקו.
- ניתן לשלוח שאלות בנוגע לתרגיל בפיאצה בלבד.
- המתרגל האחראי על תרגיל זה: **דניאל אלגריסי**.
- בקשות דחיה מוצדקות (מילואים, אשפוז וכו') יש לשלוח למתרגל האחראי (**ספיר טובול**) בלבד.
- במהלך התרגיל ייתכן שנעלה עדכונים, למסמך הנ"ל – תפורסם הודעה בהתאם.
- העדכונים הינם מחייבים, ועליכם להתעדכן עד מועד הגשת התרגיל.
- שימו לב, העתקות טטופלנה בחומרה.
- התשובות לסעיפים בהם מופיע הסימון 🍷 צריכים להופיע בדוח.
- לחלק הרטוב מסופק שלד של הקוד.
- אנחנו קשובים לפניות שלכם במהלך התרגיל ומעדכנים את המסמך הזה בהתאם. גרסאות עדכניות של המסמך יועלו לאתר. **הבהרות ועדכונים שנוספים אחרי הפרסום הראשוני יסומנו כאן בצהוב**. ייתכן שתפורסמנה גרסאות רבות – אל תיבהלו מכך. השינויים בכל גרסה יכולים להיות קטנים.

לצורך הנוחות:

הבהרות ועדכונים גרסה ראשונה סומנו ככה.
הבהרות ועדכונים גרסה שניה סומנו ככה.
הבהרות ועדכונים גרסה שלישית סומנו ככה.

שימו לב שאתם משתמשים רק בספריות הפייתון המאושרות בתרגיל (מצוינות בתחילת כל חלק רטוב)
לא יתקבל קוד עם ספריות נוספות

מומלץ לחזור על שקפי ההרצאות והתרגולים הרלוונטיים לפני תחילת העבודה על התרגיל.

חלק ב' - מבוא ללמידה (56 נק')

👉 חלק א' – חלק היבש (28 נק')

kNN – נעים להכיר

בחלק זה תכירו אלגוריתם למידה בשם kNN, או בשמו המלא k-Nearest Neighbors, כאשר ה-k הוא למעשה פרמטר!

יהי סט אימון עם n דוגמות, $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, כאשר $\forall i: x_i \in \mathbb{R}^d, y_i \in \mathcal{Y}$. כלומר הדוגמות הינן וקטורים d -ממדיים והתגיות הינן מדומיין כלשהו, הבעיה היא בעיית קלסיפיקציה (סיווג). אם לא נאמר אחרת, הקלסיפיקציה תהיה בינארית, כלומר $\mathcal{Y} = \{-, +\}$. עבור כל דוגמה בסט האימון, ניתן להסתכל על הכניסה ה- i בווקטור כעל ה- i feature של הדוגמה, קרי כל דוגמה x_i מיוצגת על ידי d -ערכים: $f_1(x_i), f_2(x_i), \dots, f_d(x_i)$. תהליך ה"אימון" של האלגוריתם הוא טריוויאלי – פשוט שומרים את סט האימון במלואו. תהליך הסיווג הוא גם פשוט למדי – כאשר רוצים לסווג דוגמה מסט המבחן מסתכלים על k השכנים הקרובים ביותר שלה במישור ה- d ממדי מבין הדוגמות בסט האימון, ומסווגים את הדוגמה על פי הסיווג הנפוץ ביותר בקרב k השכנים.

על מנת להימנע משוויון בין הסיווגים, נביח בדרך כלל כי k אי זוגי, או שנגדיר היטב שובר שוויון. אם לא נאמר אחרת, במקרה של שוויון בקלסיפיקציה בינארית, נסווג את הדוגמה כחיובית +.

שאלות הבנה

א. (3 נק') כאמור, בתהליך הסיווג אנו בוחרים עבור הדוגמה את הסיווג הנפוץ ביותר של k השכנים הקרובים ביותר, אולם עלינו להגדיר את פונקציית המרחק עבור קביעת סט שכנים זה. שתי פונקציות מרחק נפוצות הינן מרחק אוקלידי ומרחק מנהטן.

1) עבור איזה ערכים של d, k נקבל שאין תלות בבחירה בין פונקציות המרחק הנתונות בבחירה פונקציית המרחק? (נמקו)

-For both Euclidean and Manhattan distances, and for every k , when $d = 1$, the distance calculation reduces to the absolute difference between two points. In this case, in both distance functions, the distance calculation becomes the same.

2) עבור בעיית קלסיפיקציה בינארית תנו דוגמה פשוטה לערכי d, k , סט אימון ודוגמת מבחן בה השימוש בכל אחת מפונקציות המרחק הנ"ל משנה את סיווג דוגמת המבחן.

$$D = \{((5,1), +), ((4,3), -)\}$$

$$d = 2$$

$$k = 1$$

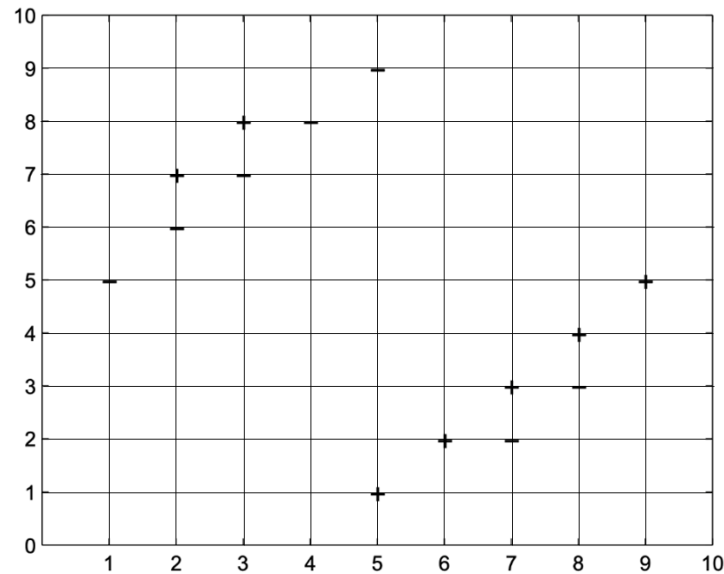
$$\text{test_set} = \{((0,0),+)\}$$

-(0,0) is closest (4,3) in Euclidian distance, so its classified as -. (Wrongly)

-(0,0) is closest (5,1) in Manhattan distance, so it's classified as +.

מעתה, אלא אם כן צוין אחרת, נשתמש במרחק אוקלידי.

נתונה קבוצת האימון הבאה, כאשר $d = 2$:



(3) (1 נק') איזה ערך של k עלינו לבחור על מנת לקבל את הדיוק המרבי על קבוצת האימון? מה יהיה ערך זה? (הדוגמא לא יכולה להיות שכנה של עצמה)

Since a sample can't be its own neighbor $k=1$ doesn't work well as we get only 4 right classifications, $k=2$ is problematic due to being even, $k=3$ will yield 8 right classifications, $k=5$ yields 10 right classifications and is the best we can achieve here.

(4) (1 נק') עבור איזה ערך של k נקבל מסווג *majority* של קבוצת האימון? קרי כל דוגמת מבחן תקבל את הסיווג הנפוץ של כלל קבוצת האימון?

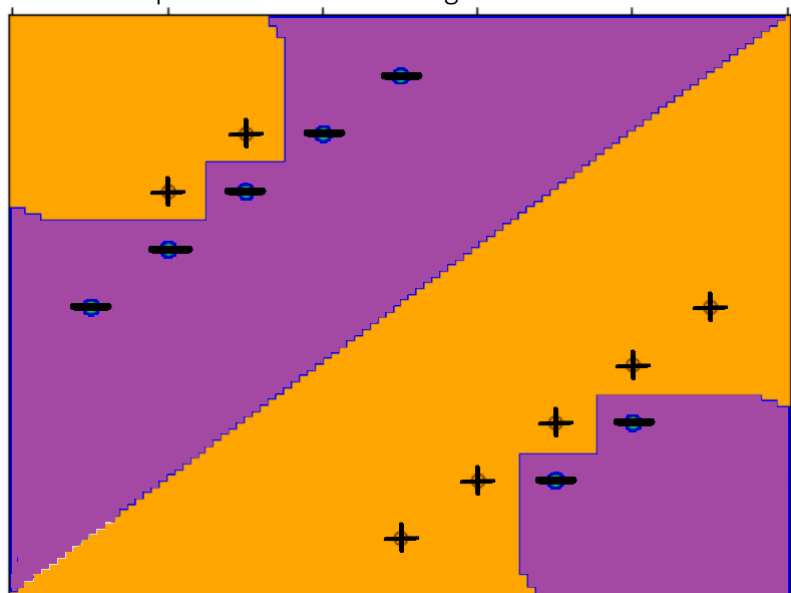
For $k \geq 13$ we get all the samples to join in on the classifications process in so making it a majority classifier, with same example not being its own neighbor, if not, then $K \geq 14$.

(5) (2 נק') נמקו מדוע שימוש בערכי k גדולים או קטנים מדי יכול להיות גרוע עבור קבוצת הדגימות הנ"ל.

-For large k values, our classifier will make a lot of wrong choices as it chooses the majority and generalizes the model too much as it takes into account far neighbours.

-For small k values, the model is more sensitive to noise, in the way our training data is distributed, and struggles to generalize.

(6) (2 נק') שרטט את גבול ההחלטה של 1-nearest neighbour עבור הגרף.



השוואה בין מודלי למידה – יש לנמק בקצרה את הפתרונות

1) (3 נק') הציגו מסווג מטרה $f(x): R^2 \rightarrow \{0,1\}$ וקבוצת אימון בעלת לכל היותר 10 דוגמאות כך שלמידת עץ ID3 תניב מסווג אשר עונה נכון עבור כל דוגמת מבחן אפשרית (כלומר יתקבל מסווג המטרה), אך למידת KNN תניב מסווג שעבורו קיימת לפחות דוגמת מבחן אחת עליה הוא יטעה, לכל ערך K שייבחר.

We define $f(x_1, x_2) = I\{x_1, x_2 > 0\}$, in words f is an indicator for every point in the upper right quarter not including the axis, otherwise returns 0.

Training_Set = $\{(-1,1,0), (1, -1,0), (-1, -1,0), (1,1,1)\}$

Here ID3 will check if either x_1 or x_2 is less or equal to 0, if so, it returns 0 otherwise 1, making it optimal for our classifier, while KNN, for every K possible, with this training set, will get the point (1,1) for example classified wrong.

2) (3 נק') הציגו מסווג מטרה $f(x): R^2 \rightarrow \{0,1\}$ וקבוצת אימון בעלת לכל היותר 10 דוגמאות כך שלמידת מסווג KNN עבור ערך K מסוים תניב מסווג אשר עונה נכון עבור כל דוגמת מבחן אפשרית (כלומר יתקבל מסווג המטרה), אך למידת עץ ID3 תניב מסווג אשר עבורו קיימת לפחות דוגמת מבחן אפשרית אחת עליה הוא יטעה.

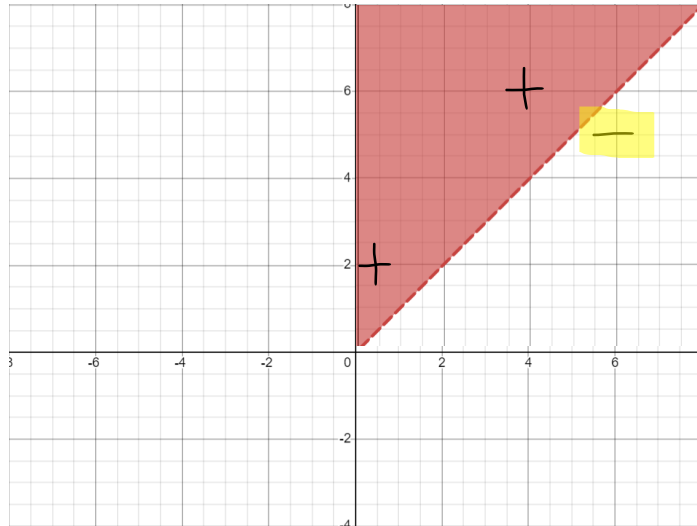
We define $f(x_1, x_2) = I\{x_1 - x_2 > 0\}$, in words f is an indicator for every point above the $y=x$ line ($x_1=x_2$), otherwise returns 0.

Training_Set = $\{(-1,1,0), (1, -1,1)\}$

Here KNN with $K=1$, and equality breaker that chooses (0) in case distances are equal, for every test point will get the right classification, as in closest neighbour to the training data, while ID3 will try to create nodes based on a single feature at a time, and in turn use the less-or-equal-to-0 that is explained above for both features, and classify for example $(-1,-0.5)$ as 0, wrongly.

(3 נק') הציגו מסווג מטרה $f(x): R^2 \rightarrow \{0,1\}$ וקבוצת אימון בעלת לכל היותר 10 דוגמאות כך שלמידת מסווג KNN עבור ערך K מסוים תניב מסווג אשר עבורו קיימת לפחות דוגמת מבחן אפשרית אחת עליה הוא יטעה, וגם למידת עץ ID3 תניב מסווג אשר עבורו קיימת לפחות דוגמת מבחן אחת אפשרית עליה הוא יטעה.

In this question we combine 1) and 2), meaning our target function will be something like this:



If we test the point (6,5), with KNN and $k=1$, we get a false classification (+), and with ID3 and its lack of expressiveness in situations like these where there is a dependency between features, we also get a false classification (+).

(4 נק') הציגו מסווג מטרה $f(x): R^2 \rightarrow \{0,1\}$ וקבוצת אימון בעלת לכל היותר 10 דוגמאות כך שלמידת מסווג KNN עבור ערך K מסוים תניב מסווג אשר עונה נכון עבור כל דוגמת מבחן אפשרית (כלומר יתקבל מסווג המטרה), וגם למידת עץ ID3 תניב מסווג עונה נכון עבור כל דוגמת מבחן אפשרית (כלומר יתקבל מסווג המטרה).

Here we can go for the trivial classifier $f(x_1, x_2) = 1$

Training_Set = $\{(-1,1,1), (1,-1,1), (-1,-1,1), (1,1,1)\}$

KNN model with $K = 1$, will always classify right (1), and for ID3 for every feature and with that training set It will also choose 1 every time, thus achieving the desired classifier.

מתפצלים ונהנים

(7 נק') כידוע, בעת סיווג של דוגמת מבחן על ידי עץ החלטה, בכל צומת בעץ אנו מחליטים לאיזה צומת בן להעביר את דוגמת המבחן על ידי ערך סף ϵ שמושווה לfeature של הדוגמה. לפעמים ערך הסף קרוב מאוד לערך feature של דוגמת המבחן. היינו רוצים להתחשב בערכים "קרובים" לערך הסף בעת סיווג דוגמת מבחן, ולא לחרוץ את גורלה של הדוגמה לתת-עץ אחד בלבד; לצורך כך נציג את האלגוריתם הבא:

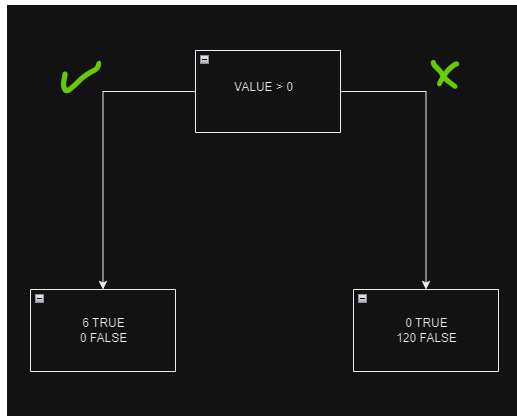
יהיו עץ החלטה T , דוגמת מבחן $x \in \mathbb{R}^d$, ווקטור $\epsilon \in \mathbb{R}^d$ המקיים $\forall i \in [1, d]: \epsilon_i > 0$. כלל אפסילון-החלטה שונה מכלל ההחלטה הרגיל שנלמד בכיתה באופן הבא: נניח שמגיעים לצומת בעץ המפצל לפי ערכי התכונה i , עם ערך הסף ϵ_i . אם מתקיים $|x_i - v_i| \leq \epsilon_i$ אזי ממשיכים **בשני** המסלולים היוצאים מצומת זה, ואחרת ממשיכי לבן המתאים בדומה לכלל ההחלטה הרגיל. לבסוף, מסווגים את הדוגמה x בהתאם לסיווג הנפוץ ביותר של הדוגמאות הנמצאות בכל העלים אליהם הגענו במהלך הסיווג על העץ (במקרה של שוויון – הסיווג ייקבע להיות **True**).

יהא T עץ החלטה לא גזום, ויהא T' העץ המתקבל מ- T באמצעות גיזום מאוחר שבו הוסרה הרמה התחתונה של T (כלומר כל הדוגמות השייכות לזוג עלים אחים הועברו לצומת האב שלהם). הוכיחו/הפריכו: **בהכרח** קיים ווקטור ϵ כך שהעץ T עם כלל אפסילון-החלטה והעץ T' עם כלל ההחלטה הרגיל יסווגו כל דוגמת מבחן ב- \mathbb{R}^d בצורה זהה.

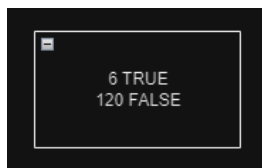
Disproof:

For the given, $d=1$, $\text{test_sample} = 2 * \epsilon$,

T :



T' :



T' , Is T after post pruning, T' always yields false including our test sample.

For every vector ϵ and our tree T we get true classification since our test sample is always larger than threshold and the epsilon threshold.

חלק ב' - היכרות עם הקוד

רקע

חלק זה הוא רק עבור היכרות הקוד, עבורו עליו במלואו ווודאו כי הינכם מבינים את הקוד. בחלק של הלמידה, נעזר ב *dataset*, הדאטה חולק עבורכם לשתי קבוצות: קבוצת אימון *train.csv* וקבוצת מבחן *test.csv*. ככלל, קבוצת האימון תשמש אותנו לבניית המסווגים, וקבוצת המבחן תשמש להערכת ביצועיהם.

בקובץ *utils.py* תוכלו למצוא את הפונקציות הבאות לשימושכם:
`load_data_set, create_train_validation_split, get_dataset_split`
אשר טוענות/מחלקות את הדאטה בקבצי ה-*csv* למערכי *np.array* (קראו את תיעוד הפונקציות).

הדאטה של ID3 עבור התרגיל מכיל מדדים שנאספו מצילומים שנועדו להבחין בין גידול שפיר לגידול ממאיר. כל דוגמה מכילה 30 מדדים באלה, ותווית בינארית **diagnosis** הקובעת את סוג הגידול (0=שפיר, 1=ממאיר). כל התכונות (מדדים) רציפות. העמודה הראשונה מציינת האם האדם חולה (M) או בריא (B). שאר העמודות מציינות כל תכונות רפואיות שונות של אותו אדם (התכונות מורכבות ואינכם צריכים להתייחס למשמעות שלהן כלל).

תיקיית *dataset – ID3*:

- תיקיה זו אלו מכילה את קבצי הנתונים עבור *ID3*.

קובץ *utils.py*:

- קובץ זה מכיל פונקציות עזר שימושיות לאורך התרגיל, כמו טעינה של *dataset* וחישוב הדיוק.
- בחלק הבא יהיה עליכם לממש את הפונקציה *accuracy*. קראו את תיעוד הפונקציות ואת ההערות הנמצאות תחת התיאור *TODO*.

קובץ *unit test.py*:

- קובץ בדיקה בסיסי שיכול לעזור לכם לבדוק את המימוש.

קובץ *DecisionTree.py*:

- קובץ זה מכיל 3 מחלקות שימושיות לבניית עץ *ID3* שלנו.
 - המחלקה *Question*: מחלקה זו מממשת הסתעפות של צומת בעץ. היא שומרת את התכונה ואת הערך שלפיהם מפצלים את הדאטה שלנו.
 - המחלקה *DecisionNode*: מחלקה זו מממשת צומת בעץ ההחלטה. הצומת מכיל שאלה *Question* ואת שני הבנים *true_branch, false_branch* כאשר *true_branch* הוא הענף בחלק של הדאטה שעונה *True* על שאלת הצומת (הפונקציה *match* של ה-*Question* מחזירה *True*).
ו-*false_branch* הוא הענף בחלק של הדאטה שעונה *False* על שאלת הצומת (הפונקציה *match* של ה-*Question* מחזירה *False*).
 - המחלקה *Leaf*: מחלקה זו מממשת צומת שהוא עלה בעץ ההחלטה. העלה מכיל לכל אחד מהמחלקות בדאטה את מספר הדוגמאות בעלה עבור כל מחלקה (למשל: {*B*: 5, *M*: 6}).

קובץ *ID3.py*:

- קובץ זה מכיל את המחלקה של *ID3* שתצטרכו לממש חלקים ממנה, עיינו בהערות ותיעוד המתודות.

קובץ *ID3 experiments.py*:

- קובץ הרצת הניסויים של ID3, הקובץ מכיל את הניסויים הבאים, שיוסברו בהמשך:
cross_validation_experiment, basic_experiment

חלק ג' – חלק רטוב ID3 (28 נק')

עבור חלק זה מותר לכם להשתמש בספריות הבאות:

All the built in packages in python, sklearn, pandas, numpy, random, matplotlib, argparse, abc, typing.

אך כמובן שאין להשתמש באלגוריתמי הלמידה, או בכל אלגוריתם או מבנה נתונים אחר המהווה חלק מאלגוריתם למידה אותו תתבקשו לממש.

1. (3 נק') השלימו את הקובץ `utils.py` ע"י מימוש הפונקציה

`.accuracy`.

קראו את תיעוד הפונקציה ואת ההערות הנמצאות תחת התיאור
TODO.

(הריצו את הטסטים המתאימים בקובץ `unit_test.py` לוודא
שהמימוש שלכם נכון).

שימו לב! בתיעוד ישנן הגבלות על הקוד עצמו, אי-עמידה
בהגבלות אלו תגרור הורדת נקודות.

בנוסף, שנו את ערך ה-ID בתחילת הקובץ מ-123456789
למספר תעודת הזהות של אחד מהמגשים.

2. (10 נק') אלגוריתם ID3:

a. השלימו את הקובץ `ID3.py` ובכך ממשו את אלגוריתם ID3 כפי שנלמד בהרצאה. TODO

שימו לב שכל התכונות רציפות. אתם מתבקשים להשתמש בשיטה של חלוקה דינמית המתוארת בהרצאה.
כאשר בוחנים ערך סף לפיצול של תכונה רציפה, דוגמאות עם ערך השווה לערך הסף משתייכות לקבוצה עם
הערכים הגדולים מערך הסף. במקרה שיש כמה תכונות אופטימליות בצומת מסוים בחרו את התכונה בעלת
האינדקס המקסימלי.

כלל המימוש הנ"ל צריך להופיע בקובץ בשם `ID3.py`, באזורים המוקצים לכך.

(השלימו את הקוד החסר אחרי שעיינתם והפנמתם את הקובץ `DecisionTree.py` ואת המחלקות שהוא מכיל).

b. ממשו את `basic_experiment` שנמצאת ב `ID3_experiments.py` TODO

והריצו את החלק המתאים ב `main` ציינו בדו"ח את הדיוק שקיבלתם. 🍌

Test Accuracy: 99.03%

גיזום מוקדם.

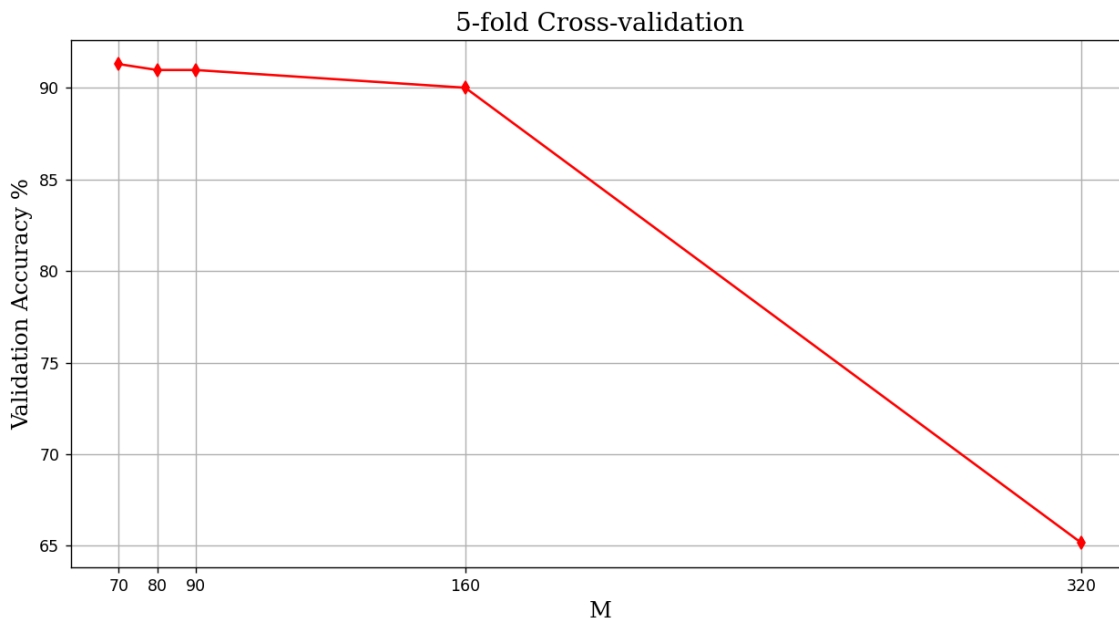
פיצול צומת מתקיים כל עוד יש בו יותר דוגמאות מחסם המינימום m , כלומר בתהליך בניית העץ מבוצע
"גיזום מוקדם" כפי שלמדתם בהרצאות. שימו לב כי פירוש הדבר הינו שהעצים הנלמדים אינם בהכרח
עקביים עם הדוגמאות. לאחר סיום הלמידה (של עץ יחיד), הסיווג של אובייקט חדש באמצעות העץ שנלמד
מתבצע לפי רוב הדוגמאות בעלה המתאים.

c. (2 נק') 🍌 הסבירו מה החשיבות של הגיזום באופן כללי ואיזה תופעה הוא מנסה למנוע?

Pruning of decision trees is done to decrease the tree size meaning less time complexity for classification and testing, and also mainly done to decrease overfitting on the training set, which in turn will increase the training error in the hope of decreasing the testing error and increasing generality.


d. (3 נק') עדכנו את המימוש בקובץ `ID3.py` כך שיבצע גיזום מוקדם כפי שהוגדר בהרצאה.
הפרמטר `min_for_pruning` מציין את המספר המינימלי בעלה לקבלת החלטה, קרי יבוצע גיזום מוקדם אם
ורק אם מספר הדוגמות בצומת קטן שווה לפרמטר הנ"ל. `TODO`

e. (8 נק') שימו לב, זהו סעיף יבש ואין צורך להגיש את הקוד שכתבתם עבורו.
בצעו כיוון לפרמטר `M` על קבוצת האימון:
1. בחרו לפחות חמישה ערכים שונים לפרמטר `M`.
2. עבור כל ערך, חשבו את הדיוק של האלגוריתם על ידי `K – fold cross validation` על קבוצת
האימון בלבד.
כדי לבצע את חלוקת קבוצת האימון ל-`K` קבוצות יש להשתמש בפונקציה
`sklearn.model_selection.KFold` עם הפרמטרים `shuffle = True, n_split = 5`
ו-`random_state` אשר שווה למספר תעודת הזהות של אחד מהשותפים.
i. השתמשו בתוצאות שקיבלתם כדי ליצור גרף המציג את השפעת הפרמטר `M` על הדיוק.
צרפו את הגרף בדו"ח. (לשימושכם הפונקציה `util_plot_graph` בתוך הקובץ `utils.py`.)



ii. הסבירו את הגרף שקיבלתם. לאיזה גיזום קיבלתם התוצאה הטובה ביותר ומהי תוצאה זו? 

The best accuracy was with pruning '70 and the accuracy were 91.29% and up until pruning 160 the accuracy is barely decreasing, this means that there are a couple of features that are very decisive in the classification process and can get us as high as 90% accuracy, (hence why we chose big m values), as for 320 we get majority classifier with accuracy 65.16% since there are less than 320 train samples after 5-fold cross validation.

f. (2 נק')  השתמשו באלגוריתם ID3 עם הגיזום המוקדם כדי ללמוד מסווג מתוך כל קבוצת האימון ולבצע חיזוי על קבוצת המבחן.

השתמשו בערך ה- M האופטימלי שמצאתם בסעיף c. (ממשו $best_m_test$ שנמצאת ב $ID3_experiments.py$ והריצו את החלק המתאים ב $main$). ציינו בדו"ח את הדיוק שקיבלתם. האם הגיזום שיפר את הביצועים ביחס להרצה ללא גיזום?

M value	Validation Accuracy
70	91.29%
80	90.97%
90	90.97%
160	90.00%
320	65.16%

=====

Best M	Validation Accuracy
70	91.29%

best_m = 70
Test Accuracy: 98.06%

The accuracy worsened actually by 1%, that is expected since we got high accuracy and pruning increases generality in hopes of getting better accuracy on the test set but not necessarily.

הוראות הגשה

- ✓ הגשת התרגיל תתבצע אלקטרונית בזוגות בלבד.
- ✓ הקוד שלכם ייבדק (גם) באופן אוטומטי ולכן יש להקפיד על הפורמט המבוקש. הגשה שלא עומדת בפורמט לא תיבדק (ציון 0).
- ✓ המצאת נתונים לצורך בניית הגרפים אסורה ומהווה עבירת משמעת.
- ✓ הקפידו על קוד קריא ומתועד. התשובות בדוח צריכות להופיע לפי הסדר.
- ✓ יש להגיש קובץ zip יחיד בשם `AI3_<id1>_<id2>.zip` (ללא סוגריים משולשים) שמכיל:
 - קובץ בשם `AI_HW3_LEARNING.PDF` המכיל את תשובותיכם לשאלות היבשות.
 - קבצי הקוד שנדרשתם לממש בתרגיל ואף קובץ אחר:
 - קובץ `utils.py`
 - בחלק של עצי החלטה – `ID3.py`, `ID3_experiments.py`

אין להכיל תיקיות בקובץ ההגשה, הגשה שלא עומדת בפורמט לא תיבדק.