

מבוא לבינה מלאכותית – תרגול 6

נושאים:

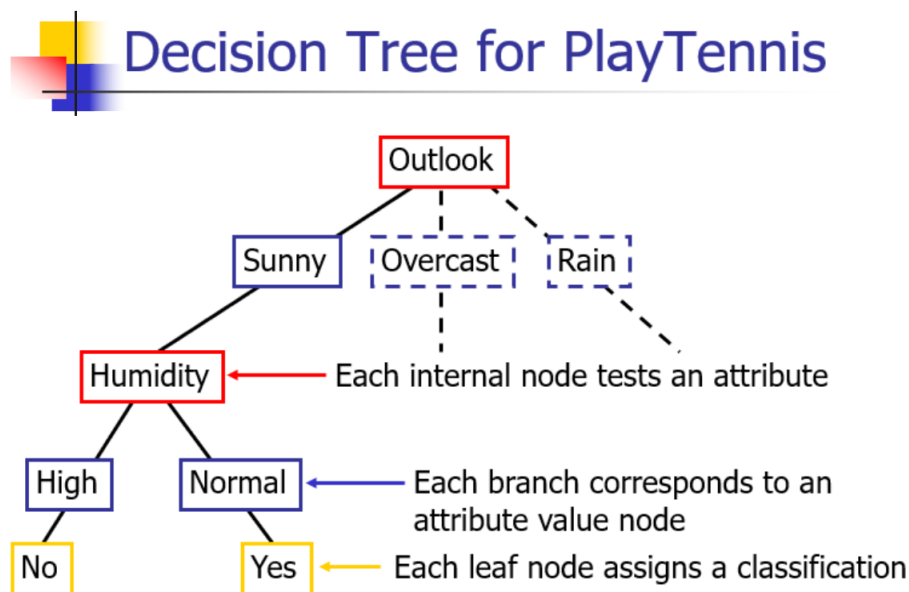
- עצי החלטה (Decision Trees)
- Random Forest

עצי החלטה – הרעיון הבסיסי:

מכירים את המשחק "נחש מי?"

עצי החלטה הם משהו כזה. מתוך אוסף הדאטא המותג שלנו, עליו אנחנו מחזיקים פיצ'רים, מנסים למצוא את התיוג של כל נקודה באמצעות שאלות על הפיצ'רים. בונים עץ, שבו בשורש יש לנו את כל הנקודות, בקדקודים פנימיים יש שאלות על פיצ'רים, מעבר על קשתות נעשה לפי ערכי הפיצ'רים והעלים של העץ ייתנו לנו סיווג.

דוגמה - נחליט האם לשחק היום טניס:



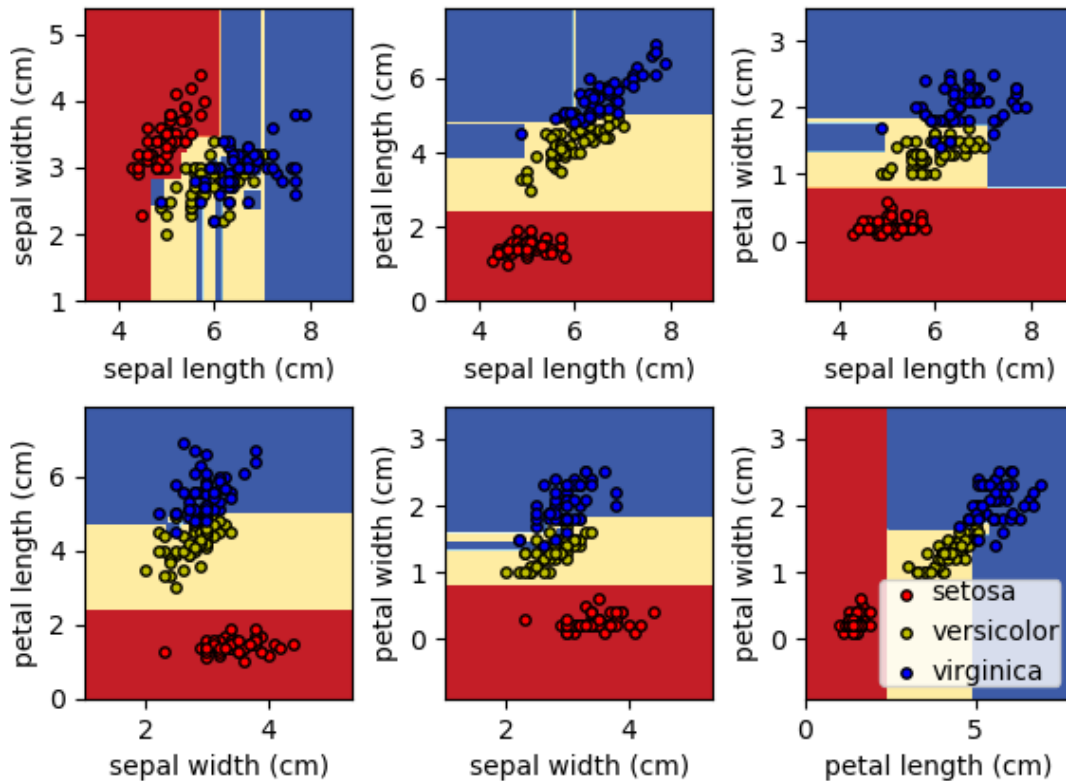
לעץ החלטה יש כל מיני יתרונות כמסווג (או ככלי לרגרסיה), בהם:

- קל להבין מה קורה בו, אפשר גם לראות זאת ממש ולהסיק מסקנות (כמו מה מוביל להחלטה לתיג נקודה באופן שתויגה).
- לא צריך לעשות הרבה פעולות להכנת הדאטא (למרות שאלגוריתם כזה לא מתמודד עם ערכי פיצ'רים חסרים, ולעתים כדאי להפעיל אלגוריתמים להורדת ממד מרחב הפיצ'רים).
- יכול לקבל פיצ'רים מסוגים שונים (רציף או בדיד, מספרי או קטגורי).
- עלות השימוש בעץ (=חיזוי הערך שניתן לנקודה) נמוכה. ליתר דיוק היא לוגריתמית במספר הנקודות שבקבוצת האימון.

כמובן שיש לו גם חסרונות:

- עצי החלטה נוטים לעשות overfitting לדאטא (ראו ציור הממחיש למה).

Decision surface of a decision tree using paired features



לכן, נעשה שימוש בשיטות להקטנת עומק העץ (הגבלה של העץ לגודל קבוע או החלטה שלא לחלק אם זה לא משתלם).

- עצי החלטה נוטים להיות לא יציבים – שינויים קטנים בדאטא עלולים להביא לשינויים גדולים בתוצאות שנותן העץ. האפקט הזה פחות חזק כשמשתמשים בעצים רבים להחלטה.
- מציאת עץ ההחלטה האופטימלי היא בעיה שפתרון מלא שלה הוא NP-complete. לכן, מימוש של בניית עצי החלטה נעשה באמצעות אלגוריתמים שלא בהכרח מבטיחי עץ אופטימלי. שוב, שימוש בעצים רבים עוזר לשיפור הלמידה.
- כאשר יש דאטא לא מאוזן מבחינת כמויות, חוסר האיזון מתבטא גם בתוצאות המודל וזה לא רצוי בהרבה מקרים.

איך בונים עצים כאלה – Entropy, Information Gain, Gini

כדי לבנות עץ החלטה, נצטרך לקבוע את סדר השאלות שנשאל על הפיצ'רים שלנו. נרצה שהן תהיינה יעילות ככל האפשר: בחזרה לדוגמת "נחש מי?", שאלה כמו "האם הדמות שלך היא אלעד?" לא תספק לנו הרבה מידע לגבי סיווג הדמות (הרי נישאר תקועים עם אלעד בענף אחד אבל כל שאר הדמויות בענף השני), בעוד ששאלה "האם הדמות שלך היא בן?" היא די טובה.

נלמד עכשיו מדדים לכמה "טובה" השאלה שאנחנו שואלים:

אנטרופיה:

נניח לשם נוחות שיש לנו שני תיוגים: +, -. האנטרופיה מודדת לנו את אי הודאות שיש לנו לגבי התיוג בקבוצה. היא מוגדרת ע"י:

$$H(S) = -p_+ \log(p_+) - p_- \log(p_-)$$

כאשר $p_{\pm}$ הם היחסים של התיוגים הרלוונטיים מתוך כלל הקבוצה. ביותר משני תיוגים אפשר לרשום סכום דומה.

נשים לב שככל שיש לנו בקבוצה S יותר מתיוג אחד יחסית לשאר, האנטרופיה תקטן וזה מה שנרצה.

Information Gain

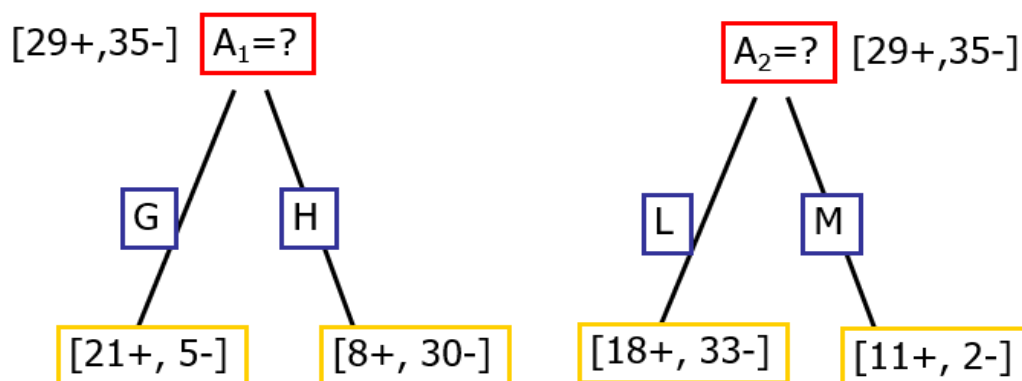
בעזרת האנטרופיה, משתמשים בממד הבא לקביעת הפיצ'ר שיהיה הקריטריון שלנו:

$$Gain(S, a) = H(S) - \sum_{v \text{ in children}} \frac{|S_v|}{|S|} H(S_v)$$

כלומר, ההפרש בין האנטרופיה של הקבוצה לפני החלוקה, לבין הסכום הממושקל (לפי יחסי הגדלים של הקבוצות החדשות) של האנטרופיות בקבוצות החדשות.

ככל שנקטין את האנטרופיה בקבוצות, כך ה-Gain יגדל. לכן, פיצ'ר שייתן לנו Gain גדול יהיה עדיף.

דוגמה:



$$\text{Entropy}([29+, 35-]) = -29/64 \log_2 29/64 - 35/64 \log_2 35/64 = 0.99$$

$$\text{Entropy}([21+, 5-]) = 0.71$$

$$\text{Entropy}([8+, 30-]) = 0.74$$

$$\text{Gain}(S, A_1) = \text{Entropy}(S)$$

$$-26/64 * \text{Entropy}([21+, 5-])$$

$$-38/64 * \text{Entropy}([8+, 30-])$$

$$= 0.27$$

$$\text{Entropy}([18+, 33-]) = 0.94$$

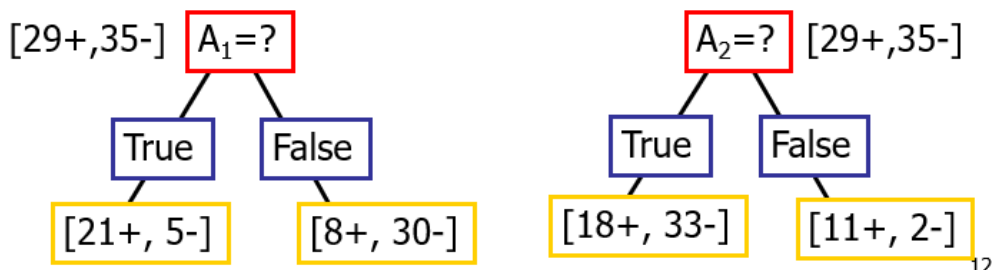
$$\text{Entropy}([11+, 2-]) = 0.62$$

$$\text{Gain}(S, A_2) = \text{Entropy}(S)$$

$$-51/64 * \text{Entropy}([18+, 33-])$$

$$-13/64 * \text{Entropy}([11+, 2-])$$

$$= 0.12$$



ג'יני:

זה מדד נוסף לכמה "טהור" הדאטא שאנחנו מחלקים. הפעם, עבור K קלאסים שונים, רושמים:

$$H(S) = \sum_{i=1}^K p_i(1 - p_i)$$

כאן, נרצה ערך נמוך ככל האפשר (אידיאלית, 1 מערך אחד היה נותן 0 בסכום). חלוקה שתיתן סכום ממושקל נמוך של ערכי ג'יני בקבוצות המחולקות תהיה מועדפת.

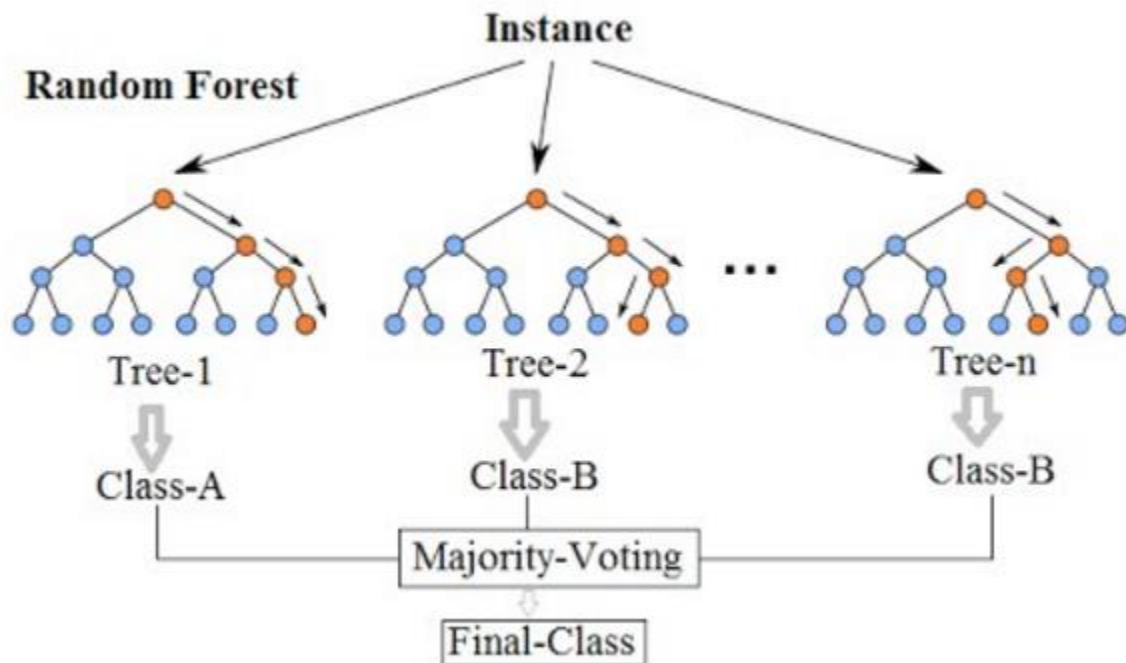
אלגוריתמים לעצי החלטה – קצת על ID3, C4.5, C5 ועל Random Forest:

אלגוריתם ID3 (Quinlan, 1986) בונה עץ החלטה כך שבכל קדקוד בוחרים להשתמש בפיצ'ר שנותן הכי הרבה Gain לחלק הדאטא שנמצא בקדקוד (כלומר, זה אלגוריתם חמדן). כאן, בונים את העץ במלואו.

אלגוריתם C4.5 הוא שיפור של הקודם (Quinlan, 1993), בכך שהוא מאפשר גם להשתמש בפיצ'רים רציפים או מספריים (האלגוריתם מוצא ערך סף אופטימלי לפיצ'ר כזה. למשל, לפיצ'ר של גובה האלגוריתם ימצא שהגובה המפריד יכול להיות 1.80, ויפריד לפי מעל לגובה זה ומתחתיו). הוא גם מאפשר להשתמש בפיצ'רים חסרים (פשוט מתעלמים מהם בחישוב ה-Gain). C4.5 בונה עץ מלא ואחר כך מבצע "גיזום" (במידה ואכן שווה לעשות זאת, כלומר בכל מקרה אין Gain משמעותי בקדקוד) כדי לטפל מעט ב-overfitting.

אלגוריתם C5.0 (Quinlan, 1996) משפר את C4.5. ההבדלים מקודמו הם בין היתר: שיפור סיבוכיות זיכרון וזמן ריצה, ייצור עצים קטנים יותר, הכנסת אפשרות למשקל סוגים שונים של אי התאמה (למשל, חשוב יותר שלא לטעות ולסווג אדם חולה בתור בריא מאשר להפך) וגם אפשרות לבצע boosting (נלמד בתרגול הבא. זה שילוב של כמה מסווגים שונים יחד).

:Random Forest



באלגוריתם זה, אנחנו בונים "יער" (אוסף של עצים). הוא יהיה מורכב מהרבה עצי החלטה שונים, כאשר כל אחד מהעצים יהיה מבוסס על חלק מהפיצ'רים. לעתים, כל עץ גם יאמן רק על חלק מקבוצת האימון, או ייקח פיצ'רים מוסטים במעט מהערכים המקוריים שלהם. הסיווג לבסוף יתבצע לפי החלטת רוב העצים או מדדים דומים.

המטרה באלגוריתם הזה לעומת עץ החלטה יחיד היא (בעזרת ריבוי העצים ויצירת השונות בפיצ'רים) מניעת overfitting ושיפור היציבות.

כמה הבדלים עיקריים מעץ החלטה רגיל ויחיד:

- מתוך עץ החלטה יחיד, ניתן להבין מהם השיקולים לסיווג דוגמה באופן שבו היא בוצעה (למשל: לא נלך לשחק טניס כי יש גשם, או כי אין פרטנר למשחק וגם המחבט שבור). לעומת זאת, בריבוי של עצים, אי אפשר לקבל דבר כזה באופן מוחלט. מה שכן אפשר לקבל הוא feature importance, כלומר כמה מכריע פיצ'ר מסוים להחלטה.
- כאמור, אפקט ה-overfitting נמנע (אבל לא תמיד באופן מוחלט). זאת בגלל ריבוי העצים והחלוקה לתת קבוצות של פיצ'רים, מה שגם יוצר עצים קטנים יותר.
- כמות עצים גדולה באלגוריתם random forest עלולה לגרום זמן חישוב ארוך בהרבה משיטות למידה אחרות, ובהן עץ החלטה יחיד.

- לא דיברנו על זה, אבל גם רגרסיה ניתן לבצע בעזרת עצים. רגרסיה נעשית קלה ויעילה יותר לביצוע בעזרת random forest מאשר עץ החלטה (אלגוריתם למשל שמאפשר רגרסיה בעזרת עץ החלטה הוא CART. זה אלגוריתם שבו גם נעשה שימוש בג'ני).