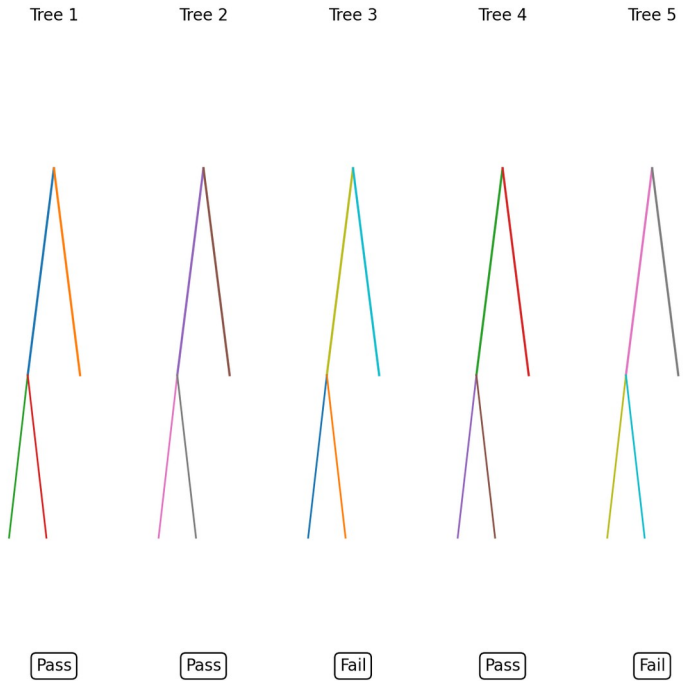


עצי החלטה ויערות אקראיים

מחברת לימוד עיונית עם דוגמאות ידניות, חישובים והסברים במעברים איטיים



Majority vote → Pass

המטרה של המחברת היא להבין לא רק כיצד משתמשים באלגוריתמים, אלא מה הם עושים בכל שלב, מדוע כל שלב נחוץ, ומה הבעיה שהוא פותר. בכל מעבר נשאל: מה כבר למדנו, מה עדיין חסר, ואיזו בעיה השלב הבא פותר.

תוכן עניינים

1. מהי בעיית סיווג ומה תפקידו של עץ החלטה
2. מבנה עץ: שורש, צומת, ענף ועלה
3. טוהר של צומת ומדד Gini
4. Entropy ו-Information Gain
5. כיצד העץ בוחר סף פיצול
6. בניית עץ שלם - דוגמה ידנית
7. מרחב התכונות והמשמעות הגיאומטרית של עץ
8. עצי רגרסיה
9. Variance, Bias ו-Overfitting
10. עצירה מוקדמת - Pre-Pruning
11. גיזום לאחר בנייה - Post-Pruning
12. Cost-Complexity Pruning
13. למה עץ יחיד אינו מספיק יציב
14. Bagging ו-Bootstrap
15. תוצאת 63.2% ו-Out-of-Bag
16. Random Feature Selection
17. Random Forest - האלגוריתם בשלבים
18. למה ממוצע של עצים מוריד שונות
19. Feature Importance
20. השוואה, יתרונות וחסרונות
21. תרגילים ידניים עם פתרונות

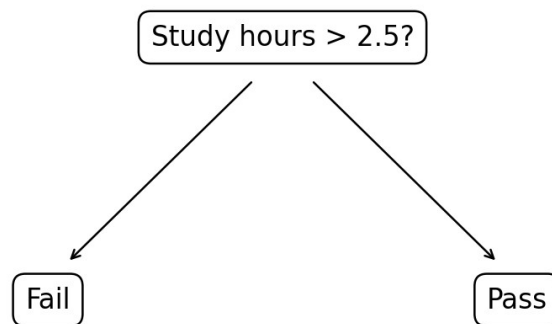
1. מהי בעיית סיווג ומה תפקידו של עץ החלטה

בלמידה מונחית אנו מקבלים אוסף דוגמאות. לכל דוגמה יש מאפיינים, ולכל דוגמה ידועה גם התשובה הנכונה. המטרה היא ללמוד מן הדוגמאות כלל שיוכל לתת תשובה גם עבור מקרה חדש שלא נראה בזמן האימון.

לדוגמה, נרצה לחזות האם סטודנט יעבור מבחן. לכל סטודנט נוכל למדוד שעות לימוד, נוכחות, מספר תרגילים שהוגשו וכדומה. המשתנה שאנו רוצים לחזות הוא עבר/נכשל.

סטודנט	שעות לימוד	נוכחות	תוצאה
1	1	נמוכה	Fail
2	2	נמוכה	Fail
3	3	גבוהה	Pass
4	4	גבוהה	Pass
5	5	נמוכה	Pass
6	6	גבוהה	Pass

עץ החלטה מנסה לפתור את הבעיה באמצעות סדרת שאלות פשוטות. כל שאלה מפצלת את הדוגמאות לשתי קבוצות. לאחר כמה שאלות אנו מגיעים לקבוצה קטנה והומוגנית יחסית, ועל פיה נקבעת התחזית.



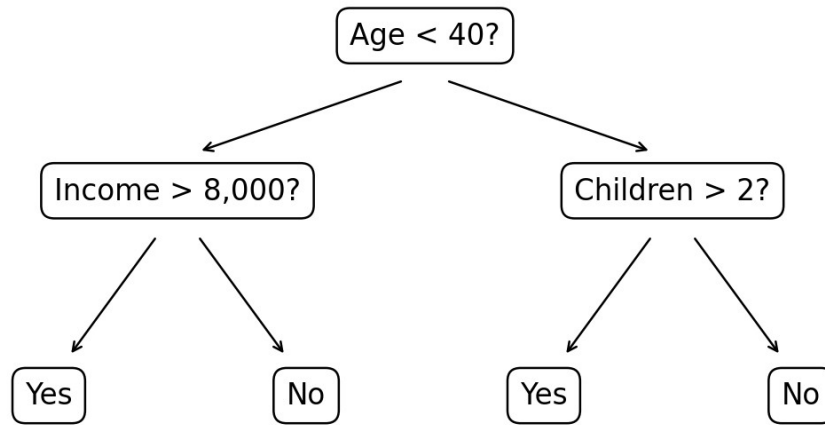
דוגמה לעץ בעל פיצול יחיד

המעבר לשלב הבא
אמרנו שהעץ שואל שאלות. אבל עדיין לא הסברנו כיצד הוא יודע איזו שאלה היא טובה. כדי לענות על כך עלינו למדוד עד כמה קבוצה היא "טהורה" או "מעורבת".

2. מבנה עץ: שורש, צומת, ענף ועלה

השורש הוא הצומת הראשון ובו נמצאות כל דוגמאות האימון. בכל צומת פנימי נשאלת שאלה. כל תשובה לשאלה מובילה לענף אחר. כאשר מפסיקים לפצל, מתקבל עלה - והוא נותן את התחזית.

- Root - שורש: נקודת ההתחלה של העץ.
- Internal Node - צומת פנימי: מכיל כלל פיצול, למשל גיל > 40 .
- Branch - ענף: תוצאת שאלה, למשל כן/לא.
- Leaf - עלה: סוף המסלול; ממנו מתקבלת התחזית.



חשוב לראות שהעץ אינו מנסה למצוא נוסחה אחת לכל הנתונים. הוא מחלק את הבעיה לאזורים שונים, ובכל אזור משתמש בכלל פשוט יותר.

3. טוהר של צומת ומדד Gini

נניח שבצומת יש עשר דוגמאות. אם כולן שייכות לאותה מחלקה, אין אי-הומוגניות: ברגע שדוגמה חדשה מגיעה לצומת הזה, התחזית ברורה. לעומת זאת, אם חמש דוגמאות חיוביות וחמש שליליות, הצומת מעורבב מאוד.

מדד Gini מודד את מידת אי-הטוהר:

$$Gini(D) = 1 - \sum p_k^2$$

כאשר p_k הוא החלק היחסי של המחלקה k בתוך הצומת. בסיווג בינארי, אם p הוא שיעור המחלקה החיובית, נקבל:

$$Gini = 1 - p^2 - (1-p)^2$$

אם כל הדוגמאות מאותה מחלקה, $p=1$ ולכן $Gini=0$. זהו צומת טוהר לחלוטין. אם החלוקה היא 50%-50%, נקבל $Gini=0.5$, שהוא הערך המקסימלי במקרה בינארי.

דוגמה מלאה: 6 עברו ו-4 נכשלו

$$Gini(parent) = 1 - (6/10)^2 - (4/10)^2 = 0.48$$

ננסה את השאלה "שעות לימוד < 3?". נניח שבצד שמאל יש 4 תלמידים: 1 עבר ו-3 נכשלו; ובצד ימין 6 תלמידים: 5 עברו ואחד נכשל.

$$Gini(L) = 1 - (1/4)^2 - (3/4)^2 = 0.375$$

$$Gini(R) = 1 - (5/6)^2 - (1/6)^2 \approx 0.278$$

אי-הטוהר לאחר הפיצול חייב להיות ממוצע משוקלל, משום שבשני הצדדים אין אותו מספר דוגמאות:

$$Gini(split) = (4/10) \cdot 0.375 + (6/10) \cdot 0.278 \approx 0.317$$

לפני הפיצול היה 0.48 ואחריו 0.317. לכן השיפור הוא:

$$\Delta Gini = 0.48 - 0.317 = 0.163$$

מה בדיוק העץ עושה?
העץ מנסה פיצולים אפשריים, מחשב לכל אחד את אי-הטוהר המשוקלל לאחר הפיצול, ומעדיף פיצול שמקטין את אי-הטוהר ככל האפשר.

4. Information Gain-Entropy

Gini אינו הממד היחיד. מדד מרכזי נוסף הוא Entropy, שמקורו בתורת האינפורמציה. הוא מודד את אי-הוודאות בהתפלגות המחלקות.

$$H(D) = - \sum p_k \log_2(p_k)$$

במקרה בינארי:

$$H(p) = -p \log_2(p) - (1-p) \log_2(1-p)$$

כאשר $p=1$ או $p=0$ אין אי-וודאות ולכן $H=0$. כאשר $p=1/2$ מתקבלת אי-וודאות מקסימלית: $H=1$.

Information Gain

גם כאן בוחנים מה קרה לפני הפיצול ומה קרה אחריו. השיפור נקרא Information Gain:

$$IG = H(\text{parent}) - [(nL/n)H(L) + (nR/n)H(R)]$$

כלומר: כמה אי-וודאות נעלמה בעקבות השאלה ששאלנו?

Gini או Entropy?

שני המדדים מבטאים רעיון דומה ולעיתים בוחרים פיצולים דומים. Gini מעט פשוט יותר חישובית; Entropy נותן פרשנות של מידע ואי-וודאות. מבחינת הבנת האלגוריתם, החשוב הוא רעיון הפחתת אי-הטוהר.

5. כיצד העץ בוחר סף פיצול

עד עכשיו הנחנו שהשאלה "שעות לימוד < 3 ?" כבר נתונה. אבל האלגוריתם צריך לבחור גם את המשתנה וגם את הסף.

נניח שערכי שעות הלימוד הם: 1, 2, 3, 4, 5. לאחר מיון, מועמדים טבעיים לסף הם נקודות האמצע בין ערכים עוקבים:

$$1.5, 2.5, 3.5, 4.5$$

עבור כל סף האלגוריתם יוצר שתי קבוצות, מחשב את Gini או Entropy המשוקלל, ומשווה בין האפשרויות.

דוגמה: בחירת סף מושלם

שעות לימוד	תוצאה
1	Fail
2	Fail
3	Pass
4	Pass
5	Pass

אם ננסה את הסף 2.5, בצד אחד נקבל שתי דוגמאות - שתיהן נכשלו; ובצד השני שלוש דוגמאות - כולן עברו. לכן שני הצדדים טהורים:

$$Gini(L)=0, Gini(R)=0 \Rightarrow Gini(\text{split})=0$$

זהו פיצול מושלם, ולכן אין צורך לבדוק פיצול עמוק יותר באותו צומת.

המעבר לשלב הבא

עד כאן למדנו לבחור פיצול יחיד. כעת נחבר את השלבים: לאחר הפיצול הראשון, מבצעים בדיוק אותו תהליך מחדש בתוך כל אחד מהצדדים. זהו האופי הרקורסיבי של העץ.

6. בניית עץ שלם - צעד אחר צעד

נניח שיש לנו שני מאפיינים: גיל והכנסה. האלגוריתם מתחיל בכל הנתונים, בודק פיצולים אפשריים בכל מאפיין ובוחר את הפיצול שמפחית ביותר את אי-הטוהר.

שלב 1: בחירת פיצול בשורש. נניח שהפיצול הטוב ביותר הוא גיל > 35 .

שלב 2: כעת מתייחסים לצד השמאלי כאל בעיה חדשה בפני עצמה. בודקים שוב פיצולים אפשריים רק בדוגמאות שהגיעו אליו.

שלב 3: עושים אותו הדבר בצד הימני.

שלב 4: ממשיכים עד שמתקיים תנאי עצירה - למשל צומת טהור, עומק מקסימלי או מספר קטן מדי של דוגמאות.

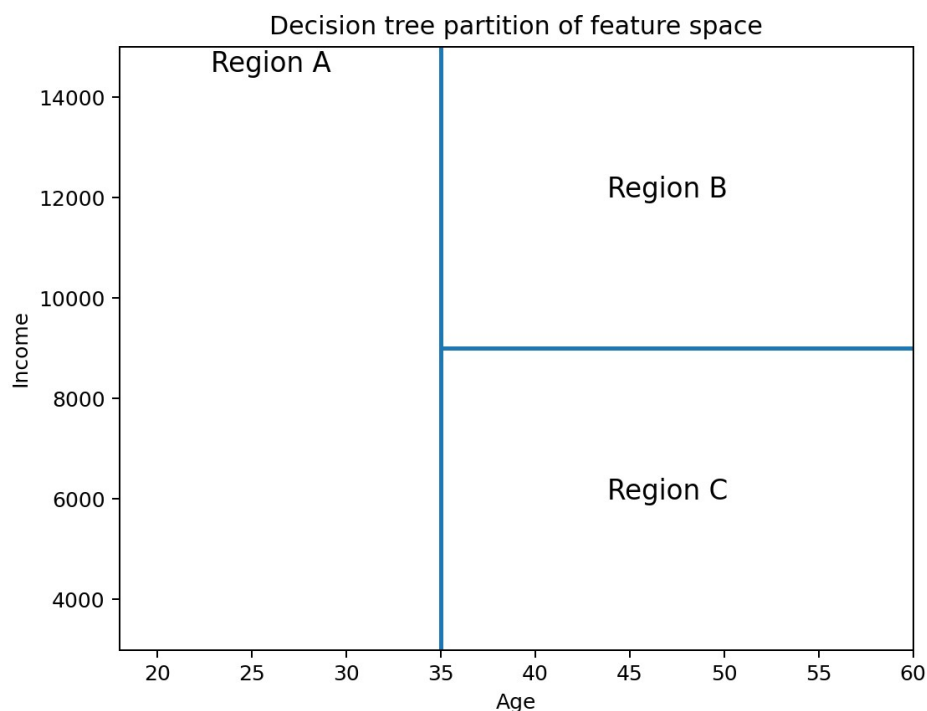
שלב 5: כל צומת שלא פוצל הופך לעלה ומקבל תחזית לפי המחלקה השכיחה בו.

עיקרון חשוב

עץ החלטה נבנה באופן חמדני (greedy): בכל צומת הוא בוחר את הפיצול הטוב ביותר כרגע. הוא אינו בוחן מראש את כל העצים האפשריים כדי למצוא עץ אופטימלי גלובלי.

7. המשמעות הגיאומטרית: העץ מחלק את מרחב התכונות

אם יש שני מאפיינים, כל פיצול מהצורה $x_j < c$ יוצר קו המקביל לאחד הצירים. פיצולים חוזרים מחלקים את המישור למלבנים. בכל מלבן התחזית קבועה לפי העלה המתאים.



לכן עץ מסוגל ליצור גבולות החלטה לא ליניאריים, אך הוא עושה זאת באמצעות סדרה של חיתוכים מקבילי-צירים. זו גם הסיבה לכך שהגבול שלו עשוי להיראות "מדרגות".

8. עצי רגרסיה

עד כה המשתנה החזוי היה מחלקה. ברגרסיה, המשתנה Y הוא מספר ממשי - למשל מחיר דירה. עקרון הפיצול נשאר דומה, אבל מדד האיכות משתנה.

שטח	חדרים	מחיר במיליונים
50	2	1.5
60	2	1.7
80	3	2.3
90	4	2.6
120	5	3.5

בעלה, התחזית הטבעית היא ממוצע ערכי Y של הדוגמאות שהגיעו אליו. למשל אם בעלה נמצאים 1.5 ו-1.7, התחזית היא 1.6.

כדי לבחור פיצול, נרצה שהערכים בכל צד יהיו דומים זה לזה. לכן משתמשים למשל בסכום ריבועי הסטיות:

$$SSE = \sum (y_i - \bar{y})^2$$

לכל פיצול מחשבים SSE בצד שמאל ובצד ימין, ובחרים את הפיצול שמקטין את הסכום הכולל.

9. Variance-ו Overfitting, Bias

כעת מתגלה הקושי המרכזי של עץ החלטה. אם נרשה לו להמשיך להתפצל ללא הגבלה, הוא יכול להגיע לעלים קטנים מאוד - אפילו עלה עם דוגמה אחת. במצב כזה הוא מסוגל כמעט לשנן את נתוני האימון. נניח שבנתונים יש אדם יחיד בן 37 עם התנהגות חריגה. עץ עמוק יכול ליצור סף כגון גיל > 37.5 כדי לבדד אותו. הפיצול מתאים לנתוני האימון, אך אין לנו סיבה לחשוב שהוא כלל אמיתי באוכלוסייה.

Variance-ו Bias

- עץ רדוד מאוד: פשוט מדי, ולכן עלול לסבול מ-High Bias ומ-Underfitting.
- עץ עמוק מאוד: מתאים היטב לנתוני האימון, אך רגיש לשינויים קטנים בנתונים - High Variance.

$$Variance \uparrow \Rightarrow Bias \downarrow \text{ כלל } Depth \uparrow$$

למה זה מוביל?

אם הבעיה היא שהעץ גדל יותר מדי, קיימות שתי אסטרטגיות טבעיות: לעצור אותו בזמן הגדילה (Pre-Pruning), או לתת לו לגדול ואז להסיר ענפים מיותרים (Post-Pruning).

10. עצירה מוקדמת - Pre-Pruning

Pre-Pruning פירושו להטיל מגבלות כבר בזמן בניית העץ. הרעיון הוא שלא כל פיצול שאפשר לבצע הוא פיצול שכדאי לבצע.

10.1 Maximum Depth - עומק מקסימלי

קובעים מספר מרבי של רמות מהשורש ועד לעלה. אם הגענו לעומק שנקבע, הצומת הופך לעלה גם אם ניתן עדיין להפחית את אי-הטוהר.

$$Maximum\ Depth = d$$

הגבלת עומק מורידה מורכבות ו-Variance, אך אם d קטן מדי נקבל Underfitting.

10.2 מינימום דוגמאות הדרוש לפיצול

נסמן את מספר הדוגמאות בצומת ב- n_split . נדרוש:

$$n_split \geq m$$

למשל, אם $m=10$, צומת בעל 7 דוגמאות אינו רשאי להתפצל. הסיבה: פיצולים על קבוצות זעירות נוטים להיות רגישים לרעש.

10.3 מינימום דוגמאות בעלה

כאן הדרישה היא לא על הצומת לפני הפיצול אלא על התוצאה שלו:

$$n_leaf \geq q$$

אם $q=5$, פיצול של 20 דוגמאות לקבוצות בגודל 19 ו-1 אסור. גם חלוקה ל-17 ו-3 אסורה. לעומת זאת, חלוקה ל-12 ו-8 מותרת.

הבדל חשוב

Minimum Samples Split שואל האם בכלל מותר לנסות לפצל את הצומת. Minimum Samples Leaf שואל האם שני הצדדים שנוצרו אחרי הפיצול גדולים מספיק.

10.4 שיפור מינימלי באי-הטוהר

נניח ש-Gini לפני הפיצול הוא 0.420 ואחריו 0.418. השיפור הוא רק 0.002. פיצול כזה מוסיף מורכבות כמעט בלי לשפר את ההפרדה.

$$\Delta Gini = Gini(parent) - Gini(split)$$

אפשר לדרוש $\Delta Gini \geq \epsilon$, למשל $\epsilon=0.01$. אם השיפור קטן מכך, לא מפצלים.

10.5 דוגמה מלאה לעצירה לפי שיפור

בצומת יש 10 דוגמאות: 6 מסוג A ו-4 מסוג B. לכן $Gini(parent)=0.48$. פיצול מסוים יוצר 8 דוגמאות בצד שמאל (5A,3B) ושתי דוגמאות בצד ימין (1A,1B).

$$Gini(L)=0.46875, \quad Gini(R)=0.5$$

$$Gini(split)=0.8 \cdot 0.46875 + 0.2 \cdot 0.5 = 0.475$$

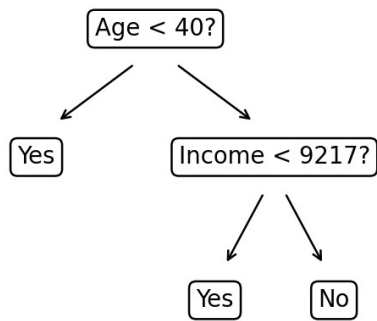
$$\Delta Gini = 0.48 - 0.475 = 0.005$$

אם דרשנו שיפור מינימלי של 0.01, הפיצול יידחה והצומת יהפוך לעלה.

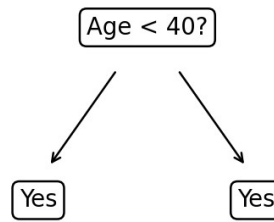
11. גיזום לאחר בנייה - Post-Pruning

Post-Pruning משתמש באסטרטגיה הפוכה: מאפשרים לעץ לגדול, ואז בודקים אילו תתי-עצים מוסיפים מורכבות בלי לשפר מספיק את ההכללה.

Before pruning



After pruning



החלפת תת-עץ שלם בעלה יחיד

נניח שעץ יצר כלל $\text{Income} < 9217$ על סמך מעט דוגמאות. המספר 9217 עשוי להתאים היטב לנתוני האימון אך להיות מקרי לחלוטין. בגיזום אפשר למחוק את הפיצול ולהחזיר תחזית יחידה עבור כל הקבוצה.

למה שגיאת האימון יכולה לעלות ובכל זאת המודל להשתפר?

עץ מלא יכול לקבל Training Error של 2%, בעוד עץ גזום יקבל 4%. אבל על נתוני בדיקה ייתכן שהעץ המלא יקבל 15% שגיאה והעץ הגזום 8%. כלומר ויתרנו מעט על התאמה לאימון כדי לקבל Generalization טוב יותר.

המעבר לשלב הבא

כדי להחליט באופן מסודר כמה לגזום, נרצה פונקציה שמענישה גם שגיאה וגם מורכבות. מכאן מגיע Cost-Complexity Pruning.

12. Cost-Complexity Pruning

נסמן ב- $R(T)$ את שגיאת העץ T וב- $|T|$ את מספר העלים. נגדיר פונקציית מטרה:

$$R\alpha(T) = R(T) + \alpha|T|$$

האיבר הראשון מבקש התאמה טובה לנתונים. האיבר השני מעניש עץ גדול. הפרמטר α קובע כמה אנחנו מוכנים לשלם על מורכבות.

דוגמה מספרית

עץ	שגיאה $R(T)$	מספר עלים $ T $
T_1	0.05	20
T_2	0.08	8

כאשר $\alpha=0$, נעדיף את T_1 כי שגיאתו קטנה יותר. כאשר $\alpha=0.005$:

$$R\alpha(T_1) = 0.05 + 0.005 \cdot 20 = 0.15$$

$$R\alpha(T_2) = 0.08 + 0.005 \cdot 8 = 0.12$$

כעת T_2 עדיף למרות ששגיאת האימון שלו גבוהה יותר, משום שהוא פשוט בהרבה.

$$\alpha \uparrow \Rightarrow \text{Tree Size} \downarrow$$

קשר ל-Regularization

הביטוי $|R(T) + \alpha|T$ הוא אותו עיקרון כללי שמופיע במודלים אחרים: Loss + Complexity Penalty. כלומר Pruning הוא צורה של Regularization לעצי החלטה.

13. למה עץ יחיד אינו מספיק יציב

גם לאחר הגבלת עומק או גיזום, לעץ יש חולשה מבנית: שינוי קטן בדאטה עלול לשנות את הפיצול הראשון, וכל הענפים שמתחתיו ישתנו בהתאם.

למשל, במדגם אחד הפיצול הראשון עשוי להיות $Age < 37.5$, ובמדגם כמעט זהה הפיצול הראשון עשוי להיות $Income < 8400$. כלומר העץ הוא מודל בעל Variance גבוה יחסית.

רעיון חדש

אם כל עץ בודד רגיש לדאטה, אולי במקום לנסות ליצור עץ אחד מושלם ניצור הרבה עצים מעט שונים, ונאגד את התחזיות שלהם. זהו הבסיס ל-Bagging.

14. Bagging ו-Bootstrap

Bagging הוא קיצור של Bootstrap Aggregating. הוא כולל שני שלבים: ליצור מדגמי Bootstrap שונים, לבנות מודל על כל מדגם, ואז לאגד את התחזיות.

נניח שהדאטה המקורי הוא:

$$D = \{A, B, C, D, E, F\}$$

מדגם Bootstrap בגודל 6 נוצר בדגימה עם החזרה. דוגמאות אפשריות:

$$D_1 = \{A, B, B, D, E, F\}$$

$$D_2 = \{A, A, C, D, D, F\}$$

$$D_3 = \{B, C, C, E, E, F\}$$

בגלל הדגימה עם החזרה, חלק מהדוגמאות מופיעות כמה פעמים וחלק אינן מופיעות כלל. כך כל עץ רואה גרסה מעט שונה של הבעיה.

למה עם החזרה?

המטרה אינה להקטין את מספר הדוגמאות, אלא ליצור שינויים אקראיים בהרכב המדגם תוך שמירה על אותו גודל מדגם. השונות בין המדגמים יוצרת שונות בין העצים.

15. תוצאת 63.2% ו-Out-of-Bag

תוצאה חשובה ב-Bootstrap: מדגם בגודל n אינו מכיל בדרך כלל את כל n הדוגמאות הייחודיות. נחשב מה הסיכוי שדוגמה מסוימת כלל לא תיבחר.

בבחירה אחת, ההסתברות שלא לבחור דוגמה מסוימת היא:

$$1 - 1/n$$

אנו מבצעים n בחירות בלתי תלויות עם החזרה, ולכן:

$$P(\text{not selected}) = (1 - 1/n)^n$$

כאשר n גדול:

$$(1 - 1/n)^n \rightarrow e^{-1} \approx 0.368$$

כלומר בערך 36.8% מהדוגמאות לא יופיעו במדגם Bootstrap מסוים. לכן החלק שיופיע לפחות פעם אחת הוא:

$$1 - 0.368 = 0.632 \Rightarrow 63.2\%$$

Out-of-Bag - OOB

הדוגמאות שלא נבחרו עבור עץ מסוים נקראות Out-of-Bag עבור אותו עץ. הן לא השתתפו באימון שלו, ולכן אפשר להשתמש בהן כמבחן פנימי.

לכל דוגמה ניתן לאסוף את התחזיות של כל העצים שבהם היא הייתה OOB, ולהשוות לתשובה האמיתית. כך מתקבלת הערכת OOB Error בלי להקדיש בהכרח סט בדיקה נוסף רק למטרה זו.

16. למה Bagging לבדו עדיין לא מספיק

נניח שיש מאפיין אחד חזק מאוד, למשל הכנסה. גם אם לכל עץ ניתן Bootstrap אחר, כמעט כל העצים עלולים לבחור הכנסה בשורש. העצים יהיו שונים מעט, אבל עדיין דומים מאוד.

אם הרבה עצים עושים את אותן טעויות, הצבעת רוב אינה עוזרת מספיק. לכן אנו רוצים לא רק עצים טובים, אלא עצים שטעויותיהם אינן מתואמות מדי.

המעבר ל-Random Forest

כדי להקטין את הדמיון בין העצים נוסיף מקור אקראיות שני: בכל צומת לא נרשה לעץ לבדוק את כל המאפיינים, אלא רק תת-קבוצה אקראית שלהם.

17. Random Feature Selection

נניח שיש $p=100$ מאפיינים. בצומת מסוים נבחר באקראי רק m מאפיינים, למשל $m=10$. העץ מחפש את הפיצול הטוב ביותר אך ורק מתוך עשרת המאפיינים שנבחרו.

בצומת הבא נבחרת שוב תת-קבוצה אקראית, וייתכן שהיא שונה. כך מאפיין חזק אינו זמין בכל מקום, ועצים שונים נאלצים להשתמש במבנים שונים.

מבחינה רעיונית אנחנו מוכנים לפעמים לקבל עץ בודד מעט פחות טוב, אם הדבר הופך את אוסף העצים למגוון יותר.

משפט מפתח

Random Forest אינו מחפש אוסף של עצים מושלמים. הוא מחפש אוסף של עצים טובים אך שונים.

18. Random Forest - האלגוריתם בשלבים

כעת כל המרכיבים מוכנים, ולכן אפשר לתאר את האלגוריתם בלי קפיצות:

שלב 1: מתחילים ב-Dataset המקורי בעל n דוגמאות ו- p מאפיינים.

שלב 2: עבור עץ מספר 1 יוצרים מדגם Bootstrap בגודל n .

שלב 3: בשורש בוחרים באקראי m מאפיינים מתוך p .

שלב 4: מבין אותם m מאפיינים בלבד בוחרים את הפיצול הטוב ביותר לפי Gini, Entropy או קריטריון רגרסיה.

שלב 5: בכל צומת חדש חוזרים על בחירת תת-קבוצה אקראית של מאפיינים ועל בחירת הפיצול הטוב ביותר מתוכה.

שלב 6: ממשיכים עד תנאי העצירה של העץ.

שלב 7: חוזרים על התהליך עבור עצים רבים, וכל עץ מקבל Bootstrap חדש והגרלות מאפיינים חדשות.

שלב 8: בסיווג משתמשים בהצבעת רוב; ברגסיה משתמשים בממוצע התחזיות.

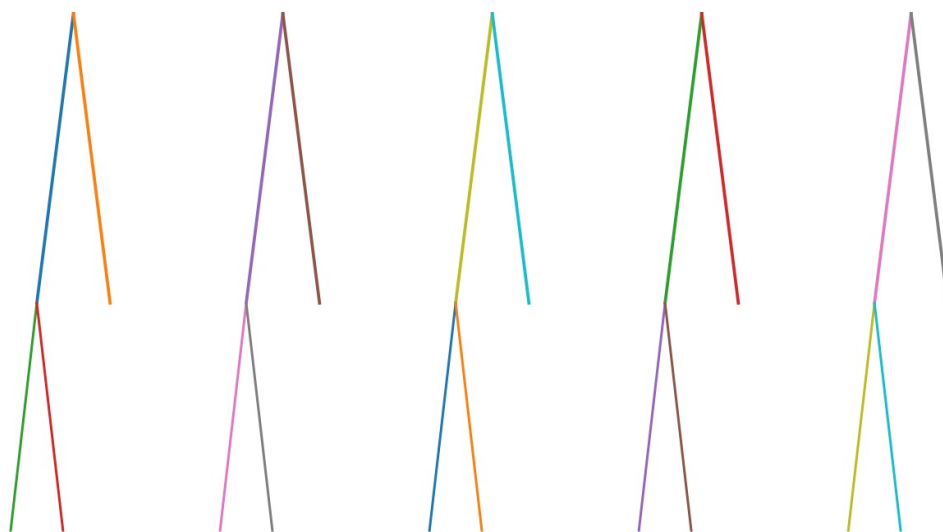
Tree 1

Tree 2

Tree 3

Tree 4

Tree 5



Pass

Pass

Fail

Pass

Fail

Majority vote → Pass

19. תחזית בסיווג וברגרסיה

סיווג

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \dots, T^B(x)\}$$

לדוגמה, חמישה עצים מחזירים: עבר, עבר, נכשל, עבר, נכשל. שלושה מתוך חמישה בחרו "עבר", ולכן זו התחזית הסופית. אפשר גם לתאר את שיעור ההצבעות כקירוב להסתברות: $0.6=3/5$.

רגרסיה

$$\hat{y} = (1/B) \sum T_b(x)$$

אם ארבעה עצים חוזים 80, 86, 81 ו-83, היער יחזיר ממוצע 82.5.

20. למה ממוצע של עצים מוריד Variance

נניח שלכל עץ שונות σ^2 . אילו העצים היו בלתי תלויים, השונות של הממוצע של B עצים הייתה:

$$\text{Var}(\text{average}) = \sigma^2 / B$$

כלומר ככל שמספר העצים גדל, התנודות האקראיות של עץ בודד מתבטלות זו מזו.

בפועל העצים אינם בלתי תלויים, כי כולם נובעים מאותו Dataset. אם הקורלציה הממוצעת ביניהם היא ρ , אפשר להבין את ההתנהגות באמצעות הקירוב:

$$\text{Var}(\text{average}) \approx \rho\sigma^2 + [(1-\rho)/B]\sigma^2$$

כאשר B גדל מאוד, האיבר השני קטן, אבל האיבר $\rho\sigma^2$ נשאר. לכן לא מספיק להוסיף עוד ועוד עצים; חשוב גם להקטין את הקורלציה ביניהם. Random Feature Selection נועד בדיוק למטרה זו.

החיבור העמוק

Bootstrap יוצר שונות בנתוני האימון של כל עץ. בחירת מאפיינים אקראית מקטינה את הדמיון המבני בין העצים. Averaging או Voting מנצלים את השונות הזו כדי להוריד Variance של התחזית הסופית.

21. Feature Importance

לאחר שבנינו יער, נרצה לעיתים לדעת אילו מאפיינים תרמו יותר לחיזוי.

21.1 Impurity-based Importance

אפשר לסכום את הירידות ב-Gini או במדד אי-הטוהר בכל הצמתים שבהם השתמשנו במאפיין מסוים. מאפיין שמופיע בפיצולים משמעותיים רבים יקבל חשיבות גבוהה.

$$\text{Importance}(X_j) \approx \sum w_t \cdot \Delta I_t$$

כאן w_t משקף את גודל הצומת ו- ΔI_t את הירידה באי-הטוהר.

21.2 Permutation Importance

גישה אינטואיטיבית יותר: מודדים ביצועים, ואז מערבבים אקראית מאפיין אחד כך שהקשר שלו לתוצאה נהרס. אם הביצועים צונחים, כנראה שהמאפיין היה חשוב.

לדוגמה: Accuracy מקורי 90%. אחרי ערבוב X_1 מתקבל 65%, ולכן X_1 חשוב. אחרי ערבוב X_2 מתקבל 89.5%, ולכן תרומתו קטנה יותר.

22. Random Forest לעומת Decision Tree

תכונה	Decision Tree	Random Forest
פרשנות	קלה מאוד	קשה יותר
מספר מודלים	עץ אחד	עצים רבים
Variance	גבוה יחסית	נמוך יותר
Overfitting	רגיש במיוחד בעץ עמוק	בדרך כלל עמיד יותר
גבול החלטה	חיתוכים מקבילי-צירים	איחוד של חלוקות רבות
סיווג	תחזית של עלה	הצבעת רוב
רגרסיה	ממוצע בעלה	ממוצע עצים

עץ יחיד מתאים במיוחד כאשר חשוב מאוד להסביר את ההחלטה. Random Forest מתאים כאשר המטרה המרכזית היא יכולת חיזוי טובה ויציבה יותר, במחיר של ירידה בפרשנות הישירה.

23. שאלות שמחדדות את ההבנה

האם יותר עצים גורמים ל-Overfitting?

הגדלת עומק של עץ בודד יכולה להגביר Overfitting. לעומת זאת, הגדלת מספר העצים ב-Random Forest בדרך כלל אינה מגדילה Overfitting באותו אופן; לאחר מספר מספיק של עצים הביצועים נוטים להתייבש. המחיר העיקרי הוא זמן וזיכרון.

מדוע ב-Random Forest אפשר להשתמש בעצים עמוקים?

עץ עמוק נותן Bias נמוך אך Variance גבוה. היער משלב הרבה עצים כאלה, וה-Averaging מפחית חלק גדול מן ה-Variance. לכן אין הכרח לגזום כל עץ באותה אגרסיביות כמו עץ יחיד.

מה ההבדל המדויק בין Bagging ל-Random Forest?

$$Bagging = Bootstrap + Aggregation$$

$$Random Forest = Bootstrap + Random Features + Tree Aggregation$$

24. תרגילים ידניים

תרגיל 1 - Gini

בצומת יש 12 דוגמאות: 8 חיוביות ו-4 שליליות. חשבו את Gini של הצומת. לאחר פיצול מתקבל צד שמאל עם 5 חיוביות ו-1 שלילית, וצד ימין עם 3 חיוביות ו-3 שליליות. חשבו את Gini המשוקלל ואת השיפור.

תרגיל 2 - בחירת סף

x	מחלקה
1	A
2	A
4	B
7	B
9	B

רשמו את ספי הביניים האפשריים ובדקו איזה סף נותן פיצול טהור ביותר.

תרגיל 3 - Pre-Pruning

קבענו $\text{min_samples_leaf}=4$. האם מותר לחלק 15 דוגמאות ל-11 ו-4? האם מותר לחלק 15 ל-12 ו-3? ומה לגבי חלוקה של 8 ל-4 ו-4? הסבירו.

תרגיל 4 - Bootstrap

יש 5 דוגמאות A,B,C,D,E. צרו ידנית שני מדגמי Bootstrap בגודל 5. סמנו בכל מדגם אילו דוגמאות הן OOB.

תרגיל 5 - Random Forest

שבעה עצים חוזים עבור דוגמה חדשה: 1,1,0,1,0,1,1. מה תחזית היער? מה שיעור ההצבעות למחלקה 1?

25. פתרונות לתרגילים

פתרון 1

$$Gini(parent)=1-(8/12)^2-(4/12)^2 = 4/9 \approx 0.444$$

בצד שמאל יש 5 מתוך 6 דוגמאות חיוביות ודוגמה אחת שלילית. לכן:

$$Gini(L)=10/36 \approx 0.278$$

בצד ימין יש 3 דוגמאות חיוביות ו-3 שליליות, ולכן:

$$Gini(R)=0.5$$

$$Gini(split)=(6/12) \cdot 0.278 + (6/12) \cdot 0.5 \approx 0.389$$

$$\Delta Gini \approx 0.444 - 0.389 = 0.055$$

פתרון 2

הערכים הממוינים הם 1,2,4,7,9 ולכן הספים האפשריים: 1.5, 3, 5.5, 8. הסף 3 מפריד {A,A} מ-{B,B,B}, ולכן נותן $Gini=0$ בשני הצדדים והוא פיצול מושלם.

פתרון 3

חלוקה של 15 דוגמאות ל-11 ו-4 מותרת, משום ששני העלים מכילים לפחות 4 דוגמאות. חלוקה ל-12 ו-3 אסורה, משום שאחד העלים מכיל רק 3. חלוקה של 8 ל-4 ו-4 מותרת.

פתרון 4

דוגמה אפשרית למדגם ראשון: A,A,C,D,E - כאן B הוא OOB. מדגם שני: B,B,C,C,E - כאן A ו-D הם OOB. אין תשובה יחידה, משום שהדגימה אקראית.

פתרון 5

יש חמש הצבעות למחלקה 1 ושתי הצבעות למחלקה 0. לכן תחזית היער היא 1, ושיעור ההצבעה למחלקה 1 הוא $5/7 \approx 0.714$.

26. סיכום - רצף הרעיונות כולו

עץ החלטה מתחיל בשאלה פשוטה: כיצד לחלק את הנתונים כך שכל צד יהיה טהור יותר. $Gini$ או Entropy נותנים לנו מדד כמותי לכך. לאחר שבחרים את הפיצול הטוב ביותר, חוזרים על אותו רעיון בכל צומת וכך נבנה העץ כולו.

כאשר נותנים לעץ לגדול ללא הגבלה הוא עלול לשנן רעש. לכן משתמשים ב-Pre-Pruning או ב-Post-Pruning כדי לשלוט במורכבות. אך גם עץ גזום נשאר מודל לא יציב יחסית: שינוי קטן בדאטה יכול לשנות את מבנהו.

Bagging פותר חלק מהבעיה באמצעות הרבה מדגמי Bootstrap והרבה עצים. Random Forest מוסיף עוד שכבת אקראיות - בחירת תת-קבוצה אקראית של מאפיינים בכל צומת - וכך מקטין את הקורלציה בין העצים.

בסופו של דבר התחזית אינה נשענת על עץ יחיד אלא על הסכמה של קבוצה גדולה של עצים. זהו המעבר הרעיוני המרכזי:

Single tree: easy to explain, but relatively high variance

Random Forest = diverse trees + aggregation → lower variance

משפט מסכם

Decision Tree לומד סדרת חוקים מקומיים. Random Forest לומד הרבה מערכות חוקים שונות ומנצל את חוסר ההסכמה ביניהן כדי לקבל החלטה יציבה יותר.