

## PROBLEM SET 3

### 1) Decision tree – finding if someone will default

- העץ נבנה בשיטת OOP והינו עץ בינארי. בניית העץ אורכת כ-10 דקות. הגרסה הסופית מביאה לידי ביטוי מספר תובנות שהגעתי אליהן על מנת לחסוך בזמן ריצה ובחישובים מיותרים:
- על מנת למקסם את  $\text{Information gain} = \text{total\_entropy} - \text{weighted\_entropy}$ , נמזער את האנתרופיה המשוקללת עבור שדה מסוים, זאת משום שבשלב ספציפי ברקורסיה  $\text{total\_entropy}$  הינו קבוע.
  - עבודה עם מערכים של numpy בעלת performance טוב יותר מאשר עבודה עם דאטה-פריים של pandas לכמות זו של נתונים.
  - בחירה חכמה של thresholds – זהו החלק העיקרי בבניית העץ. השיטה מקבלת עמודה כלשהי מהמטריצה ואת עמודת התשובות –  $\text{default\_pay}$ . נסתכל על כל אחד מהערכים השונים בעמודה כעל תנאי סף פוטנציאלי. נחשב את האנתרופיה עבור כל תנאי סף כזה, ונבחר את הטוב ביותר עבור השדה, כלומר נבחר במינימלי.
  - Online pruning – החלטתי לעשות קטימה תוך כדי בניית העץ, זאת משום שלא יכול להיות ילד שיפצל את הנתונים טוב יותר מאביו.
  - פעולת מחיקת שדה אשר בחרנו בו לפיצול משאירה אחריה אובייקט מסוג None, ועל כן, כדי להימנע מ-Deep copy בכל שלב ברקורסיה (יכול להיות שיש דרך אחרת, זולה יותר במונחי זמן, אך לא מצאתי), הנחתי ששדה לא יכול להיבחר פעמיים ולכן השארתי אותו במערך. זאת משום שהסיכוי ששדה יבחר בפעם השנייה לא גבוה.
- בנוסף אפרט על הפונקציה tree\_error (אורכת כ-50 דקות עבור  $k=5$ ):
- הפונקציה מערבבת את כלל הנתונים בצורה רנדומלית.
  - לאחר מכן, מחלקת את המידע ל-k חלקים.
  - בצורה איטרטיבית נעבור על כל אחד מ-k החלקים, נקבע שהחלק ה-i הינו ה- $\text{test\_data}$  שלנו ואילו שאר החלקים הינם  $\text{train\_data}$ .
  - בסיום שלב זה יהיו לנו ארבעה מערכים, כל אחד עם k איברים: מידע לאימון העץ, תשובות לאימון העץ, מידע לבדיקת העץ ותשובות לבדיקת העץ, כאשר הם מסודרים בהתאמה.
  - בצורה איטרטיבית נבנה k עצים ונחשב את השגיאה. לבסוף נחזיר את ממוצע השגיאות. כך נדע מהי רמת הדיוק של האלגוריתם שכתבתי.

**2) Neuron net – Hebbian learning****א) נראה דוגמה לרשת נוירונים א-סימטרית אשר איננה מתכנסת:**

$$w = \begin{pmatrix} 0 & 3 \\ -4 & 0 \end{pmatrix}$$

 $x^0 = (+ -)$  – המצב ההתחלתי

$$x^1 = \text{sign}(w_{12} * x_2) = \text{sign}(3 * (-1)) = - \rightarrow (- -)$$

$$x^2 = \text{sign}((-4) * (-1)) = + \rightarrow (- +)$$

$$x^3 = \text{sign}(3 * 1) = + \rightarrow (+ +)$$

$$x^4 = \text{sign}((-4) * 1) = - \rightarrow (+ -)$$

...

נשים לב שבכל שתי איטרציות אנחנו חוזרים למצב ההתחלתי. כמו כן מתקיים:

$$\begin{aligned} E(x^0) - E(x^1) &= -\frac{1}{2}(1 * 3 * (-1) + (-1) * (-4) * 1) \\ &\quad + \frac{1}{2}((-1) * 3 * (-1) + (-1) * (-4) * (-1)) = -\frac{1}{2} * 1 + \frac{1}{2} * (-1) = -1 < 0 \end{aligned}$$

כלומר, פונקציית האנרגיה של רשת הנוירונים אינה מונוטונית יורדת חזק  $\leftarrow$  אין לה נקודת שיווי משקל.

**ב) נראה דוגמה לרשת נוירונים סימטרית שבה העדכון קורה במקביל אשר איננה מתכנסת:**

$$w = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

 $x^0 = (+ -)$  – המצב ההתחלתי

$$x^1 = \text{sign}(w_{12} * x_2) = \text{sign}(1 * (-1)) = - \text{ and } x^1 = \text{sign}(w_{21} * x_1) = \text{sign}(1 * 1) = +$$

$$x^1 = (- +)$$

$$x^2 = \text{sign}(w_{12} * x_2) = \text{sign}(1 * 1) = + \text{ and } x^2 = \text{sign}(w_{21} * x_1) = \text{sign}(1 * (-1)) = -$$

$$x^2 = (+ -)$$

...

נשים לב שבכל איטרציה כל נוירון מתעדכן. כמו כן מתקיים:

$$\begin{aligned} E(x^0) - E(x^1) &= -\frac{1}{2}(1 * 1 * (-1) + (-1) * 1 * 1) + \frac{1}{2}((-1) * 1 * 1 + 1 * 1 * (-1)) \\ &= -\frac{1}{2} * (-2) + \frac{1}{2} * (-2) = 0 \end{aligned}$$

כלומר, פונקציית האנרגיה של רשת הנוירונים אינה מונוטונית יורדת חזק  $\leftarrow$  אין לה נקודת שיווי משקל.

**(3) Learning:**

- כפי שראינו בהרצאה 10, אכן ניתן להשתמש ברשתות נוירונים ובעצי החלטה יחד ב- **boosting algorithm**. המוטיבציה לשילוב מספר מודלים יחד הינה יצירת מודל חזק בהרבה יותר, רגיש הרבה פחות לרעשים ובעל רמת דיוק גבוהה יותר. איכות המודל החדש גבוהה בהרבה מאיכותם של כל אחד המודלים בנפרד, זאת משום שהוא טומן בחובו את היתרונות של כל אחד מהם, ובכך מניב תוצאות טובות יותר
- נסביר כיצד נשתמש במשקולות באימון של רשת נוירונים:  
 בכל איטרציה הפלט מהווה את הקלט לאיטרציה הבאה. בתחילה, כל נוירון יקבל משקל זהה, ועם התקדמות האלגוריתם נבחר את השאילות הקשות – אלו שהשגיאה שלהן הייתה גבוהה יותר מאחרות – ונגדיל את המשקל שלהן. כך, נקבל סט אימון הולך וקטן אשר שם דגש על צמצום השגיאה ולא על שאילות שלא מוסיפות מידע חדש למודל, ומקטין את זמן הריצה.