מבוא לבינה מלאכותית - תרגול 5.

רגב יחזקאל אימרה.

15 בדצמבר 2025

1 סיווג לא בינארי $Multi-class\ Classification$.

כעת, בשונה מסיווג בינארי, כאן יש לנו כמות מחלקות גדולה מ־2, כלומר $y_i \in \{c_1, c_2, c_3, \ldots, c_k\}$ (זכרו כי יש לי כמות סופית של מחלקות).

בסיווג בינארי הייתי צריך מודל סיווג בודד, שיחזה לי מה הסיכויים להיות במחלקה א' (ומכאן גם מה הסיכויים לא להיות במחלקה א'). כאן, נצטרך יותר ממסווג בודד, וכמות המסווגים תלוייה בטכניקת הסיווג שלנו:

- $One\ vs.\ All$ שאומר שעבור k מחלקות נצטרך לאמן k מסווגים בינאריים.
- $One\ vs.\ One$ שאומר שעבור k מחלקות נצטרך לאמן $\frac{k(k-1)}{2}$ מסווגים לא בינאריים.

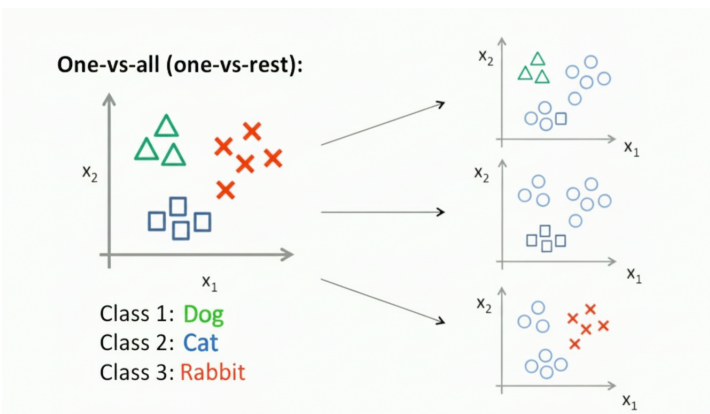
אפשר גם לסווג מודל שישר יעשה סיווג ליותר מ־2 מחלקות בלי לעבור דרך סיווג בינארי.

1.1 $One\ vs.\ All$.

בשיטת אחד מול כולם, נייצר k מסווגים בינאריים עבור k מחלקות אפשריות, כך שהמסווג ה־i יסווג בין המחלקה ה־i לבין כל שאר המחלקות.

דוגמה 1.1. נניח ויש לנו תמונה ואנחנו רוצים לדעת אם זו תמונה של כלב, חתול או ארנב, אזי בשיטת OvA נייצר 3 מסווגים:

1. {כלב} מול {חתול, ארנב}.
2. {חתול} מול {כלב, ארנב}.
3. {ארנב} מול {כלב, חתול}.



דוגמה לאיך שזה נראה בפועל, יש לנו את השאתה המקורי וכל פעם אנחנו בוחרים קבוצה להיות קבוצה א' וכל שאר הדאטה להיות קבוצה ב', ומאמנים 3 מודלי סיווג בינארי על שלושת החלוקות האפשריות.

איך זה נראה בדאטה שלנו? נניח הדאטהסט שלנו הוא

חיה y	פיצ'רים X			
כלב	-	-	-	-
ארנב	-	-	-	-
ארנב	-	-	-	-
חתול	-	-	-	-
כלב	-	-	-	-
חתול	-	-	-	-

אז נמיר את עמודת y ב־3 עמודות חדשות, כל אחת עם ערכים $0/1$ שיסמנו האם הדגימה שלנו היא המחלקה הרצויה:

ארנב	חתול	כלב	פיצ'רים X			
0	0	1	-	-	-	-
1	0	0	-	-	-	-
1	0	0	-	-	-	-
0	1	0	-	-	-	-
0	0	1	-	-	-	-
0	1	0	-	-	-	-

ואז המודל הראשון, שמסווג כלב מול כל השאר יתאמן על הדאטהסט

כלב	פיצ'רים X			
1	-	-	-	-
0	-	-	-	-
0	-	-	-	-
0	-	-	-	-
1	-	-	-	-
0	-	-	-	-

והמודל השני, שמסווג בין חתול וכל השאר יתאמן על הדאטהסט

חתול	פיצ'רים X			
0	-	-	-	-
0	-	-	-	-
0	-	-	-	-
1	-	-	-	-
0	-	-	-	-
1	-	-	-	-

והמודל השלישי, שמסווג בין ארנב וכל השאר יתאמן על הדאטהסט

ארנב	פיצ'רים X			
0	-	-	-	-
1	-	-	-	-
1	-	-	-	-
0	-	-	-	-
0	-	-	-	-
0	-	-	-	-

הערה 2.1. בעצם לקחנו את עמודת ה־y ועשינו לה $one-hot\ encoding$, ואימנו מודל אחד על כל עמודה חדשה שיצאה.

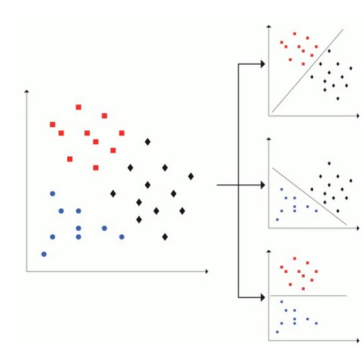
הערה 3.1. לעשות $one-hot\ encoding$ לעמודה y עם מחלקות $c_1,\dots,c_k$ זה פשוט לקחת את כל המחלקות שבה, לכל אחת לתת עמודה חדשה משלה (k עמודות חדשות) ולכל דגימה, בעמודה ה־i לשים 1 אם בעמודה המקורית y היא הייתה c_i ו־0 אחרת.

נניח וקיבלנו על תמונה חדשה שהמסווג של כלב החזיר סיכוי 0.9 להיות כלב, המסווג של חתול החזיר סיכוי של 0.5 להיות חתול והמסווג של ארנב החזיר סיכוי של 0.4 להיות ארנב. קל, פשוט ניקח את המחלקה עם הערך הכי גבוה להיות המחלקה שאנחנו חוזים, לכן נגיד שהתמונה הזאת היא תמונה של כלב.

1.2 $One\ vs.\ One$.

בשיטת אחד מול אחד, ניקח כל 2 מחלקות, ונאמן מסווג בינארי רק על הדגימות של שתיהן, ככה שנתמקד בלהפריד בין שתי המחלקות האלו (בשונה מאחד מול כולם ששם התמקדנו בלהפריד מחלקה משאר המחלקות). כיוון שאנחנו לוקחים כל מחלקה מול כל מחלקה, נצטרך לאמן $\binom{k}{2}=\frac{k(k-1)}{2}$ מסווגים.

בדוגמה מעל, עם הכלב, חתול, וארנב, זה יראה כך:



כאן ספציפית יש לנו גם 3 מקרים כי $\binom{3}{2}=3$, אבל אם היינו מוסיפים מחלקה "תרנגולת", בשיטת OvA היינו מאמנים עוד מודל של תרנגולת מול השאר, בעוד שפה היינו צריכים לאמן עוד 3 מודלים (תרנגולת מול כלב, תרנגולת מול חתול, ותרנגולת מול ארנב).

מבחינת הדאטה שלנו נפרק אותו שוב לטבלאות שונות:

כלב מול חתול:

חיה y	פיצ'רים X			
כלב	-	-	-	-
חתול	-	-	-	-
כלב	-	-	-	-
חתול	-	-	-	-

חתול מול ארנב:

חיה y	פיצ'רים X			
ארנב	-	-	-	-
ארנב	-	-	-	-
חתול	-	-	-	-
חתול	-	-	-	-

כלב מול ארנב:

חיה y	פיצ'רים X			
כלב	-	-	-	-
ארנב	-	-	-	-
ארנב	-	-	-	-
כלב	-	-	-	-

עכשיו אנחנו מקבלים וקטור לא של סיכויים מכל מסווג, אלא של תחזיות מכל מסווג, לדוגמה כאן בהינתן דוגמה חדשה נקבל וקטור סיווגים

$$\begin{pmatrix} dog \\ cat \\ dog \end{pmatrix}$$

לכן נסווג סה"כ את התמונה להיות המחלקה עם רוב הסיווגים, כלומר שוב פעם נסווג כאן את התמונה להיות תמונה של כלב.

1.3 $Softmax\ regression$.

נניח ואני לא רוצה לחזור שוב למסווגים בינאריים. פה אנחנו רוצים לאמן מודל אחד שיעבור על כל הדאטה שלנו, ובסוף יוציא לנו וקטור מאורך k שיגיד מה הסיכוי להיות בכל מחלקה.

כמו ברגרסיה לוגיסטית, גם כאן יש לי דרישות מהמודל: עבור מחלקות $\{c_1, \dots, c_k\}$ נרצה שהמודל יוציא פלט $(p_1, \dots, p_k)^T$ שיקיים:

1. $\forall i \in \{1, \dots, k\} : p_i \in [0, 1]$.
2. $\sum_{i=1}^{k} p_i = 1$.

פונקציה שאנחנו מכירים שנותנת לנו את זה היא פונקציית סופטמקס:

$$s(z_1, \dots, z_k) = \left(\frac{e^{z_1}}{\sum_{i=1}^{k} e^{z_i}}, \dots, \frac{e^{z_k}}{\sum_{i=1}^{k} e^{z_i}} \right)$$

מכאן שהאיבר ה־i ב־$s(\overrightarrow{v})$ הוא $\mathbb{P}(y = i|x, \theta)$. כמו ברגרסיה לוגיסטית, נגדיר את המודל שלי להיות

$$p_\theta(x) = s(w^T x + b)$$

1.3.1 פונקציית $loss$.

נזכור כי כל מודל הסתברותי שואף למזער את $\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log \mathbb{P}(y_i|x_i; \theta)$. נסמן $\mathbb{P}(y_i|x_i; \theta) = p_{i,y_i}$ כאשר $p_{i,k} = [p_\theta(x_i)]_k$ זה האיבר במקום ה־k בוקטור הסופטמקס של הדגימה ה־i. נסמן

$$y_{i,k} = \begin{cases} 1 & y_i = k \\ 0 & else \end{cases}$$

לכן

$$\mathbb{P}(y_i|x_i, \theta) = \prod_{j=1}^{k} p_{i,j}^{y_{i,j}}$$

נציב ב־$loss$ ונקבל

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N}\sum_{j=1}^{k} y_{i,j}\log p_{i,j}$$

הגדרה 4.1. אם נמצא את $\mathcal{L}(\theta)$ נקבל את **הפסד האנטרופיה הצולבת** (באנגלית־ $Cross\ Entropy\ Loss$) להיות

$$CE(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k} y_{i,j}\log p_{i,j}$$

1.4 איך מאמנים?

נתחיל דווקא מ־$Softmax\ regression$: כמובן שנרצה להשתמש ב־GD. בשביל זה נצטרך לחשב את הנגזרות לפי w ולפי b:
נסמן, עבור הדגימה ה־i והמחלקה ה־j:

- $z_{i,j} = w_j^T x_i + b_j$.
- $p_{i,m} = \frac{e^{z_{i,m}}}{\sum_{r=1}^{k} e^{z_{i,r}}}$.
- $\ell_i(\theta) = -\sum_{m=1}^{k} y_{i,m}\log p_{i,m}$.

לכן ה־$loss$ יהיה

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\ell_i(\theta)$$

לכל i קבוע:

$$\ell_i(\theta) = -\sum_{m=1}^{k} y_{i,m}\log p_{i,m}$$

לכן

$$\frac{\partial}{\partial p_{i,j}}\ell_i(\theta) = \frac{\partial}{\partial p_{i,j}}\left(-\sum_{m=1}^{k} y_{i,m}\log p_{i,m}\right) = -\sum_{m=1}^{k} y_{i,m}\frac{\partial}{\partial p_{i,j}}\log p_{i,m}$$

אבל אם $m \neq j$ נקבל כי $\log p_{i,m}$ לא תלוי ב־$p_{i,j}$ ולכן $\frac{\partial}{\partial p_{i,j}}\log p_{i,m} = 0$, כלומר נישאר רק עם $m = j$ ומכאן

$$-\sum_{m=1}^{k} y_{i,m}\frac{\partial}{\partial p_{i,j}}\log p_{i,m} = -y_{i,j}\frac{\partial}{\partial p_{i,j}}\log p_{i,j} = -\frac{y_{i,j}}{p_{i,j}}$$

כעת:

$$p_{i,m} = \frac{e^{z_{i,m}}}{\sum_{r=1}^{k} e^{z_{i,r}}}$$

נסמן $S_i = \sum_{r=1}^{k} e^{z_{i,r}}$. לכן $p_{i,m} = \frac{e^{z_{i,m}}}{S_i}$. נגזור:

$$\frac{\partial p_{i,m}}{\partial z_{i,j}} = \frac{\partial}{\partial z_{i,j}}\left(\frac{e^{z_{i,m}}}{S_i}\right) = \frac{S_i\frac{\partial}{\partial z_{i,j}}e^{z_{i,m}} - e^{z_{i,m}}\frac{\partial}{\partial z_{i,j}}S_i}{S_i^2}$$

אם $m = j$:

$$\frac{\partial}{\partial z_{i,j}}e^{z_{i,m}} = \frac{\partial}{\partial z_{i,j}}e^{z_{i,j}} = e^{z_{i,j}}$$
$$\frac{\partial}{\partial z_{i,j}}S_i = \frac{\partial}{\partial z_{i,j}}\sum_{r=1}^{k} e^{z_{i,r}} = e^{z_{i,j}}$$

לכן

$$\frac{\partial p_{i,m}}{\partial z_{i,j}} = \frac{e^{z_{i,j}} S_i - e^{z_{i,j}} e^{z_{i,j}}}{S_i^2} = \frac{e^{z_{i,j}}}{S_i}\left(1 - \frac{e^{z_{i,j}}}{S_i}\right) = p_{i,j}(1 - p_{i,j})$$

ואם $m \neq j$:

$$\frac{\partial}{\partial z_{i,j}} e^{z_{i,m}} = \frac{\partial}{\partial z_{i,j}} S_i = e^{z_{i,j}}$$

לכן

$$\frac{\partial p_{i,m}}{\partial z_{i,j}} = \frac{0 \cdot S_i - e^{z_{i,m}} e^{z_{i,j}}}{S_i^2} = -\frac{e^{z_{i,m}}}{S_i}\frac{e^{z_{i,j}}}{S_i} = -p_{i,m} p_{i,j}$$

כלומר

$$\frac{\partial p_{i,m}}{\partial z_{i,j}} = p_{i,j}(\delta_{j,m} - p_{i,m})$$

סה"כ

$$\begin{aligned}
\frac{\partial \ell_i(\theta)}{\partial z_{i,j}} &= \sum_{m=1}^{k} \frac{\partial \ell_i(\theta)}{\partial p_{i,m}} \frac{\partial p_{i,m}}{\partial z_{i,j}} \\
&= \sum_{m=1}^{k} -\frac{y_{i,m}}{p_{i,m}} p_{i,j}(\delta_{j,m} - p_{i,m}) \\
&= -p_{i,j} \sum_{m=1}^{k} \frac{y_{i,m}}{p_{i,m}} (\delta_{j,m} - p_{i,m}) \\
&= -p_{i,j} \sum_{m=1}^{k} \left(\frac{y_{i,m}}{p_{i,m}} \delta_{j,m} - y_{i,m}\right) \\
&= -p_{i,j} \frac{y_{i,j}}{p_{i,j}} + p_{i,j} \underbrace{\sum_{m=1}^{k} y_{i,m}}_{=1} \\
&= -y_{i,j} + p_{i,j}
\end{aligned}$$

וכעת לחלק הקל

$$\frac{\partial z_{i,j}}{w_j} = x_i, \frac{\partial z_{i,j}}{b_j} = 1$$

כלומר

$$\begin{aligned}
\frac{\partial \ell_i(\theta)}{\partial w_j} &= \frac{\partial \ell_i(\theta)}{\partial z_{i,j}} \frac{\partial z_{i,j}}{w_j} = (p_{i,j} - y_{i,j}) x_i \\
\frac{\partial \ell_i(\theta)}{\partial b_j} &= \frac{\partial \ell_i(\theta)}{\partial z_{i,j}} \frac{\partial z_{i,j}}{b_j} = p_{i,j} - y_{i,j}
\end{aligned}$$

ומכאן

$$\begin{aligned}
\frac{\partial L(\theta)}{\partial w_j} &= \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \ell_i(\theta)}{\partial w_j} = \frac{1}{N} \sum_{i=1}^{N} (p_{i,j} - y_{i,j}) x_i \\
\frac{\partial L(\theta)}{\partial b_j} &= \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \ell_i(\theta)}{\partial b_j} = \frac{1}{N} \sum_{i=1}^{N} (p_{i,j} - y_{i,j})
\end{aligned}$$

וכלל העדכון נשאר כרגיל.

עבור שיטת $One\ vs.\ All$: אם לדוגמה יש לי 3 מחלקות: כלב חתול וארנב, ואנני מכניס למודל שלי תמונה של כלב, הייתי רוצה שהמודל יוציא וקטור $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ כלומר שבהסתברות 1 הוא באמת יחזה שזה כלב, ובהסתברות 0 הוא יחזה שזה חתול ובהסתברות 0 הוא יחזה שזה ארנב. אותו

היגיון כשאני מכניס תמונה של חתול, הייתי רוצה מודל שמוציא וקטור הסתברויות $\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$, וכשאני מכניס תמונה של ארנב הייתי רוצה שהמודל יוציא וקטור $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$. נחזור לטבלת ה־$one-hot$ שיצרנו

ארנב	חתול	כלב	פיצ'רים X			
0	0	1	-	-	-	-
1	0	0	-	-	-	-
1	0	0	-	-	-	-
0	1	0	-	-	-	-
0	0	1	-	-	-	-
0	1	0	-	-	-	-

ונראה שבמילים אחרות אני רוצה שהמודל יפלוט לי את וקטור ה־$one-hot$ של אותה השורה. מכאן שהמסווג של כלב יתאמן על $\{(x_i, y_i == dog)\}_{i=1}^N$, המודל של חתול יתאמן על $\{(x_i, y_i == cat)\}_{i=1}^N$ והמודל של ארנב יתאמן על $\{(x_i, y_i == rabbit)\}_{i=1}^N$.

באופן יותר פורמלי, בהינתן k מחלקות $\{c_1, \ldots, c_k\}$ ו־N דגימות $\{(x_i, y_i)\}_{i=1}^N$, המסווג של המחלקה ה־j מול השאר יתאמן על

$$\{(x_i, \delta_{y_i,j})\}_{i=1}^N$$

כאשר

$$\delta_{y_i,j} = \begin{cases} 1 & y_i = j \\ 0 & y_i \neq j \end{cases}$$

וכל אחד מהמודלים הללו יתאמן כרגיל.

עבור $One\ vs.\ One$:

כאן אנחנו מאמנים $\frac{k(k-1)}{2}$ מודלים, אבל לכל אחד אנחנו לא מביאים את כל הדאטה, אלא רק דגימות (x, y) עם $y \in \{i, j\}$ (כשעושים מחלקה i מול מחלקה j). לכן כעת בעמודת y החדשה שלי יהיו רק שני ערכים, i ו־j, כלומר אפשר להתייחס לזה כתור אימון מודל סיווג בינארי לכל מודל בנפרד. פורמלית, אם אנחנו עכשיו עושים מחלקה i מול מחלקה j, אז הדאטהסט שלנו יהיה

$$\{(x_k, y_k) | y_k \in \{i, j\}\}_{k=1}^N$$

כאשר נסמן מחדש

$$y'_k = \begin{cases} 1 & y_k = i \\ 0 & y_k = j \end{cases}$$

ואז המודל יתאמן על $\{(x_k, y'_k)\}$ כמו מודל סיווג בינארי רגיל.

2 הערכת מודל הסיווג הבינארי

נניח ויש לנו דאטה עד הרבה אנשים שבאו להיבדק למחלת לב, ואנחנו רוצים לנבא בהינתן מטופל חדש האם הוא יחלה במחלת לב או לא. כיוון שאנחנו לא מדעני נתונים עצלנים, אנחנו לא רוצים סתם לאמן מודל עלוב על הדאטה שלנו ולסיים בזה, אנחנו רוצים לבנות את המודל הכי טוב על הדאטה שלנו.

נשאלת השאלה, מה הכוונה בהכי טוב? איך אנחנו יכולים לכמת במספר כמה המודל שלנו טוב, בדרך שתאפשר לנו להשוות בין כמה מודלים?

2.1 מטריצת בלבול - $Confusion\ Martix$.

נחלק את הדאטה שלנו ל־$train$ ו־$test$ ונאמן מודל סיווג בינארי על ה־$train$. עכשיו אני רוצה לראות כמה צדקתי על ה־$test$.

דרך אחת לעשות את זה, היא בעזרת שימוש במטריצת בלבול ($Confusion\ Martix$ באנגלית). היא נראית ככה:

		תיוג אמיתי	
		יש מחלת לב	אין מחלת לב
תיוג מהמכונה	יש מחלת לב		
	אין מחלת לב		

השורות במטריצת הבלבול הן מה שהמכונה חזתה, והעמודות במטריצת הבלבול תהיינה התיוג האמיתי של המטופל.

כיוון שיש רק 2 תיוגים לבחור מהם, התא השמאלי העליון יכיל דגימות True Positive- אלו מטופלים שבאמת היה להם מחלת לב והמכונה זיהתה שיש להם מחלת לב.

דגימות בתא הימני התחתון יהיו True Negative- אלו מטופלים שבאמת אין להם מחלת לב והמכונה אמרה שאין להם מחלת לב.
דגימות בתא השמאלי התחתון יהיו False Negative- אלו מטופלים שיש להם מחלת לב אבל המכונה קבעה שאין להם מחלת לב.
ולבסוף, דגימות בתא הימני העליון יהיו False Positive- אלו מטופלים שאין להם מחלת לב אבל המכונה קבעה שיש להם מחלת לב.
לסיכום, מטריצת בלבול תיראה כך

	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

הבחנות:

1. הערכים על האלכסון אומרים לנו כמה פעמים המודל סיווג נכון.
2. הערכים שלא על האלכסון אומרים לנו כמה פעמים המודל לא סיווג נכון.
3. סך כל הערכים במטריצת הבלבול ($TP + FP + TN + FN$) זה כך כל הדגימות שלי, כלומר כל דגימה הולכת לתא אחד ויחיד במטריצה.

דוגמה 1.2. אחרי שסיווגנו לחולה/ לא חולה לב, קיבלנו את התוצאות הבאות

תיוג אמיתי

		יש מחלת לב	אין מחלת לב
תיוג מהמכונה	יש מחלת לב	142	22
	אין מחלת לב	29	110

לכן אנחנו יודעים להגיד על המסווג שלנו שהוא סיווג נכון 142 מטופלים שהיה להם מחלת לב True Positive (וסיווג שהיה להם מחלת לב), ושהוא סיווג נכון 110 מטופלים שלא היה להם מחלת לב True Negative (והוא סיווג שלא היה להם מחלת לב).
לעומת זאת, האלגוריתם טעה בסיווג של 29 מטופלים בכך שאמר שאין להם מחלת לב כשבעצם יש להם מחלת לב False Negative, ובעוד 22 מטופלים שאמר שיש להם מחלת לב כשבעצם לא היה להם False Positive.

עכשיו אנחנו יכולים לאמן כמה מודלים, ולהשוות ביניהם בעזרת מטריצות הבלבול שלהם:
נאמן את שני המודלים על אותו $train$ ונקבל את התוצאות הבאות:

ממודל רגרסיה לוגיסטית:

תיוג אמיתי

		יש מחלת לב	אין מחלת לב
תיוג מהמכונה	יש מחלת לב	142	22
	אין מחלת לב	29	110

לעומת

מודל KNN:

תיוג אמיתי

		יש מחלת לב	אין מחלת לב
תיוג מהמכונה	יש מחלת לב	107	53
	אין מחלת לב	64	79

המודל KNN גרוע יותר בלסווג חולים עם מחלת לב, אנחנו שמים לב לזה כי כמות ה־True Positive שלנו ירדה ב־35 חולים, ולעומת זאת הם עברו ללהיות מסווגים כתור לא בעלי מחלת לב, כלומר כמות ה־False Positive עלתה ב־35 מטופלים.
בנוסף, מודל ה־KNN גרוע גם בלסווג חולים בלי מחלת לב, כי הוא סיווג 31 מטופלים יותר כתור False Negative מאשר מודל הרגרסיה הלוגיסטית שלנו.
לכן, אם נצטרך לבחור בין מודל ה־KNN למודל הרגרסיה הלוגיסטית, נבחר במודל הרגרסיה הלוגיסטית.

עכשיו נניח שיש לנו דאטה חדש, תמונות של בעלי חיים ואני רוצה לדעת האם התמונה מציגה חתול, כלב, או ארנב. מטריצת הבלבול שלי תיראה ככה:

תיוג אמיתי

		כלב	חתול	ארנב
תיוג המכונה	כלב	112	6	0
	חתול	12	102	4
	ארנב	8	60	98

בדיוק כמו בסיווג בינארי, הערכים על האלכסון אומרים לי כמה דגימות נכונות סה"כ סיווגתי, ושאר הערכים (אלו שלא על האלכסון) אומרים לי כמה דגימות סיווגתי לא נכון.

2.2 ROC, AUC ושאר חבריהם.

עכשיו נניח יש לי עבור מודל מסויים את ה־TP, TN, FP, FN שלו, אני יכול לשאול שאלות יותר מתקדמות על המודל שלי, כגון:

1. $Precision$: מבין כל התחזיות שחוזו חיוביות לפי המודל שלי, כמה מהן באמת חיוביות? אז ניקח כל מי שסומן True Positive ונחלק בכל מי שסומן בכללי Positive:

$$Precision = \frac{TP}{TP + FP} = \mathbb{P}\left(y = 1 | \hat{y} = 1\right)$$

2. $Recall$: מבין כל התחזיות שבאמת חיוביות, כמה מהן חוזו חיוביות לפי המודל שלי? אז ניקח כל מי שסומן True Positive ונחלק בכל מי שבאמת Positive:

$$Recall = \frac{TP}{TP + FN} = \mathbb{P}\left(\hat{y} = 1 | y = 1\right)$$

לעיתים גם יקרא $TPR - True\ Positive\ Rate$.

3. FPR: מביו כל הדגימות שבאמת שליליות, כמה מהן המודל שלי סיווג לא נכון? אז ניקח כל מי שסומן False Negative ונחלק בכל מי שבאמת Negative:

$$FPR = \frac{FP}{FP + TN} = \mathbb{P}\left(\hat{y} = 1 | y = 0\right)$$

$FPR - False\ Positive\ Rate$.

4. $Specificity$: מבין כל הדגימות שבאמת שליליות, כמה מהן המודל שלי סיווג נכון?

$$Specificity = \frac{TN}{FP + TN} = 1 - FPR = \mathbb{P}\left(\hat{y} = 0 | y = 0\right)$$

לעיתים גם יקרא $TNR - True\ Negative\ Rate$

5. $F_1 \backslash F - Score$: ממוצע הרמוני של $Precision$ ו־$Recall$:

$$F_1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

6. $Accuracy$: החלק היחסי של התחזיות הנכונות מתוך כלל הדגימות.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \mathbb{P}\left(y = \hat{y}\right)$$

7. $ROC\ curve$: בסיווג בינארי, ראינו שכדי לסווג דגימה כחיובית היינו צריכים לקבוע $threshold$ מסויים שלפיו מסווגים את הדגימה. ב־SVM זה היה 0, וברגרסיה לוגיסטית זה היה 0.5. אבל, זה רק ערך בודד מפני ∞ ערכי $threshold$ אפשריים. נרצה לבחון את ביצועי המודל לאורך מספר $thresholds$ כי אנחנו נרצה את ה־$threshold$ הכי טוב עבור הבעיה שלנו (אבל אין באמת ערך שהוא טוב תמיד, אלא תמיד נבחר פר בעיה). נגדיר את ה־$ROC\ curve$ להיות עקומה המצייגת על ציר ה־y את ה־TPR, ועל ציר ה־x את ה־FPR, לכל ערכי ה־$threshold$ האפשריים.

8. AUC: השטח מתחת לעקומת ה־ROC.

$$AUC = \mathbb{P}(model\ ranks\ a\ random\ positive\ higher\ than\ a\ random\ negative)$$

אומר לנו כמה המודל יכול להבחין בין מחלקות. AUC גבוה יותר אומר שהמודל מסווג יותר טוב (כלומר בכל מודל אנחנו מעוניינים שה־AUC שלו יהיה כמה שיותר קרוב ל־1).

קיימת הקבלה של זה לסיווג רב מחלקתי, למרות שזה לא הכי יפה:

נזכור כי עבור סיווג של תמונות לכלב, חתול או ארנב המודל שלי קיבל את מטריצת הבלבול

תיוג אמיתי

		כלב	חתול	ארנב
תיוג המכונה	כלב	112	6	0
	חתול	12	102	4
	ארנב	8	60	98

אז אני באמת יכול לחשב $FPR, TPR, accuracy$ כאן, פשוט שאני אצטרך לעשות את זה לכל מחלקה בנפרד. אם לדוגמה אני ארצה לחשב FPR לכלב, נקבל שאני צריך לעבוד עם התוצאות

תיוג אמיתי

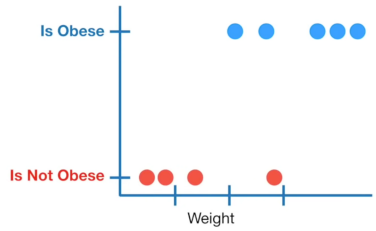
		כלב	חתול+ארנב
תיוג המכונה	כלב	$TP = 112$	$FP = 6$
	חתול+ארנב	$FN = 12$	$TN = 264$

לכן

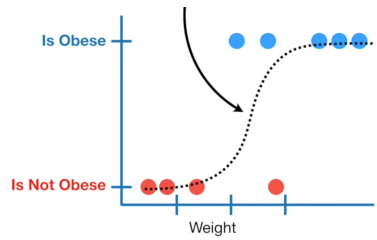
$$FPR_{dog} = \frac{6}{6 + 264} = 0.022$$

שאלה 2.2. בהינתן מודל סיווג בינארי עם $AUC < \frac{1}{2}$, מדוע אומרים שמודל יותר טוב הוא אחד שמנחש ברנדומליות 0 או 1 ($\mathbb{P}(0) = \mathbb{P}(1) = 0.5$)?

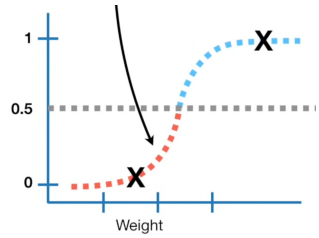
דוגמה 3.2. אינטואיציה מאחורי AUC ו־$ROC\ curve$: נניח כי שוב פעם יצאתי למדוד משקל של עכברים ואני רוצה לסווג לפי זה האם הם שמנים או לא שמנים, ויצאו לי הנקודות האלו:



יש לנו עכבר שלא שמן, למרות שיש לו משקל גבוה (כנראה שזה עכבר ממש שריר), ויש לנו עכבר רזה שכן שמן. נתאים מודל רגרסיה לוגיסטית לדאטה שלנו ונקבל את הדבר הבא:

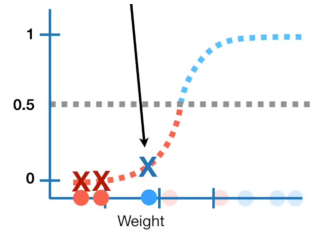


עכשי וציר ה־y הפך להיות ההסתברות של דגימה חדשה להיות שמנה. ככל שאני דוגם עכבר כבד יותר, סיכוי גבוה יותר שנסווג אותו להיות שמן, וככל שאני דוגם עכבר קל יותר, ככה סיכוי נמוך יותר שאני אסווג אותו כשמן. אמרנו כבר שבשביל שאני אסווג לשמן/ לא שמן צריך לקבוע $threshold$ מסויים, לדוגמה נקבע $threshold = 0.5$ ונקבל



כלומר כל מי שמקבל סיכוי להיות שמן יותר מ־0.5 נסווגו להיות שמן, ומי שמקבל סיכוי להיות שמן ≤ 0.5 נסווגו לא להיות שמן.

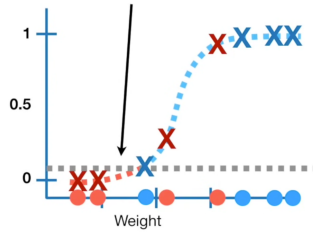
כדי להעריך את יעילות הרגרסיה הלוגיסטית ביחד עם $threshold = 0.5$, נבחן את המודל על ה־$test\ set$:



שני העכברים השמאליים הם לא שמנים, ואכן אנחנו מסווגים אותם להיות לא שמנים, אבל הנקודה הכחולה לידם מסווגת להיות לא שמנה למרות שזה כן עכבר שמן. העכבר שאחריו מסווג כמו שצריך, אך העכבר הבא לא מסווג נכון. שלושת העכברים האחרונים סווגו נכון. לכן אנחנו יכולים לבנות מטריצת בלבול כדי לסכם את התוצאות שלנו:

		תיוג אמיתי	
		שמן	לא שמן
תיוג מהמכונה	שמן	3	1
	לא שמן	1	3

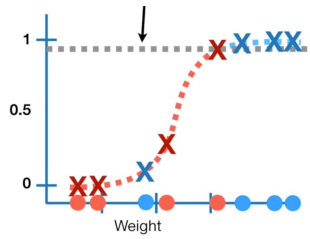
איך התוצאות ישתנו אם נשנה את ה־$threshold$ שלנו? לדוגמה, אם ממש חשוב לנו לסווג נכון את כל העכברים השמנים כשמנים, נקבע את ה־$threshold$ להיות 0.1 ונקבל



ונקבל סיווג נכון לכל ארבעת העכברים השמנים, כלומר ה־TP שלנו יעלה. כמובן שזה יגרום ל־TN שלנו לרדת, אבל כאן יצא שגם ה־FP עלה, כי לקחנו שני עכברים לא שמנים וסיווגנו אותם להיות שמנים. נקבל את מטריצת הבלבול הבאה:

		תיוג אמיתי	
		שמן	לא שמן
תיוג מהמכונה	שמן	4	2
	לא שמן	0	2

אם מצד שני היינו קובעים את ה־$threshold$ שלנו להיות 0.9 היינו מקבלים

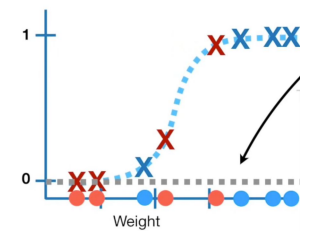


עדיין ה־TP שלנו לא ישתנה מהאחד שהיה לנו עבור $threshold = 0.5$, אבל ה־FN שלנו עלה כי סיווגנו את כל העכברים הלא זמנים כלא שמנים, כמו שצריך. מכאן גם שירד ה־FP.

אלו רק 3 ערכי $threshold$ בין 0 ל־1, יש ∞ כאלו. איך אנחנו קובעים איזה הוא הכי טוב? תחילה, נשים לב כי צריך רק לבדוק את מטריצות הבלבול רק על $threshold$ שמשנים את מטריצת הבלבול (נגיד לא נצטרך לבדוק כאן בין 0.6 ל־0.7 כי זה לא ישנה את הסיווגים שלנו. אם ניצור מטריצת בלבול לכל ערך כזה שמשנה, נקבל מספר מבלבל של מטריצות בלבול (☻). לכן נשרטט את ה־$ROC\ curve$: על ציר ה־y נשרטט את ה־TPR כתלות ב־$threshold$, ובציר ה־x נשרטט את ה־FPR כתלות ב־$threshold$.

נשרטט:

נתחיל מ־$threshold$ שמסווג את כל העכברים כשמנים:



ונקבל מטריצת בלבול

		תיוג אמיתי	
		שמן	לא שמן
תיוג מהמכונה	שמן	4	4
	לא שמן	0	0

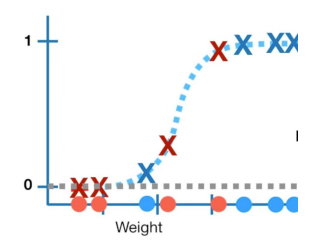
נחשב FPR, TPR:

$$TPR = \frac{TP}{TP + FN} = \frac{4}{4+0} = 1$$
$$FPR = \frac{FP}{TN + FP} = \frac{4}{4+0} = 1$$

כלומר עבור $threshold$ נמוך קיבלנו שאנחנו מסווגים נכון את כל העכברים השמנים, אך טועים לגמרי בכל העכברים הלא שמנים בכך שאנחנו מסווגים אותם כתור עכברים שמנים.

נשרטט נקודה ב־$(1,1)$. נקודה ב־$(1,1)$ אומרת שלמרות שסיווגנו נכון את כל הדגימות של העכברים השמנים, סיווגנו לא נכון את כל הדגימות של העכברים שלא שמנים.

נגדיל את ה־$threshold$ עד שיהיה לנו רק עכבר אחד שנסווג כלא שמן:



נקבל מטריצת בלבול

		תיוג אמיתי	
		שמן	לא שמן
תיוג מהמכונה	שמן	4	3
	לא שמן	0	1

כלומר נקבל

$$TPR = \frac{TP}{TP+FN} = \frac{4}{4+0} = 1$$
$$FPR = \frac{FP}{TN+FP} = \frac{3}{3+1} = 0.75$$

ונשרטט נקודה ב־$(0.75, 1)$. זה אומר שעדיין אנחנו מסווגים נכון 100% מהדגימות של העכברים השמנים, אבל עכשיו רק 75% מהעכברים הלא שמנים מסווגים כעכברים שלנים, כלומר זהו $threshold$ טוב יותר (אובייקטיבית). נגדיל את ה־$threshold$ ככה שהפעם שני עכברים יסווגו כתור לא שמנים ונקבל מטריצת בלבול

		תיוג אמיתי	
		שמן	לא שמן
תיוג מהמכונה	שמן	4	2
	לא שמן	0	2

כלומר נקבל

$$TPR = \frac{TP}{TP+FN} = \frac{4}{4+0} = 1$$
$$FPR = \frac{FP}{TN+FP} = \frac{2}{2+2} = 0.5$$

ומכאן נשרטט $(0.5, 1)$ כלומר שוב קיבלנו $threshold$ טוב יותר.

נגדיר את ה־$threshold$, הפעם קיבלתי עכבר שמן שאסווג אותו כתור עכבר לא שמן, כלומר נקבל

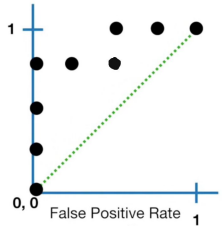
		תיוג אמיתי	
		שמן	לא שמן
תיוג מהמכונה	שמן	3	2
	לא שמן	1	2

ומכאן

$$TPR = \frac{TP}{TP+FN} = \frac{3}{3+1} = 0.75$$
$$FPR = \frac{FP}{TN+FP} = \frac{2}{2+2} = 0.5$$

כלומר נשרטט נקודה ב־$(0.5, 0.75)$.

נמשיך ככה ונקבל את הגרף הזה:



אם נחבר אותו בקו נקבל את עקומת ה־ROC לתרגיל שלנו, והשטח מתחת לגרף (ה־AUC) יהיה 0.875. אם כעת יבוא דוד ויגיד לי שהוא הריץ $Random\ forest$ על הדאטה שלי ויצא לו $AUC = 0.7$, אני אדע שהמודל שלי הוא הטוב מבין השניים.