סדנת Machine Learning

מפגש שני- Classification

חלק מהשקפים נלקחו מהספר:
"An Introduction to Statistical Learning"

חזרה

<u>בפעם הקודמת עסקנו ברגרסיה לינארית-</u>

זהו מודל בסיסי שמטרתו לאפיין את הקשר בין המשתנים/הפיצ'רים (X) לבין המשתנה החזוי (Y).

- ראינו את מדידת השגיאה של המודל (MSE) והצגנו את בעיית ה- $overfitting$.
- ניסחנו את אמידת הפרמטרים השונים ואת אמידת הדיוק שלהם
- הצגנו את הדרכים לבדיקת טיב המודל
- הרחבנו את המודל למשתנים מרובים
- העלינו בעיות שונות והצענו דרכים לפתור אותן

<u>האם רגרסיה לינארית מתאימה לכל סוגי הנתונים?</u>

תזכורת – Regression Vs Classification

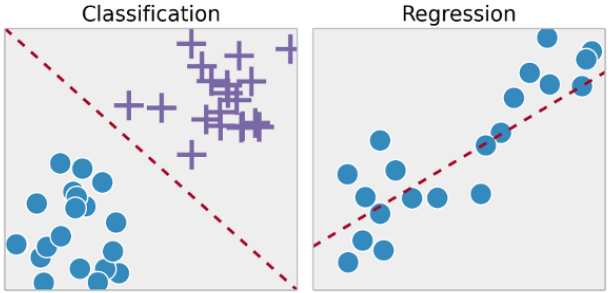
במפגש החשיפה התייחסנו לשני סוגים שונים של נתונים : נתונים מספריים (נומריים) ונתונים איכותיים.

<u>נתונים מספריים</u> – גיל, גובה, ציון פסיכומטרי (ועוד...)

<u>נתונים איכותיים</u> – מין , סיווג מחלה , קבוצה

רגרסיה לינארית מתייחסת לנתונים מספריים, בעוד עבור נתונים איכותיים אנו זקוקים למודלים מסוג אחר.

<u>מהו ההסבר הבסיסי לכך?</u>



ברגרסיה לינארית אנו מנסים לחזות <u>ערך</u> כאשר לערכים ישנו <u>סדר כרונולוגי</u> ($1000 יותר גדול מ$500),לעומת זאת, כאשר אנו מתעסקים בנתונים איכותיים , למשל מין , אין סדר כרונולוגי בין זכר לנקבה.

לכן, היום נעסוק במודלים העוסקים <u>בבעיות קלסיפיקציה</u>.

Logistic Regression

המודל הראשון בו נעסוק היום הוא רגרסיה לוגיסטית

ניקח כדוגמא נתונים המתארים את החריגה החודשית בכרטיס האשראי (Y) אל מול המשכורת השנתית (X_1) והמסגרת החודשית (X_2).

נשים לב כי המשתנה אותו אנו מנסים לחזות (החריגה החודשית) הוא משתנה איכותי (קרתה חריגה/לא קרתה חריגה) ולכן מודל הרגרסיה הלינארית אותו אנו מכירים לא יעבוד כראוי.

מודל זה מתאר את ההסתברות עבור דגימה כלשהי לשייכות למחלקה בצורה הבאה:

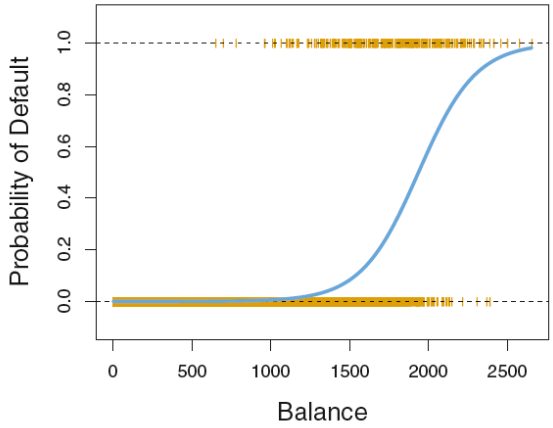
$$\Pr(\texttt{default} = \texttt{Yes} | \texttt{balance})$$

כלומר, ההסתברות שקרתה חריגה (Default = Yes) במקרה של מסגרת חודשית ספציפית (Balance)

ניתן לקצר ולכתוב: $p(\texttt{balance})$ כאשר ההסתברות היא כמובן בטווח [0,1].

מאיזו הסתברות נחליט לסווג את הדגימה כחריגת כרטיס אשראי?

Logistic Regression



כיצד נמדל את הקשר בין X לבין ההסתברות $p(X) = \Pr(Y = 1|X)$?

נשתמש בפונקציית ה- $logistic$. זוהי פונקציה ש"מכווצת" את הערכים בין 0 ל-1 בצורה רציפה.

נשתמש בפונקציה הזו על המודל הלינארי המוכר- $p(X) = \beta_0 + \beta_1 X$ ונקבל:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

לאחר מעט מניפולציות נוכל לקבל:

$$\underbrace{\frac{p(X)}{1 - p(X)}}_{odds\ function} = e^{\beta_0+\beta_1 X} \quad \Longrightarrow \quad \underbrace{\log\left(\frac{p(X)}{1 - p(X)}\right)}_{log-odds\ function} = \beta_0 + \beta_1 X$$

קיבלנו כי הקשר בין X לבין המודל הלוגיסטי הוא לינארי כתלות בפונקציית ה-$log - odds$, כלומר עלייה ביחידה אחת ב-X שקולה לעלייה ב-β_1 בפונקציית ה$log - odds$ (דוגמא בהמשך)

Logistic Regression

כיצד נעריך את הפרמטרים?

ברגרסיה לינארית השתמשנו ב$least\ squares$ על מנת להעריך את הפרמטרים.

במקרה שלנו אנו ננקוט בגישה סטטיסטית ידועה הנקראת MLE
Maximum Likelihood Estimation

בצורה אינטואטיבית:

אנו מחפשים את הפרמטרים β_0 ו - β_1 כך שכאשר "נחבר" אותם למודל $p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$ נקבל הסתברות גבוהה ככל הניתן לדגימות שהתיוג שלהן הוא 1 והסתברות נמוכה ככל האפשר לדגימות עבורן הסיווג הוא 0

בצורה מתמטית :

$$likelihood\ function\colon \ell(\beta_0,\beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1-p(x_{i'}))$$

כאשר המטרה שלנו למקסם את הפונקציה (לא ניכנס לדרכים כיצד לפתור משוואה זו)

Logistic Regression

פלטי המודל

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.6513	0.3612	−29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

כמו במודל הרגרסיה הלינארית, גם כאן אנו יכולים לבצע הסקה על הנתונים ועל המובהקות שלהם.

לצורך כך,

- אנו מחשבים את ערך הפרמטרים מתוך אמד הנראות המקסימלית (MLE) –מה ערכי הפרמטרים אומרים לנו?
- מוצאים שגיאת תקן לכל פרמטר ומחשבים משתנה Z
- מבצעים מבחן סטטיסטי (הפעם מבחן Z במקום מבחן t ברגרסיה לינארית)
- מקבלים P_value (ההסתברות לקבל ערך פרמטר כפי שקיבלנו בהנחה שהפרמטר שווה ל-0)

Logistic Regression

דוגמא לחיזוי באמצעות רגרסיה לוגיסטית

נחזור לדוגמא שלנו: נניח ואנו מעוניינים לחזות האם תתבצע חריגה בכרטיס אשראי שתקציבו $1000

נציב את הנתונים שקיבלנו במודל ונקבל:

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1{,}000}}{1+e^{-10.6513+0.0055\times 1{,}000}} = 0.00576$$

כלומר, קיבלנו שההסתברות לחריגה היא פחות מ1% (!)

לעומת זאת, אם נציב את הנתונים בתקציב של $2000 נקבל כבר 0.586 , מעל 58%

רגרסיה לוגיסטית מרובה

בדומה לרגרסיה לינארית, גם כאן אנו יכולים להרחיב את המודל כך שיעסוק במספר פרמטרים:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}} \longrightarrow \log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

גם כאן, נשתמש בMLE על מנת לקבל הערכה לגבי כל פרמטר.

<u>נשים לב לדבר מעניין:</u>

המקדם של המשתנה סטודנט הוא חיובי ברגרסיה הפשוטה , ואילו ברגרסיה המרובה המקדם שלו שלילי!

במילים אחרות, כשאנו מנתחים אותו בנפרד הוא <u>מעלה</u> את ההסתברות לחריגה ואילו כשאנו מנתחים אותו ביחד עם שאר המשתנים הוא <u>מוריד</u> את ההסתברות לחריגה

<u>מה ההגיון בכך?</u>

רגרסיה מרובה

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.8690	0.4923	−22.08	<0.0001
balance	0.0057	0.0002	24.74	<0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	−0.6468	0.2362	−2.74	0.0062

רגרסיה פשוטה

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−3.5041	0.0707	−49.55	<0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

רגרסיה לוגיסטית מרובה

ההסבר לתופעה הזו נעוץ בהבנה הבאה:

הפרמטר ברגרסיה המרובה מציין את ההשפעה יחסית לערכים קבועים של תקציב והכנסה- בהינתן ערכים קבועים, כאשר אנו מתייחסים לסטודנט ההסתברות שהוא יחרוג עם ערכי תקציב והכנסה כאלו הוא נמוך ביחס לשאר האוכלוסיה.

לעומת זאת, הפרמטר ברגרסיה פשוטה איננו מתחשב בפרמטרים האחרים. במקרה זה, מכיוון שסטודנטים ככלל נוטים לחריגה יותר משאר האוכלוסיה אנו רואים כי הפרמטר המתייחס אליהם הוא בעל ערך חיובי!

זוהי תופעה שאינה חריגה וממחישה את הצורך להבין את הנתונים שלנו בצורה טובה ולהבין את המבחנים שאנו מבצעים ואת התוצאות שלהם.

רגרסיה לוגיסטית מרובה

<u>ומה אם יש לנו יותר משתי מחלקות?</u>

במקרה כזה אנו נדרשים להכליל את הפונקציה שהגדרנו עבור 2 מחלקות למחלקות מרובות:

$$P(y=j|x_i)=\frac{e^{w_j\cdot x_i}}{\sum_k e^{w_k\cdot x_i}}$$

זוהי פונקציית $Softmax$ שהיא המקרה הכללי של פונקציית ה$logit$.

עבור כל מחלקה ישנו וקטור של משקולות (w או β שאנו מכירים) המחושב באמצעות MLE. כאשר נרמול כל הערכים להסתברויות בין 0 ל-1 מתבצע באמצעות פונקציית ה- $Softmax$.

לבסוף, נבחרת המחלקה בעלת ההסתברות הגבוהה ביותר.

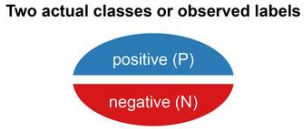
(מה קורה כאשר ישנם ערכי תיקו?)

הערכת תוצאות המודל

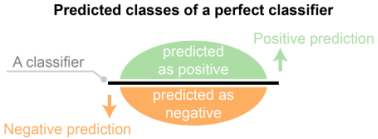
לאחר שבנינו את המודל נרצה לבחון את התוצאות המתקבלות ולדעת כמה המודל מדויק. בפעם הקודמת הצגנו בקצרה שני מדדים, היום נחזור עליהם ונרחיב מעט יותר.

בבעיות סיווג (אנו נתייחס למודל בינארי אך ניתן להכליל גם לסיווג מרובה מחלקות)

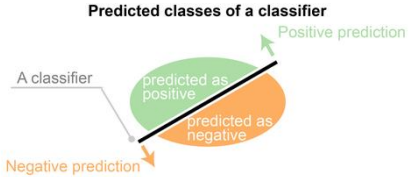
הנתונים מורכבים משתי מחלקות:



המסווג האופטימלי, ידע להבחין בצורה מושלמת בין המחלקות ויסווג כל דגימה למחלקה הנכונה:

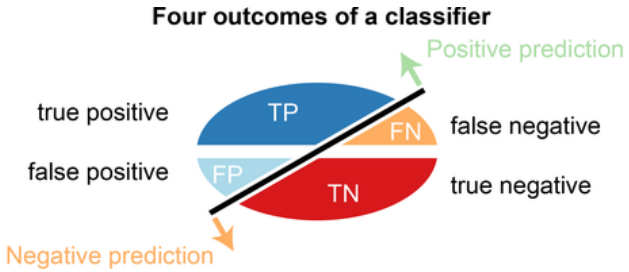


מסווג אופטימלי לרוב איננו קיים ולכן נקבל מסווג בעל שגיאה מסוימת:



הערכת תוצאות המודל

אם ננסה לבחון את התוצאות של המסווג שקיבלנו ביחס למסווג האופטימלי נבחין ב-4 תכונות:



1. TP- True Positive : דוגמאות שסווגו כחיוביות בצורה נכונה (התחזית צדקה)
2. FP- False Positive : דוגמאות שסווגו כחיוביות בצורה לא נכונה (התחזית טעתה)
3. TN- True Negative : דוגמאות שסווגו כשליליות בצורה נכונה (התחזית צדקה)
4. FN- False Negative : דוגמאות שליליות שסווגו בצורה לא נכונה (התחזית טעתה)

ניתן לסכם את כל המאפיינים הללו בסכמה שנקראת מטריצת בלבול (confusion matrix):

		Predicted	
		Positive	**Negative**
Observed	**Positive**	TP (# of TPs)	FN (# of FNs)
	Negative	FP (# of FPs)	TN (# of TNs)

הערכת תוצאות המודל

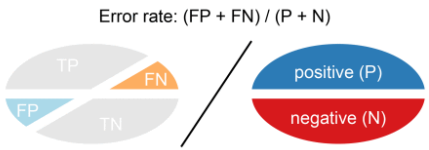
		Predicted	
		Positive	**Negative**
Observed	**Positive**	TP (# of TPs)	FN (# of FNs)
	Negative	FP (# of FPs)	TN (# of TNs)

מתוך מטריצת הבלבול שראינו ניתן להפיק מספר מדדים להערכת המודל:

- ERR- Error Rate:

אחוז התחזיות השגויות מכלל התחזית-

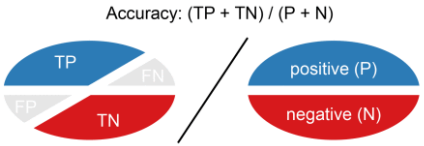
$$ERR = \frac{FP + FN}{TP + TN + FN + FP} = \frac{FP + FN}{P + N}$$



- Accuracy:

אחוז התחזיות הנכונות מכלל התחזיות-

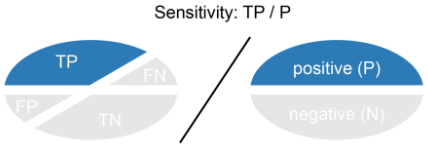
$$ACC = \frac{TP + TN}{TP + TN + FN + FP} = \frac{TP + TN}{P + N}$$



- Sensitivity (Recall):

אחוז התחזיות החיוביות הנכונות מכלל הדגימות החיוביות-

$$SN = \frac{TP}{TP + FN} = \frac{TP}{P}$$



הערכת תוצאות המודל

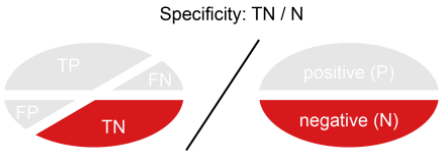
		Predicted	
		Positive	**Negative**
Observed	**Positive**	TP (# of TPs)	FN (# of FNs)
	Negative	FP (# of FPs)	TN (# of TNs)

מתוך מטריצת הבלבול שראינו ניתן להפיק מספר מדדים להערכת המודל:

- Specificity-

אחוז התחזיות השליליות הנכונות מכלל הדגימות השליליות-

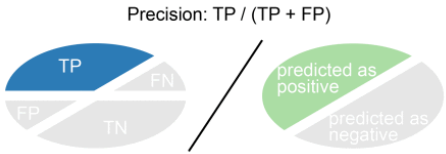
$$SP = \frac{TN}{TN + FP} = \frac{TN}{N}$$



- Precision-

אחוז התחזיות החיוביות הנכונות מכלל התחזיות החיוביות-

$$PREC = \frac{TP}{TP + FP}$$



- FPR- False Positive Rate-

אחוז התחזיות החיוביות הלא-נכונות מכלל הדגימות השליליות-

$$FPR = \frac{FP}{TN + FP} = \boxed{1 - SP}$$

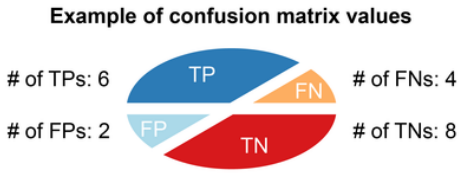
הערכת תוצאות המודל

דוגמא

נניח וברשותינו מדגם של 20 דגימות כך ש-10 חיוביות ו-10 שליליות, המסווג אותו הפעלנו הביא את התוצאות הבאות:

מדד		ערך מחושב
Error rate	ERR	6 / 20 = 0.3
Accuracy	ACC	14 / 20 = 0.7
Sensitivity True positive rate Recall	SN TPR REC	6 / 10 = 0.6
Specificity True negative rate	SP TNR	8 / 10 = 0.8
Precision Positive predictive value	PREC PPV	6 / 8 =0.75
False positive rate	FPR	2 / 10 = 0.2

נבנה את המדדים השונים



Confusion Matrix

		Predicted	
		Positive	Negative
Observed	Positive	6	4
	Negative	2	8

הערכת תוצאות המודל

ROC Curve

כעת, לאחר שהבנו את המדדים השונים, נרצה לחזור ולהבין כיצד אנו משתמשים במדדים אלו על מנת להעריך את המודל ולהשוות בין מודלים שונים.

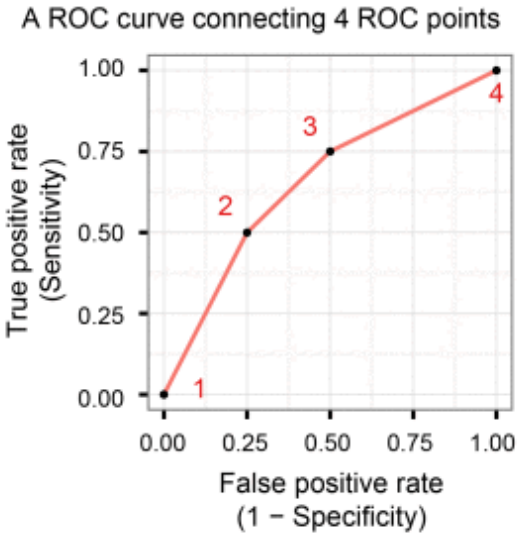
אחת הדרכים הנפוצות היא ROC Curve.

כאשר בנינו את מודל הרגרסיה הלוגיסטית קיבלנו בהינתן דגימה הסתברות עבור כל מחלקה. מהי ההסתברות ש"תספק" אותנו על-מנת לקבוע שהדגימה מסווגת כחיובית?

ההסתברות הזו מוגדרת כ-*threshold*, זהו סף הקובע האם הדגימה תסווג כחיובית או שלילית. עבור כל threshold כזה יווצר לנו יחס אחר במטריצת הבלבול בין המדדים השונים-למשל, אם אני מחמיר יותר (דורש הסתברות גבוהה יותר לסיווג חיובי) אני **אטה** יותר לצדוק בסיווגים החיוביים אך "אפספס" דגימות חיוביות רבות יותר

אם נאסוף את כל ערכי המדדים השונים עבור threshold שונים עבור המדדים Sensitivity: כפונקציה של 1-Specificity נקבל את ה-ROC Curve

גרף זה מאפשר לנו לדעת ולבחון מהו המחיר אותו אנו "מסכימים" לשלם עבור שינוי ביחסים השונים בין המדדים של המודל (למשל, באבחון רפואי אני כנראה אעדיף שרק חולים שמסווגים בוודאות כחולים ייטלו תרופה מסוימת)



הערכת תוצאות המודל

ROC Curve

לאחר שהבנו כיצד לבנות את הגרף נתבונן בגרפים הבאים:

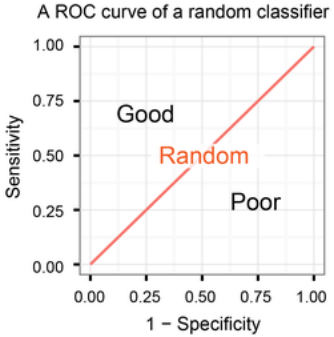
ניתן לראות כי מודל רנדומלי אמור לתת לנו קו לינארי (לכל הסתברות שנבחר הטעויות יתחלקו בצורה שווה –אין הגיון)

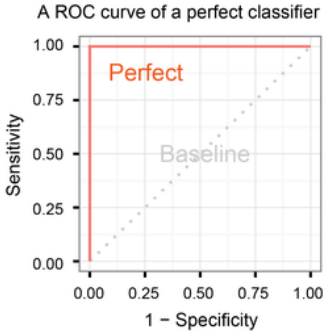
לעומת זאת, מודל מושלם היודע להפריד בצורה ברורה בין המחלקות יקבל יחס של 1 (המודל מסווג רק בצורה נכונה)

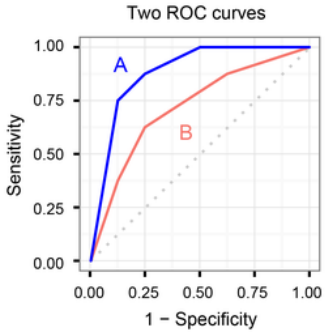
במציאות, אנו צריכים לשאוף כמובן שהגרף שלנו ייראה כמה שיותר דומה לגרף של המסווג המושלם (אם כי במקרה שהוא ממש דומה כנראה שיש לנו בעיה במודל)

באמצעות הגרף נוכל להשוות בין מודלים שונים ולראות עדיפות של אחד על פני השני:

מי עדיף?







הערכת תוצאות המודל

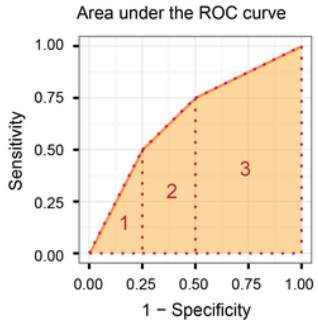
ROC Curve

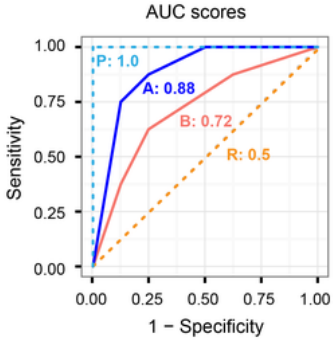
על מנת לכמת את יכולת ההשוואה בין מודלים, עלינו להסתכל בשטח שמתחת לגרף או בשמו המוכר יותר AUC

זהו בעצם מדד נוסף לטיב המודל שלנו. מודל מושלם יפיק AUC=1 בעוד מודל רנדומלי יפיק AUC=0.5

אנו מצפים שמודל סביר יפיק תוצאות בין הערכים הללו, כאשר כמובן שככל שהמדד יותר גבוה המודל יותר טוב.

אם נחזור לדוגמא מהשקף הקודם נוכל לראות את מה שהבחנו בו בצורה אינטואיטיבית. מודל A אכן מדויק יותר ממודל B.



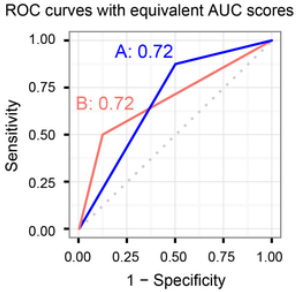


הערכת תוצאות המודל

ROC Curve

נקודות בהן עלולות לצוץ בעיות וחשוב להיזהר:

- השוואות- חשוב להשוות את כל המודלים על נתונים זהים (או מפולגים בצורה זהה מבחינת יחס הדגימות החיוביות והשליליות. השוואת מודלים מנתונים שונים תייצג הערכה שאיננה תקפה



- AUC- ייתכן ויהיו שני מודלים בעלי AUC זהה, לכן חשוב להסתכל על גרף הROC ולבחון בעיניים את הדברים (וכן להשתמש במדדים נוספים)

- נתונים שאינם מאוזנים- במידה והנתונים שלנו אינם מאוזנים ייתכן ונקבל תוצאות מאוד טובות אך בפועל המודל שלנו איננו נכון. (השאלה כיצד להתמודד עם מצב כזה לא תדון בסדנה)

Naïve Bayes Classifier

זהו מודל הסתברותי הדומה במקצת למודל הרגרסיה הלוגיסטית שראינו.

גם כאן אנו מעוניינים למצוא את ההסתברות $P\left(Y_i|X_1,X_2 \dots X_p\right)$ רק שבניגוד לרגרסיה הלוגיסטית, במקרה זה אנחנו מוצאים את ההערכה להסתברות בצורה מעט שונה.

<u>תזכורת:</u> <u>חוק בייס</u> - $P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B)}$ ובמקרה שלנו : $P\left(Y_i|X_1,X_2 \dots X_p\right) = \frac{P(Y_i)P\left(X_1,X_2 \dots X_p|Y_i\right)}{P(X_1,X_2 \dots X_p)}$

מכיוון שהמכנה הוא זהה ללא תלות בY_i לכן נוכל להתעלם ממנו <u>ולהתמקד בשני האיברים במונה</u>:

א. <u>ההסתברות הכללית של המחלקה</u>- $P(Y_i)$

ב. <u>ההסתברות המשותפת של הפיצ'רים בהינתן המחלקה</u> : $P\left(X_1,X_2 \dots X_p|Y_i\right)$- זהו ביטוי מורכב שמייצג הסתברות משותפת של הפיצ'רים בהינתן המחלקה.

Naïve Bayes Classifier

א. <u>ההסתברות הכללית של המחלקה</u>- $P(Y_i)$

על מנת למצוא את ההסתברות הזו פשוט נשתמש (שוב) בגישת ה- MLE שבמקרה הזה היא החלק היחסי של הדגימות השייכות למחלקה מתוך כלל הדגימות

<u>אם נכתוב זאת מבחינה מתמטית:</u>

$$\hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\}}{|D|}$$

כאשר D מייצג את מספר הדגימות שברשותינו (כמובן שאנו מתייחסים לדגימות ממדגם האימון)

Naïve Bayes Classifier

ב. <u>ההסתברות המשותפת של הפיצ'רים בהינתן המחלקה</u> : $P(X_1,X_2 \dots X_p|Y_i)$

מכיוון שהביטוי טיפה מורכב אנו נציג את הנחת אי-תלות בין הפיצ'רים. הנחה זו תפשט את הביטוי ותראה כיצד המודל עובד.

<u>הנחת אי-תלות</u>

זוהי הנחה הסתברותית (שבד"כ איננה נכונה בנתונים מן העולם האמיתי) שבהינתן המחלקה Y_i אין תלות בין הפי'צרים השונים.

<u>נניח:</u> $X = \langle X_1, X_2 \rangle$, אם אנו מניחים אי-תלות בהינתן המחלקה <u>נקבל:</u>

$$\begin{aligned} P(X|Y) &= P(X_1,X_2|Y) \\ &= P(X_1|X_2,Y)P(X_2|Y) \\ &= P(X_1|Y)P(X_2|Y) \end{aligned}$$

<u>כעת נשאל:</u> כיצד נוכל להעריך את הביטוי $P(X_j|Y_i)$?

Naïve Bayes Classifier

כיצד נוכל להעריך את הביטוי $P(X_j|Y_i)$?

כפי שראינו בשלב א' גם פה נשתמש באומדן לפי הכמות היחסית של ערך הפיצ'ר מתוך כלל הדגימות השייכות למחלקה.

בכתיב מתמטי:

$$\hat{P}(X_i = x_{ij}|Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\}}{\#D\{Y = y_k\}}$$

כאשר המונה מייצג את כמות הפעמים ש- X_{ij} , ערך מסוים של הפיצ'ר X_i הופיע מתוך כלל הדגימות במחלקה

דוגמא:

נגדיר: X- חום/ללא חום , Y- חולה/לא חולה , N=30 דגימות

כאשר, מתוך 30 הדגימות 20 חולים, ומתוך החולים ל-14 ישנו חום.

$$\hat{P}(\text{חולה}|\text{חום}) = \frac{\text{מספר חולים עם חום}}{\text{מספר חולים}}$$

מה הבעיה שעלולה לצוץ בשימוש בהגדרה כזו?

Naïve Bayes Classifier

לאחר שראינו כיצד ניתן לבודד ולנסח כל ביטוי ולאמוד אותו נחזור למודל שלנו

ההסתברות המשותפת של הפיצ'רים בהינתן המחלקה :

$$P(X_1,X_2 \dots X_p|Y_i) = P(X_1|Y_i) \times P(X_2|Y_i) \dots \times P(X_p|Y_i) = \prod_{k=0}^{p} P(X_k|Y_i)$$

עכשיו נחבר הכל ביחד

$$P(Y=y_k|X_1 \dots X_n) = \frac{P(Y=y_k)P(X_1 \dots X_n|Y=y_k)}{\sum_j P(Y=y_j)P(X_1 \dots X_n|Y=y_j)}$$

הנחת אי תלות

$$P(Y=y_k|X_1 \dots X_n) = \frac{P(Y=y_k)\prod_i P(X_i|Y=y_k)}{\sum_j P(Y=y_j)\prod_i P(X_i|Y=y_j)}$$

מציאת המחלקה עם ההסתברות הגבוהה ביותר

$$Y \leftarrow \arg\max_{y_k} \frac{P(Y=y_k)\prod_i P(X_i|Y=y_k)}{\sum_j P(Y=y_j)\prod_i P(X_i|Y=y_j)}$$

ניתן להזניח את המכנה

ההסתברות של המחלקה

ההסתברות של הפיצ'רים בהינתן המחלקה

$$\boxed{Y \leftarrow \arg\max_{y_k} P(Y=y_k)\prod_i P(X_i|Y=y_k)}$$

Naïve Bayes Classifier

דוגמא:

נניח ובידינו הנתונים הבאים:

Fruit	Long	Sweet	Yellow	Total
Banana	400	350	450	500
Orange	0	150	300	300
Other	100	150	50	200
Total	**500**	**650**	**800**	**1000**

אנו מעוניינים על סמך נתונים אלו לבנות מודל שידע לחזות את סוג הפרי בהינתן המאפיינים שלו.

נשתמש באלגוריתם שלמדנו. ראשית נמצא את הסתברויות המחלקות השונות:

נזכור כי אנו החישוב הוא - $\hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\}}{|D|}$

$$P(y = other) = \frac{200}{1000} = 0.2 \ , \ P(y = orange) = \frac{300}{1000} = 0.3, \ P(y = banana) = \frac{500}{1000} = 0.5$$

Naïve Bayes Classifier

Fruit	Long	Sweet	Yellow	Total
Banana	400	350	450	500
Orange	0	150	300	300
Other	100	150	50	200
Total	**500**	**650**	**800**	**1000**

דוגמא:

בנוסף, אנו יודעים לכל פרי את הסתברויות המאפיינים שלו:

נזכור כי החישוב מתבצע בצורה הבאה - $\hat{P}(X_i = x_{ij} | Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\}}{\#D\{Y = y_k\}}$

כמובן כי לכל ערך שונה של מאפיין (נניח לא מתוק), ההסתברות היא פשוט המשלים ל-1

$$P(yellow|banana) = \frac{450}{500} = 0.9,\ \ P(sweet|banana) = \frac{350}{500} = 0.7\ \ ,\ P(long|banana) = \frac{400}{500} = 0.8$$

$$P(yellow|orange) = \frac{300}{300} = 1,\ \ P(sweet|orange) = \frac{150}{300} = 0.5\ \ ,\ P(long|orange) = \frac{0}{300} = 0$$

$$P(yellow|other) = \frac{50}{200} = 0.25,\ \ P(sweet|other) = \frac{150}{200} = 0.75\ \ ,\ P(long|other) = \frac{100}{200} = 0.5$$

Naïve Bayes Classifier

דוגמא:

כעת, לאחר שבידינו כל החישובים נוכל בהינתן פרי חדש לסווג אותו.

נניח והגיע פרי בעל המאפיינים הבאים – מתוק, צהוב וארוך כיצד נסווג אותו?

עבור כל פרי נחשב את ההסתברות להיות שייך למחלקה, נזכיר כי החישוב מתבצע בצורה הבאה: $Y \leftarrow \arg\max_{y_k} \; P(Y = y_k) \prod_i P(X_i | Y = y_k)$

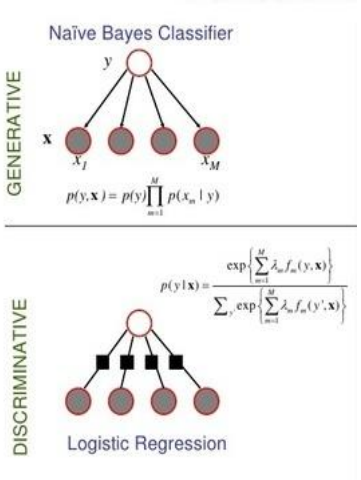
$$P(banana|yellow, sweet, long) = P(sweet|banana) \times P(long|banana) \times P(yellow|banana) \times P(banana) = 0.8 \times 0.9 \times 0.7 \times 0.5 = 0.252$$

$$P(orange|yellow, sweet, long) = P(sweet|orange) \times P(long|orange) \times P(yellow|orange) \times P(orange) = 1 \times 0 \times 0.75 \times 0.3 = 0$$

$$P(other|yellow, sweet, long) = P(sweet|other) \times P(long|other) \times P(yellow|other) \times P(other) = 0.25 \times 0.5 \times 0.75 \times 0.2 = 0.01875$$

לכן, נבחר לסווג את הפרי כבננה

Naïve Bayes VS Logistic Regression



<u>הבדלים בין Naïve Bayes ל-Logistic Regression:</u>

1. LR מעריך בצורה ישירה את ההסתברות $P(Y|X)$, לעומת זאת NB מעריך את ההסתברות המשותפת $P(Y,X)$ זוהי הבחנה חשובה בין $Discreminitive\ Models$ (LR) לבין $Generative\ Models$ (NB).
2. LR אינו מניח את הנחת האי-תלות בצורה מובהקת כמו NB ולכן נוטה לביצועים יותר טובים ככל שהמשתנים יותר תלויים אחד בשני (דבר שאינו קורה לרוב במציאות). כאשר המשתנים אינם תלויים NB יבצע טוב יותר
3. LR מציג הסתברויות למחלקה בצורה נוחה וממדל את הקשר בין Y ל-X במונחים לינאריים וכתוצאה מכך ניתן להסיק על עוצמת השינוי הדרוש במשתנה מסויים ולבצע מבחנים סטטיסטיים, NB אינו נותן את הסתברויות אמפיריות אלא בוחר את הערך ההסתברותי היותר גבוה (calibration)
4. LR ניתן לטיוב ואופטימיזציה באמצעות רגולריזציות שונות ושיטות לבחירת פרמטרים (יילמד בהמשך הסדנה)
5. NB מתכנס לפתרון בצורה מהירה יותר (דורש מעט דוגמאות) לעומת RL שדורש דוגמאות רבות.
6. NB בעל $bias$ גבוה אבל $variance$ נמוך וLR בעל $bias$ נמוך אבל $variance$ גבוה (למדנו במפגש הראשון)

K-Nearest Neighbors

זהו אלגוריתם פשוט אך יעיל המשמש הן לבעיות סיווג והן לבעיות רגרסיה.

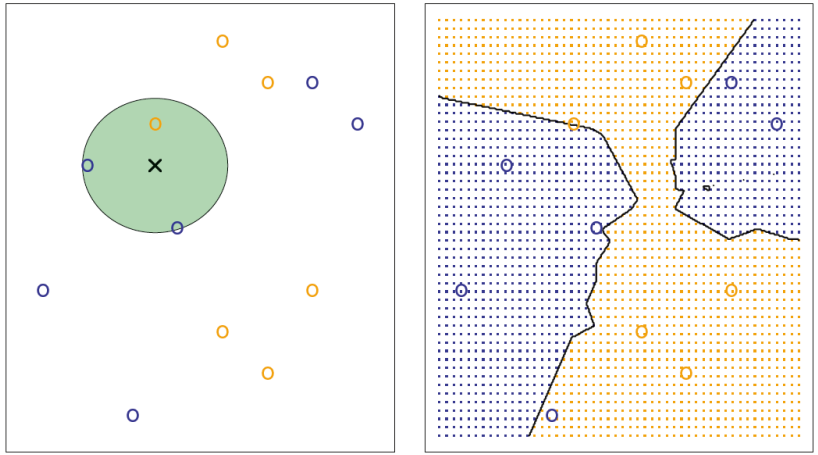
<u>העיקרון העומד מאחוריו מאוד פשוט:</u>

אנו מקבלים כקלט דגימות שונות בעלות מאפיינים (פיצ'רים) X וסיווג Y כלשהו (מחלקה כלשהי). כאשר מתקבלת דגימה חדשה אנו נבחן את K השכנים הקרובים אליה ביותר (מבחינת המאפיינים) ונסווג את הדגימה <u>לפי המחלקה של רוב השכנים הקרובים.</u>

<u>מבחינה מתמטית:</u>

$$\Pr(Y=j|X=x_0)=\frac{1}{K}\sum_{i\in\mathcal{N}_0} I(y_i=j)$$

כאשר אנו לוקחים את המחלקה בעלת <u>ההסתברות הגבוהה ביותר</u>



<u>דוגמא ל-KNN עם K=3</u>

<u>משמאל:</u> דגימה חדשה(מסומנת ב-x) מסווגת למחלקה הכחולה (2/3 לכחולה מול 1/3 לכתומה)

<u>מימין:</u> גבול ההחלטה לסיווג לפי K=3

K-Nearest Neighbors

<u>על אף שזהו אלגוריתם מאוד פשוט ישנן שתי נקודות חשובות לציון:</u>

- זהו אלגוריתם *עצלן* , הוא איננו בונה מודל מייצג אלא שומר את כל הדגימות ובזמן אמת (או קרוב לכך) מחשב את המרחקים השונים. כתוצאה מכך זמן האימון הוא קצר מאוד (בעיקר ייצוג של הדגימות ועיבוד מקדים) אך זמן המבחן הוא ארוך מאוד יחסית למודלים *חרוצים* (שאינם *עצלנים)*

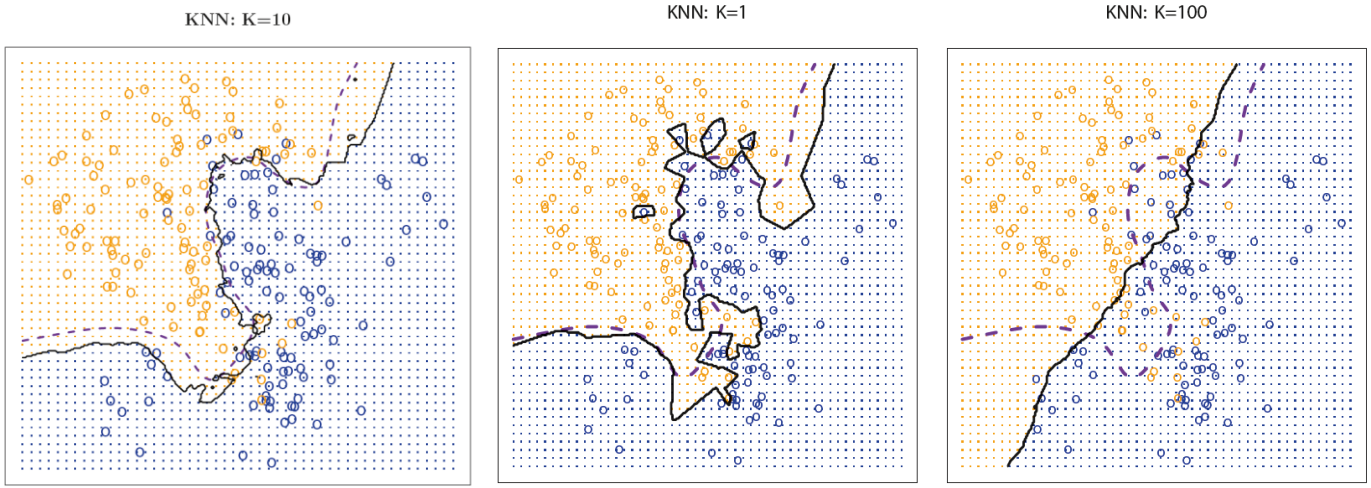
- האלגוריתם אינו-פרמטרי במובן זה שאינו מניח ידע מוקדם על הנתונים (למשל: הנתונים ניתנים להפרדה לינארית, התפלגות נורמלית/גאוסיאנית) <u>זוהי תכונה חשובה</u>, כפי שלמדנו מקרים רבים בעולם האמיתי אינם מתנהגים בצורה "נחמדה" ולכן שיטות כאלו יכולות להיות עדיפות במקרים רבים על פני שיטות פרמטריות

K-Nearest Neighbors- נקודות חשובות

- <u>בחירת K:</u>

לבחירת מספר השכנים השלכות נרחבות על ביצועי האלגוריתם. <u>נסתכל על הדוגמא הבאה:</u>

<u>זוהי הדגמה של פעולת האלגוריתמים עבור K שונים בנתונים בעלי n=200 דגימות ושתי מחלקות. מה תוכלו להסיק מכך?</u>

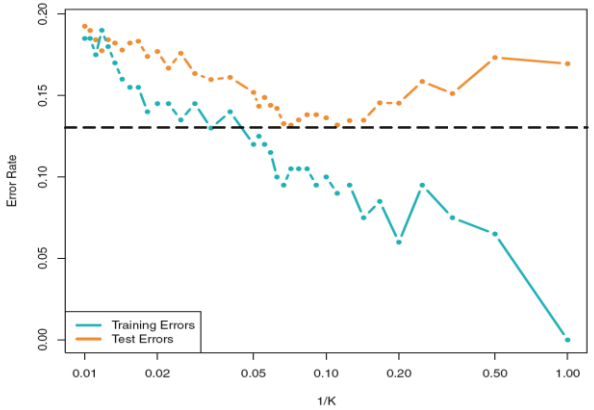


הקו השחור מציין את ההפרדה הנוצרת ע"י האלגוריתם ואילו הקו הסגול המקווקו הוא השגיאה המינימלית הניתנת להשגה (זהה בכולם כמובן)

K-Nearest Neighbors- נקודות חשובות

<u>מסקנות:</u>

- כאשר $K = 1$ – אנו מתייחסים לשכן הקרוב בלבד , זוהי דוגמא ל$bias$ נמוך אבל $variance$ גבוה (הקו מאוד פתלתל וגמיש –מותאם מדי לנתונים הספציפיים)
- ככל שK גדל הקו נהיה קשיח יותר ויותר עד שכאשר $K = 100$ אנו מקבלים קו המזכיר בצורתו קו לינארי -$bias$ גבוה אבל $variance$ נמוך (מכליל אבל איננו מותאם לנתונים הקרובים)
- כאשר $K = 10$– אנו מקבלים קו המזכיר מאוד את הקו האופטימלי. ניתן לראות את השוואת שגיאת המבחן והאימון, נשים לב שגם פה (בדומה למפגש הקודם) אנו יכולים להבחין בצורת ה-U כאשר שגיאת המבחן יורדת עד לאזור $K = 10$ ומשם (לאור גמישות המודל) עולה
- <u>ככלל אצבע ניתן לומר</u> שהK הרצוי הוא $\sqrt{N}$ - מתוך הנחה שגודל הK תלוי במספר הדגימות (ככל שיש יותר דגימות נרצה להסתכל יותר במרחב ולהפך)
- כאשר מספר המחלקות הוא 2 (סיווג בינארי) נבחר את K להיות מספר אי-זוגי



K-Nearest Neighbors- נקודות חשובות

- בחירת מטריצת המרחק

על-מנת להעריך את הקרבה בין הדגימות השונות אנו זקוקים למדד אותו ניתן להשוות. לצורך כך ישנן מספר אופציות:

מרחק מנהטן: $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$ ניתן לכתוב גם כך (לוקטור בעל מספר איברים´

$$d(x, x') = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + \ldots + (x_n - x_n')^2}$$

מרחק אוקלידי: $\sum_{i=1}^{k}|x_i - y_i|$

מרחק מינקאוסקי (הכללה של שני המקרים הקודמים): $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

ישנם עוד המון סוגים של השוואת מרחקים (cosine similarity,Jaccard ועוד) אך כל המרחקים האלו שימושיים במשתנים רציפים, כאשר למשתנים קטגוריאליים ישנו מדד אחר:

מרחק Hamming: $D_H = \sum_{i=1}^{k}|x_i - y_i|$

$$x = y \Rightarrow D = 0$$
$$x \neq y \Rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1

K-Nearest Neighbors -נקודות חשובות

- Weighted KNN

עד כה, התייחסנו לכל השכנים בצורה דומה, נניח ובחרנו K=10 המשקל של השכן העשירי שווה למשקל של השכן הראשון.

האם זוהי הנחה נכונה?

לא תמיד. בדרך-כלל, ככל שהשכן יותר קרוב הוא נוטה להיות יותר משמעותי בבחירת הסיווג של הדגימה. לכן, הפתרון הוא להשתמש בגרסה מעודכנת של KNN הממשקלת את תרומת השכנים לסיווג שכנים לפי קירבתם לדגימה.

מבחינה מתמטית נכתוב כך:

$$w'_i = \begin{cases} \frac{d(x',x_k^{NN})-d(x',x_i^{NN})}{d(x',x_k^{NN})-d(x',x_1^{NN})} & , \text{ if } \quad d(x',x_k^{NN}) \neq d(x',x_1^{NN}), \\ 1 & , \text{ if } \quad d(x',x_k^{NN}) = d(x',x_1^{NN}). \end{cases}$$

המשקל של השכן ה-i'th יהיה ביחס למרחק שלו מהדגימה אל מול המרחק המינימלי לדגימה (סוג של נרמול בין 0 (הכי רחוק) ל-1 (הכי קרוב)

$$y' = \arg\max_y \sum_{(x_i^{NN},y_i^{NN})\in T'} w'_i \times \delta(y = y_i^{NN})$$

המחלקה תיקבע לפי השכנים (כמו KNN רגיל) רק שהפעם לכל שכן ישנו משקל שונה בהתאם לקרבה

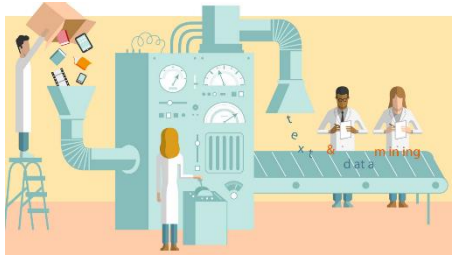
K-Nearest Neighbors- נקודות חשובות

- הכנת הנתונים ל-KNN

ישנם 2 דברים שעשויים להשפיע על ביצועי האלגוריתם:

1. נתונים שאינם מנורמלים- נניח ובנתונים שלנו ישנם שני מאפיינים: גובה, משקל. כאשר נבצע מדידת מרחק (באמצעות אחת מן המטריקות שהצגנו מקודם) המרחקים של המאפיינים השונים יהיו לא פרופורציונליים. זוהי תופעה שיכולה לגרום להדגשת מאפיין מסוים על פני מאפיין אחר בגלל טווח הנתונים שלו. על-מנת לפתור בעיה זו נבצע נרמול לנתונים בין 0 ל-1: עבור כל מאפיין נחסר את ממוצע הערך ונחלק בסטיית התקן (ראינו נרמול דומה במפגש על רגרסיה לינארית)

2. מימדים גבוהים- לאלגוריתם KNN (ולא רק לו) ישנה בעיה להתמודד עם מספר גדול של פיצ'רים. לדוגמא, אם יש לנו 20 פיצ'רים ו- n=100 דגימות ,אז קרוב לוודאי שמדדי הקרבה השונים לא יצליחו למצוא שכנים קרובים (באמת) לדגימה שלנו ובשל כך נקבל ביצועים גרועים של האלגוריתם. בעיה זו נקראת *Curse of Dimensionality*. על-מנת לפתור בעיה זו נשתמש בטכניקות הורדת מימד (יילמד בהמשך הסדנה) כך שנוכל לקבל ייצוג מוקטן ודחוס של הפיצ'רים שאיתם יוכל האלגוריתם להתמודד בצורה טובה



K-Nearest Neighbors- יתרונות וחסרונות

- <u>יתרונות</u>

1. קל ופשוט להבנה ומימוש
2. משמש כאלגוריתם בסיסי (baseline) להשוואה ראשונית
3. מודל א-פרמטרי – איננו מניח דבר על הנתונים (מגלה יותר גמישות)
4. יעיל הן בבעיות סיווג והן בבעיות רגרסיה

- <u>חסרונות</u>

1. סיבוכיות חישוב גבוהה (חישוב בזמן-אמת)
2. רגיש במיוחד לכמות הפיצ'רים ולערכי הנתונים (ללא-נרמול)
3. נוטה לבחירת הרוב בנתונים שאינם מאוזנים (לדוגמא, אם מחלקה ספציפית מכסה 90% מהדגימות רוב הסיכויים שהדגימה תתוייג ככזו)
4. דיוק יחסית נמוך ביחס לאלגוריתמים יותר מורכבים (כמו כן, אם הנתונים אכן באים ממודל לינארי-מודלים לינאריים עדיפים)