הטכניון – מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל

המעבדה לבקרה, רובוטיקה ולמידה חישובית

ניסוי מעבדות 2-3 :

מבוא למערכות לומדות

Introduction to Machine Learning

נכתב על ידי : מעין הראל, אורלי אבנר - אוגוסט 2011

עדכון :

מעין הראל, אורלי אבנר – אוקטובר 2012
טל דניאל – יולי 2019

תיקונים :

איתמר כץ – אוגוסט 2013
מרק לוין- יולי 2016
רון עמית – מאי 2017
אדוארד מורושקו – אוגוסט 2017
רון עמית – יוני 2020

עדכונים נוספים באתר המעבדה : http://eewebt.technion.ac.il/LABS1/control

תוכן עניינים

רקע למעבדה – מפגש ראשון

רקע על התחום

ניסוי זה מהווה מבוא למערכות לומדות (Machine Learning), תחום העוסק בפיתוח ותכנון אלגוריתמים המאפשרים מיצוי אוטומטי של מידע מתוך נתונים אמפיריים. לפי אחת ההגדרות, מערכת לומדת היא מערכת אשר משפרת את ביצועיה בביצוע משימה נתונה ככל שהיא מבצעת משימה זו.

לתחום יישומים רבים ומגוונים: זיהוי כתב יד ודיבור, סיווג מסמכים (כפי שתראו בהמשך), לימוד במשחקים, רובוטיקה, ביולוגיה חישובית, כריית מידע, חיזוי פיננסי ועוד.

בניגוד לפתרון אלגוריתמי "מסורתי", בו האלגוריתם מפורש וקבוע וכל פרטי הפתרון ידועים למתכנן, אלגוריתם לומד מוכתב עד כדי מאפיינים (פרמטרים) תלויי מידע, המכוונּנים במהלך הלימוד.

לגישה לומדת יתרונות רבים, ביניהם הקניית יכולות שהן מעבר ליכולת הניתוח של מפתח המערכת, והסתגלות לסביבה משתנה.

בעיה פונדמנטלית במערכות לומדות הינה בעיית הסיווג (בה מסווגים נתונים למחלקות מוגדרות מראש וזאת לעומת בעיית הרגרסיה בה חוזים ערכים רציפים), בה נעסוק בניסוי. דרך בעיה בסיסית זו תחשפו להיבטים שונים של התחום ולחלק מהאתגרים שהוא מציב. אנו מקווים כי מבוא זה ישמש לכם צוהר לעולם המרתק של מערכות לומדות.

בעיית הסיווג

1. הגדרות

בבעיית הסיווג אנו נדרשים לתכנן מסווג באמצעות סט דוגמאות מתויגות כך שיסווג בצורה הטובה ביותר קלט חדש.

נשתמש בהגדרות הבאות לתיאור בעיית הלמידה:

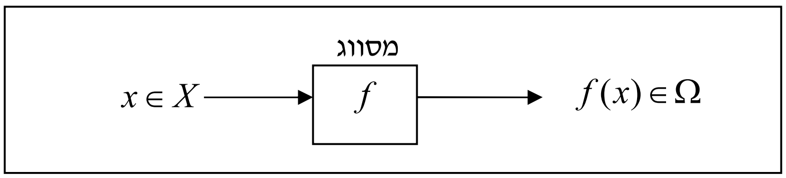
- מרחב הקלט: $X \subset \mathbb{R}^d$, כך שכל דגימה $x = (x_1, x_2, \ldots, x_d) \in X$. כאשר $d > 1$, נקרא ל- x "וקטור המאפיינים" או "וקטור הפיצ'רים (features)".
- מרחב הפלט: $\Omega = \{1, 2, \ldots, C\}$ מכיל את אוסף המחלקות האפשריות.
- מסווג: העתקה (פונקציה) $f : X \to \Omega$ אשר נותנת לכל קלט במרחב הקלט תיוג.
- סדרת הלימוד (training set): סט של n דוגמאות מתויגות (labeled) $\{x_k, y_k\}_{k=1}^{n}$, כאשר $y_k \in \Omega$ הוא הסיווג הנכון של תבנית הקלט.
- סדרת הבוחן (test set): סט של m דוגמאות (שאינו שייך לסדרת הלימוד) $\{x_k\}_{k=n+1}^{n+m}$, עם תיוג לא ידוע, אותן נרצה לסווג. את הדוגמאות מהסט הזה האלגוריתם לא רואה בשום שלב של

האימון. הביצועים על הסט הזה ייתנו לנו מידע על שגיאת ההכללה – השגיאה האמיתית על דוגמאות חדשות שהאלגוריתם לא ראה מעולם.

נגדיר שנית את בעיית הסיווג תוך שימוש במושגים הנ"ל:

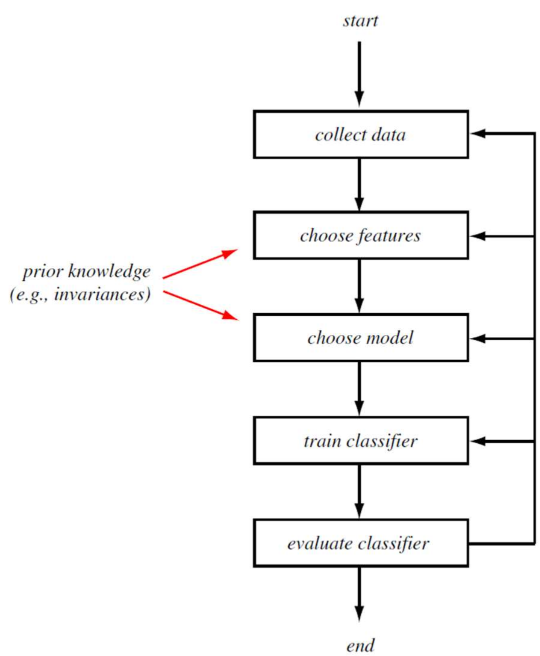
בהינתן סדרת לימוד $\{x_k, y_k\}_{k=1}^{n}$ נרצה למצוא מסווג $f : X \to \Omega$ כך שיסווג את סדרת הבוחן $\{x_k\}_{k=n+1}^{n+m}$ למחלקה המתאימה עם שגיאה קטנה ככל הניתן.

באופן סכמתי מסווג מוגדר בצורה הבא:



הלמידה המתוארת הינה למידה אינדוקטיבית: הכללה מהפרט- סדרת הלימוד, אל הכלל – קלט חדש. בפרט נשים לב כי נדרש לתכנן מסווג בעל שגיאה קטנה על דוגמאות חדשות שלא שייכות לסט הדוגמאות בו נעזרנו לתכנון.

התרשים הבא מתאר בצורה סכמתית את תהליך הלימוד בבעיית הסיווג.



איור 1 – תיאור סכמתי של תהליך הלימוד, מתוך [1]

לאחר שלב איסוף המידע, יש לבחור את המאפיינים הרלוונטיים לתיאורו ואת המודל המתאים למערכת; בשלב זה ניתן לשלב ידע מקדים אודות המערכת. לאחר מכן מגיע שלב אימון המסווג ובחינת ביצועיו.

2. <u>דוגמא פשוטה</u>

במפעל אריזת דגים מעוניינים להפריד באופן אוטומטי בין הדגה היומית. בפרט, יש להפריד בין דגי הסלמון לדגי הלברק על סמך תמונה של הדג, דהיינו למצוא מסווג שמוציא עבור כל תמונה פלט המציין את סוג הדג.

למשימה מסוג זה קודם כמובן שלב של מיצוי מאפיינים "מעניינים" מתוך התמונה (שלב זה נקרא עיבוד מקדים או pre-processing), כמתואר באיור 1. מאפיינים אילו יסייעו לנו בסיווג הדגים.

הוחלט כי ימוצו מתוך התמונה שני המאפיינים הבאים: אורך הדג ובהירות הדג (בהינתן שהתמונה נתונה בשחור-לבן). נציין שמיצוי המאפיינים מתוך תמונה דורש הפעלת עיבוד תמונה על-מנת לנקות את התמונה מרעש ולבצע סגמנטציה של הדג מתוך הרקע. במעבדה זו לא נעסוק בשלב זה ונניח כי בידינו שני המאפיינים הרצויים עבור כל תמונה.

תחילה נכתוב את בעיית הלמידה הנתונה באמצעות ההגדרות לעיל:

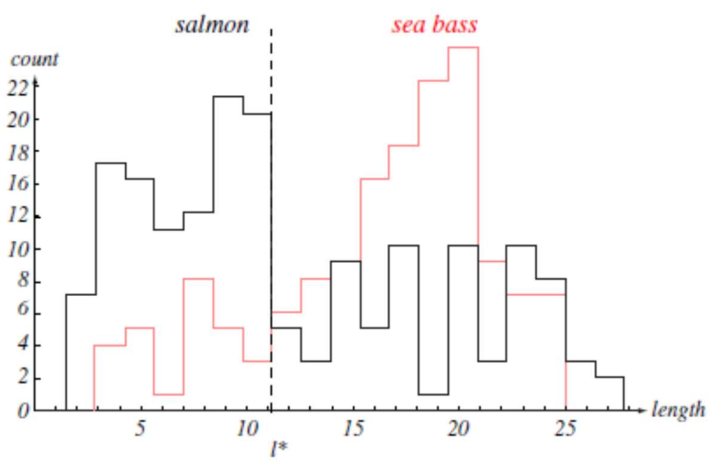
מרחב הקלט - $X \subset \mathbb{R}^2$ כך שעבור כל דגימה $x \in X$ יש שני מאפיינים: אורך ובהירות.

מרחב הפלט - $\Omega = \{-1,+1\}$ מכיל שתי מחלקות לסיווג: סלמון ולברק. בעיית סיווג מסוג זה, עם שתי מחלקות בלבד, נקראת בעיית סיווג בינארית.

כדי ללמוד את המסווג שלנו נקבל סדרת דוגמאות אימון $\{x_k, y_k\}_{k=1}^{n}$, כלומר תמונות דגים אשר מחלקתם תויגה באופן ידני.

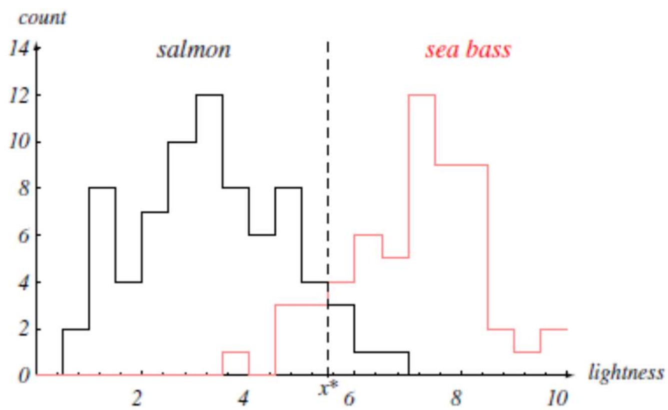
בשלב ראשון נתבונן בדוגמאות שלנו, ובהתפלגות המאפיינים השונים.

באיור 2 מוצגת ההיסטוגרמה של הדוגמאות ע"פ המאפיין של אורך הדג (כל האיורים נלקחו מתוך [1]). ניתן לראות כי אין הפרדה טובה בין המחלקות מכיוון שהחפיפה גדולה מידי.



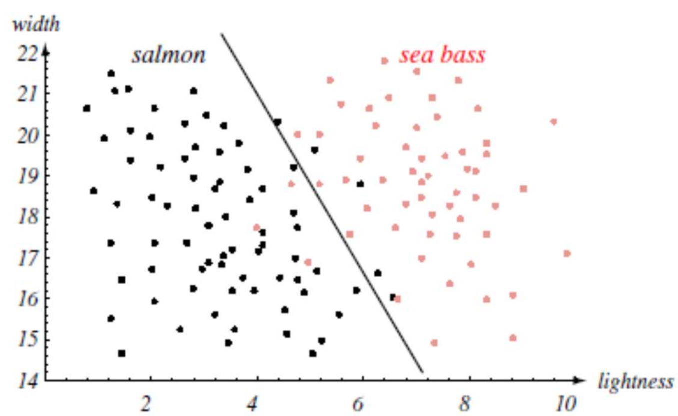
איור 2 – היסטוגרמה ע"פ אורך הדג

באיור 3 ניתן לראות את ההיסטוגרמה של הדוגמאות ע"פ המאפיין של בהירות הדג. כאן תחום החפיפה קטן יותר שכן ברוב המקרים הלברק בהיר יותר.



איור 3 - היסטוגרמה ע"פ בהירות הדג

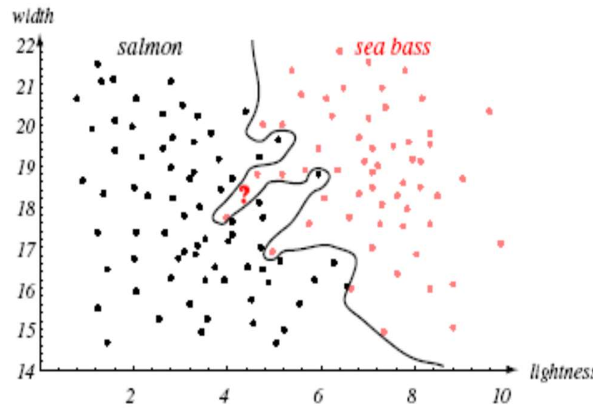
נשאלת השאלה האם שימוש בשני המאפיינים יחד יכול להביא לייצוג טוב יותר של הבעיה. על מנת לענות על שאלה זו, ניתן לצייר את ערך המאפיינים בתרשים דו-ממדי, כפי שמוצג באיור 4. כפי שניתן לראות ניתן להעביר קו ישר בין שתי המחלקות המסווג באופן נכון את רוב דוגמאות סדרת הלימוד. ניתן להבחין שבעזרת הייצוג הדו-מימדי שתי המחלקות ניתנות להפרדה טובה יותר מאשר בשימוש באחד מהמאפיינים.



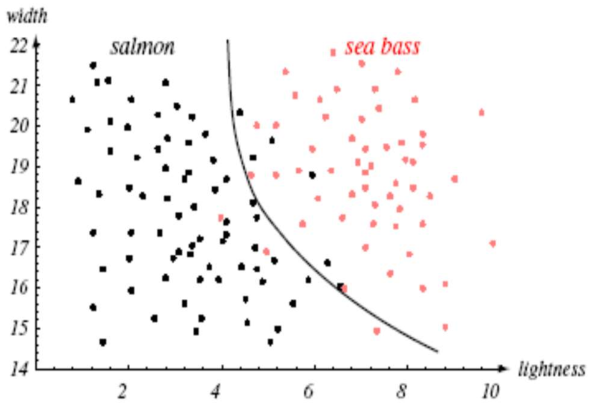
איור 4 – הצגה דו-מימדית של המאפיינים

כמובן שקיימות אפשרויות נוספות להעברת הקו לסיווג המחלקות. שתי דוגמאות מוצגות בהמשך. באיור 5 ניתן להבחין כי מושג סיווג מושלם על סדרת האימון אך אזורי ההחלטה מאוד מסובכים. אזורי החלטה כאלו יכולים להשיג אפס שגיאה על סדרת האימון, אך עלולים להוביל לשגיאה גדולה יותר על דוגמאות חדשות . באיור 6 מוצג מסווג בעל אזור החלטה עם פחות שגיאות על סדרת האימון מזה שבאיור 4, אך מעט יותר מסובך.

בתכנון מסווג יש להתייחס ל-tradeoff בין סיבוכיות המודל והביצועים על סדרת האימון.



איור 5



איור 6

3. <u>מדד ביצועים</u>

נשתמש במדד ביצועים כדי למדוד את מידת ההצלחה של המסווג שלנו. נעריך את שגיאת המסווג באמצעות סדרת הבוחן (test set) $\{x_k\}_{k=1}^{m}$:

$$Err(f)=\frac{1}{m}\sum_{k=1}^{m}\ell\left(f(x_k),y_k\right)$$

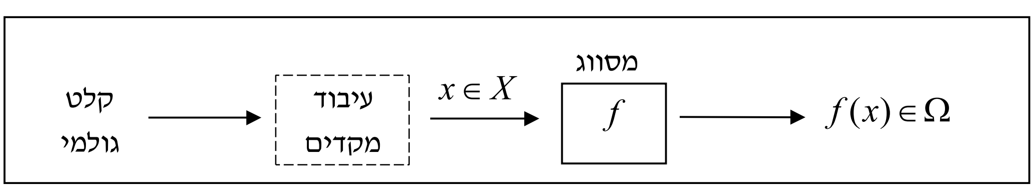
כאשר ℓ הינה פונקציית השגיאה : $\ell\left(f(x_k),y_k\right)=\begin{cases}0 & f(x_k)=y_k\\ 1 & f(x_k)\neq y_k\end{cases}$.

שגיאת המסווג הינה השגיאה האמפירית על סדרת הבוחן.

4. <u>תהליך התכן של המסווג</u>

אלגוריתם הסיווג בדרך כלל אינו מופעל על הקלט הגולמי. קלט זה יעבור שלבים מוקדמים של עיבוד מקדים לפני למידת המסווג והפעלתו.

באופן סכמתי, נוסיף את השלב של העיבוד המקדים לתכן המסווג :



בדוגמת הדגים לעיל, חלק מהעיבוד המקדים היה שלב מיצוי המאפיינים, אורך ובהירות הדג, מתוך התמונות. שלב העיבוד המקדים נעשה בהתאם לאופי הקלט הגולמי ולסוג אלגוריתם הלמידה (אופי המסווג).

עיבוד מקדים של קבוצת הלימוד חיוני לטובת:
1. התאמת הקלט למודל הלמידה. כלומר, ישנם סוגי מסווגים המתאים לקלט מסוג מסוים.
2. האצת שלב הלמידה או האימון של המסווג.
3. שיפור רמת הביצועים של המסווג.

5. <u>בעיית הסיווג במעבדה זו: סיווג מסמכי דואר אלקטרוני</u>

בעיה זו שייכת לקבוצה של בעיות למידה העוסקות בעיבוד שפה טבעית (Natural Language Processing-NLP). בבעיות מסוג זה יש לדון בשאלת ייצוג השפה טבעית. אחת מצורות הייצוג האפשריות היא ייצוג של אוסף מילים (bag of words), כלומר שמירת תוכן המסמך בעזרת שמירת כמות ההופעות של המילים השונות. כמובן שיש אובדן של אינפורמציה בייצוג זה שכן איננו שומרים על סדר המילים. למרות חסרון זה, צורת ייצוג זו מאוד נפוצה ובה נשתמש במעבדה זו. בהינתן אוסף של מסמכים (פריטי דוא״ל במקרה שלנו) תחילה יש למצוא את אוסף המילים שהופיעו לפחות פעם אחת לפחות במסמך אחד. בצורה זו אנו יוצרים מילון של מילים המייצג את המסמכים שלנו. נניח כי יש d מילים במילון זה. לאחר קביעת המילון, כל מסמך מיוצג כווקטור x באורך d. האיבר ה- x_i מכיל את מספר ההופעות של המילה ה-i במסמך.
את אוסף המסמכים נייצג במטריצה דו-מימדית כאשר כל שורה מייצגת מסמך (דגימה) וכל עמודה מילה (מאפיין). נציין כי המטריצה המתקבלת היא מאוד דלילה (sparse), כלומר בעלת הרבה מקומות עם הערך אפס. הסיבה לכך היא שבעוד השפה מאוד עשירה ומכילה מילים רבות, רק חלקן הקטן מופיע בכל מסמך.

עיבוד מקדים:

- בחירת מאפיינים ליצירת המילון - עיבוד מקדים על המסמכים כדי לנסות להקטין את כמות המילים במילון. לדוגמא, ניתן לעשות זאת ע״י העברת סף על כמות הפעמים שמופיע המילה בסט האימון, והשארת מאפיינים ששכיחותם גבוהה מערך סף הנקבע מראש. עם זאת יש לציין שהיות והייצוג דליל, רוב המילים מופיעות מספר קטן של פעמים, אך אין זה אומר שהופעתן חסרת משמעות. על-כן בחירת הסף הינה משימה לא טריוויאלית.
- יצירת מאפיינים (ייצוג וקטור הקלט) - ניתן לבחור מאפיינים אחרים מהייצוג של מספר הפעמים בו הופיעה המילה במסמך (Term count) . לדוגמא:

1. ייצוג בינארי – עבור כל מילה ומסמך לציין באמצעות מספר בינארי אם המילה הופיעה במסמך.
2. TFIDF - Term frequency Inverse document frequency: ייצוג מאוד נפוץ עבור קלט של מסמכים המחושב באופן הבא:
   - Term frequency- מספר הפעמים שהמלה מופיעה במסמך, מנורמל באורך המסמך; נספור את n_{ji} (מספר הפעמים שמילה i מופיע במסמך j) ונחשב:
   $$tf_{ji} = \frac{n_{ji}}{\sum_k n_{jk}}$$
   - Inverse document frequency – יחס הופעת המאפיין במסמך:
   $$idf_i = \log\left(\frac{|D|}{|d_i|}\right)$$
   כאשר $|D|$ הינו סך מספר המסמכים, ו-$|d_i|$ סך המסמכים בהם מופיעה המילה ה-i.
   - הערך הסופי של המאפיין TFIDF נקבע ע"י המכפלה: $tfidf_{ji} = tf_{ji} \times idf_i$

   סכמה זו מעלה את הערך של מילים שכיחות במסמך אך יותר נדירות בכלל המסמכים. הסכמה מתבססת על ההנחה שמילים מאוד נפוצות, הנמצאות בכל המסמכים פחות חשובות לביצוע הסיווג.

- עיבוד של סדרת הלימוד וסדרת הבוחן – חשוב לזכור כי את ההחלטות עבור קביעת המילון ועיבודו המקדים יש לבצע על סדרת האימון. כלומר, יש לחשוב על סדרת הבוחן במקרה הנ"ל כעל אוסף מסמכים שהגיעו לידיך רק לאחר תכנון המסווג ולכן אינה יכולה לעזור בקבלת החלטות אלו. יחד עם זאת, אם הוחלט על עיבוד מקדים כלשהו (לדוגמא הורדת מילים מסוימות) יש לזכור להפעיל את אותו עיבוד על סדרת הבוחן לפני הפעלת המסווג.

אלגוריתמים

1. <u>סיווג בייסיאני נאיבי אמפירי (Naïve Bayes)</u>

תחילה נסביר מהו *סיווג בייסיאני* , לאחר מכן מהו *סיווג בייסיאני אמפירי*, ולבסוף מהו *סיווג בייסיאני נאיבי אמפירי* אותו נבצע במעבדה זו.

<u>סיווג בייסיאני</u>

בסיווג מסוג זה אנו מניחים כי ידוע לנו המבנה ההסתברותי של הבעיה. בפרט נניח כי אנו יודעים את הפילוגים על מרחב הקלט והפלט:

- *פילוג ההסתברות* $P(\omega)$ על מרחב Ω המחלקות האפשריות. זהו הפילוג האפריורי של המחלקות, ומציין את שכיחות המחלקות (ללא תלות בקלט). נקרא prior.

- *פילוג ההסתברות המותנה* $p(x\,|\,\omega)$ של כל מחלקה $\omega\in\Omega$ על מרחב הקלט. פילוג זה נקרא גם פונקציית הסבירות (likelihood function), ומציין את הסבירות של קלט בהינתן מחלקה מסוימת.

נשים לב, שביחד הפילוגים הנ"ל מגדירים באופן מלא את הפילוג המשותף $p(x,\omega)=p(x\,|\,\omega)P(\omega)$.

כדי לקבל את המסווג הבייסיאני נשתמש בחוק בייס (Bayes):

$$p(\omega\,|\,x)=\frac{p(x\,|\,\omega)P(\omega)}{p(x)} \qquad (1)$$

ההסתברות $p(\omega\,|\,x)$ מתארת את ההסתברות של מחלקה ω לאחר שראינו את הקלט x, ועל-כן מכונה ההסתברות בדיעבד (a-posteriori). נרצה לבחור את המחלקה עם ההסתברות הגבוהה ביותר לקלט הנתון ולכן מסווג בייס הוא:

$$f_{Bayes}(x)=\arg\max_{\omega\in\Omega}\, p(\omega\,|\,x) \qquad (2)$$

כלומר, על-פי מסווג בייס יש למצוא את המחלקה הממקסמת את הביטוי.
נפשט את הבעיה ע"י הצבה של חוק בייס (1):

$$f_{Bayes}(x)=\arg\max_{\omega\in\Omega}\frac{p(x\,|\,\omega)P(\omega)}{p(x)}=\arg\max_{\omega\in\Omega}\, p(x\,|\,\omega)P(\omega)$$

כאשר במעבר האחרון השמטנו את התלות ב- $p(x)$ שכן ערך זה אינו תלוי במחלקה.
כאמור, נניח כי יש מידע הסתברותי מלא על ההתפלגויות $P(\omega)$ ו- $P(x\,|\,\omega)$.

נציין כי מסווג בייס ידוע כמסווג האופטימאלי, שכן ניתן להראות כי הוא ממזער את הסתברות השגיאה הממוצעת. בזכות תכונה זו, נרצה להשתמש במסווג זה גם כאשר אין לנו מידע מלא על ההתפלגויות. לשם כך נציג כעת את המסווג הבייסיאני האמפירי. מסווג זה נקרא גם MAP: Maximum A-Posteriori.

<u>מסווג בייסיאני אמפירי</u>

נרצה להשתמש באותו עקרון של מסווג בייסיאני אך ללא הנחת ידיעה של הפילוגים הדרושים. במקום הנחה זו נעזר בסט דוגמאות. הלמידה תעשה בשני שלבים:

א. שערוך של הפילוגים $\hat{P}(\omega)$ ו- $\hat{P}(x\,|\,\omega)$ בעזרת הדוגמאות $\{x_k,y_k\}_{k=1}^{n}$.

ב. הפעלת מסווג בייס על הפילוגים שהתקבלו בשלב הראשון.

הערכת הפילוגים הנדרשים:

- שערוך ההתפלגות האפריורית $\hat{P}(\omega)$: נבצע שערוך זה על פי השכיחות היחסית של הדוגמאות במחלקות השונות: $\hat{P}(\omega)=\frac{1}{n}\sum_{k=1}^{n}1\{y_k=\omega\}$.
- שערוך הסבירות $\hat{P}(x\mid\omega)$: ישנן מספר שיטות לעשות הערכת פילוגי הסתברות. במעבדה זו ההערכה תעשה בעזרת משערך הסבירות המרבית (Maximum Likelihood Estimator -MLE).

משערך הסבירות המרבית (MLE): זוהי גישה לשערוך פרמטרים של התפלגות בהינתן מדגם של דגימות ממנה.

בהינתן סדרת דוגמאות $\left\{x_k\right\}_{k=1}^{n}$, נניח מודל פרמטרי של פונקציית הסבירות:

$$Lik(\theta)=p(x_1,x_2,...,x_n\mid\theta)$$

זוהי ההסתברות לראות את סדרת הדוגמאות כפונקציה של וקטור פרמטרים θ.

משערך הסבירות המרבית של θ- הינו הערך של θ הממקסם את $Lik(\theta)$. כלומר, זהו הערך עבורו סדרת הדוגמאות היא "הסבירה ביותר". ניתן לכתוב זאת בצורה הבאה:

$$\hat{\theta}_{MLE}=\arg\max_{\theta\in\Theta} p\left(x_1,x_2,...,x_n\mid\theta\right)$$

כאשר Θ מסמן את סט הערכים ש-θ יכול לקבל.

הפרמטרים שנשערך תלויים בהנחת פילוג מסוים. לדוגמא, עבור פילוג גאוסי, הפרמטרים θ שיש לשערך הם הממוצע והשונות: $\theta=[\mu,\Sigma]$.
נציין כי באופן כללי הערכת פילוג רב מימדי אינה בעיה פשוטה ולכן נשתמש בהנחת "נאיביות" אשר תוצג מיד.

<u>מסווג בייסיאני נאיבי אמפירי</u>

הבעייתיות בגישה של המסווג הבייסיאני האמפירי נמצאת בהערכת הפילוג $p(x\mid\omega)$, עבור קלט x רב מימדי. ניתן לפשט בעיה זו ע"י הנחת אי-תלות בין הרכיבים $x=(x^1,x^2,\ldots,x^d)$, שכן תחת הנחה זו:

$$p(x\mid\omega)\approx p(x^1\mid\omega)p(x^2\mid\omega)\cdots p(x^d\mid\omega)$$

ההנחה מביאה לקירוב של ההתפלגות לצורך השערוך, ושוויון מתקבל רק אם באמת קיימת אי תלות בין הרכיבים. כעת, יש לשערך את הפילוגים השוליים (החד-מימדיים) $\left\{p(x^i\mid\omega)\right\}_{i=1}^{d}$ על סמך סדרת הלימוד $\left\{x_k\right\}_{k=1}^{n}$.

הערה: הפעלת ההנחה ביחס לסיווג מסמכים ופריטי דוא"ל -
כפי שנראה בהמשך, הפעלת ההנחה הנאיבית של אי תלות בהינתן המחלקה מניבה תוצאות סיווג סבירות בהחלט. הסיבה לכך היא שעבור מסמכים הנחה זו אינה חסרת בסיס (באופן אינטואיטיבי), שכן בהינתן המחלקה יתכן שיש לנו אינפורמציה מספקת לידיעת שאר המילים שתמצאנה במסמך.

נציג כעת את משערכי הסבירות המרבית של $p(x \mid \omega)$ עבור ההתפלגויות שנבחן בניסוי.

התפלגות גאוסית

נניח כי לכל מחלקה $\omega \in \Omega$ הפילוג $p(x \mid \omega)$ ניתן לקירוב באמצעות פילוג גאוסי חד ממדי- $p(x \mid \omega) \sim N(\mu_\omega, \Sigma_\omega)$. כמו כן, נניח אי-תלות בין הדוגמאות. נעריך את הממוצע μ_ω והשונות σ_ω^2 באמצעות משערך ה-MLE. נסמן ב- $\{z_k\}_{k=1}^{n(\omega)}$ את תת-הסדרה של $\{x_k\}_{k=1}^{n}$ שעבורה $y_k = \omega$.

נחשב את הסבירות עבור הפילוג השולי $p(x^i \mid \omega)$:

$$Lik(\theta_i) = p(z_1^i, z_2^i, ..., z_n^i \mid \theta_i) = \prod_{k=1}^{n(\omega)} \frac{1}{\sqrt{2\Pi\sigma_i^2}} \exp\left(-\frac{1}{2}\left(\frac{z_k^i - \mu_i}{\sigma_i}\right)^2 \right)$$

כאשר במעבר הראשון הכנסנו את האי-תלות בין הדגימות ואת הנחת הגאוסיות.

על-מנת לפשט את החישוב נפעיל פונקציית log על הסבירות:

$$\log Lik(\theta_i) = -n \log\sqrt{2\Pi} - n \log \sigma_i - \frac{1}{2\sigma_i^2} \sum_{k=1}^{n(\omega)} \left(z_k^i - \mu_i\right)^2$$

כעת, יש להביא למקסימום את הביטוי ועל כן נגזור לפי המשתנים:

$$\frac{\partial \log Lik(\theta_i)}{\partial \mu_i} = \frac{1}{\sigma_i^2} \sum_{k=1}^{n(\omega)} \left(z_k^i - \mu_i\right)$$

$$\frac{\partial \log Lik(\theta_i)}{\partial \sigma_i} = -\frac{n(\omega)}{\sigma_i} - \frac{1}{\sigma_i^3} \sum_{k=1}^{n(\omega)} \left(z_k^i - \mu_i\right)^2$$

מהשוואת ביטויים אלו לאפס נקבל את הערכים המוכרים:

$$\hat{\mu}_{\omega(i)} = \frac{1}{n(\omega)} \sum_{k=1}^{n(\omega)} z_k^i$$

$$\hat{\sigma}_{\omega(i)}^2 = \frac{1}{n(\omega)} \sum_{k=1}^{n(\omega)} (z_k^i - \hat{\mu}_i)^2$$

כאשר $\hat{\mu}_{\omega(i)}$, $\hat{\sigma}^2_{\omega(i)}$ מסמנים את הממוצע והשונות של ההתפלגות השולית של האיבר ה-i עבור מחלקה ω.

התפלגות ברנולי

נניח כי לכל מחלקה $\omega \in \Omega$ הפילוג השולי $p(x^i \mid \omega)$ של כל קואורדינטה i (מילה) ניתן לקירוב באמצעות התפלגות ברנולי. נזכיר כי התפלגות זו יכולה לקבל שני ערכים בלבד $x^i \in \{0,1\}$. כפועל יוצא, כאשר מניחים התפלגות זו יש לדאוג לבינאריזציה של המאפיינים. בפרט עבור מסמכים, יש לשנות את הייצוג כך שכל איבר ייצג הופעה או אי הופעה של המילה. כמובן שבייצוג מסוג זה יש אובדן של אינפורמציה בנוגע למספר ההופעות של המילה.

יש לשערך את הערך $p(x^i=1 \mid \omega)=\theta$ (את הערך $p(x^i=0 \mid \omega)$ כבר נמצא מההסתברות המשלימה $p(x^i=0 \mid \omega)=1-p(x^i=1 \mid \omega)$).

נסמן ב- $\{z_k\}_{k=1}^{n(\omega)}$ את תת-הסדרות של $\{x_k\}_{k=1}^{n}$ שעבורן $y_k=\omega$, ונניח כי הדוגמאות אינן תלויות.

נחשב את הסבירות עבור הפילוג השולי $p(x^i \mid \omega)$:

$$Lik(\theta_i)=p(z_1^i,z_2^i,...,z_n^i \mid \theta_i)=\prod_{k=1}^{n(\omega)} p(z_k^i \mid \theta_i)=\prod_{k=1}^{n(\omega)} \theta_i^{z_k^i}(1-\theta_i)^{1-z_k^i}$$

גם כאן נפעיל פונקציית log על מנת לפשט את החישוב :

$$\log Lik(\theta_i)=\sum_{k=1}^{n(\omega)} z_k^i \log\theta_i+(1-z_k^i)\log(1-\theta_i)$$

מגזירה והשוואה לאפס נקבל כי :

$$\widehat{\theta^i}=\frac{1}{n(\omega)}\sum_{k=1}^{n(\omega)} z_k^i$$

כלומר, החלק היחסי של הדגימות ששוות ל-1 מתוך סך הדגימות במחלקה.

התפלגות מולטינומית

נניח כי לכל מחלקה $\omega \in \Omega$ הפילוג $p(x \mid \omega)$ ניתן לקירוב באמצעות התפלגות מולטינומית :

$$p(x \mid \theta)=\frac{m!}{\prod_i x^i!}\prod_{i=1}^{d}\theta_i^{x^i}$$

כאשר m הינו סכום המאפיינים: $m = \sum_{i=1}^{d} x^i$.

ניתן לפרש את ההתפלגות כמקרה שבו יש סדרה של m הגרלות זהות ובלתי תלויות עם d תוצאות אפשריות בכל הגרלה. θ_i היא ההסתברות לתוצאה i בהגרלה כלשהי.

x הוא וקטור באורך d, כאשר הקואורדינטה x^i סופרת את מספר ההופעות של תוצאה i.
יש לדרוש ש- θ_i יהיה אי-שלילי ושסך כל ההסתברויות יהיה $\sum_i \theta_i = 1$. התפלגות זו הינה הרחבה של התפלגות בינומית למקרה של יותר משתי תוצאות אפשריות בהגרלה.

נחשב את משערך הסבירות המרבית של וקטור הפרמטרים θ בנפרד לכל מחלקה ω.
נסמן ב- $\{z_k\}_{k=1}^{n(\omega)}$ את תת-הסדרה של $\{x_k\}_{k=1}^{n}$ שעבורן $y_k = \omega$.
משערך הסבירות המרבית של כל איבר בווקטור הפרמטרים הוא:

$$\hat{\theta}_i = \frac{\sum_{k=1}^{n(\omega)} z_k^i}{T}$$

כאשר המונה הוא סך הערכים של המאפיין i בדוגמאות סדרת הלימוד עבורן $y_k = \omega$.

המכנה $T = \sum_{i=1}^{d} \sum_{k=1}^{n(\omega)} z_k^i$ הוא סך ערכי כל המאפיינים בדוגמאות סדרת הלימוד עבורן $y_k = \omega$.
כלומר, נשערך באופן אמפירי את ההסתברות להופעת התוצאה i על פי השכיחות היחסית שלה בדוגמאות המתאימות בסדרת האימון.

החלקה של ההתפלגות המולטינומית

נשים לב שעבור המקרה בו $\theta_i = 0$ מתקבל, מהצבה בנוסחה של ההתפלגות המולטינומית, שכל דגימה חדשה (מסמך במקרה שלנו) עם $x^i > 0$ היא בעלת הסתברות אפס, וזאת ללא קשר לשאר המאפיינים (המילים) בדגימה. מסיבה זו נרצה להימנע ממצבים של הסתברות השווה לאפס, ונדרוש $\theta_i > 0$ לכל i. ניתן לעשות זאת באמצעות החלקה (smoothing) עם משתנה c כלהלן:

$$\hat{\theta}_i = \frac{1}{T'}\left(c + \sum_{k=1}^{n(\omega)} z_k^i\right)$$

באופן אינטואיטיבי, המשתנה c הוא מספר ההופעות המינימלי של מאפיין (מילה) בין אם הופיעה בסדרת האימון ובין אם לא. T' הוא עדכון של הנרמול וערכו $T' = dc + T$.

קריאה נוספת

[1] *Pattern Classification* (2nd ed.), Richard O. Duda, Peter E. Hart and David G. Stork (John Wiley and Sons, 2001)

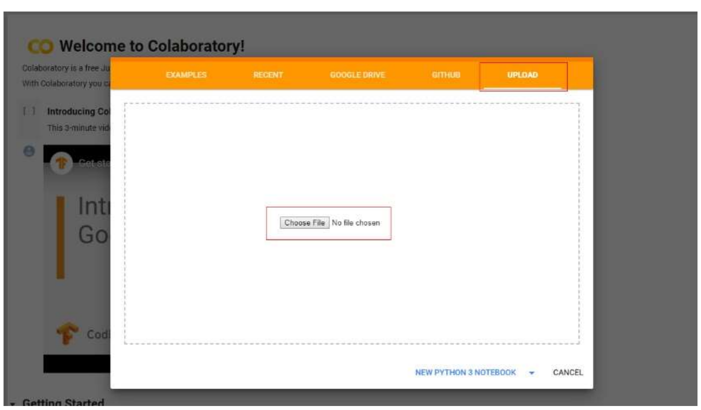
היכרות עם Python וספריות העבודה: NumPy, Pandas, Matplotlib, Scikit-Learn

על מנת להגיע מוכנים למעבדה, עליכם לבצע היכרות עם שפת פייתון והספריות שנעבוד איתן. שפת פייתון היא השפה המובילה כיום בתחום של מערכות לומדות ואיתה מבצעים גם Deep Learning, לכן, חשוב שתלמדו לעבוד איתה (היא מאד פשוטה!). עליכם להגיש את קובץ המחברת עם הקוד שהשלמתם (שם הקובץ: **ml_preparation_part_1a.ipynb**).

1. הרצת המחברת (באופן מקומי או אונליין)
   a. **הרצת המחברת אונליין באמצעות Google Colab:** ניתן להריץ את מחברת הפייתון בצורה מקוונת על הפלטפורמה של Google הנקראת Google Colab. הפלטפורמה מספקת **בחינם** שירותי ענן להרצת קוד (המיועד ל- Machine Learning) עם חומרה חזקה למדי. שימו לב כי יש מגבלת שימוש של 12 שעות (אבל אתם לא אמורים לבלות במחברת למעלה מ-30 דקי'). השימוש ב-Colab מחייב חשבון Google, לכן וודאו כי יש לכם אחד.

      הוראות הרצה:

      - כנסו ל- https://colab.research.google.com
      - בסרגל העליון בחרו ב- Upload ובחרו בקובץ ההכנה - **ml_preparation_part_1a.ipynb**

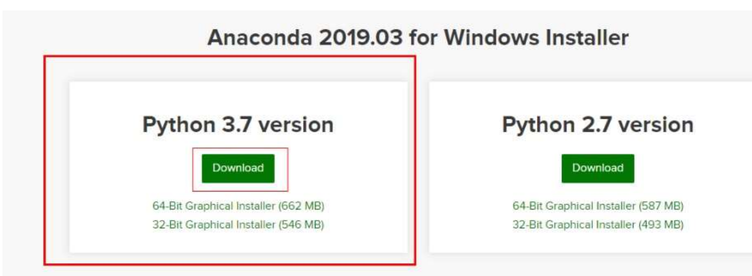


- כעת תיפתח המחברת. הוסיפו את קבצי העזר הדרושים להרצת המחברת כך: לחצו על סמל התיקייה בצד שמאל, לחצו על Upload, נווטו אל הקבצים הדרושים ובחרו אותם.
- כעת ניתן להריץ את הקוד (כדי להריץ בלוק לחצו Ctrl + Enter).
- אחרי שסיימתם את התרגילים, שמרו את המחברת (Ctrl+S) והורידו אותה למחשבכם. הגישו אותה יחד עם השאלות לדו"ח ההכנה. **שימו לב שהשינויים שתבצעו במחברת לא ישמרו בקובץ המחברת המקומי שטענתם! לכן בסיום העבודה יש להוריד את המחברת שעבדתם עלייה אונליין על ידי File->Download .pynb ולהגיש את הקובץ הנ"ל.**

b. **<u>הרצת המחברת באופן מקומי עם Anaconda:</u>** כדי להריץ את המחברות באופן לוקלי, בדומה לצורה שתעבדו איתה במעבדה, יש להתקין את סביבת Anaconda על המחשב, המספקת כלים מצוינים כגון Jupyter Notebook לעבודה עם פייתון. דרך זו מומלצת כיוון שגם בקורסים המועברים בטכניון ובצורה מקוונת, נהוג לעבוד עם המחברות וכדאי שתוכלו להריץ אותן על המחשב האישי. **הערה חשובה: יש לוודא כי דפדפן ברירת המחדל הוא Chrome או Firefox (ולא Internet Explorer).**

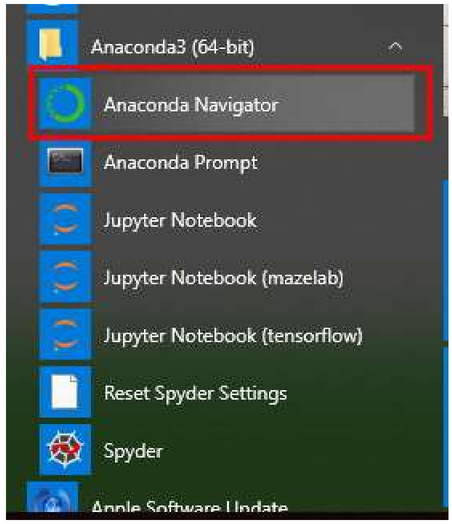
<u>הוראות התקנה והרצה:</u>

- הורידו את חבילת Anaconda בהתאם למערכת ההפעלה שלכם בכתובת - https://www.anaconda.com/distribution/

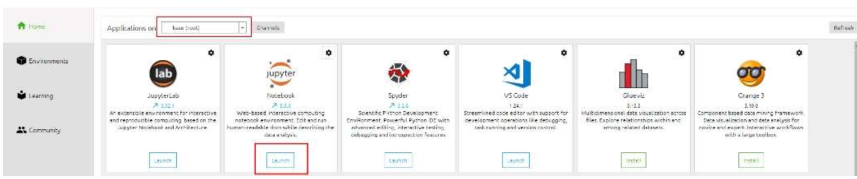


יש לוודא כי אתם מורידים את גרסת **Python 3.7!**

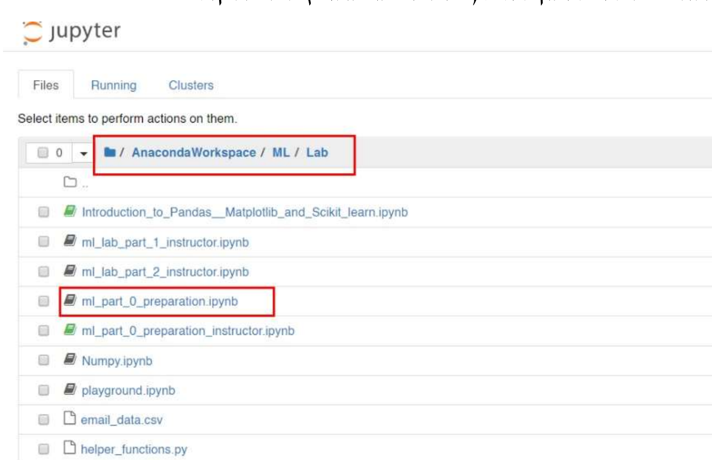
- לאחר ההתקנה, מצאו את Anaconda Navigator מהתפריט והפעילו אותו.



- בחלון שיפתח יש לוודא כי סביבת העבודה היא base, ולאחר מכן להפעיל את Jupyter Notebook.



- כעת יפתח הדפדפן ותוכלו לנווט אל הקובץ של המחברת. לחצו עליו והוא יפתח בחלון חדש, בו תוכלו להריץ את הקוד.



- ודאו שקבצי העזר הדרושים להרצת המחברת נמצאים באותה תיקייה עם קובץ המחברת.

2. טיפים שימושיים
   a. כדי להריץ בלוק של קוד, בחרו אותו עם העכבר ולחצו Ctrl + ENTER. כדי להריץ ולעבור לבלוק הבא לחצו Shift + ENTER.
   b. כדי לקבל מידע על פונקציה, צרו קוד בלוק חדש (Cell חדש) והריצו help(name_of_function/class/module).
   c. כדי להציג שורות בבלוק (כדי לדעת להכווין למספר שורה של קוד מסוים), בחרו אותו עם העכבר, לחצו ESC ולאחר מכן L (לא ביחד).
   d. כדי להציג מידע על פונקציה תוך כדי כתיבת קוד, כתבו את שם הפונקציה ולחצו Shift + TAB.
   e. כדי להוסיף debug breakpoint במקום כלשהו בקוד הוסיפו את שורת הקוד הבאה:

```
import pdb;pdb.set_trace()
```

הרצת התוכנית תעצור בשורה זו, לאחר מכן אפשר להריץ פקודות (כמו הדפסת משתנים) או להתקדם לשורה הבאה עם הפקודה c .

מפגש ראשון

שאלות הכנה

עליכם להגיש 3 קבצים – קובץ עם תשובות לשאלות ההכנה ושני קבצי מחברות הפייתון (ראו למטה) עם הקוד שהשלמתם.

0. לפני שאתם מתחילים, ודאו שאתם מכירים סינטקס בסיסי של שפת פייתון (כולל מבני הנתונים List, String ו- Dictionary).
   מקורות לדוגמא:
   a. https://docs.python.org/3/tutorial/introduction.html
   b. https://docs.python.org/3/tutorial/controlflow.html
   c. https://docs.python.org/3/tutorial/datastructures.html.

1. עליכם להגיש את קובץ המחברת עם הקוד שהשלמתם **ml_preparation_part_1a.ipynb**, כפי שתואר בחלק הקודם.

2. נתונה בעיית הסיווג הבאה: על סמך רשומות מספריות נתונות, יש להפריד בין נשים וגברים. המאפיינים המרכיבים את הרשומות הם:
   a. גובה
   b. משקל
   c. צבע שיער
   d. צבע עיניים
   e. אורך שיער
   f. מידת נעליים
   g. מספר טלפון

   א. באילו מהמאפיינים הייתם בוחרים על מנת לבנות מסווג? האם יש סיבה לא להשתמש בכל המאפיינים?
   ב. רשמו את הבעיה בצורה פורמלית: מהו מרחב הקלט? מהו מרחב הפלט?

3. בשאלה זו נבחן שיטות שונות לעיבוד מקדים בעזרת הדוגמא של קטעי הטקסט הבאים:

Here is Edward Bear, coming downstairs now, bump, bump, bump, on the back of his head.

Sometimes Winnie-the-Pooh likes a game of some sort when he comes downstairs.

Winnie-the-Pooh sat down at the foot of the tree, put his head between his paws and began to think.

   א. פתחו את מחברת **ml_preparation_part_1b.ipynb**
   ב. השלימו את הקוד ליצירת מטריצת הייצוג.
   ג. השלימו את הקוד עבור הפעלת עיבוד מקדים בצורת סף: השאירו רק את המאפיינים שמופיעים שלוש פעמים או יותר בכל המסמכים יחד.
   ד. השלימו את הקוד עבור הפעלת עיבוד מקדים דמוי tf-idf: השאירו רק את המאפיינים שמופיעים פעמיים או יותר בכל המסמכים, אולם אינם מופיעים בכל אחד מהמסמכים.
   ה. מי מהשיטות מתאימה יותר למיצוי המאפיינים הרלוונטיים מתוך הטקסט? נמקו.
   ו. הציעו שיטה נוספות של עיבוד מקדים המתאימה לקטעים הנתונים.
   ז. בסיום התרגיל זכרו לצרף לדו"ח המכין את המחברת **ml_preparation_part_1b.ipynb** .

4. השאלה עוסקת בסיווג בייסיאני לפי מאפיין בודד. נתונות שלוש מחלקות בעלות פילוג אפריורי ידוע:

$$P(\omega_1) = 0.5, P(\omega_2) = P(\omega_3) = 0.25$$

פונקציות הסבירות ידועות אף הן:

$$p(x|\omega_1) \sim N(0,1)$$
$$p(x|\omega_2) \sim N(0.5,1)$$
$$p(x|\omega_3) \sim N(1,1)$$

א. לאיזו מחלקה תסווג הדגימה $x = 0.6$?

ב. בהינתן סדרת הדגימות $x = 0.6, 0.1, 0.9$, ובהינתן ששלוש הדגימות שייכות לאותה מחלקה, מיהי המחלקה הסבירה ביותר?

5. יהי x משתנה אקראי מפולג אחיד עם פרמטר θ:

$$p(x|\theta) = \begin{cases} 1/\theta, & 0 \le x \le \theta \\ 0, & otherwise. \end{cases}$$

נניח כי נדגמות n דגימות בלתי תלויות $D = \{x_1, x_2, \dots, x_n\}$ מהפילוג הנתון. הראו כי משערך הסבירות המרבית עבור הפרמטר θ במקרה זה הוא $\max[D]$, כלומר – ערך האיבר המקסימלי ב-D.

6. שערוך הפרמטרים שנדון בפרק המבוא נוגע למצב בו מתקבלות כל הדגימות בבת אחת, והשערוך מתבצע על סמך כולן יחד. סוג כזה של טיפול בנתונים קרוי Batch (אצווה). לעומת זאת, קיימים מצבים רבים בהם הדגימות מתקבלות אחת אחת, בהפרשי זמן מסוימים. במקרים כאלה מתעדכן שערוך הפרמטר עם כל דגימה, באופן סדרתי. טיפול זה קרוי Online. פתחו ביטויים לשערוך אמפירי נאיבי של התוחלת של פילוג כלשהו מתוך דגימות $D = \{x_1, x_2, \dots, x_n\}$ באופן סדרתי.

הבהרות:

א. לצורך שערוך אמפירי של התוחלת, השתמשו בממוצע הדגימות.

ב. הביטוי הנדרש הוא מהצורה $\hat{\mu}_{i+1} = f(\hat{\mu}_i, x_{i+1})$.

מהלך הניסוי

שאלות הדו״ח המסכם מופיעות לאורך הניסוי. יש לענות עליהן בכיתה, במהלך ביצוע הניסוי. בנוסף, עליכם לצרף לדו״ח המסכם את כל קבצי הקוד (המחברות Python) שיצרתם במהלך הניסוי.

1. <u>סביבת העבודה</u>

במהלך הניסוי תעבדו בסביבת Anaconda Jupyter Notebook בשפת Python.

יש שתי אפשרויות עבודה: סביבה מקומית או מקוונת עם Colab.

הנחיות להרצת סביבה מקומית:

- נא לוודא כי Google Chrome מוגדר כדפדפן ברירת מחדל.
- הפעילו את תכנת ה-Anaconda Navigator מתפריט ה-Start. עקבו אחר ההוראות בחלק של "הרצת המחברות".
- נווטו אל הקבצים של המחברות, נקראים: ml_lab_part_1.ipynb, ml_lab_part_2.ipynb בהתאם לחלק הניסוי עליו אתם עובדים.

הנחיות להרצת סביבה מקוונת:

- היכנסו עם חשבון המשתמש שלכם ל - https://colab.research.google.com/
- לחצו על Upload ונווטו אל הקבצים של המחברות, נקראים: ml_lab_part_1.ipynb, ml_lab_part_2.ipynb בהתאם לחלק הניסוי עליו אתם עובדים.
- לחצו על סמל התיקייה בצד שמאל, לחצו על Upload ונווטו אל הקבצים email_data.csv ו- helper_functions.py ובחרו אותם.
- **שימו לב שהשינויים שתבצעו במחברת לא ישמרו בקובץ המחברת המקומי שטענתם! לכן בסיום העבודה יש להוריד את המחברת שעבדתם עלייה אונליין על ידי File->Download .pynb ולהגיש את הקובץ הנ"ל עם הדו"ח המסכם.**

2. <u>היכרות עם מסד הנתונים</u>

כאמור בפרק המבוא, ניסוי מעבדה זה עוסק בסיווג מסמכי דואר אלקטרוני.

מסד הנתונים העומד לרשותכם מורכב מ-3,502 מיילים, המכילים את כתובות הנמענים (To), כתובת השולח (From), נושא הדוא"ל (Subject) ותוכנו (Content). לכל מייל מצורף גם תיוג לאחת משתי מחלקות – ספאם (S) ואמיתי (H). באנגלית משימה זו נקראת Spam/Ham Classification. המיילים הם אמיתיים ונלקחו מ-Spam Assassin מתוך מטרה ללמוד אילו מילים מוכלים בדואר זבל, האם כשיש הרבה קישורים (לינקים) מדובר בדואר זבל וכו'...

- טענו את מסד הנתונים באמצעות הפקודה email_data = pd.read_csv('./email_data.csv') ולאחר מכן כתבו email_data.sample(15) והריצו את הבלוק. התבוננו ב15 דגימות מהמסד. מבנה הטבלה:

5	4	3	2	1
תיוג המסמך Label	תוכן הדוא"ל Content	נושא הדוא"ל Subject	כתובת השולח From	כתובות הנמענים To

בשלב זה שמור המידע במחרוזות. כפי שהוסבר בפרק המבוא, עלינו להמיר את המידע לפורמט המאפשר למידה וסיווג, כלומר – לייצג אותו באופן מספרי. במעבדה זו נעבוד עם תוכן (Content) המייל המכיל בתוכו את הנושא בנוסף.

3. <u>ייצוג המידע</u>

בעיות למידה בתחום של עיבוד שפה טבעית מציבות אתגר כבר בשלב ייצוג המידע. בסעיף זה תבחנו מספר שיטות לייצוג המידע העומד לרשותכם; מיותר לציין כי לאופן הייצוג השפעה משמעותית על ביצועי המערכת.

- כעת נעביר את המידע מספר טרנספורמציות כדי להפוך אותו לצורה מספרית שנוכל לעבוד איתה. פתחו את הקובץ helper_functions.py (באמצעות הסייר ב-Anaconda או ע"י Text Editor) ועיינו בתיעוד של הטרנספורמציות: EmailToWords ו-WordCountToVector. האובייקט email_pipeline מאפשר לנו להעביר את המידע דרך 2 הטרנספורמציות ברצף. הקריאה מתבצעת באופן הבא:

```
# transform the data
X_sample_augmented = email_pipeline.fit_transform(X_sample)
```

שימו לב כי המשתנים הם משתנים דלילים (sparse), שערכם אפס ברב הכניסות, ולכן הם אינם מיוצגים כמשתנים רגילים.

```
<class 'scipy.sparse.csr.csr_matrix'>
  (0, 1)	2
  (0, 6)	1
  (0, 10)	1
  (0, 16)	1
  (0, 18)	1
  (0, 19)	1
  (0, 32)	1
  (0, 35)	1
  (0, 50)	1
  (0, 82)	1
  (0, 83)	1
  (0, 84)	1
  (0, 113)	1
  (0, 114)	1
  (0, 186)	1
  (0, 187)	1
  (0, 361)	1
  (0, 362)	1
  (0, 363)	1
  (0, 364)	1
  (0, 365)	1
  (0, 366)	1
  (0, 367)	1
  (0, 368)	1
  (0, 369)	1
```

איור 7

המספרים המסומנים בצהוב באיור 7 מצביעים על כך שבשורה ה-0, בעמודה הראשונה, ערכו של המשתנה הוא 2. תרגום משתנה דליל למשתנה "רגיל", מתבצע באמצעות הפקודה ()todense, למשל: ()X_augmented.todense.

- **דו"ח מסכם – שאלה 1:** הסבירו בקצרה מהו המידע שמכיל המשתנה dictionary.
  מילונים ב-Python הם מהצורה: {key: value}. מה הוא "המפתח" (key) של המילון ומה הם ערכיו (values).
- **דו"ח מסכם – שאלה 2:** הסבירו בקצרה מהו המידע שמכיל המשתנה y_sample.
- **דו"ח מסכם – שאלה 3:** הסבירו בקצרה מהו המידע שמכיל המשתנה X_sample_augmented. מה מייצגות שורותיו? מה מייצג מספר עמודותיו?

4. סדרת הלימוד וסדרת הבוחן

כפי שהוסבר בפרק המבוא, לימוד המסווג ובחינת ביצועיו מתבצעים על קבוצות (סדרות) זרות.

- ממשו פונקציה היוצרת מתוך מסד הנתונים, סדרת לימוד וסדרת בוחן.
שלד הפונקציה הוכן עבורכם ועליכם להשלים את הקוד, בבלוק המכיל את הפונקציה train_test_split. הפונקציה תקבל שלושה פרמטרים, לפי הסדר הבא: מסד הנתונים (וקטור של מחרוזות X והווקטור Y), גודל סדרת המבחן (באחוזים) מתוך כל הסט. בחירת המסמכים לשתי הסדרות תתבצע באופן אקראי, תוך שמירה על תנאי הזרות. הפונקציה תחזיר את המשתנים X_train, X_test, y_train, y_test.
רמז: השתמשו במחולל המספרים הרנדומלי: rand_gen = np.random.RandomState
והיעזרו בפונקציית ()rand_gen.permutation בשביל לערבב את האינדקסים בצורה רנדומלית.
הערה: בסעיפים הבאים חלקו את מסד הנתונים לסט אימון וסט מבחן ביחס 80% ו 20% בהתאמה, אלא אם נאמר אחרת.

- **דו"ח מסכם – שאלה 4:** הסבירו מדוע הצורה הנכונה ליצירת סדרת אימון ובוחן היא לפי מקטע קוד מספר 1 להלן, ולא לפי מקטע קוד מספר 2.

 # Option 1

```
# split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
# transform
X_train_augmented = email_pipeline.fit_transform(X_train)
X_test_augmented = email_pipeline.transform(X_test)
```

 # Option 2

```
# transform
X_augmented = email_pipeline.fit_transform(X)
# split
X_train_augmented, X_test_augmented, y_train, y_test =
train_test_split(X_augmented, y, test_size=0.2)
```

5. סיווג בייסיאני נאיבי אמפירי (Naïve Bayes)

שיטת הסיווג הראשונה בה תשתמשו היא סיווג בייסיאני נאיבי אמפירי (ר' פרק המבוא). במהלך העבודה עם מסווג זה תבחנו מספר היבטים, חלקם נוגעים למסווגים כולם וחלקם ייחודיים לסיווג הבייסיאני.

נתחיל מבניית המחלקה והפונקציות הדרושות למסווג עצמו: פונקציה המבצעת את שלב האימון ופונקציה המבצעת את שלב הסיווג. נעבוד לפי המבנה של Scikit-Learn (ספריית האלגוריתמים של מערכות לומדות), כך שלכל מסווג יש 2 פונקציות עיקריות:

1. Fit – מבצעת את שלב האימון. מקבלת את סט האימון והתיוגים שלו (X_train, y_train) ולומדת את הפרמטרים הדרושים לסיווג.
2. Predict – מבצעת סיווג. מקבלת את סט המבחן (X_test) ומבצעת תיוג על בסיס הפרמטרים שנלמדו בשלב האימון.

צורת העבודה עם האלגוריתמים במערכות לומדות היא מהצורה:

```
# create classifier
clf = MlabNaiveBayes(dist_type="gaussian", num_classes=2)
# train on train set
clf.fit(X_augmented_train, y_train)
# predict on test set
y_pred = clf.predict(X_augmented_test)
```

בחלק הבא תדרשו להשלים את מימוש הפונקציות fit ו-predict של המחלקה MlabNaiveBayes בהתאם לסוג הפילוג. צורות הפילוג האפשריות כקלט עבור הסבירות (likelihood) $\hat{P}(x\,|\,\omega)$:

"gaussian", "bernoulli", "mulinomial", "multinomial_smooth".

- **<u>מימוש פונקציית הלימוד fit.</u>** כמוסבר בפרק המבוא, על הפונקציה שלכם לשערך שני פילוגי הסתברות: ההתפלגות האפריורית, $\hat{P}(\omega)$, והסבירות, $\hat{P}(x\,|\,\omega)$. את קטע הקוד המשערך את ההתפלגות האפריורית תכתבו בעצמכם, ואת הסבירות תחשבו באמצעות קריאה לפונקציה מוכנה, estimate_likelihood_params, שתחזיר לכם את הפרמטרים הרלוונטיים לחישוב הסבירות. הקדישו מספר דק' כדי להבין איך היא עובדת.

- **<u>מימוש פונקציית החיזוי predict.</u>** הפונקציה predict תחזיר וקטור המכיל את תיוגי המסמכים עליהם החליט המסווג. עליכם להשלים את מקטעי הקוד הבאים, כפי שמוסבר בפרק המבוא:
  - השלימו את חישוב הפילוג המותנה $\hat{P}(x\,|\,\omega)$ בפונקציית העזר eval_sample_likelihood עבור הפילוג "gaussian".

    הערה: ניתן להשתמש בפונקציה np.prod המקבלת וקטור ומחזירה את מכפלת איבריו.
  - השלימו את חישוב הפילוג האפוסטריורי $\hat{P}(\omega\,|\,x)$ בפונקציה predict.
  - השלימו את כלל ההחלטה עצמו בפונקציה predict.

- השלימו את הקוד בפונקציה calc_err לחישוב שגיאת הסיווג, הנתונה על ידי החלק היחסי של הדגימות בו טעה המסווג.

- הפעילו את פונקציית הסיווג עם צורה גאוסית על סדרת הבוחן. למרבה הצער, אתם צפויים לקבל הודעת שגיאה, לפיה מסד הנתונים שבידיכם אינו מתאים לתיאור על ידי פילוג גאוסי. טכנית, הודעת השגיאה מתקבלת מפני שישנן מילים (מאפיינים) בעלות מספר קבוע של מופעים בכל המסמכים השייכים למחלקה מסויימת בסדרת האימון. בשלב שערוך הפרמטרים על סמך סדרת האימון מקבלות מילים אלה שונות אפס, ובשלב חישוב הסבירות עבור קבוצת המבחן הדבר גורם לשגיאה.

- **דו"ח מסכם – שאלה 5:** הציעו פתרון שיאפשר שימוש בסבירות גאוסית.

בשלב הבא תבדקו את התנהגות המסווג עבור צורות סבירות אחרות.

- השלימו את חישוב הפילוג המותנה $\hat{P}(x \mid \omega)$ בפונקציית העזר eval_sample_likelihood עבור הפילוג "multinomial" ו- "multinomial_smooth".
<u>הערה</u>: שימו לב שבחישוב הסבירות עצמה אין צורך לחשב פרמטרים שאינם תלויים במחלקה - האם אתם יכולים להסביר מדוע? (רמז: במה תלויה הפעולה $argmax_\omega$? האם עבור סיווג למחלקה מסוימת יש צורך באיברים שלא תלויים במחלקה ω?)

- הפעילו את פונקציית האימון ואת פונקציית הסיווג תחת הנחת סבירות מולטינומיאלית. התבוננו בערכי הפילוג האפוסטריורי המחושב בשלב הסיווג, ע"י הסתכלות על המשתנה ששומר את התוצאות: clf.last_scores. שימו לב שהמבנה שלו הוא התוצאה שכל מחלקה מקבלת לכל דוגמה בסט המבחן.

- **דו"ח מסכם – שאלה 6:** מהי שגיאת הסיווג? מה הבעייתיות שמציבים ערכי הפילוג האפוסטריורי? כיצד משפיעה בעייתיות זו על הסיווג המתקבל? ממה היא נובעת, וכיצד שימוש בסבירות מולטינומיאלית מוחלקת יכולה לפתור את הבעיה?

- הפעילו את פונקציית האימון ואת פונקציית הסיווג תחת הנחת סבירות מולטינומיאלית מוחלקת.

- **דו"ח מסכם – שאלה 7:** מהי שגיאת הסיווג?
הביטו בערכי הפילוג האפוסטריורי החדשים. מה הבעייתיות שאתם מזהים כעת?
כיצד חישוב לוגריתם סבירות מולטינומיאלית מוחלקת יכולה לפתור את הבעיה?

רמז 1. חשבו על ייצוג מספרים במחשב, כמה ביטים נדרשים לייצוג מספרים מאד קטנים ואיך זה משפיע על הדיוק.

רמז 2. חשבו מה קורה למכפלות כשמפעילים עליהן לוגריתם.

- הפעילו את פונקציית האימון ואת פונקציית הסיווג תחת הנחת סבירות מולטינומיאלית מוחלקת ועם שימוש בלוגריתם.
- **דו"ח מסכם – שאלה 8:** הביטו בערכי לוגריתם הפילוג האפוסטריורי (clf.last_score) . מהי שגיאת הסיווג? האם הבעייתיות שזיהיתם קודם נפתרה?

כעת תבצעו בחינה יסודית יותר של ביצועי המסווג הבייסיאני המולטינומיאלי המוחלק עם שימוש בלוגריתם. תוך שאתם בוחרים סדרות אימון ובוחן מתאימות, תבצעו את תהליך האימון והסיווג עבור גדלים שונים של סדרת האימון. חלקו את המסד לסט אימון וסט מבחן לפי יחס 0.8 ו-0.2 בהתאמה, ומתוך ה-0.8 השתמשו רק בחלק מסט האימון לפי היחסים הבאים: [0.1, 0.2, 0.4, 0.8, 1].

- בצעו עשרים חזרות, לצרכי מיצוע, על הרצף הבא, עבור כל אחד מגדלי סדרת האימון:
  - הגרלת סדרות לימוד ובוחן בגודל הרצוי
    1. אימון
    2. סיווג
    3. בחינת ביצועים

  בסופה של הריצה, מצעו על החזרות ושרטטו גרף של שגיאת הסיווג הממוצעת וסטיית התקן לכל גודל סדרת לימוד. השלימו את הקוד במקום הנדרש.
- **דו"ח מסכם – שאלה 9:** נתחו את הגרף המתקבל.
- **דו"ח מסכם – שאלה 10:** מדוע עלינו למצע על גבי חזרות? מה מקור האקראיות בהרצות?

  טיפ לחרוצים: עשרים חזרות הן מספר קטן יחסית עבור בעיות מסובכות כגון זו שאנו עוסקים בה. השוו את תוצאותיכם עד כה לאלה המתקבלות עבור הרצה של חמישים חזרות.

נתון נוסף בו נעשה שימוש בעת הערכת איכות המסווג הוא שגיאת האימון.

- הריצו את הרצף מהסעיף הקודם, כאשר סדרת הלימוד משמשת גם כסדרת בוחן. בסופה של הריצה, מצעו על החזרות ושרטטו גרף של שגיאת האימון הממוצעת וסטיית התקן לכל גודל סדרת לימוד.
- **דו"ח מסכם – שאלה 11:** נתחו את הגרף המתקבל והשוו אותו לגרף שהתקבל משימוש בסדרת בוחן שונה מסדרת הלימוד.

בזאת מסתיים המפגש הראשון. בשבוע הבא נעסוק בשתי שיטות סיווג נוספות ונסכם בהשוואה בין השיטות השונות.

רקע למעבדה – מפגש שני

לפני שתיגשו לחומר הרקע למפגש השני מומלץ להתרענן במושגי היסוד המופיעים בהכנה למפגש הראשון.

נמשיך בבחינת אלגוריתמים שונים והתאמתם לבעיית סיווג מסמכי הדוא״ל.

אלגוריתמים - המשך

1. מסווג K השכנים הקרובים ביותר (K nearest neighbors-KNN)

מסווג זה שייך לקבוצה של מסווגים לא-פרמטרים אשר מוגדרים ישירות בעזרת סדרת הלימוד ללא שלב ביניים של כוונון פרמטרים.

מסווג השכן הקרוב ביותר

בהינתן סדרת לימוד $\{x_k, y_k\}_{k=1}^{n}$ נסווג קלט חדש x ע״י מציאת הקלט x_k הקרוב ביותר ל- x , ונסווג את x בהתאם לתיוג שלו - y_k . ניתן לכתוב זאת בצורה הבא :

$$NN_x = \arg\min_{x_k} d(x, x_k) \quad \Rightarrow \quad \hat{y} = y_{NN_x}$$

כאשר $d(x, x')$ הינה פונקציית מרחק כלשהי בין שתי דוגמאות x ו- x' .

כחלק מהפעלת מסווג זה, יש לבחור את פונקציית המרחק הטובה ביותר לבעיה.

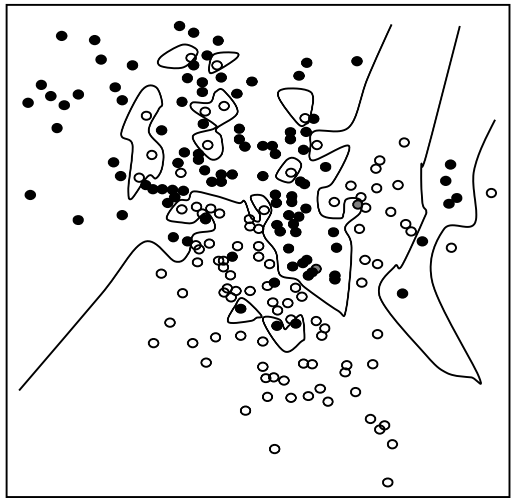
דוגמאות לפונקציות מרחק אפשריות בין שתי דוגמאות $x, x' \in \mathbb{R}^d$:

מרחק אוקלידי L_2 : $d_2(x, x') = \sqrt{\sum_{i=1}^{d} \left(x_i - x_i^{'}\right)^2}$

מרחק מנהטן L_1 : $d_1(x, x') = \sum_{i=1}^{d} \left| x_i - x_i^{'} \right|$

מרחק זוויתי (cosine distance) : $d_\theta(x, x') = 1 - \dfrac{x^T x^{'}}{\|x\|_2 \|x'\|_2}$

דוגמא לסיווג של דוגמאות ב-2D ע״פ השכן הקרוב ביותר :

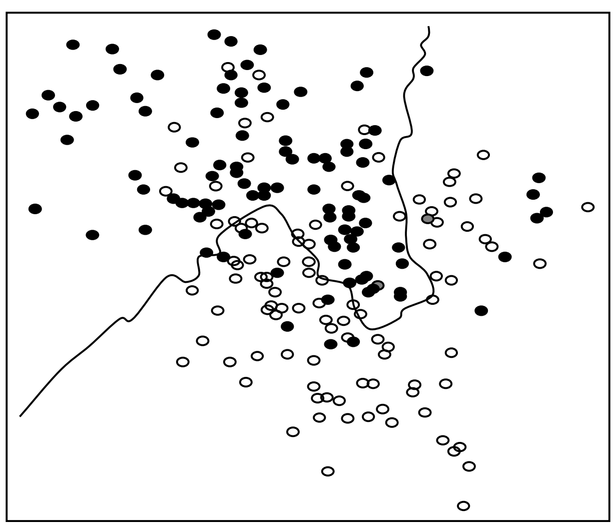


איור 8 - סיווג לשתי קטגוריות באמצעות מסווג K-NN עם K=1 לפי Hastie et al. (2001)

כפי שניתן לראות מהדוגמא מסווג השכן הקרוב ביותר עשוי לחלק את מרחב הקלט בצורה לא חלקה. בנוסף לכך, בגלל שאנו מסתמכים רק על דוגמא בודדה להחלטת הסיווג, קיימת רגישות לטעויות בסדרת הלימוד.

מסווג K השכנים הקרובים ביותר

מסווג זה הוא הכללה של מסווג השכן הקרוב ביותר למספר שכנים. כעת, נסווג קלט חדש x ע"י מציאת Kהדוגמאות מסדרת הלימוד הקרובות ביותר ל- x. נסווג את הדוגמא החדשה ע"פ התווית השכיחה ביותר מבין השכנים. במקרה של תיקו בין מספר תוויות ניתן להכריע ביניהן באמצעות כלל המוחלט מראש, כגון בחירה אקראית.



איור 9 - סיווג לשתי קטגוריות באמצעות מסווג K-NN עם K=15 לפי Hastie et al. (2001)

נשים לב שלמסווג K-NN אין שלב של חישוב מקדים בו מתכננים מסווג שיופעל בשלב הבחינה על קלט חדש, אלא רוב פעולת החישוב (מציאת המרחקים) מתבצעת כאשר מגיע הקלט החדש.
כלומר, מתבצעת העברה של העומס החישובי משלב תכנון המסווג לשלב הסיווג. אופן למידה זה נקרא Lazy Learning.

פועל יוצא של אופן הלמידה זה הוא שיש צורך לשמור בזיכרון את סדרת הלימוד $\{x_k, y_k\}_{k=1}^{n}$ לשלב הסיווג. כאשר מספר הדוגמאות גדול נדרש להקצות זיכרון גדול בהתאם.

2. <u>סיווג באמצעות פרספטרון בודד</u>

הפרספטרון הוא הרכיב הבסיסי של מערכת למידה רב-שכבתית המנסה לדמות את תהליך הלמידה במערכת נוירונים. במעבדה זו נעסוק רק ברכיב בסיסי זה, אשר יהיה לנו למסווג.
הפרספטרון הוא רכיב המפעיל פונקציה לינארית על המאפיינים השונים של הקלט ועליה פונקציית הפעלה לא –לינארית.
ניתן לתאר את פעולת הפרספטרון בצורה הבאה:

$$y = \varphi\left(\sum_{i=1}^{d} w_i x_i + b\right)$$

כאשר φ היא פונקציית ההפעלה הלא לינארית, $w = (w_1, w_2, ..., w_d)$ הינו וקטור המשקלים של הפונקציה הלינארית ו- b הינו פרמטר הטיה (bias).
הערה על סימונים: ניתן להוסיף איבר נוסף לקלט $x_{d+1} = 1$ ולסמן $w_{d+1} = b$, המאפשר שימוש בכתיב מקוצר: $y = \varphi\left(\sum_{i=1}^{d+1} w_i x_i\right) = \varphi\left(w^T x\right)$.

פונקציית ההפעלה φ שבה נבחר בניסוי זה היא פונקציית הפעלה מסוג Hard Limiter, והיא מותאמת *לסיווג בינארי* בין שתי מחלקות:

$$y = \varphi_{HL}\left(w^T x\right) = \begin{cases} -1: & w^T x < 0 \\ +1: & w^T x \geq 0 \end{cases}$$

מה המשמעות של פונקציה זו?

נתבונן בפונקציה $g(x) = w^T x$. המשוואה $g(x) = 0$ מגדירה משטח החלטה המפריד בין שתי מחלקות: $g(x) < 0$ ו- $g(x) \geq 0$, ובהתאם נקבע הסיווג.
ננסה לקבל קצת אינטואיציה גיאומטרית על הבעיה:

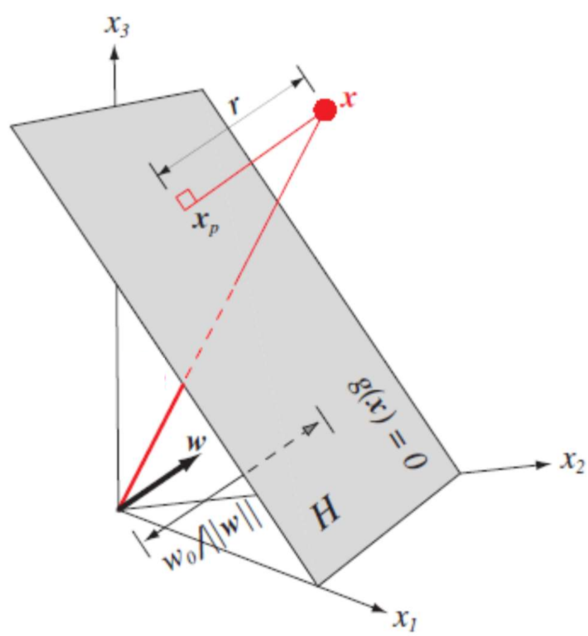
משטח ההחלטה הוא על-מישור (hyperplane), והווקטור w מאונך למישור זה (היות ולכל דוגמא הנמצאת על המשטח $w^T x = 0$). נשים לב כי ערך הפונקציה $g(x)$ נותן אינדיקציה על המרחק של נקודה x מהמישור. ניתן לראות זאת ע"י ייצוג x כסכום של ההטלה של x על המישור $x_p = P(x)$ ווקטור המאונך למישור: $x = P(x) + r\frac{w}{\|w\|}$. הערך r מתאר את המרחק מן המישור, ונרצה למצוא את ערכו באמצעות $g(x)$:

$$g(x) = w^T x = w^T\left(P(x) + r\frac{w}{\|w\|}\right) = 0 + r\|w\|$$

$$\Rightarrow r = \frac{g(x)}{\|w\|}.$$

(בפרט, המרחק של המישור מראשית הצירים $x = (0,0...,0)$ הוא $\frac{b}{\|w\|}$).

באופן גיאומטרי עבור $x \in \mathbb{R}^3$:



איור 10 - משטח הפרדה לינארי מתוך [1]

אם כן, פונקציית ההפעלה שלנו $\varphi_{HL}\left(w^T x\right)$ מסווגת כל קלט x בהתאם לצד של הנקודה ביחס לעל-מישור $g(x) = 0$.

כדי לסווג קלט חדש תחילה עלינו ללימוד בעזרת סדרת הלימוד את הפרמטרים של פונקציית ההפעלה, בפרט את הפרמטרים $w=\left(w_1,w_2,...,w_d,w_{d+1}\right)$. בשביל כך נשתמש באלגוריתם האיטרטיבי הבא.

אלגוריתם אימון הפרספטרון של רוזנבלט

האלגוריתם מעדכן באופן איטרטיבי את וקטור המשקלים במטרה להקטין את שגיאת הסיווג על סדרת הדוגמאות. וקטור המשקלים באיטרציה t יסומן w^t .

<u>קלט</u>: סדרת דוגמאות מתויגות לאימון $\left\{x_k,y_k\right\}_{k=1}^{n}$, פרמטר $\alpha>0$, מספר חזרות מקסימאלי על סדרת הלימוד m .

<u>פלט</u>: וקטור פרמטרים $w=\left(w_1,w_2,...,w_d,w_{d+1}\right)$.

<u>שלבי הלימוד</u>:

1. אתחול: אתחל את הווקטור w^0 בערך כלשהו (למשל, $w^0=\left(1,1,...,1\right)$).
2. רוץ על דוגמאות סדרת האימון $k=1,2,3,..,n$:
   1) קבל את הווקטור $x\in\mathbb{R}^{d+1}$ והתיוג $y\in\{-1,+1\}$ של הדוגמא ה- k .
   2) חשב את יציאת הפרספטרון עם וקטור המשקלים הנוכחי w^t : $\hat{y}=\varphi_{HL}\left(w^{tT}x\right)$.
   3) עדכן את וקטור המשקלים w^{t+1} (שהוא **וקטור** המשקלים אשר התקבל לאחר איטרציה t):

$$w^{t+1}=\begin{cases} w^t+\alpha\left(y-\hat{y}\right)x & ,y\neq\hat{y} \\ w^t & ,y=\hat{y}\end{cases}$$

3. עבור שוב על סדרת הלימוד (שלב 2) עד שמתקיים אחד משני תנאי העצירה הבאים:
   1) אין יותר עדכון של w - האלגוריתם מנבא נכון את התיוג של כל הדוגמאות.
   2) הושלמו m חזרות על סדרת הלימוד (epochs).

הערות:

- נשים לב כי האלגוריתם מעדכן את וקטור הפרמטרים רק אם התקבלה שגיאה בניבוי הדוגמא (ועל-כן נקרא אלגוריתם *פסיבי*). כאשר מתקבלת שגיאה משנים את וקטור הפרמטרים בכיוון הנכון, לצד החיובי של המישור אם הדוגמא חיובית או לצד השלילי אם הדוגמא שלילית.
- הפרמטר $\alpha>0$ מציין את קצב ההתקדמות או גודל הצעד של האלגוריתם, ויש לבחור אותו בהתאם לבעיית הלמידה.
- התכנסות האלגוריתם: ניתן להוכיח כי אם סדרת דוגמאות האימון ניתנת להפרדה ע"י פונקציה לינארית $g(x)=w^Tx$ (קיים וקטור פרמטרים המפריד את הדוגמאות ללא

שגיאה) אז האלגוריתם לעיל יתכנס במספר סופי של פעמים, ולפתרון אשר מסווג ללא שגיאה את כל דוגמאות האימון.

- אין הבטחה לגבי ביצועי האלגוריתם כאשר לא ניתן להגיע להפרדה מושלמת של הדוגמאות.
- גם כאשר קיימת הפרדה לינארית מושלמת הפתרון אינו יחיד.

סיווג למספר מחלקות:

אלגוריתם הפרספטרון המוצג לעיל מאפשר סיווג של דוגמאות לשתי מחלקות – סיווג בינארי.
מה קורה כאשר יש יותר מחלקות ורוצים להשתמש באלגוריתם זה?
אפשרות אחת היא לשלב מספר כללי החלטה של מסווגים בינאריים. כמובן שמידת ההצלחה של שילוב זה תלויה הן בבחירת כלל ההחלטה והן בבעיית הסיווג הנתונה.

קריאה נוספת

[1] *Pattern Classification* (2nd ed.), Richard O. Duda, Peter E. Hart and David G. Stork (John Wiley and Sons, 2001)

מפגש שני

שאלות הכנה

1. כתבו קטע פסאודו-קוד המקבל כקלט סדרת לימוד מתוייגת באורך N , $\left\{\left(x_i, y_i\right)\right\}_{i=1}^{N}$, כאשר $x_i \in \mathbb{R}, y_i \in \{1,...,K\}$ ודוגמא בודדת לא מתוייגת, $x \in \mathbb{R}$, ומחזיר את תיוג הדוגמא לאחת מ- K המחלקות על סמך מסווג 1-NN עם פונקציית המרחק $d(x,x') = |x-x'|$. מהי סיבוכיות זמן הריצה של הקוד שכתבתם?
2. בדומה לשאלה הקודמת, יש לכתוב קוד המסווג דגימה, $x \in \mathbb{R}$, על סמך מסווג 1-NN עם פונקציית המרחק $d(x,x') = |x-x'|$. הפעם ניתן לבצע בנפרד עיבוד מראש על סדרת האימון $\left\{\left(x_i, y_i\right)\right\}_{i=1}^{N}$, כדי שבזמן קבלת הדגימה החדשה x , הסיווג יתבצע מהר יותר. נדרש לכתוב אלגוריתם שסיבוכיות זמן הריצה שלו בעת קבלת הדגימה לסיווג היא **לוגריתמית** ב- N . נתחו את סיבוכיות זמן הריצה של התהליך כולו.
3. הוכיחו או הפריכו: המרחק הזוויתי, כפי שהוגדר בפרק המבוא, מקיים את הגדרות פונקציית המרחק (מטריקה) המקובלות בכל מימד $d \geq 1$.
4. הסבירו מדוע רצוי לבצע יותר ממעבר אחד על סדרת האימון באלגוריתם אימון הפרספטרון של רוזנבלט.
5. נניח שדוגמת אימון (x, y) בעלת תיוג $y = +1$ מסווגות באופן שגוי ידי פרספטרון עם וקטור משקלים w^t , כלומר $\hat{y} = \varphi_{HL}\left(w^{tT} x\right) = -1$. הראו כי עבור $\alpha > 0$ גדול מספיק, וקטור המשקלים המעודכן w^{t+1} יוביל לסיווג נכון, כלומר $\hat{y} = \varphi_{HL}\left(w^{t+1T} x\right) = +1$.
6. הראו כי עבור האלגוריתם של רוזנבלט, בחירה של משקלים התחלתיים w^0 וגודל צעד α שקולה לבחירת משקלים התחלתיים w^0/α וגודל צעד 1. במינוח "שקולה" הכוונה לכך ששני המסווגים יתנו תוצאה זהה עבור כל סדרת בוחן.
7. הציעו מימוש למסווג תלת מחלקתי הבנוי ממספר מסווגי פרספטרון בינאריים. לכמה מסווגים בינאריים תידרשו על מנת לממש מסווג n-מחלקתי?

הנחיות

שאלות הדו"ח המסכם מופיעות לאורך הניסוי. יש לענות עליהן בכיתה, במהלך ביצוע הניסוי. בנוסף, עליכם לצרף לדו"ח המסכם את כל קבצי הקוד (המחברות Python) שיצרתם במהלך הניסוי.

מהלך הניסוי

במפגש זה נמשיך לעסוק בהיבטיה השונים של בעיית הסיווג. הקדישו מספר דקות לריענון זיכרונכם. ודאו כי זכור לכם כיצד לטעון את מאגר המידע, כיצד להמירו לפורמט מספרי וכיצד ליצור ממנו סדרות אימון ולימוד. העזרו בהנחיות המפגש הראשון. מלאו את הבלוקים הראשונים בהתאם למחברת מהמעבדה הקודמת, בה טענתם את המסד, ביצעתם עיבוד מקדים וחילקתם לסט אימון וסט מבחן.

1. <u>סיווג K השכנים הקרובים ביותר (K-NN – K Nearest Neighbors)</u>

חלק זה של הניסוי עוסק במסווג K-NN. יחודו של מסווג זה בכך שלא מתקיים שלב לימוד מובחן, הקודם לשלב הסיווג. במקום זה, מקבל המסווג כקלט סדרת לימוד מתויגת וסדרת בוחן לא מתויגת. הסיווג מתבצע על סמך המידע שמספקת סדרת האימון.

תחילה תעסקו בסיווג לפי השכן הקרוב ביותר (K=1).

- השלימו את מימוש הפונקציה `evaluate_classifier` המקבלת אובייקט מסווג `clf` , מטריצת מאפיינים X ווקטור תיוגים y. הפונקציה מגריל סדרות אימון ובוחן, ומחזירה ממוצע וסטיית תקן של שגיאת בוחן. תוכלו להשתמש בפונקציה זו בהמשך.
- הפעילו את המסווג 1-NN באמצעות יצירת המסווג KNeighborsClassifier מ-Scikit-Learn, עבור כל אחת מפונקציות המרחק המוצעות. השתמשו בחלוקה של 80% ו-20% בהתאמה וחשבו את הממוצע וסטיית תקן של שגיאת הבוחן על פני **10** חזרות.
  תזכורת ליצירת אובייקט המסווגים :

L2
```
KNeighborsClassifier(n_neighbors=K, p=2)
```
L1
```
KNeighborsClassifier(n_neighbors=K, p=1)
```
Cosine distance
```
KNeighborsClassifier(n_neighbors=K, metric='cosine')
```

- **דו"ח מסכם – שאלה 1:** מהי פונקציית המרחק המתאימה ביותר לבעיה בה אנו עוסקים? מדוע?

בשלב הבא תכמתו את השפעת הפרמטר K על ביצועי המסווג.

- כתבו קוד שישתמש במסווג KNeighborsClassifier עם מרחק זוויתי עבור ערכי K הבאים : [1,3,5,7,15]. הפעילו את המסווג על סדרת לימוד ומבחן בחלוקה של 80% ו-20% בהתאמה. מצעו על פני **10** חזרות ושרטטו גרף של שגיאת הסיווג הממוצעת וסטיית התקן.
- **דו"ח מסכם – שאלה 2:** מהו ערך K האופטימלי בבעיה הנתונה? הסבירו כיצד משפיעה הגדלתו של K על הסיווג. ציינו יתרון אחד וחסרון אחד של בחירת ערך K גדול.

דו"ח מסכם – שאלה 3:

א. הסבירו מדוע מקובל להשתמש ב K אי-זוגי.

ב. במידה והייתה לנו כמות כפולה של דוגמאות לאימון, כיצד לדעתכם היה משתנה ערך ה-K האופטמלי? מדוע?

כעת תבחנו את השפעתם של המאפיינים השונים על ביצועי המסווג. עד כה, למעט ההמרה לפורמט מספרי, לא ביצעתם עיבוד מקדים למידע שברשותכם. עם זאת, במקרים רבים לעיבוד כזה יש השפעה רבה על איכות הסיווג. העיבוד המקדים, כפי שמצוין בפרק המבוא, מופעל הן על סדרת האימון והן על סדרת הבוחן. ישנן שלוש גישות לעיבוד מקדים עבור המשימה שלפנינו.
האחת היא גישת tf-idf, המנרמלת את השפעתה של מילה מסוימת במספר המופעים הכולל שלה; השנייה היא גישת הסף, הכוללת בסט המאפיינים רק מילים שהופיעו לפחות מספר מסוים של פעמים; והשלישית ממירה את מבנה הנתונים המוזן לה למבנה נתונים בינארי, כך שאם מילה מסוימת הופיעה יותר מערך סף של פעמים, המאפיין המתאים יקבל את הערך '1, אחרת יקבל '0.

דו"ח מסכם – שאלה 4: האם, באופן אינטואיטיבי, המרת מבנה הנתונים שבידיכם למבנה בינארי צפויה לשפר את ביצועי המסווג או לפגוע בהם?

מיד תפעילו את הטרנספורמציה TfidfTransformer ותבחנו את השפעת העיבוד המקדים על ביצועי המסווג. העיבוד המקדים מופעל תחילה על סדרת האימון ואחר כך על סדרת הבוחן. קראו את הפונקציה ואת תיעודה (ע"י (help(TfidfTransformer)) על מנת להבין באילו פרמטרים להשתמש ואיך לקרוא לה. על מנת להקל על השימוש במחלקה, להלן דוגמא של אופן הקריאה לה:

```
tfidf_transformer=TfidfTransformer(smooth_idf=True,use_idf=True)

X_augmented_tfidf_train = tfidf_transformer.fit_transform(X_augmented_train)

X_augmented_tfidf_test = tfidf_transformer.transform(X_augmented_test)
```

הערה: בדיקה בסיסית של העיבוד המקדים היא השוואת מספר המאפיינים בסדרת הלימוד ובסדרת הבוחן – אם הופעל על שתי הסדרות אותו העיבוד המקדים, מספר המאפיינים בשתי הסדרות צריך להיות זהה.

כעת תשוו את הביצועים המתקבלים אחרי הפעלת עיבוד מקדים מסוג tf-idf לאלה המתקבלים ללא הפעלת עיבוד מקדים.

כתבו קטע קוד המבצע את השלבים הבאים:

- מגריל סדרות אימון ובוחן, השתמשו בחלוקה של 80% ו-20% בהתאמה. .

1. יוצר שני עותקים של סדרות האימון והבוחן ומפעיל עיבוד מקדים מסוג tf-idf על סדרת האימון ועל סדרת הבוחן של אחד העותקים. העותק השני נותר ללא עיבוד מקדים, לשם השוואה.
2. מפעיל את המסווג 3-NN באמצעות יצירת המסווג KNeighborsClassifier מ-Scikit-Learn, תוך שימוש במרחק זוויתי (ר' תיעוד הפונקציה), על כל אחד מעותקי סדרות האימון והבוחן.
3. בוחן את איכות הסיווג עבור כל אחד מהתרחישים (עם וללא עיבוד מקדים).
4. מריץ את סעיפים 1-4 **10** פעמים ומחשב את הממוצע וסטיית התקן של התוצאות.

- **דו"ח מסכם – שאלה 5** : מהי שגיאת הסיווג הממוצעת עבור כל אחת מהשיטות? מה מסקנותיכם לגבי השפעת העיבוד המקדים?

הערה: בכל הסעיפים הבאים אין לבצע עיבוד מקדים.

2. סיווג באמצעות פרספטרון (Perceptron)

חלק זה של הניסוי עוסק במסווג פרספטרון. יחודו של מסווג זה בכך שהוא בינארי, כלומר מאפשר לסווג לשתי מחלקות בלבד. הרחבתו למקרה הרב מחלקתי (שלוש מחלקות, במקרה בו אנו עוסקים) מצריכה התאמה מסוימת, כפי שראיתם בדו"ח המכין.
בחלק זה תממשו ותפעילו את אלגוריתם הפרספטרון על בעיה הבינארית של ספאם (+1) ודוא"ל אמיתי (-1).

- ממשו את הפונקציות fit ו-predict של המחלקה MlabPerceptron , כאשר זכרו כי הראשונה מבצעת אימון ואילו השנייה מבצעת תיוג. בחרו וקטור משקלים התחלתי $w^0 = (1, \dots ,1)$. הריצו את פונקציית האימון על סדרת לימוד המהווה 80% מסך כל הדוגמאות. השתמשו בערכים $\alpha = 0.5$, m=10. m מציין את מספר ה-epochs.

- **דו"ח מסכם – שאלה 6:** מדוע מוגדר אורך וקטור המשקלים להיות מספר המאפיינים ועוד אחד (`self.w = np.ones((1, num_features + 1))`) ?
- **דו"ח מסכם – שאלה 7:** התבוננו בווקטור w המתקבל לאחר תהליך האימון. מה המשמעות של משקולות הגדולות בערך מוחלט ומשקולות הקטנות (הקרובות ל-0) בערך מוחלט?

- **דו"ח מסכם – שאלה 8:** בצעו הרצה סדרתית של אימון וסיווג על פני 20 חזרות, תוך הגרלה מחדש של סדרות הלימוד והבוחן (80% ו-20% בהתאמה) בכל חזרה. השתמשו

בערכים $\alpha = 0.5$, m=50. חשבו את שגיאת הסיווג הממוצעת ואת סטיית התקן של השגיאה על פני כל ההרצות.

3. סיכום והשוואה בין האלגוריתמים

בחלק זה נשווה בין האלגוריתמים שלמדנו בניסוי (בייס נאיבי, K-NN, פרספטרון) על הבעיה של ספאם ודוא״ל אמיתי.

- אמנו והפעילו כל אחד משלושת המסווגים שהכרתם בניסוי על הבעיה. עבור מסווג בייס בחרו צורת פילוג multinomial_smooth והשתמשו ב use_log_prob=True ; עבור מסווג K-NN בחרו פונקציית מרחק זוויתית ו-K=3. הריצו את האלגוריתמים עשרים פעמים, בכל פעם דיגמו מחדש את מאגרי הנתונים כמוסבר לעיל. ודאו כי סדרת לימוד המהווה 80% מסך כל הדוגמאות. חשבו את שגיאת הסיווג הממוצעת ואת סטיית התקן עבור כל מסווג. שימו לב, אם שמרתם את כל התוצאות מההרצות, אינכם חייבים לאמן שוב, אלא אתם יכולים להעתיק את התוצאות לטבלה מטה.
- **דו״ח מסכם – שאלה 9:** מלאו את ערכי שגיאת הסיווג בטבלה.

		בייס	K-NN	פרספטרון
שגיאת סיווג				
סטיית תקן				

במקרים מסוימים (במיוחד שגודל המחלקות לא מאוזן), מדד הביצועים הרצוי של מסווג הוא לא אחוז השגיאה הכולל, אלא המדדים precision ו recall. מדדים אלו מציגים מידע על השגיאה בהינתן מחלקה.

נגדיר את המדדים כך :

$$\text{Precision} = \frac{\text{Number of spam mails classified as 'spam'}}{\text{Number of spam mails}}$$

$$\text{Recall} = \frac{\text{Number of spam mails classified as 'spam'}}{\text{Number of mails classified as 'spam'}}$$

- **דו״ח מסכם – שאלה 10:** חשבו את מדדי ה precision ו recall על סט הבוחן עבור מסווג אחד לבחירתכם. עבור אחד המסווגים שראינו, הציעו דרך לשנות את פעולת המסווג כך שמדד ה- precision יגדל. כיצד ישתנה לדעתכם מדד ה recall?

רשימת הפונקציות

הפונקציות המסומנות בכחול מומשו, בעוד שאת הפונקציות המסומנות באדום יש להשלים.

ניתן להשתמש ב-()help כדי לקבל הסבר מפורט במחברת עצמה.

הסבר	פלט	קלט	שם הפונקציה\מחלקה
פונקציה זו ממירה את מסד הנתונים מייצוג מילולי לייצוג מספרי. הקלט הוא מסד הנתונים.	▪ X_augmented – numeric representation of the data, using Bag of Words.	▪ X– database of e-mail (rows are document samples, columns are features - strings)	**email_pipeline**
פונקציה המגרילה סדרות לימוד ובוחן מתוך מסד הנתונים, בייצוגו המספרי. הקלט הוא מסד הנתונים בצירוף גדלי הסדרות הדרושים. הפלט הוא סדרת הלימוד וסדרת הבוחן.	▪ X_train – training set samples ▪ Y_train – training set labels ▪ X_test – test set samples ▪ Y_test – test set labels	▪ X - dataset representation matrix (rows are document samples, columns are features) ▪ y - dataset labels ▪ test_size- fraction of the data to allocate for the test set	**train_test_split**
פונקציית האימון של מסווג בייסיאני אמפירי נאיבי - הפונקציה מקבלת קלט מתויג (סט האימון) ולומדת את הסבירות p(x|w) ואת ההסתברות האפריורית P(w) של כל אחת מהמחלקות.	▪ .fit() – trains the classifier ▪ .predict() – returns labels for unlabeled data.	▪ X - training set matrix ▪ Y - training set labels ▪ dist_type - assumed distribution of likelihood; ‘gaussian', 'bernoulli', 'multinomial', 'multinomial_smooth' ▪ num_classes - number of classes ▪ use_log_prob – whether or not to use the log of the probabilities	**MlabNaiveBayes**
פונקציה המחשבת את פרמטרי הפילוג המותנה בהתאם לצורת הפילוג הנבחר. הקלט הוא מסד הנתונים, הפילוג הנבחר ומספר המחלקות. הפלט הוא ההפילוג המותנה.	▪ params - parameters of likelihood distribution for each class	▪ X - training set matrix ▪ Y - training set labels ▪ dist_type - assumed distribution of likelihood : 'gaussian', 'bernoulli', 'multinomial', 'multinomial_smooth' ▪ num_classes - number of classes	**estimate_likelihood_params**
פונקציה המבצעת סיווג לפי אלגוריתם KNN. הקלט הוא סדרת האימון (עם תיוגיה) וסדרת הבוחן הלא מתויגת, בצירוף פרמטרים המאפיינים את המסווג. הפלט הוא סדרת התיוגים של סדרת הבוחן.	▪ .fit() – trains the classifier ▪ .predict() – returns Y2 - label vector for samples X2	▪ X1,X2 - data matrices, each row represents a document and each column a feature ▪ Y1 - the labels corresponding to X1 ▪ K - number of neighbors for classification ▪ metric - type of distance measure	**KNeighborsClassifier**

MlabPerceptron	▪ X - training set samples ▪ Y - training set labels ▪ alpha - step size for perceptron training algorithm ▪ nun_epochs - maximal number of epochs for perceptron training algorithm	▪ .fit() – training, learns w - weight vector for perceptron classifier ▪ .predict() -returns labels for unlabeled data	פונקציה המממשת את אלגוריתם אימון הפרספטרון. הקלט הוא סדרת האימון, גודל הצעד ומספר החזרות המקסימלי. הפלט הוא וקטור המשקלים הנחוץ לסיווג באמצעות פרספטרון.
TfidfTransformer	▪ X – samples	▪ X – preprocessed samples with TF-IDF transformation.	פונקציה המבצעת עיבוד מקדים על סדרת האימון או הבוחן. הסבר מפורט מופיע בגוף הפונקציה.