מבוא ללמידת מכונה - תרגיל 3

מגיש: יואב לוי 314963257

Bayes Optimal and LDA

שאלה 1

$$\text{argmax}_{y\in\{\pm1\}}\Pr(x|y)\cdot\Pr(y)\overset{\text{Bayes Thm.}}{=}\text{argmax}_{y\in\{\pm1\}}\frac{\Pr(x\cap y)}{\Pr(y)}\cdot\Pr(y)=$$

$$\overset{\Pr(x)\neq0}{=}\text{argmax}_{y\in\{\pm1\}}\frac{\Pr(x\cap y)}{\Pr(x)}\cdot\Pr(x)=\text{argmax}_{y\in\{\pm1\}}\Pr(y|x)\cdot\Pr(x)=$$

$$\overset{*}{=}\text{argmax}_{y\in\{\pm1\}}\Pr(y|x)=\begin{cases}+1 & \Pr(y=1|x)\geq\frac{1}{2}\\ -1 & otherwise\end{cases}=h_{\mathcal{D}}(x)$$

(* מכיוון שהסתברות $\Pr(x)$ היא חיובית ואינה תלוייה ב־y שעברו ה־argmax מוגדר, ניתן להתעלם מהמכפלה בו)
הערה: נשים לב למקרי קצה בהם אם $\Pr(y=1)=0$
אז מכיוון ש־$Supp(y)=\{1,-1\}$ אז $\Pr(y=-1)=1$ ואז

$$\text{argmax}_{y\in\{\pm1\}}\Pr(x|y)\cdot\Pr(y)=-1=h_{\mathcal{D}}(x)$$

וזה נכון לכל x, בנוסף אם $\Pr(y=-1)=0$ אז בלי הגבלת הכלליות נקבל אותו דבר כמו מקודם רק הפוך.

שאלה 2

$$\begin{aligned}
h_{\mathcal{D}}(x)&=\text{argmax}_{y\in\{\pm1\}}\left\{\Pr(\boldsymbol{x}|y)\Pr(y)\right\}\\
&=\text{argmax}_{y\in\{\pm1\}}\left\{\frac{1}{\sqrt{(2\pi)^d\det(\Sigma)}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\mu_y)^\top\Sigma^{-1}(\boldsymbol{x}-\mu_y)\right\}\cdot\Pr(y)\right\}\\
&\overset{\text{(removed constants)}}{=}\text{argmax}_{y\in\{\pm1\}}\left\{\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\mu_y)^\top\Sigma^{-1}(\boldsymbol{x}-\mu_y)\right\}\cdot\Pr(y)\right\}\\
&\overset{\text{(Log is monotone increasing)}}{=}\text{argmax}_{y\in\{\pm1\}}\left\{ln\left(\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\mu_y)^\top\Sigma^{-1}(\boldsymbol{x}-\mu_y)\right\}\cdot\Pr(y)\right)\right\}\\
&=\text{argmax}_{y\in\{\pm1\}}\left\{\left(-\frac{1}{2}(\boldsymbol{x}-\mu_y)^\top\Sigma^{-1}(\boldsymbol{x}-\mu_y)\right)+ln(\Pr(y))\right\}\\
&=\text{argmax}_{y\in\{\pm1\}}\left\{\left(\left(-\frac{1}{2}\boldsymbol{x}^\top\Sigma^{-1}+\frac{1}{2}\mu_y^\top\Sigma^{-1}\right)(\boldsymbol{x}-\mu_y)\right)+ln(\Pr(y))\right\}\\
&=\text{argmax}_{y\in\{\pm1\}}\left\{\left(-\frac{1}{2}\boldsymbol{x}^\top\Sigma^{-1}x+\frac{1}{2}\boldsymbol{x}^\top\Sigma^{-1}\mu_y+\frac{1}{2}\mu_y^\top\Sigma^{-1}\boldsymbol{x}-\frac{1}{2}\mu_y^\top\Sigma^{-1}\mu_y\right)+ln(\Pr(y))\right\}\\
&\overset{\left(-\frac{1}{2}\boldsymbol{x}^\top\Sigma^{-1}\mathbf{x}\text{ is not depend on y}\right)}{=}\text{argmax}_{y\in\{\pm1\}}\left\{\left(\mu_y^\top\Sigma^{-1}\boldsymbol{x}-\frac{1}{2}\mu_y^\top\Sigma^{-1}\mu_y\right)+ln(\Pr(y))\right\}\\
&=\text{argmax}_{y\in\{\pm1\}}\delta_y(x)
\end{aligned}$$

שאלה 3

$$\mu_{+1} = \frac{1}{m}\sum_{i=1}^{m} x_i \cdot (1_{y_i}) \qquad where\ 1_{y_i} = \begin{cases} 1 & if\ y_i = 1 \\ 0 & otherwise \end{cases}$$

$$\mu_{-1} = \frac{1}{m}\sum_{i=1}^{m} x_i \cdot (-1_{y_i}) \qquad where\ -1_{y_i} = \begin{cases} 1 & if\ y_i = -1 \\ 0 & otherwise \end{cases}$$

$$\Sigma = \frac{X_{centered} X_{centered}^T}{m-1} \qquad where\ X_{centered}^T = \begin{pmatrix} - & x_1 & - \\ & \vdots & \\ - & x_m & - \end{pmatrix} - \begin{pmatrix} \frac{1}{m}\sum_{i=1}^m x_i^1 & \dots & \frac{1}{m}\sum_{i=1}^m x_i^d \\ \vdots & \vdots & \vdots \\ \frac{1}{m}\sum_{i=1}^m x_i^1 & \dots & \frac{1}{m}\sum_{i=1}^m x_i^1 \end{pmatrix}$$

$$\Pr(y=1) = \frac{\sum_{i=1}^m 1_{y_i}}{m} \qquad where\ 1_{y_i} = \begin{cases} 1 & if\ y_i = 1 \\ 0 & otherwise \end{cases}$$

$$\Pr(y=-1) = \frac{\sum_{i=1}^m -1_{y_i}}{m} \qquad where\ -1_{x_i} = \begin{cases} 1 & if\ y_i = -1 \\ 0 & otherwise \end{cases}$$

הערה: נשים לב שמטריצת ה־COV מוגדרת בעזרת $X_{centered}^T$ כאשר זאת מוגדרת בעזרת מטריצת של m דגימות על d פיצ'רים,
פחות מטריצה שבכל עמודה יש ממוצע של פיצ'ר לפי מספר העמודה (כלומר בעמודה הראשונה יופיעו בכל השורות הממוצע על הפיצ'ר הראשון).

Spam

שאלה 4

- הטעויות שהמסווג שלי עלול לעשות הן, לסווג אימייל כספאם כאשר הוא אינו ספאם, וההפך לסווג אימייל כלא ספאם כאשר הוא למעשה ספאם.
- הטעות שלא נרצה לעשות היא לסווג אימייל כספאם כאשר הוא אינו ספאם.
- spam=1
  not-spam $= -1$

SVM-Formulation

שאלה 5

$$\text{argmin}_{\boldsymbol{v}\in\mathbb{R}^n} \left(\frac{1}{2} v^\top \cdot 2 \cdot I \cdot v + \overbrace{\vec{0}^\top}^{\boldsymbol{a}^\top} v \right)$$
$$s.t. \quad Av \leq \boldsymbol{d}$$

כאשר,

$$Q = 2\cdot I = \begin{pmatrix} 2 & 0 & \dots & 0 \\ \vdots & 2 & 0 & \vdots \\ 0 & 0 & \ddots & 0 \\ 0 & \dots & 0 & 2 \end{pmatrix} \in \mathbb{R}^{n\times n}$$

$$A = \overbrace{\begin{pmatrix} -y_1 & \overbrace{(-y_1x_1)^\top}^{n-1} \\ \vdots & \vdots \\ y_m & (-y_mx_m)^\top \end{pmatrix}}^{n} \in \mathbb{R}^{m\times n}$$

$$\boldsymbol{a} = \vec{\mathbf{0}} \in \mathbb{R}^n$$

$$\boldsymbol{d} = \vec{-\mathbf{1}} \in \mathbb{R}^m$$

<u>הערות</u>: שורה i במטריצה A היא $row_i = \left[-y_1, -y_1x_i^1, -y_1x_i^2 - y_1x_i^3, \dots, -y_1x_i^{n-1}\right] \in \mathbb{R}^n_{row}$
כאשר x_i^j הוא ה־feature ה־j בדגימה ה־i. בנוסף $I \in \mathbb{R}^{n\times n}$ היא מטריצת הזהות.

שאלה 6

ראשית נשים לב שהבעיות שקולות פרט לכך שבבעיה הראשונה אנחנו ממצעים על ξ_i לכל $i \in [m]$
כאשר יש לנו את התנאי $\forall i, y_i \langle w, x_i\rangle \geq 1 - \xi_i$ and $\xi_i \geq 0$ (נסמנו ♪) ולכן כדי להראות שהבעיות שקולות
יש להראות שלכל i הבחירה של ξ_i עבור **<u>אותו</u>** w_i (כאשר ξ_i יגרור לעמידה בתנאי ♪), תהיה שקולה
להפעלת פונקציית ה־ℓ^{hinge} על $y_i \langle w, x_i\rangle$.
קרי שמתקיים $\forall i \in [m], \ell^{hinge}(y_i \langle w, x_i\rangle) = \xi_i$ וגם שזה גורר לכך שעומדים בתנאי ♪.
<u>כדי להראות זאת אחלק לשני מקרים</u>: (עבור $i \in [m]$)

- במידה ומתקיים $y_i \langle w, x_i\rangle \geq 1$, נשים לב שלפי הגדרת פונקציית ה־ℓ^{hinge} אנחנו נקבל

$$\ell^{hinge}(y_i \langle w, x_i\rangle) = \max\{0, 1 - y_i \langle w, x_i\rangle\} \overset{y_i\langle w,x_i\rangle \geq 1}{=} 0$$

כעת, נתבונן בבעיה הראשונה ונשים לב שבמקרה הנ"ל אם נחבר $\xi_i > 0$ אז $\ell^{hinge}(y_i \langle w, x_i\rangle) = 0 < \xi_i$
אבל מכיוון שאנחנו מחפשים להגיע לערך המינימלי של $\frac{\lambda}{2}||w||^2 + \frac{1}{m}\Sigma_{i=1}^m \xi_i$ אז ברור שעדיף ש־$\xi_i = 0$,
בנוסף עבור אותו i גם יתקיים התנאי ♪,

$$y_i \langle w, x_i\rangle \geq 1 - \xi_i = 1 - 0 = 1 \text{ and } \xi_i \geq 0$$

כלומר במקרה זה פונקציה ה־ℓ^{hinge} תהיה שקולה לבחירת $\xi_i = 0$ עבור ה־i הנ"ל, וגם אנחנו עומדים בתנאי ♪.

- במידה ומתקיים $y_i \langle w, x_i\rangle < 1$, נשים לב שנקבל

$$\ell^{hinge}(y_i \langle w, x_i\rangle) = \max\{0, 1 - y_i \langle w, x_i\rangle\} = 1 - y_i \langle w, x_i\rangle$$

כעת, אם נבחר ב־$\xi_i = 1 - y_i \langle w, x_i\rangle$ אז תנאי ♪ יתקיים כי,

$$y_i \langle w, x_i\rangle = y_i \langle w, x_i\rangle = 1 \overbrace{-1 + y_i \langle w, x_i\rangle}^{-\xi_i} = 1 - \xi_i \tag{1}$$

$$\textbf{and} \quad \xi_i = 1 - y_i \langle w, x_i\rangle \overset{y_i\langle w,x_i\rangle<1 \iff 0<1-y_i\langle w,x_i\rangle}{>} 0 \tag{2}$$

בנוסף מכיוון שאנחנו רוצים להקטין כמה שיותר את פונקציית המטרה שלנו בבעיית ה־Soft-SVM, אז נרצה לבחור ξ_i קטן ככל שאפשר

אבל נניח בשלילה שיש כזה $\xi' < \xi_i = 1 - y_i \langle w, x_i \rangle$ אז נקבל

$$y_i \langle w, x_i \rangle = y_i \langle w, x_i \rangle = 1 \overbrace{-1 + y_i \langle w, x_i \rangle}^{-\xi_i} = 1 - \xi_i < 1 - \xi'$$

בניגוד לתנאי ♪ $(\forall i, y_i \langle w, x_i \rangle \geq 1 - \xi_i)$, ולכן בהכרח מצאנו את ה־$\xi$ הנכון עבור ה־i הנ"ל.

כלומר למעשה הוכחנו ששני הבעיות שקולות.

■

Implemention and simulation-comparison of different classifiers

שאלה 7

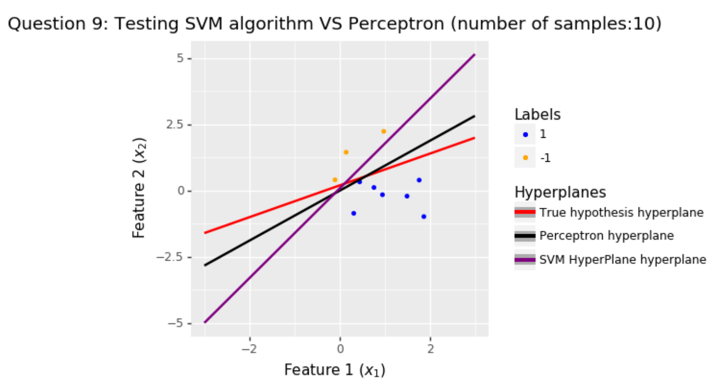
This is a coding question, the code is provided in the 'models.py' python file.

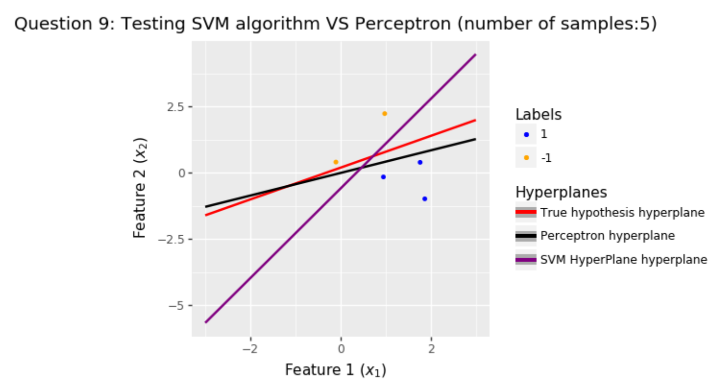
שאלה 8

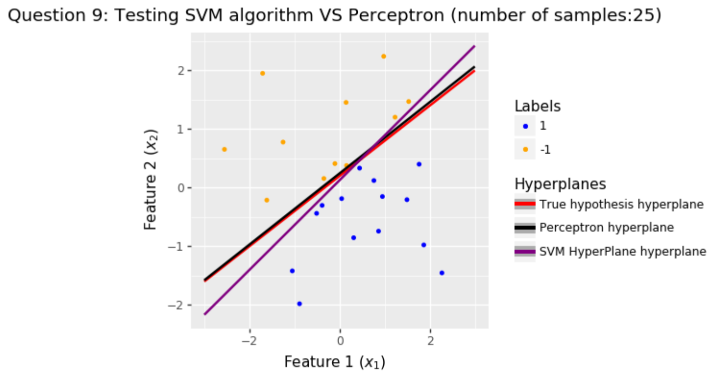
This is a coding question, the code is provided in the 'comparison.py' python file.

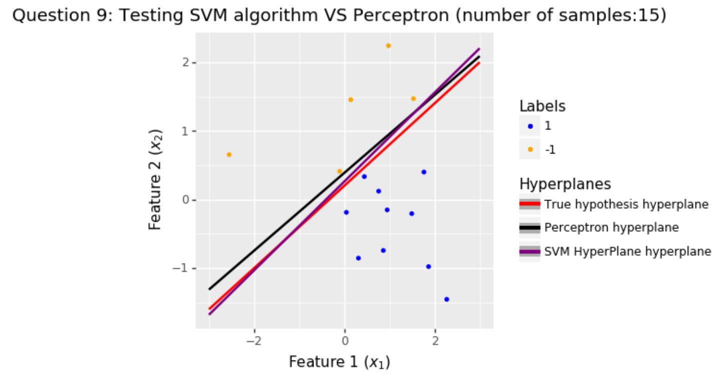
שאלה 9

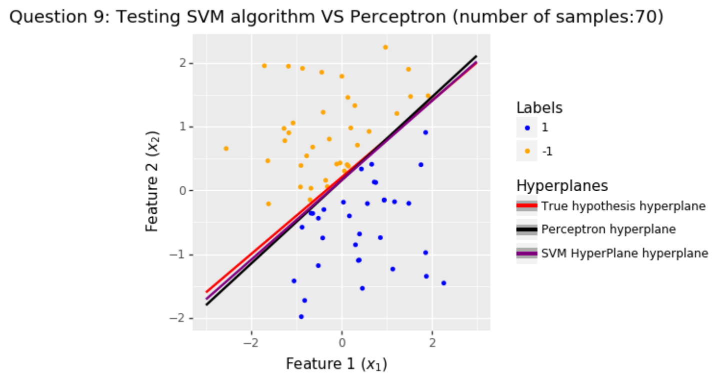
הגרפים הבאים מציגים את העל־מישורים שנבחרו ע"י האלגוריתמים SVM, Perceptron, כאשר הפונקציה האמיתית שמתארת את העל מישור היא $\left\langle \begin{pmatrix} 0.3 \\ -0.5 \end{pmatrix}, \boldsymbol{x} \right\rangle + 0.1 = 0$, והשוני בכל גרף הוא מספר הדגימות $(samples)$ שעליו כל אלגוריתם קלסיפיקציה אומן.







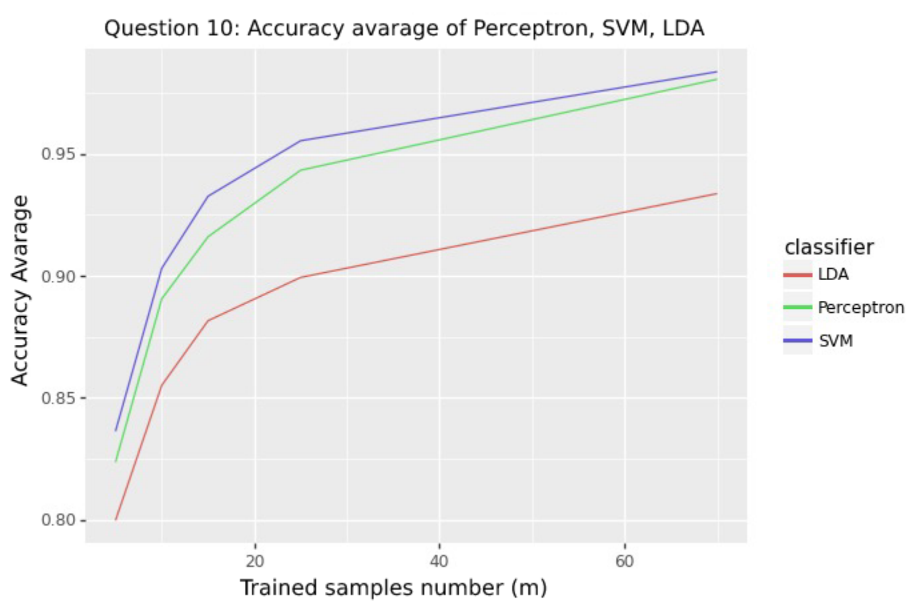




*בנוסף הקוד של הגרפים ויצירתם נמצא בקובץ 'comparison.py'.

שאלה 10

בשאלה זו נדרשנו לבדוק את ה־Accuracy של האלגוריתמים Perceptron, LDA, SVM, עבור מספר דגימות משתנה שעליהן אימנו את ה־DATA, בנוסף ה־Testing data נעשה על $10,000$ דגימות נוספות, ומיצענו על 500 איטרציות עבור
כל מספר דגימות שעליו אימנו את ה־$DATA$.
בתשובה לשאלה הבאה אני מסביר על התוצאות.



*בנוסף הקוד של הגרפים ויצירתם נמצא בקובץ 'comparison.py'.

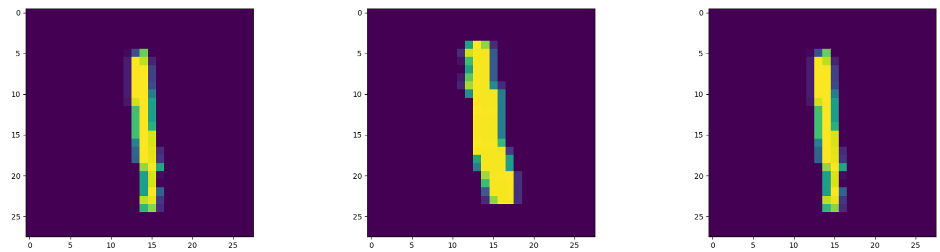
שאלה 11

ניתן לראות בגרף שהמסווג SVM היה המוצלח ביותר מבין שלושת המסווגים, וה־Perceptron גם מאוד קרוב אליו.
לדעתי זה נובע מהעובדה שה־Perceptron מנסה למצוא את ההפרדה הטובה ביותר, בעוד שה־SVM מנסה למצוא את ה־margin
הטוב ביותר, נשים לב שכל האלגוריתמים מתכנסים יחסית ככל שאנחנו מאמנים על יותר DATA.
בנוסף אלגוריתם ה־LDA אינו מצליח להגיע לרמת דיוק של ה־SVM and Perceptron.

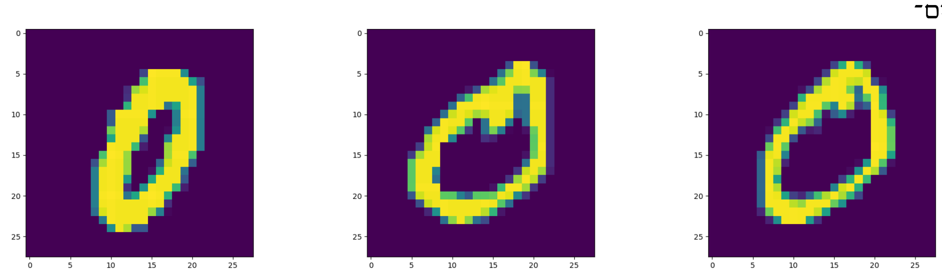
שאלה 12

<u>דגימות מה־DATA SET:</u>

אחדים־



אפסים־



שאלה 13

This is a coding question, the code is provided in the 'mnist_data.py' python file.

שאלה 14

- הגרף הבא מתאר את הדיוק (Accuracy) של כל אחד מהאלגוריתמי למידה שנדרשנו לבדוק.
  מדדנו את התוצאות על DATA של כתבי יד של מספרים (0 או 1) וציר ה־X הוא מספר הדגימות שדגמנו כדי לאמן את האלגוריתמים.
  נשים לב שבמקרה של KNN דווקא חלה ירידה קלה מאוד ככל שאימנו על יותר וזה מכיוון שהגדרתי את שהאלגוריתם יבדוק על מספר שכנים שהוא שליש ממספר הדגימות, כלומר כשיש יותר דגימות ככה הוא מתחשב ביותר שכנים.
  דבר זה כנראה מביא לבעיה מסויימת כי הוא מתחיל להחשיב שכנים שכנראה מטעים אותו.
  למרות ששינוי זה הוא מינורי (במאית אחוז).
  בקשר לשאר האלגוריתמים, הם די יציבים וגם ככה רובם קרובים ל־100% רוב הזמן.
- **הזמן שלקח לפונקציה לרוץ הוא: 136.6 שניות בסה"כ.**

בנוסף הפילוח של הזמנים היה כזה שהזמנים שלה אלגוריתמים הבאים היו בסדר הזה (מהזמן הארוך ביותר לקצר)

$$\overbrace{\text{KNN}}^{\text{50 Seconds}} > \overbrace{\text{SVM}}^{\text{34 Seconds}} \cong \overbrace{\text{Logistic Regression}}^{\text{33 Seconds}} > \overbrace{\text{Decision Tree}}^{\text{19 Seconds}}$$ ־ה

ראשית ה־KNN עובד הכי לאט משום שהגדרתי לו לעבור על שליש מהשכנים בכל פעם,
ומכיוון שמספר השכנים גדל אז גם זמן החישוב שלו יגדל בצורה משמעותית.
בנוסף ה־SVM גם איטי אך פחות מה־KNN זאת מכיוון שהוא נפתר בצורה איטרטיבית,

כלומר הוא מריץ על כל הנקודות ומעדכן את כל המשקולות
קצת, וחוזר שוב לעבור על כל הנקודות. לכן הוא יהיה איטי יותר.
נשים לב ש־Logistic Regression פעל לאט כמעט כמו ה־SVM, זאת כנראה בגלל שעליו להריץ את
הפונקציה על כל נקודה ולסווגה מחדש.
לבסוף ה־Decision Tree היה המהיר ביותר, וזאת מכיוון שבכל איטרציה שלו אנחנו מקטינים את מרחב העבודה שלו
כלומר האופן שבו האלגוריתם פועל הוא שהוא חוצה את המרחב לשני חלקים בכל פעם, ואז עובד על כל חלק קטן יותר
ויותר.