[illegible] חזן
203375828

# IML תרגיל 3

1 (a)

$$TPR = \mathbb{P}_D(h_t(x)=1 \mid y=1) \underset{\text{הסתברות מותנית}}{=} \frac{\sum_{i=1}^{m} \mathbb{P}(y_i=1 \wedge h(x_i)=1)}{\sum_{i=1}^{m}\mathbb{P}(y_i=1)} \underset{\text{h בלתי תלוי ב y}}{=} \mathbb{P}(h_t(x_i)=1)$$

לכן, בהגדרת $h_t$ נקבל:

$$\mathbb{P}(h_t(x_i)=1)=\mathbb{P}(z_i \geq t)=1-t$$

ומכיוון ש $TPR=p$ נקבל: $\boxed{t=1-p}$

(b) כמו בסעיף (a) רק נקבל:

$$FPR = \frac{\sum_{i=1}^{m} \mathbb{P}(y_i=0, h(x_i)=1)}{\sum_{i=1}^{m}\mathbb{P}(y_i=0)} \underset{\text{אי-תלות}}{=} \mathbb{P}(h_t(x_i)=1) = p$$

כלומר, במקרה זה קיבלנו $FPR=TPR$, כלומר לינארי, המודל לא תלוי בכלל בהתפלגות D.

2 $\Leftarrow$ נניח $\Sigma \in \mathbb{R}^{d\times d}$ מטריצה סימטרית בעלת ע"ע אי-שליליים.

נרשום את הפירוק SVD של $\Sigma$:

$$u^T\Sigma u = u^T \underbrace{\sum_{i=1}^{d} \sigma_i v_i v_i^T}_{\Sigma} u = \sum_{i=1}^{d}\sigma_i u^T v_i \cdot v_i^T u = \sum_{i=1}^{d}\sigma_i (v_i^T u)(v_i u) =$$

$$= \sum_{i=1}^{d} \underbrace{\sigma_i}_{\geq 0} \underbrace{(v_i u)^2}_{\geq 0} \geq 0$$

כנדרש.

$\Rightarrow$ נניח שלכל $u\in\mathbb{R}^d$ שקיים $u^T\Sigma u \geq 0$, כאשר $\Sigma = \mathbb{R}^{d\times d}$

בפרט, נבחר $u=v_i^T$ ונקבל כמו בכיוון הראשון:

$$\sum_{i=1}^{d}\sigma_i (u v_i v_i^T u^T) = \sum_{i=1}^{d}\sigma_i (v_i^T v_i)(v_i^T v_i)$$

נניח, אם נניח בשלילה שקיים $\sigma_i$ eigenvalue כך ש $\sigma_i<0$ ונבחר $u=\begin{pmatrix}0\\ \vdots \\ 1 \\ \vdots \\ 0\end{pmatrix}-i$ אזי

$$u^T\Sigma u = \sum_{i=1}^{d}\sigma_i u_i^2 = \sigma_i < 0 \Rightarrow$$ סתירה

$\Leftarrow$ כלומר, כל ה eigenvalues של $\Sigma$ הם אי-שליליים ∎

③ עבור התפלגות נורמלית $N(0,\Sigma)$ נקבל

$$p(x) = \det(2\pi\Sigma)^{-0.5} \cdot \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right)$$

כאשר, $v_i$ ו- $\lambda_i$, $x^T\Sigma^{-1}x = \sum_{i=1}^{d} \lambda_i^{-1} \cdot x_i^2$

ולכן, אם נסמן: $C = p(x)$, נקבל:

$$C = \det(2\pi\Sigma)^{-0.5} \cdot \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right) \Rightarrow -2\log\left(C\cdot\det(2\pi\Sigma)^{0.5}\right) = \sum_{i=1}^{d} \lambda_i^{-1} x_i^2 =$$

$$\Rightarrow 1 = \sum_{i=1}^{d} \frac{x^2}{-2\lambda_i \cdot \log\left(C\cdot\det(2\pi\Sigma)^{0.5}\right)}$$

וקיבלנו משוואת היפר-אליפסואיד (מהצורה $(a_1x^2+a_2x^2+\dots a_dx^2$ כנדרש.

④ לפי הגדרת ה kernel:

$$\langle\psi(x),\psi(x')\rangle = \sum_{J\in[d]^k} \prod_{i=1}^{k} x_{J_i}\cdot x'_{J_i}$$

מהגדרה שקיבלנו לפי הנוסחה של האיברים כמכפלה של המונומים מדרגה $k$ עם מקדם $a=1$:

$$\sum_{\sum_{j=1}^{d} k_j = k} \binom{k}{k_1,k_2,\dots,k_d} \prod_{t=1}^{d} (x_t\cdot x'_t)^{k_t}$$

אולם זו בדיוק ההגדרה של $(\langle x,x'\rangle)^k$, ולכן:

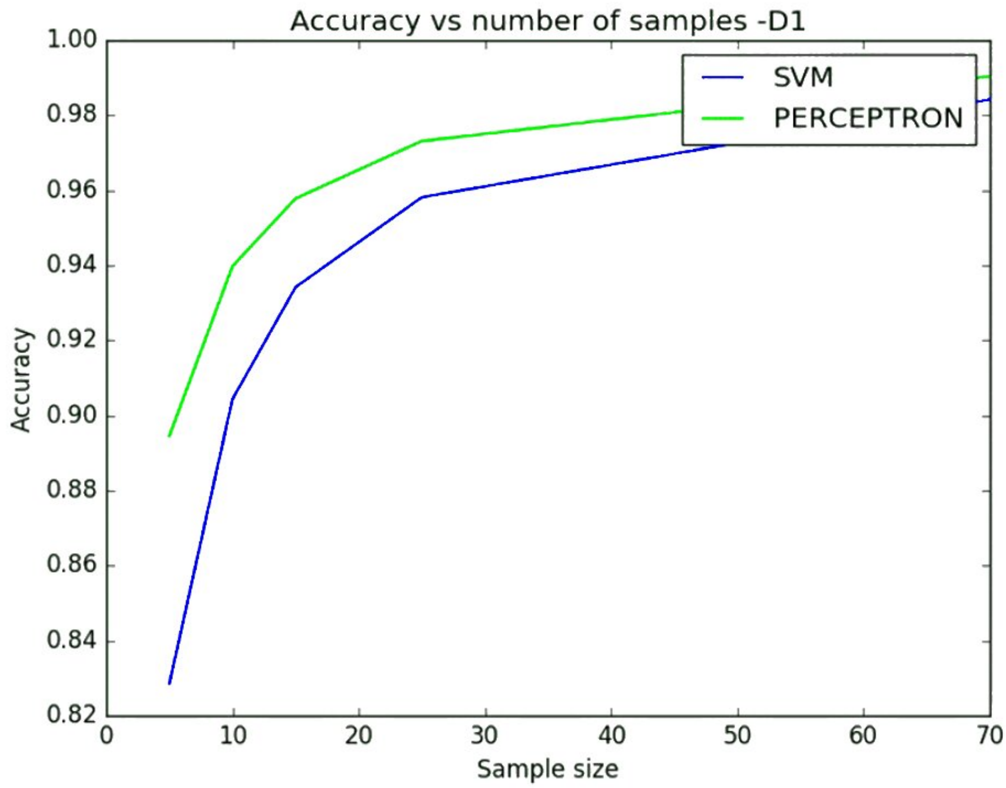
$$\langle\psi(x),\psi(x')\rangle = (\langle x,x'\rangle)^k$$

כנדרש ∎

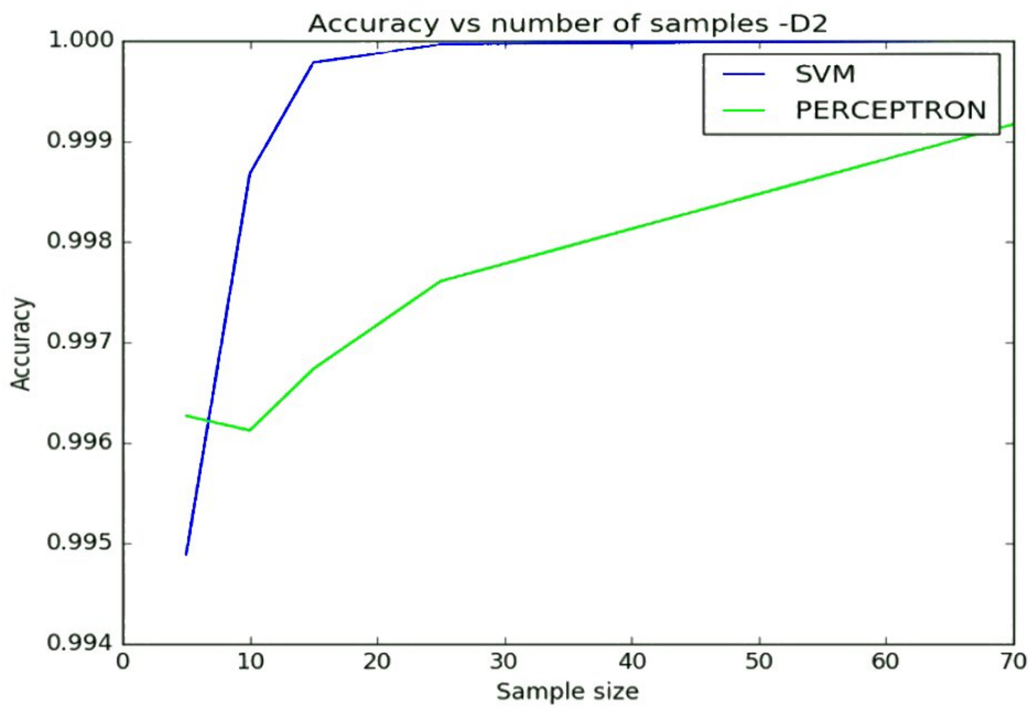
## שאלות 6-7 :

Perceptron vs SVM (See code on Perceptron.py)

**Graph 1**: Perceptron vs Svm – accuracy as a function of training size on normal Dist.




**Graph 2** : Perceptron vs SVM – accuracy as a function of training size on Rectangle Dist.



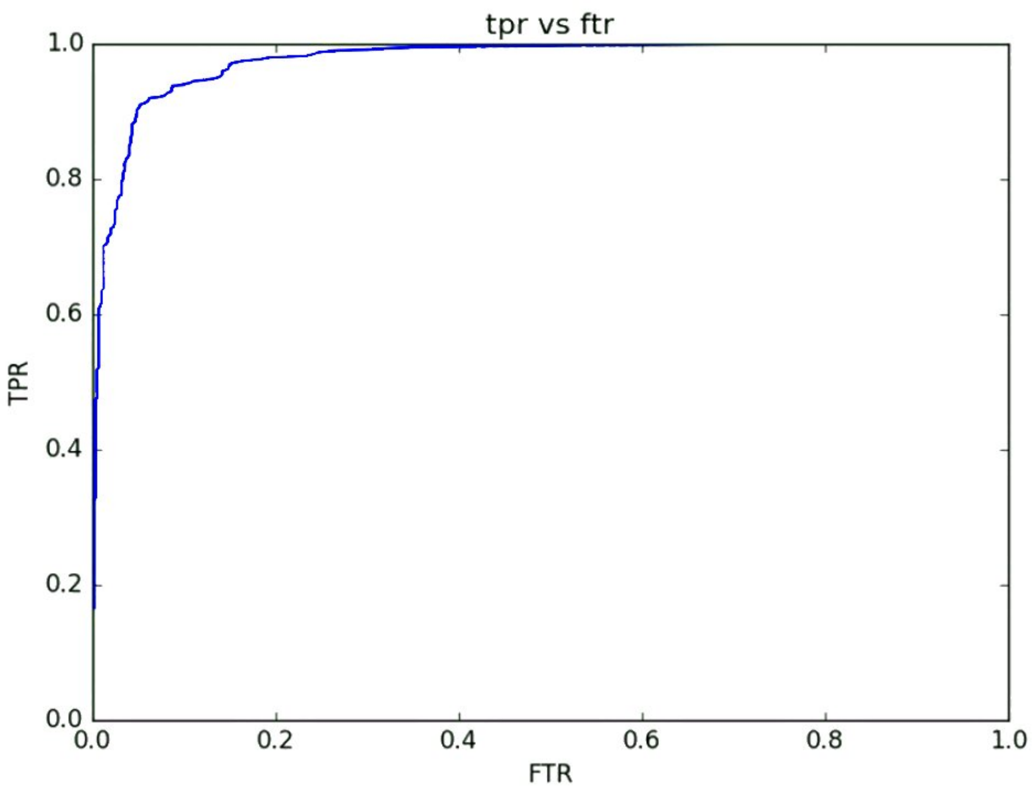
הסבר: נשים לב כי כשדוגמים מהריבועים (ההתפלגות השניה, גרף 2) יש מישור מפריד ברור בין הדגימות, ואולם בדגימות המתקבלות מהגאוסיאן (גרף 1) הואיל והדגימות צפופות ורצופות יותר, קשה יותר למצוא את הקו הספציפי שמפריד, ולכן לפרספטרון היה "קל" יותר למצוא קו מפריד בדגימות שהתקבלו מהגאוסיאן, הואיל וכל דגימה חדשה שינתה אותו במעט, כך שבקירוב, ככל שאלגוריתם הפרספטרון דוגם מחדש מאותו מדגם ומבצע התאמה ושקלול מספר רב של פעמים, אזי מידת הדיוק שלו תגדל במציאת הקו המפריד. לעומתו, אלגוריתם ה SVM- ינסה למקסם את השוליים ללא הצלחה, עקב המרחק הקטן בין הדגימות ולכן ביצועיו יהיו פחות מוצלחים

בגרף השני, ברור כי הפרספטרון משמעותית מציג תוצאות פחות טובות מבחינת הדיוק עקב המרחק הרב בין הדגימות , כלומר, כל דגימה חדשה שתתקבל , הפרספטרון יתאים את עצמו אליה (הולך אחרי הרעש) ולכן לא מתמודד טוב עם דגימות חדשות

**שאלה 8:**

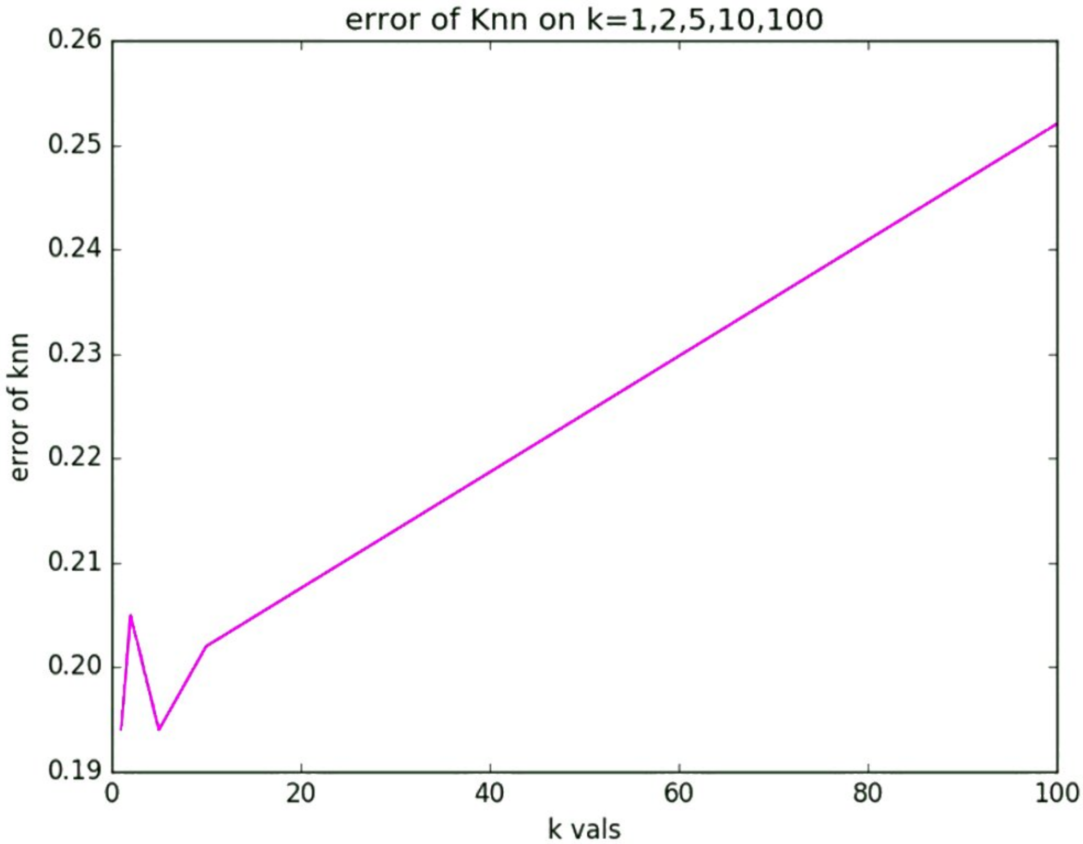
(a , b) ROC- Curve for Logistic Regression

**Graph 3**: TPR vs FPR (Code on ROC.py)



( c ) K-nearest neighbors : See code on file knn.py

**Graph 4** : K nearest – neighbors : Accuracy as a function of k




ניתן להבחין מהגרף כי עבור מספר שכנים קטן יותר נקבל את התוצאות הטובות ביותר, כלומר עלינו לקבל מידע מוקדם על אופי ההפלגות והדגימות כדי שנוכל לקבוע מהם השכנים הטובים ביותר לצורך הקלסיפיקציה , כלומר אלגוריתם כפי שמימשנו מכיל רעש רב ויתכן כי במקרים מסוימים יבחר את אותם הנקודות אך ילך אחר הרעש כשיקבל דגימה חדשה, רחוקה מהשתיים הנ"ל.