עבודה מסכמת – גישות חישוביות במדעי המח

רקע – מאגר הנתונים

בחרנו במאגר "$Rice\ Cammeo\ Osmancik\ DataSet$" [1] , המתאר שני סוגי דגנים של אורז הגדלים בטורקיה.
המאגר מכיל 3810 דוגמאות המכילות 8 משתנים:

- 7 פיצ'רים מורפולוגיים המתארים את צורת הדגן בערכים נומריים.
- משתנה תיוג קטגורלי ($Cammeo\backslash Osmancik$) לסיווג הדגן.

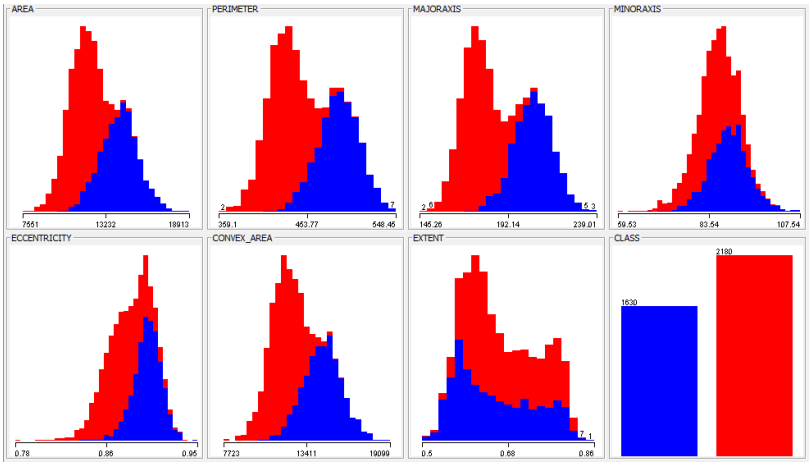
פירוט המשתנים בטבלה הבאה:

N	Variable	Details
1	Area	Returns the number of pixels within the boundaries of the rice grain.
2	Perimeter	Calculates the circumference by calculating the distance between pixels around the boundaries of the rice grain.
3	Major Axis Length	The longest line that can be drawn on the rice grain, i.e. the main axis distance, gives.
4	Minor Axis Length	The shortest line that can be drawn on the rice grain, i.e. the small axis distance, gives.
5	Eccentricity	It measures how round the ellipse, which has the same moments as the rice grain, is.
6	Convex Area	Returns the pixel count of the smallest convex shell of the region formed by the rice grain.
7	Extent	Returns the ratio of the region-formed by the rice grain to the bounding box pixels.
8	Class	Cammeo and Osmancik rice

ערכי הפיצ'רים נקבעו בעזרת תמונות של הדגנים ומיוצגים בפיקסלים, כפי שמצוין במאמר [1].
לטובת ניתוח הנתונים בוצעה המרה של התיוג הקטגורלי לתיוג בינארי:

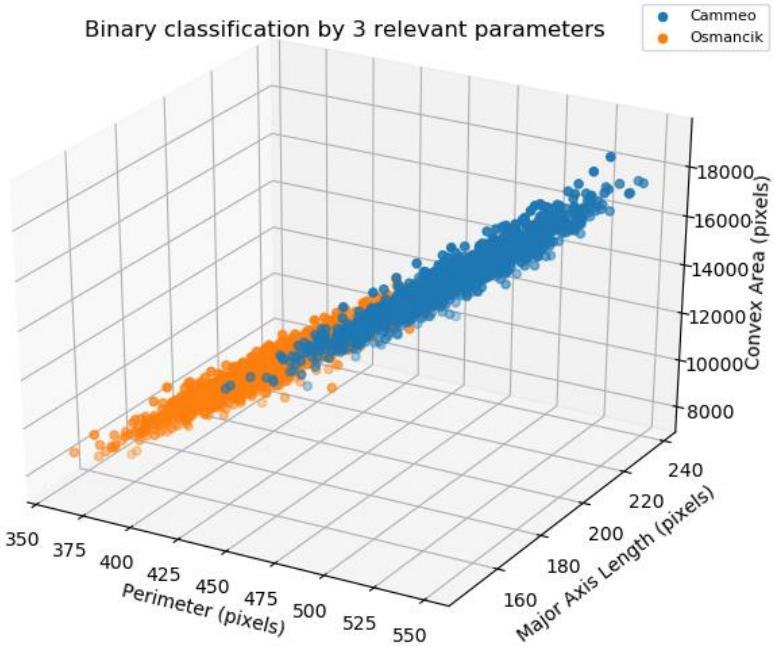
$Classification\ value$	$Class$	$\#\ Samples$
0	$Cammeo$	1630
1	$Osmancik$	2180

מכיוון שאנו לא מתמצאים בסוגי האורז, על מנת לקבל הבנה כללית של ה$DataSet$ ומשתניו השתמשנו בתצוגה ויזואלית להפרדת ה$data$ על פי כל פיצ'ר בנפרד בעזרת $weka$ [2]. בהמחשה הויזואלית בחיפוש אחר המשתנים שמפרידים את ה$data$ בצורה טובה, נראה כי המשתנים 1,2,3,6 (מהטבלה מעלה) יוכלו להיות רלוונטיים בהמשך לניבוי סוג האורז, כפי שניתן לראות בתמונה משמאל.



ב$DataSet$ אין נתונים חסרים, נראה שהמכשול העיקרי עשוי להיות חוסר איזון בכמות הדוגמאות, הן מתפלגות ביחס של 43:57.

לטובת בניית מודלים לסיווג ה$data$ בהמשך, ראשית בוצעה חלוקה של ה$DataSet$ לסט אימון וסט מבחן ביחס של 1 :9, ובוצע נרמול $zScore$ לכל הדוגמאות. על מנת להתרשם מה$data$ בחרנו להציג את דוגמאות האימון ב$scatter\ plot$ תלת מימדי בעזרת משתנים 2,3,6 בהפרדה למשתנה מנובה. ציר x מייצג את משתנה 2 ($Perimeter$), ציר y מייצג את משתנה 6 ($Convex\ Area$) וציר z מייצג את משתנה 3 ($Major\ Axis\ Length$). ניתן לראות שכבר במימד שלישי בעזרת המשתנים הנ"ל קיימת הפרדה מסוימת הנראית לעין בין שני המשתנים המנובים.



שיטות אימון ובניית מסווגים

על מנת לסווג את ה$Data$ בחרנו ב3 שיטות קלסיפיקציה - $SVM,\ Logistic\ regression,\ NN$.
כדי לבחון את מידת ההצלחה של כל שיטת סיווג על ה$data$, ראשית בחרנו היפר-פרמטרים סבירים, פונקציית $Loss$ פשוטה ומדד $Accuracy$ להערכת המודל. לאחר מכן, בעזרת כל שיטת קלסיפיקציה חילקנו את ה$TrainingSet$ ל5 קבוצות שוות בגודלן, וביצענו למידה בשתי דרכים:

- סט ולידציה בודד: למידה על 4 מתוך 5 הקבוצות ואבלואציה על הקבוצה הנותרת.
- $Cross-validation$ ומיצוע.

בשלב זה חזרנו על התהליך עם מקדם רגולריזציה, ולבסוף השתמשנו בתוצאות כדי לבצע $tunning$ נוסף להיפר פרמטרים הראשוניים שבחרנו לטובת מודל סופי.

כל המודלים שנציג משתמשים במדד דיוק להערכת המודל:

$$Accuracy=\frac{1}{N}\sum_{k=1}^{N}I(y_k,\widehat{y_k})\ \ ,\ \ where\ N=Number\ of\ Samples\ ,\ \ I(y_k,\widehat{y_k})=\begin{cases}1, & y_k=\widehat{y_k}\\ 0, & else\end{cases}$$

הערכת המודל מבוצעת כפונקציה של משך הלמידה, לפי מס' דוגמאות מתוך סט האימון, באחוזים: $20\%\ ,40\%,60\%,80\%,100\%$. הערכת המודל מבוצעת בעזרת מעקב אחר הביצועים על סט האימון והביצועים על סט הולידציה, בעזרת מדד הדיוק כפי שפורט לעיל, שהוא המדד המשלים ל$error$: $acc(x)=1-err(x)$ (הופכיים אחד לשני)

לטובת מימוש המודלים השתמשנו בחבילת $sklearn$, הקוד נכתב ב$python$.

$SVM\ Classification$

עבור מסווג SVM בחרנו בשימוש ב$kernel$ פולינומי, מהצורה הבאה:

$$K(x_i, x_j) = (r + \gamma \cdot x_i x_j)^d$$

כאשר עבור סיפוק ההיפר-פרמטרים בתנאי הראשוני השתמשנו בערכים $d = 1, r = 0, \gamma = 1$ לקבלת פונקציית ה$kernel$: $K(x_i, x_j) = x_i \cdot x_j$ (המכפלה הפנימית הסטנדרטית). עבור המסווג בחרנו בפונקציית $hingeLoss$:

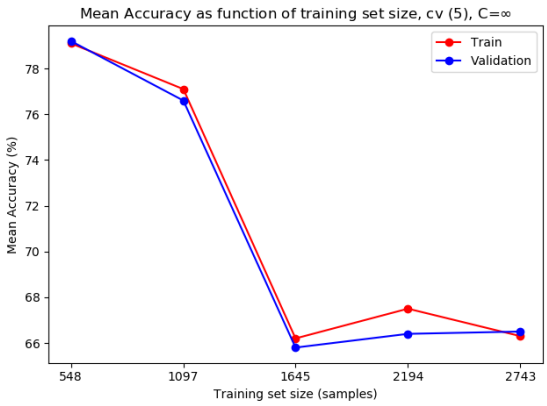
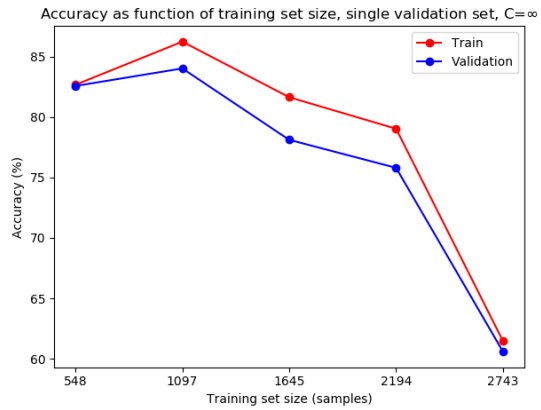
$$Hinge - Loss(x, \theta, y) = \max\{0, 1 - y\theta^T x\}$$

בעיית האופטימיזציה שמנסים לפתור :

$$\min Loss(\theta) = \min\left[\frac{1}{N}\sum_i \max(0, 1 - y_i\theta^T x_i) + C \cdot \sum_i \theta_i^2\right]$$

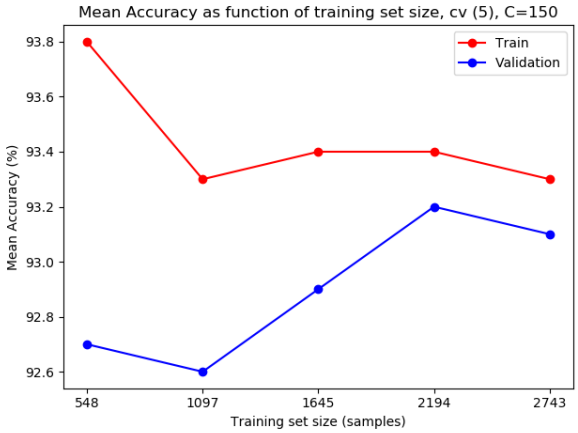
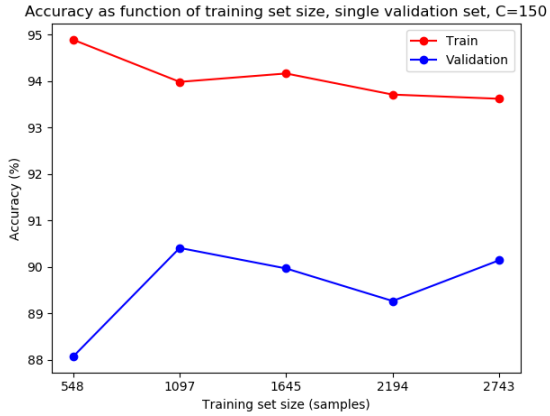
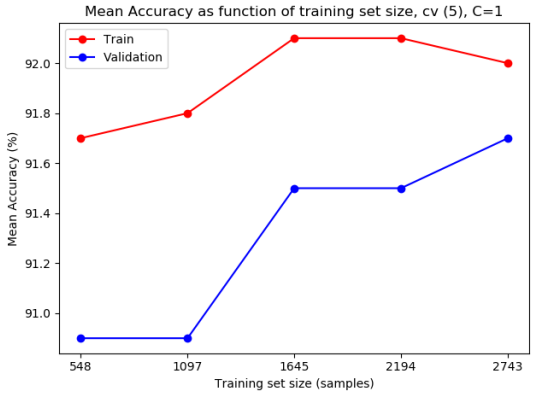
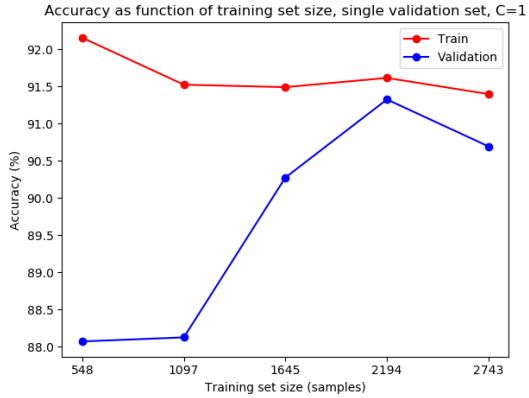
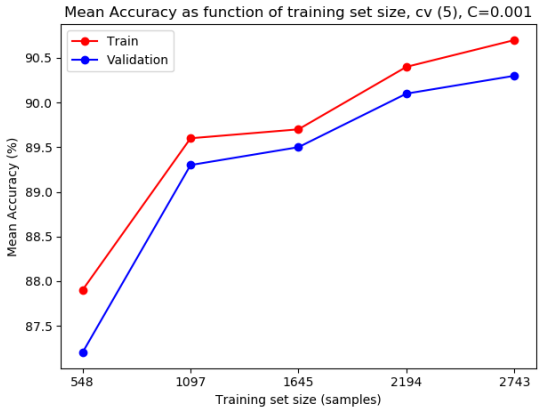
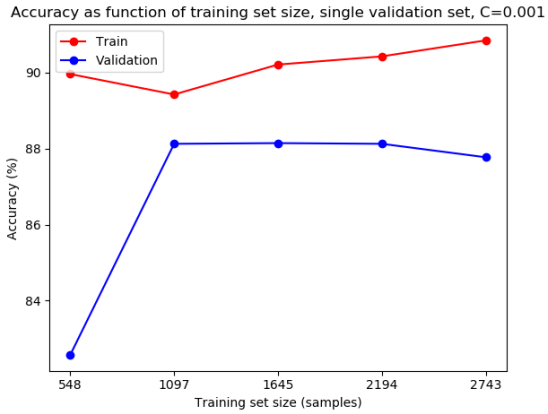
בשלב הראשון ניסינו לבחון את המודל ללא רגולריזציה, קיבענו $C \leftarrow \infty$ על מנת לא לאפשר דוגמאות סוררות.
במצב זה עבור הפרמטרים הראשוניים שבחרנו המודל לא התכנס, לכן הקטנו את C להיות 150, ועבורו בחנו את המודל – זה עדיין מקדם גדול מאד שלא מאפשר הרבה טעויות על הדוגמאות ויכול הגיע ל$overfitting$. הגבלנו את מס' האיטרציות ל$2{,}000{,}000$. בשלב השני הוספנו מקדמי רגולריזציה נוספים, $C \in \{1, 0.001\}$, ובדקנו האם מדד הערכת המודל משתפר בתוספת מקדמים בערך נמוך יותר, שיאפשרו יותר דוגמאות סוררות.
שיטת הרגולריזציה של המודל היא $ridge\ regularization$ ($C \cdot \Sigma\, \theta_i^2$), נציג את מדד הערכת המודל כפונקציה של מס' הדוגמאות שסיפקנו למודל.

בטבלה הבאה מוצגות התוצאות שהתקבלו עבור $C \to \infty$. כיוון שעבור $C \to \infty$ המודל לא מתכנס, אחוזי הדיוק עבור C זה נמוכים (המודל לא מגיע לפיתרון).

C	$cross\ validation\ (split\ to\ 5)$	$single\ validation\ set\ \ (4:1)$
∞		

כעת לתוצאות עבור $C \in N^+$.
עבור $C = 150$ ניתן לראות בשורה הראשונה בטבלה שמקבלים $over\ fitting$ על הדוגמאות: אחוזי הדיוק גבוהים מאד על סט האימון ופחות על סט הולידציה, זאת כיוון שמקדם הרגולריזציה הוא גדול ומדמה מצב "חסר רגולריזציה". רואים זאת בעיקר בלמידה שמבוצעת על 80% מה$data$ ומוערכת על סט ולידציה בודד, שבה אין

מיצוע וקיים חוסר איזון יחסי, לעומת כאשר הלמידה היא בעזרת $cross\ validation$ שמאזנת מעט את התוצאות על סט האימון. עבור C קטן יותר ניתן לראות כיצד שימוש ברגולריזציה מסייע ללמידה – בשורה השניה בטבלה רואים למידה עם מקדם רגולריזציה $C = 1$, ובשורה השלישית עם מקדם רגולריזציה של $C = 0.001$. עבור $C = 0.001$ רואים ירידה באחוזי הדיוק לעומת $C = 1$, כיוון שאנחנו מאפשרים יותר טעויות על דוגמאות סוררות, ולכן ייתכן שהוא מקדם נמוך מדי. מהתוצאות עושה רושם ששימוש במקדם $C = 1$ מביא לאחוזי דיוק טובים עבור מסווג SVM ל$data$ שלנו.

C	$cross\ validation\ (split\ to\ 5)$	$single\ validation\ set\ \ (4:1)$
150		
1		
0.001		

על מנת לבחור היפר-פרמטרים טובים ולמקסם את יכולות המודל נשתמש במקדם $C = 1$ שמצאנו לטובת הצגת $Grid\ search$ עבור 3 הפרמטרים של $kernel$ פולינומי : d דרגת הפולינום, r משתנה חופשי, γ מקדם למכפלה הפנימית. נציג בטבלה את אחוזי הדיוק שקיבלנו עבור הפרמוטציות השונות של משתנים אלה בערכים:

$d \in \{1, 2, 3, 4, 5\}$
$\gamma \in \{0.5, 1, 2, 4\}$
$r \in \{-1, 0, 1\}$

מתוך כלל האפשרויות המודל יבחר את הפרמטרים הטובים ביותר.

#	r	degree	γ	Accuracy	#	r	degree	γ	Accuracy
1	1	1	1	93.1%	31	0	4	0.5	81.9%
2	0	1	1	93.1%	32	-1	5	0.5	81.9%
3	-1	1	1	93.1%	33	0	2	2	80.2%
4	1	1	4	93.1%	34	0	2	1	79.7%
5	0	1	4	93.1%	35	1	3	2	77.9%
6	-1	1	4	93.1%	36	0	2	0.5	77.5%
7	1	1	0.5	93.0%	37	1	5	1	77.2%
8	0	1	0.5	93.0%	38	0	2	4	75.9%
9	-1	1	0.5	93.0%	39	0	5	1	75.4%
10	1	1	2	93.0%	40	0	4	1	74.9%
11	0	1	2	93.0%	41	0	3	2	72.7%
12	-1	1	2	93.0%	42	1	4	4	72.6%
13	1	2	0.5	92.8%	43	1	5	2	72.6%
14	1	2	1	92.8%	44	1	4	1	72.3%
15	1	2	2	92.8%	45	1	5	4	71.2%
16	1	3	0.5	92.8%	46	0	5	2	70.8%
17	1	4	0.5	92.3%	47	0	5	4	70.8%
18	0	3	0.5	91.7%	48	1	4	2	70.4%
19	0	3	1	91.6%	49	0	3	4	68.2%
20	1	2	4	91.5%	50	1	3	4	67.0%
21	1	3	1	91.5%	51	0	4	2	66.6%
22	1	5	0.5	90.2%	52	0	4	4	64.9%
23	-1	3	4	87.7%	53	-1	4	4	21.8%
24	0	5	0.5	86.9%	54	-1	4	2	19.4%
25	-1	3	2	86.4%	55	-1	2	4	18.5%
26	-1	3	1	85.0%	56	-1	4	1	17.3%
27	-1	3	0.5	84.6%	57	-1	2	2	17.1%
28	-1	5	2	84.6%	58	-1	4	0.5	17.1%
29	-1	5	4	84.3%	59	-1	2	0.5	16.4%
30	-1	5	1	82.9%	60	-1	2	1	16.4%

Logistic Regression

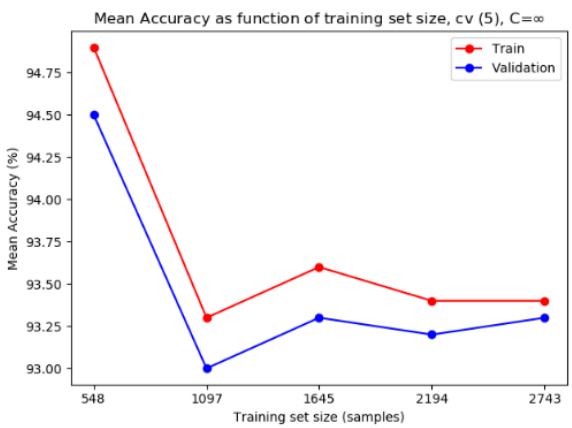
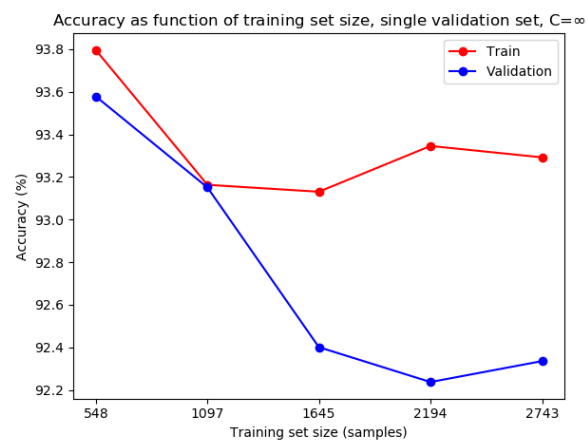
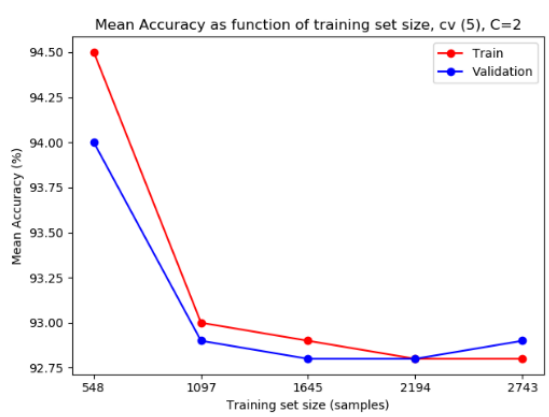
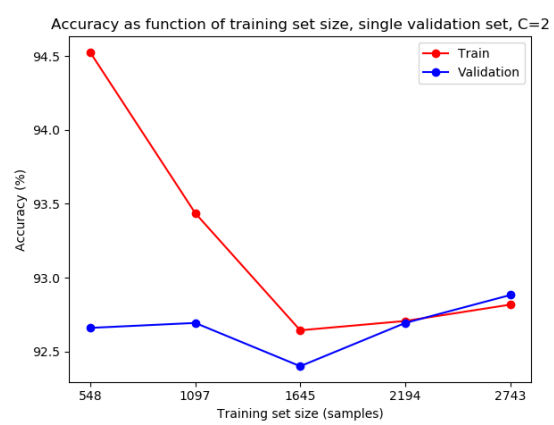
ביצוע $logistic\ regression$ למציאת קו הרגרסיה. המסווג משתמש בפונקציית $loss$:

$$loss(x,\theta,y) = \begin{cases} -\log\left(g(x)\right) & ,y=1 \\ -\log\left(1-g(x)\right) & ,y=0 \end{cases} \quad , \qquad where\ g(x) = \frac{1}{1+e^{-\theta x}}$$

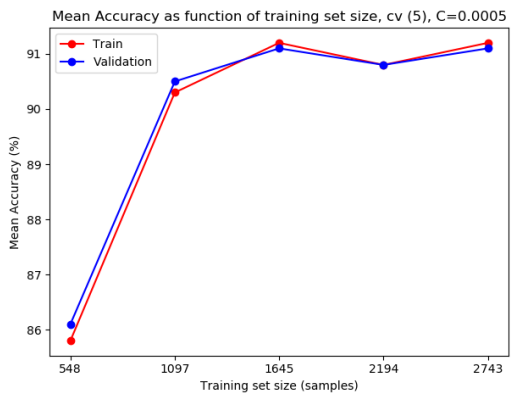
בעיית האופטימיזציה שמנסים לפתור :

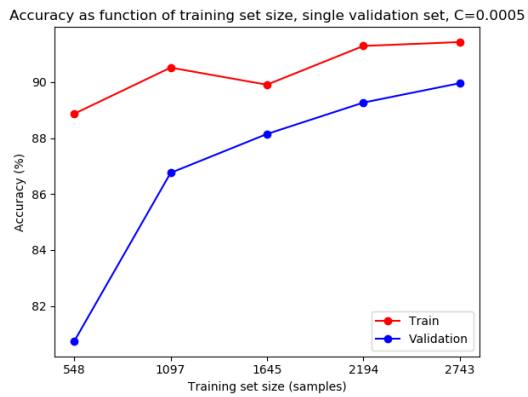
$$\min Loss(\theta) = \min\left[-\frac{1}{N}\sum_i y_i\, log\big(g(x_i)\big) + (1-y_i)\, log\big(1-g(x_i)\big)\right] + C\cdot\sum_i \theta_i^2$$

גם כאן בחנו את המודל ללא רגולריזציה, במודל lr של $sklearn$ [3] ניתן לממש רגרסיה לוגיסטית ללא רגולריזציה על ידי שימוש בערך $None$ במקום $\sum_i f(\theta_i)$ כלשהו. עשינו זאת, והגבלנו את מס' האיטרציות ל20,000, זה הספיק כדי שהמודל התכנס בהרצות שביצענו. כשהוספנו מקדם רגולריזציה בחרנו את המקדמים $\{2, 0.0005\}$, גם כאן השתמשנו בשיטת $ridge\ regularizaion$ להוספת רגולריזציה.
נציג את מדד הערכת המודל כפונקציה של מס' הדוגמאות שסיפקנו למודל.

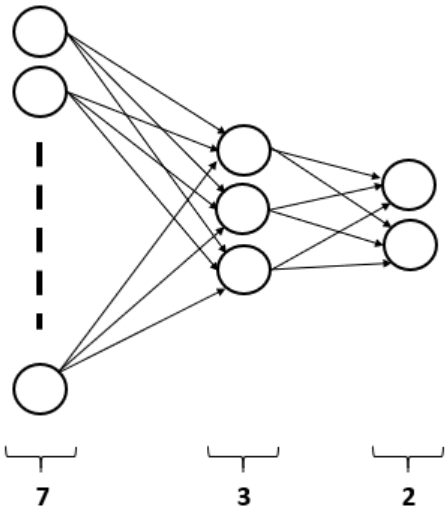
C	$cross\ validation\ (split\ to\ 5)$	$single\ validation\ set\ \ (4:1)$
$None$		
2		

0.0005





השימוש ב$cross\ validation$ במודל הנוכחי לא השפיע בצורה ניכרת על התוצאות, אך ראינו שללא שימוש ברגולריזציה, בסט ולידציה בודד אחוזי הדיוק על סט הולידציה מתחילים לרדת ככל שמשתמשים ביותר דוגמאות, בעוד שבסט האימון האחוזים יותר יציבים, דבר שייתכן שמצביע על $over\ fitting$. באשר לשימוש במקדם רגולריזציה, עבור ה$dataset$ שלנו ניתן להעריך מהתוצאות ששימוש ברגולריזטור או חוסר שימוש מביאים לביצועים די דומים, כל עוד קבוע הרגולריזציה בגודל סביר (למשל, בגרפים מעלה הוצג שימוש ב $C = 2$). עם זאת, עבור קבוע קטן ($C \leq 0.0005$) אחוזי הדיוק על הדוגמאות מתחילים להיפגע כיוון שמאפשרים יותר מדי דוגמאות סוררות.



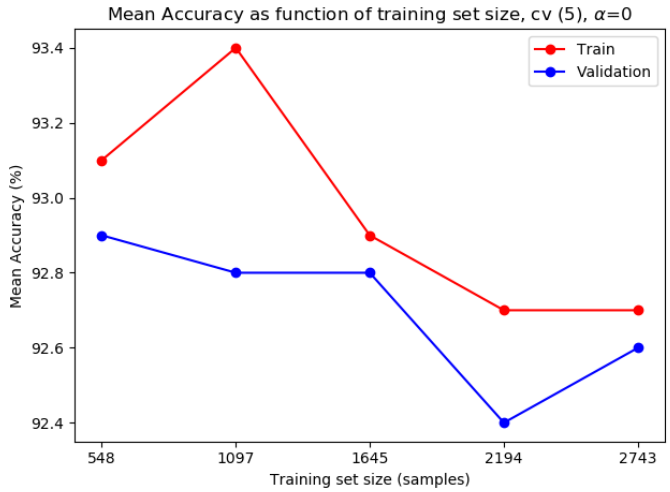
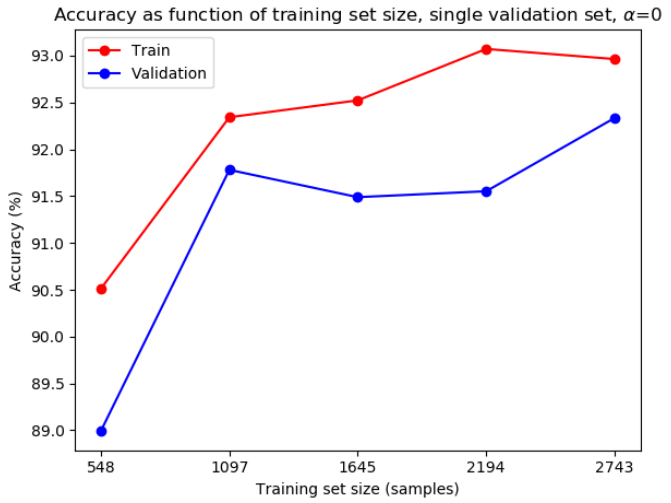
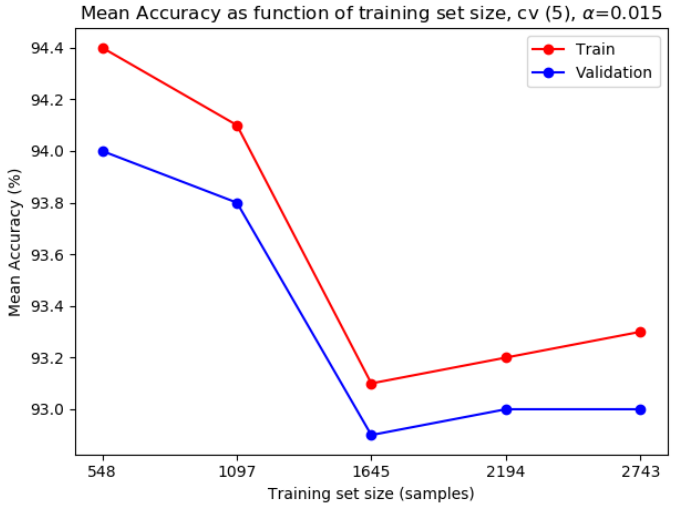
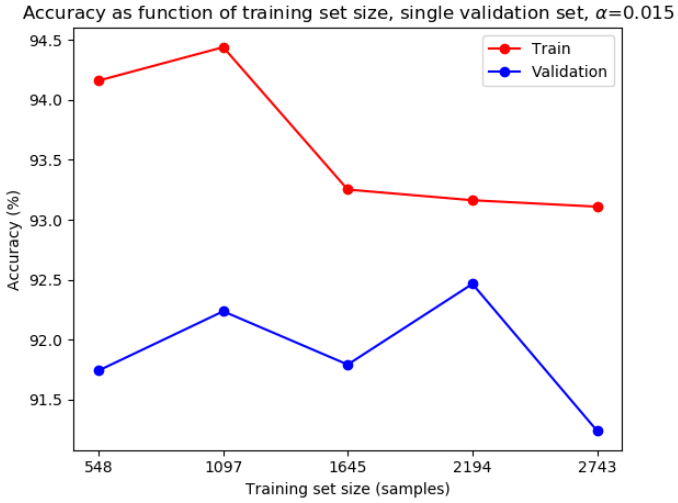
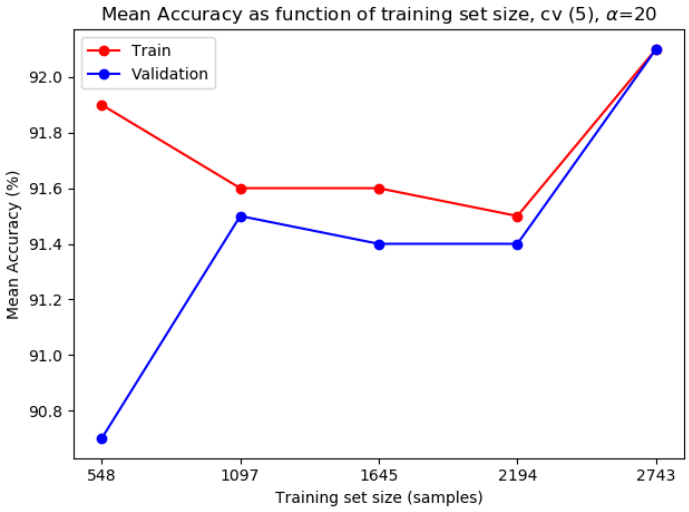
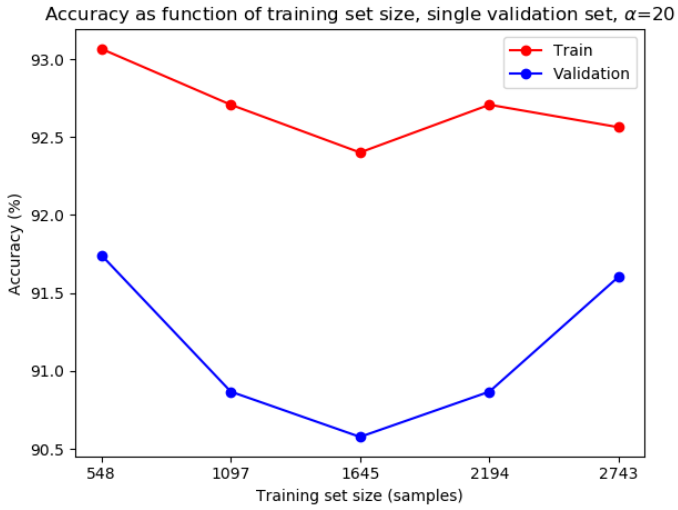
$\boldsymbol{Neural\ Network}$

בנינו רשת נוירונים בעלת 3 שכבות – שכבת קלט בעלת 7 נוירונים (כמס' הפיצ'רים), שכבת ביניים ($hidden\ layer$) עם 3 נוירונים ושכבת פלט בעלת 2 נוירונים (כמס' אפשרויות התיוג). השכבות הן $fully\ connected$ [5].

לטובת היפר-פרמטרים ראשוניים לאתחול המודל, בחרנו בפונקציית אקטיבציה מסוג $relu$ שפועלת על שכבת הקלט ושכבת ה $hidden$, ועל שכבת הפלט פועלת פונקציית $softmax$ לקבלת הסתברויות על התיוגים. המודל פועל בשיטת sgd. הגבלנו את מס' האיטרציות ל20,000 אם המודל לא התכנס. הערך ההתחלתי עבור $momentum$ היה 0.9, ומקדם הלמידה ($learning\ rate$) היה 0.001.

במודל של רשת נוירונים מקדם הרגולריזציה הוא α, ערכו הדיפולטיבי במודל nn של $sklearn$ הוא 0.0001 והוא מתפקד כהופכי למקדם C, כך שמתקיים $C = \frac{1}{\alpha}$ [4]. לכן, על מנת לבחון את המודל ללא שימוש ברגולריזציה, כיוון ש $C \to \infty \ \Rightarrow \ \alpha \to 0$ השתמשנו ב $\alpha = 0$. לאחר מכן, על מנת לבחון את הרשת עם מקדם רגולריזציה ($ridge\ regularization,\ \Sigma{\theta_i}^2$) בחרנו במקדמים $\{0.015,20\}$. המודל התכנס לכל ערכי המקדם שבחרנו. נציג את מדד הערכת המודל כפונקציה של מס' הדוגמאות שסיפקנו למודל.

מהתוצאות שקיבלנו, המודל מציג אחוזים יפים על סט הולידציה עם ובלי רגולריזציה, אך מקדם רגולריזציה טוב ($\alpha = 0.015$) מסייע להעלאת אחוזי הדיוק על סט הולידלציה כשמשתמשים ב $cross\ validation$.
לכל $\alpha \geq 20$ המודל מאפשר יותר מדי טעויות ואחוזי הדיוק על סט הולידציה מתחילים לרדת.

α	*cross validation (split to 5)*	*single validation set* (4:1)
0		
0.015		
20		

על מנת לבחור היפר-פרמטרים טובים ולמקסם את יכולות המודל נשתמש במקדם $C = 0.015$ שמצאנו לטובת הצגת $Grid\ search$ עבור 3 הפרמטרים הבאים של רשת נוירונים: בחירת פונקציית אקטיבציה, מקדם הלמידה ($learning\ rate$), ומומנטום (תחת sgd). נציג בטבלה את אחוזי הדיוק שקיבלנו עבור הפרמוטציות השונות של משתנים אלה בערכים:

$$Activation\ function \in \{relu, tanh, logistic, identity\}$$
$$Momentum \in \{0.5, 0.8, 0.9, 0.99\}$$
$$lr \in \{0.01, 0.001, 0.0001\}$$

כאשר:

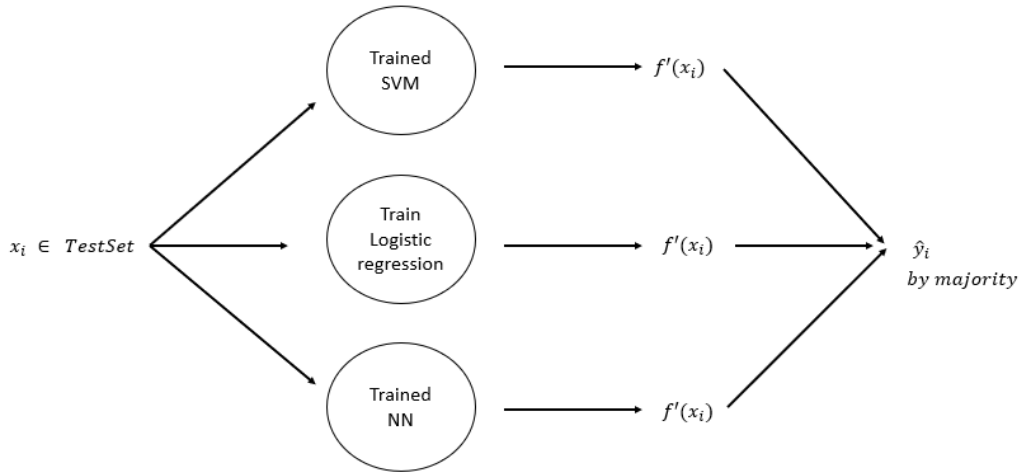
$$relu(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$
$$tahn(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$
$$logistic(x) = sigmoid(x) = \frac{1}{1 + e^{-x}}$$
$$identity(x) = x$$

מתוך כלל האפשרויות המודל יבחר את הפרמטרים הטובים ביותר.

#	Activation	lr	Momentum	Accuracy	#	Activation	lr	Momentum	Accuracy
1	relu	0.01	0.8	**0.93**	25	identity	0.0001	0.99	**0.927**
2	tanh	0.001	0.99	**0.929**	26	tanh	0.01	0.99	**0.926**
3	identity	0.01	0.5	**0.929**	27	tanh	0.01	0.9	**0.926**
4	identity	0.001	0.9	**0.929**	28	tanh	0.001	0.8	**0.926**
5	tanh	0.0001	0.99	**0.928**	29	relu	0.001	0.9	**0.926**
6	relu	0.01	0.9	**0.928**	30	identity	0.01	0.99	**0.926**
7	logistic	0.01	0.9	**0.928**	31	tanh	0.001	0.5	**0.925**
8	logistic	0.01	0.5	**0.928**	32	relu	0.001	0.5	**0.925**
9	logistic	0.001	0.99	**0.928**	33	identity	0.001	0.5	**0.925**
10	logistic	0.0001	0.99	**0.928**	34	relu	0.01	0.5	**0.924**
11	identity	0.001	0.8	**0.928**	35	logistic	0.001	0.5	**0.924**
12	tanh	0.01	0.8	**0.927**	36	relu	0.001	0.8	**0.923**
13	tanh	0.01	0.5	**0.927**	37	identity	0.0001	0.9	**0.922**
14	tanh	0.001	0.9	**0.927**	38	tanh	0.0001	0.9	**0.921**
15	relu	0.01	0.99	**0.927**	39	logistic	0.0001	0.8	**0.921**
16	relu	0.001	0.99	**0.927**	40	tanh	0.0001	0.8	**0.92**
17	relu	0.0001	0.99	**0.927**	41	logistic	0.0001	0.9	**0.92**
18	logistic	0.01	0.99	**0.927**	42	identity	0.0001	0.8	**0.92**
19	logistic	0.01	0.8	**0.927**	43	identity	0.0001	0.5	**0.917**
20	logistic	0.001	0.9	**0.927**	44	relu	0.0001	0.8	**0.913**
21	logistic	0.001	0.8	**0.927**	45	tanh	0.0001	0.5	**0.911**
22	identity	0.01	0.9	**0.927**	46	relu	0.0001	0.9	**0.911**
23	identity	0.01	0.8	**0.927**	47	relu	0.0001	0.5	**0.909**
24	identity	0.001	0.99	**0.927**	48	logistic	0.0001	0.5	**0.505**

הערכת המודלים על סט המבחן

בידינו $test\ set$ בגודל 10% מכלל הדוגמאות. נרצה לבדוק את מידת הצלחת הסיווג של המודלים שיצרנו על סט המבחן הזה. על מנת לעשות זאת, הרצנו כל אחד מהמודלים בנפרד על סט האימון (שלב האימון), כחלק ממנו בוצעה הרצת $grid\ search$ לקבלת היפר פרמטרים טובים ביותר. לאחר מכן, עם המודל שהתקבל ביצענו חיזוי לדוגמאות מסט המבחן, ובדקנו את אחוז הדיוק. חזרנו על התהליך 100 פעמים, ומיצענו את התוצאות. בנוסף, ייצרנו מודל $Ensemble$ בשיטת "הרוב קובע" לסיווג דוגמאות סט המבחן על פי הסיווגים המתקבלים משלושת המודלים. להלן תרשים של הרצת 1 מתוך 100 ההרצות למיצוע:



מכיוון שה $dataset$ שלנו מבצע הכרעה בינארית, בשיטת "הרוב קובע" על דוגמא x_j :

$$\hat{y}_j = 1 \ \Leftrightarrow \text{at least } {}^{2}/_{3}\ models\ declare\ f'(x_j) = 1$$

להלן התוצאות הממוצעות שקיבלנו עבור כל מודל בנפרד ועבור מודל ה $Ensemble$:

$Model$	$AVG\ Accuracy\ on\ TestSet$
SVM	$93.175\ \%$
$Logistic\ Regression$	$93.438\ \%$
NN	$93.042\ \%$
$Ensebmle$	$93.209\ \%$

ניתן לראות שאחוזי הדיוק של ה$Ensemble$ דומים לשל שאר המודלים. קשה להצביע על מודל אחד שסיפק תוצאות טובות משמעותית משל שאר המודלים, כל הביצועים טובים באופן יחסי.

רפרנסים

[1] CINAR, I. and KOKLU, M., (2019). “Classification of Rice Varieties Using Artificial Intelligence Methods.” *International Journal of Intelligent Systems and Applications in Engineering, 7*(3), 188-194.

[2] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explor. Newsl. 11 (1) (2009) 10–18.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[4] https://scikit-learn.org/stable/modules/svm.html#svc

[5] https://scikit-learn.org/stable/modules/neural_networks_supervised.html#neural-networks-supervised