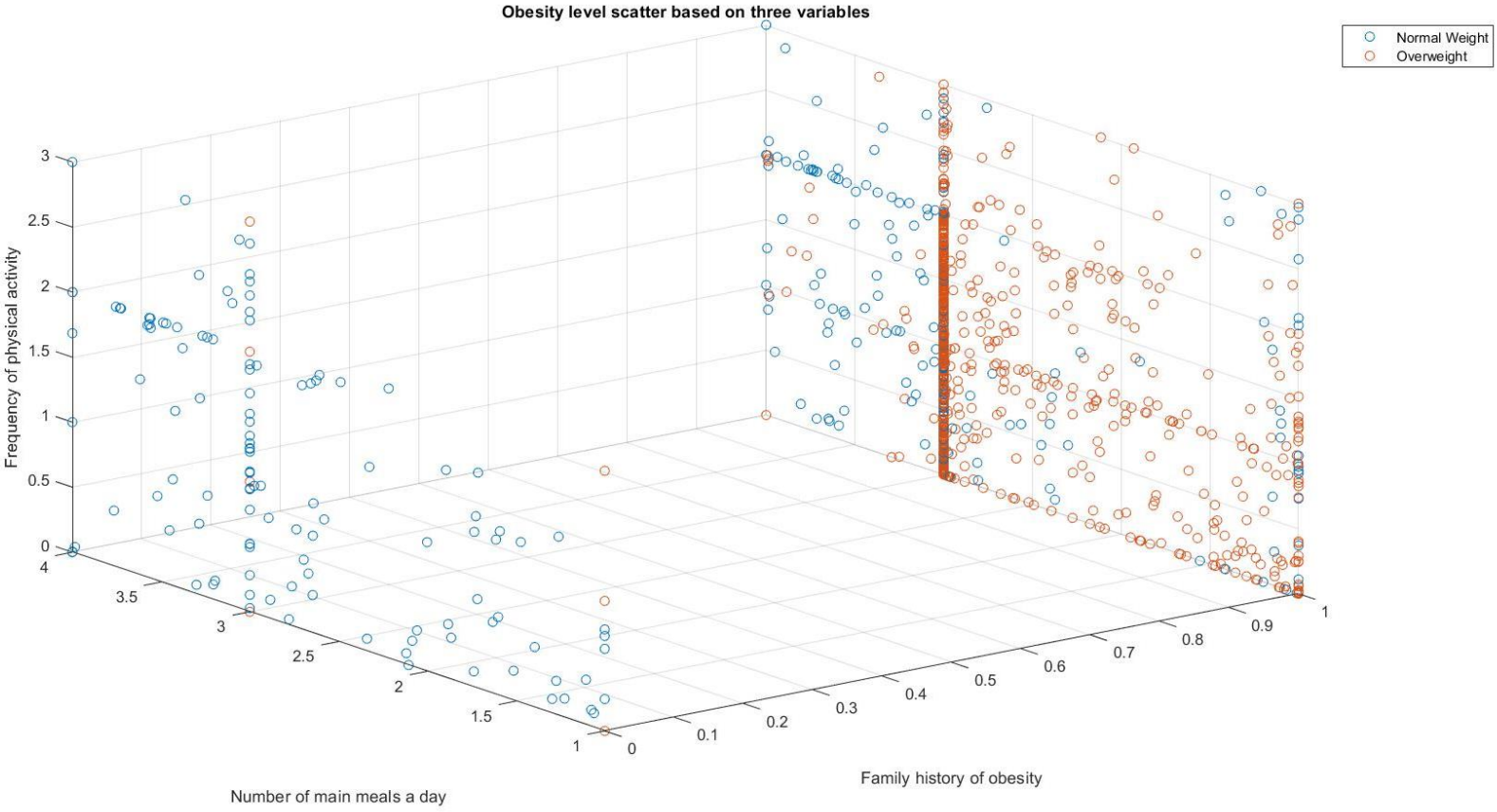
רננה מובאס 206698441
מור דניאלי-זולוטוב 205612484
שחף קובל 203962717

<u>תרגיל מסכם – קורס גישות חישוביות מס' 27-431-01 (הרצת הקוד 'mainScript.m' תריץ את כל התרגיל – צריך מטלב 2019/2020) ***הדאטה אצלנו עובר רנדומליזציה, לכן הגרפים והתוצאות יצאו מעט שונים בכל הרצה*</u>**

1) המאגר מכיל משתנים שונים הקשורים לאורח החיים של הנבדק. יש מגוון משתנים מסוגים שונים לדוגמה גיל כמשתנה רציף, מין כמשתנה בינארי, שימוש באמצעי תחבורה כמשתנה בדיד, עישון כמשתנה בדיד וכו'. סך הכל יש כשישה עשר משתנים מסוגים שונים, המשתנים הללו מטרתם היא לחזות משתנה אחד שהוא רמת הObesity. אנחנו בחרנו להכניס ללמידת המודל את כל המשתנים למעט – גיל, מין, גובה ומשקל.
רמת הobesity המשתנה אותו אנו מנבאים, הינו משתנה בינארי כאשר 0 משמעותו משקל תקין ו1 משקל עודף. לדעתנו המשתנים הרלוונטיים ביותר לניסוי הינם – היסטוריה משפחתית של השמנת יתר, מספר ארוחות עיקריות ביום ותדירות פעילות גופנית.

2) נעשה בקוד – חלוקה ל 20% סט מבחן ו80% סט אימון. **(פונקציית 'prepareData.m')**

3) בחרנו במשתנים של היסטוריה משפחתית של השמנת יתר, מספר ארוחות עיקריות ביום ותדירות פעילות גופנית. בגרף ניתן לראות כי היסטוריה משפחתית מצליחה להפריד בצורה טובה יחסית בין משקל תקין למשקל עודף. בנוסף רואים כי רוב האנשים בעלי משקל עודף אוכל בין 1 ל3 ארוחות עיקריות ביום, בעוד משקל תקין מפוזרים יחסית על הגרף אבל ניתן לראות ריכוז טיפה יותר גדול בין 3 ל4 ארוחות. ניתן להסיק מהגרף כי שלושה משתנים אינם מספיקים לסיווג של דאטה זה, לכן יש צורך להשתמש בעוד משתנים על מנת לסווג את הדאטה טוב יותר.
גרף scatter – (פונקציית 'plotScatter.m')



רננה מובאס 206698441
מור דניאלי-זולוטוב 205612484
שחף קובל 203962717

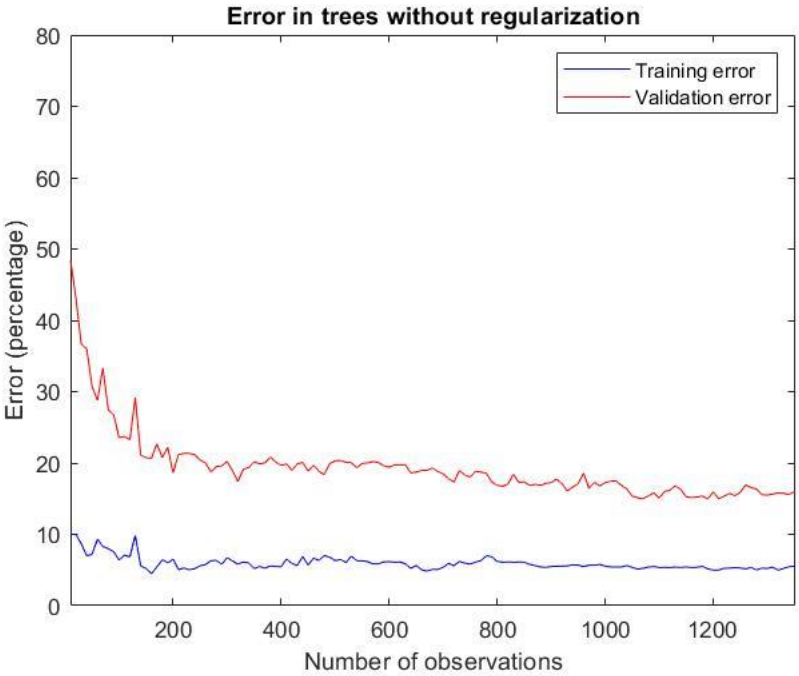
שיטת קלסיפיקציה : עצי החלטה – פונקציה 'decisionTreeModel.m'

4) לדעתנו פונקציית השגיאה המתאימה בשיטה זו הינה מיסקלסיפיקציה, בה על כל נקודה שסווגה לא נכון יש עלייה של נקודה אחת בשגיאה. פונקציית שגיאה זו עלולה לגרום למודל לעשות overfitting, היות ותמיד תהיה שגיאה אם יהיו נקודות שמסווגות לא נכון. לכן, בחרנו היפרפרמטרים מתאימים אשר יגבילו את הoverfitting של המודל. בחרנו Accuracy כמדד להצלחת המודל, הוא מציג את אחוז הדוגמאות אשר מסווגות נכון במודל.
רשימת ההיפרפרמטרים שבחרנו וההסבר:

- **אלגוריתם פרדיקציה של הפיצול הטוב ביותר** – 'PullLeft'
- מדד להערכת פיצול של ענף – Gini הוא המדד שבחרנו. הוא בודק מהו הפיצול שיניב לנו את החלוקה הטובה ביותר של הדאטה.
- **מספר מקסימלי של פיצולים** – השארנו את ברירת המחדל שהיא [1 – # of obsevations].
- **מספר מינימלי של פרדיקציות בleaf** – השארנו את ברירת המחדל שהיא מינימום תצפית אחת לעלה.
- **מספר מינימלי של פרדיקציות בbranch node** – השארנו את ברירת המחדל שהיא מינימום 10 תצפיות לענף.

הסיבה שהשארנו את אפשרויות ברירת המחדל של מטלב היא שרצינו שהמודל שלנו לא יעבור רגולריזציה ויוכל להתאמן על האימון ללא הפרעה. ברירות המחדל של מטלב הינם המספרים המינימלים על מנת שלא תעשה רגורליזציה והגבלה של למידת המודל (כאשר נדרשנו להוסיף רגולריזציה הגדרנו מספרים אלו על מנת להקשות על למידת המודל).

חילקנו את סט האימון לחמש סאבסטים (5 fold) ועשינו קרוס וולידציה. על מנת לחשב את שגיאת האימון השתמשנו בפונקציה מובנית של מטלב שנקראת 'resubloss'. חישוב של שגיאת הוולידציה גם הוא נעשה בעזרת פונקציה מובנית של מטלב הנקראת 'loss'. על מנת למצוא הבדלים בין הלמידה של חמש הסאבסטים חישבנו שונות של השגיאות. קיבלנו שונות שהיא 0.04 לשגיאת האימון ו0.08 לשגיאת הוולידציה. זה אומר כי הלמידה של הסאבסטים דומה מאוד. לכן על מנת להציג את גרף שגיאת האימון ושגיאת הוולידציה, עשינו ממוצע לשגיאות האימון של כל חמשת הסאבסטים וכך גם לשגיאת הוולידציה.



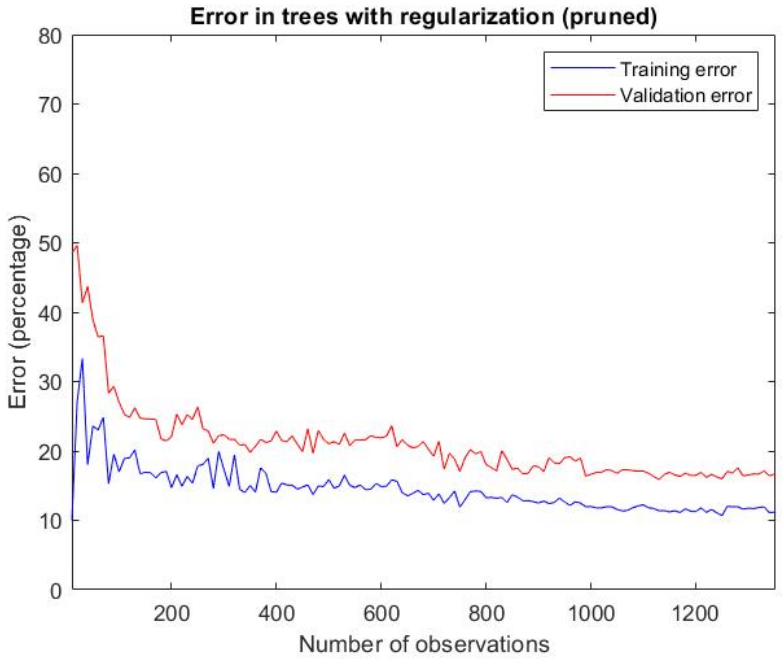
ניתן לראות בגרף השגיאה משמאל –

שגיאת האימון נשארת נמוכה בין 5% ל10% אחוז במהלך הלמידה, היא אינה מושפעת ממספר התצפיות שמהן המודל למד. בלמידה על כל סט האימון שגיאת האימון הינה 5%.
לעומת זאת שגיאת הוולידציה גבוהה יחסית ויורדת מהר בין 0 ל200 תצפיות. מ200 תצפיות והלאה ישנה ירידה מתונה מאוד עד לא מורגשת של שגיאת הוולידציה, כך שבלמידה על כל סט האימון שגיאת הוולידציה הינה 16%.

רננה מובאס 206698441
מור דניאלי-זולוטוב 205612484
שחף קובל 203962717

5) עשינו רגולריזציה בעזרת קיצוץ הענפים של העץ שנוצר. קיצוץ נעשה לענפים בעלי impurity הכי גבוהה. מה שאומר שמדד הgini שלהם הכי קרוב ל1. מדד gini קרוב ל1 מצביע על ענף שיוצר חלוקה לא טובה של הדאטה ולכן אין צורך בחלוקה זו, כי היא אינה מוסיפה הרבה מידע לעץ. בקוד הוספנו קריטריון לpruning שהוא impurity. בחרנו רמת קיצוץ שהיא 75% הרמת הקיצוץ האפשרית הגבוהה ביותר. לדוגמה עץ בעל 20 שכבות ורמת הקיצוץ הגבוהה ביותר היא 15, אז רמת הקיצוץ שאנו בחרנו היא בערך 11.

ניתן לראות בגרף כי רגולריזציה גרמה לשגיאת האימון להיות גבוהה יותר, מה שאומר שאכן הצלחנו להגביל את למידת המודל לסט האימון באופן ספציפי מידי. אולם שגיאת הוולידציה עלתה מעט גם כן. זה מצביע על כך שהרגולריזציה שהחלנו על העץ הינה חזקה מידי, מה שגורם למודל לunderfitting זאת אומרת להיות פחות טוב לוולידציה גם כן. לאחר למידה של המודל על כל התצפיות הגענו לשגיאת אימון של 11% (לעומת 5% ללא רגולריזציה) ולשגיאת וולידציה של 17% (לעומת 16% ללא רגולריזציה). גם הפעם הירידה המשמעותית ביותר בשגיאת האימון והוודליציה נעשית בין 0 ל200 תצפיות. גם כאן חישבנו שונות על מנת לבדוק שהשונות אכן קטנה. השונות לשגיאת האימון שהתקבלה היא 0.06, ואילו השונות לשגיאת הוולידציה היא 0.1.



6) השתמשנו בפונקציה של מטלב על מנת למצוא את ההיפרפרמטרים הטובים ביותר בשביל כל מודל מבין חמשת המודלים של הקרוס וולידציה. ההיפרפרמטרים שבחנו הינם – מספר מקסימלי של פיצולים, מספר תצפיות מינימלי בעלה, קריטריון לפיצול. השתמשנו בrandom search. מודל העץ בעל ההיפרפרמטרים האופטימליים ביותר ושגיאת הוולידציה הקטנה ביותר נבחר בשביל ההמשך בשאלות 9 ו10.

טבלה של ההיפרפרמטרים האופטימליים שהתקבלו בשביל כל fold:

מספר מודל / fold	מינימום תצפיות בעלה	מקסימום פיצולים	קריטריון לפיצול
1	7	875	gini
2	1	171	deviance
3	5	278	deviance
4	1	968	deviance
5	1	122	gini

רננה מובאס 206698441
מור דניאלי-זולוטוב 205612484
שחף קובל 203962717

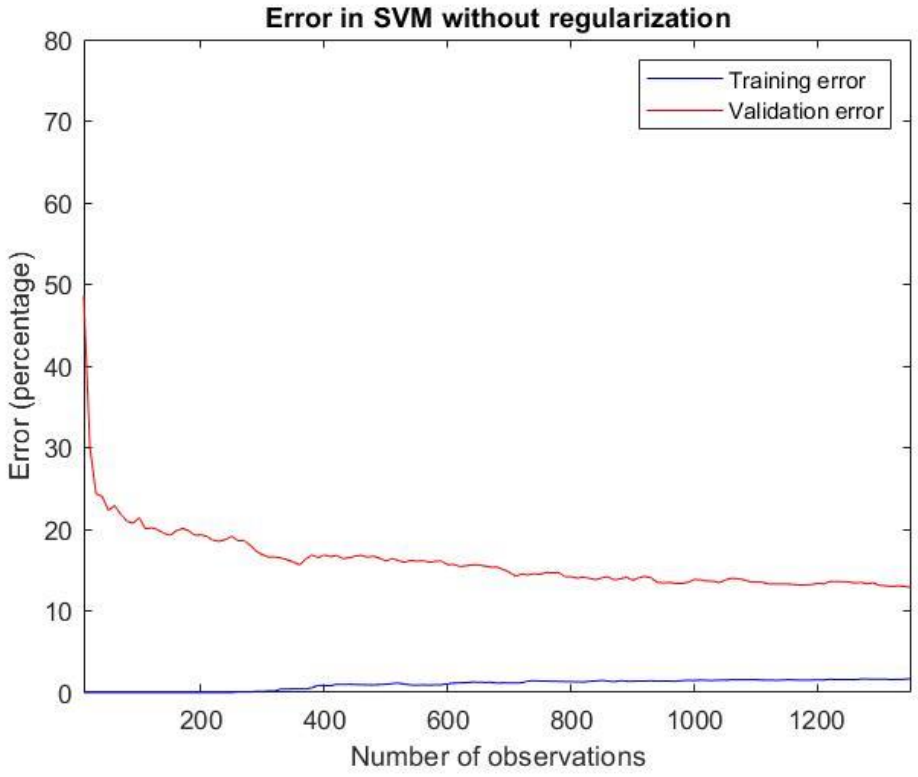
שיטת קלסיפיקציה : SVM – פונקציה 'svmModel.m'

7) 4. בסעיף זה בחרנו פונקציית שגיאה של מיסקלסיפיקציה, כאשר נקודות אשר לא מסווגות נכון מוסיפות 1 לערך השגיאה. מדד הצלחת המודל שבחרנו הינו accuracy, הוא מייצג את אחוז התצפיות אשר מסווגות נכון.
ההיפרפרמטרים שבחרנו הינם כתוצאה מברירת מחדל של מטלב, אשר מחייבת להגדיר אותם.
רשימת ההיפרפרמטרים שבחרנו להגדיר וההסבר:

- **'BoxConstraint'** – הינו היפרפרמטר אשר מגדיר את העונש על כל נקודה שמסווגת לא נכון. היות וברירת המחדל עבור היפרפרמטר זה היא 1, ברירת מחדל זו מקשה על המודל ללמוד. לכן בחרנו להגדיל אותו ל50 מה שמאפשר למודל ללמוד טוב ובקלות את סט האימון, מכיוון שהעונש על כל טעות הוא גדול. מה שנוצר לנו הוא מודל מאוד ספציפי לסט האימון.
- **'KernelFunction'** – השארנו לפי ברירת מחדל את הקרנל הגאוסיאני. לאחר הרצה של כמה קרנלים ובחינה של גרפי שגיאות האימון והווליידציה, ראינו כי קרנל זה הינו הטוב ביותר עבורנו למידת המודל שלנו.

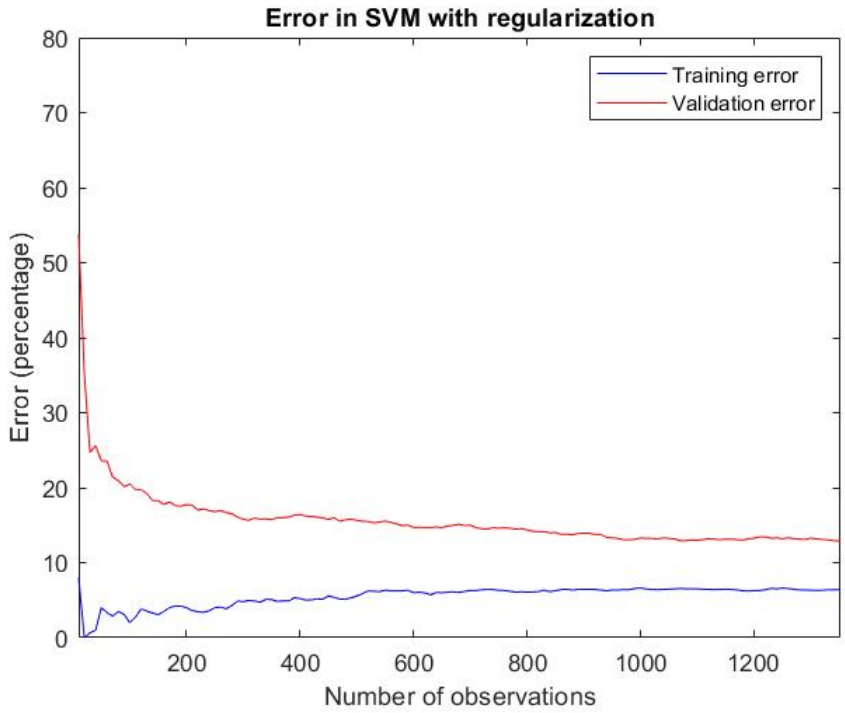
על מנת לחשב את שגיאת האימון השתמשנו בפונקציה מובנית של מטלב שנקראת 'resubloss'. חישוב של שגיאת הווליידציה גם הוא נעשה בעזרת פונקציה מובנית של מטלב הנקראת 'loss'.

בגרף – ניתן לראות שבתחילת אימון המודל שגיאת הוליידציה גבוהה מאוד וככל שנוספות יותר תצפיות כך שגיאת הוליידציה יורדת (ירידה חדה עד ל-200 תצפיות ולאחר מכן ירידה מינורית) ושגיאת האימון עולה, אך באופן מינורי יחסית. לאחר למידה על כל התצפיות שגיאת האימון הגיעה ל1% ושגיאת הוולידציה ל12.5%.
מאחר ובדקנו את השונות בין 5 המודלים של קרוס ווליידציה ולא נמצאה שונות משמעותית ביניהם, בחרנו לייצג את הגרף הממוצע של כולם. השונות של שגיאת האימון הייתה 0.009 ושל שגיאת הוולידציה הייתה 0.128.



רננה מובאס 206698441
מור דניאלי-זולוטוב 205612484
שחף קובל 203962717

5. רגולריזציה נעשתה בעזרת שינוי היפרפרמטר BoxConstraint, ל1. מה שאומר שההענשה על נקודות שלא סווגו נכון הינה קטנה ומתונה יותר. דבר זה מאפשר למודל שלנו ללמוד את סט האימון בפחות ספציפיות, מכיוון שהוא מאפשר לו לעשות טעויות. כמו שניתן לראות מהגרף, הרגולריזציה גרמה אכן לשגיאת האימון לעלות מכיוון שכעת המודל פחות ספציפי לסט האימון, בעוד ששגיאת הולדיציה נשארה יחסית זהה למודל בלי רגולריזציה. בין 0 ל200 תצפיות ניתן לראות את הירידה הגדולה ביותר בשגיאת הווליידציה (אבל לא בשגיאת האימון). לאחר מכן שגיאת הווליידציה והאימון נשארות יחסית קבועות, שגיאת אימון בין 2.5% ל5% ושגיאת הווליידציה בין 15% ל12.5%. העובדה ששגיאת האימון עלתה ושגיאת הווליידציה נשארה ללא שינוי, יכולה להצביע על כך שהרגולריזציה אשר ביצענו אינה מספיק חזקה והמודל עדיין יחסית ספציפי לסט האימון.
גם כאן חישבנו שונות על מנת לוודא שהיא אכן זניחה וניתן להציג את חמשת המודלים כממוצע בגרף אחד.
השונות של שגיאת האימון שהתקבלה היא 0.0019, השונות של שגיאת הווליידציה הינה 0.065.



6. השתמשנו בפונקציה של מטלב על מנת למצוא את ההיפרפרמטרים הטובים ביותר בשביל כל מודל מבין חמשת המודלים של הקרוס ווליידציה. ההיפרפרמטרים שבחנו הם - 'BoxConstraint', 'KernelScale' 'KernelFunction'. השתמשנו בrandom search. מודל הSVM בעל ההיפרפרמטרים האופטימליים ביותר ושגיאת הווליידציה הקטנה ביותר נבחר בשביל ההמשך בשאלות 9 ו10.

טבלה של ההיפרפרמטרים האופטימליים שהתקבלו בשביל כל fold:

מספר מודל / fold	Box constraint	Kernel Scale	Kernel function
1	50.45	1.55	גאוסיאני
2	16.78	none	פולינומיאלי
3	0.0039	none	פולינומיאלי
4	0.0086	none	פולינומיאלי
5	51.31	0.522	גאוסיאני

רננה מובאס 206698441
מור דניאלי-זולוטוב 205612484
שחף קובל 203962717

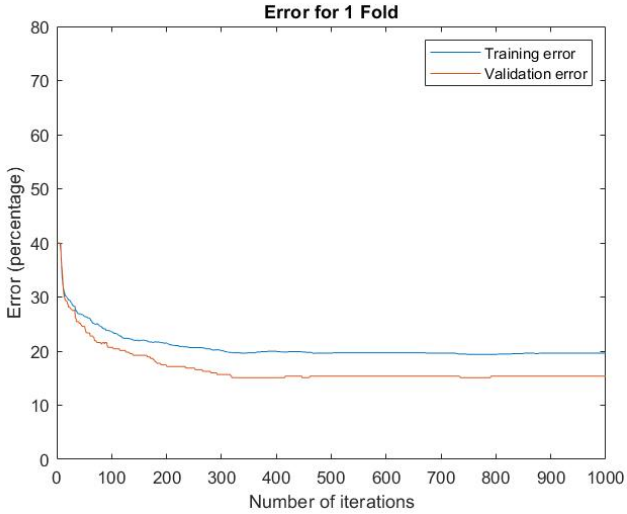
שיטת קלסיפיקציה : רגרסיה לוגיסטית – פונקציות 'logisticRegressionModel.m', 'convertData.m', 'cost.m', 'costRegu.m', 'gridSearch.m', 'predTest.m', 'sigmoid.m'

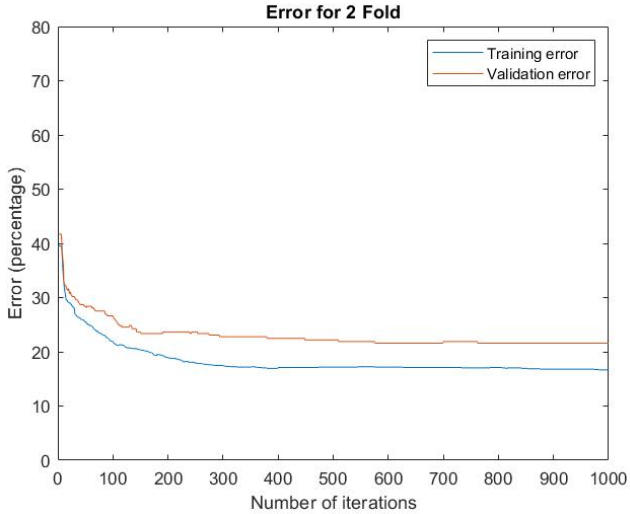
8) 4. רגרסיה לוגיסטית הינה פונקציה מסוג supervised learning כלומר אנחנו מספקים למודל את הX והמטרה שלנו הוא לבצע פרדיקציה על Y. זוהי שיטת רגרסיה שנוכל להשתמש בה כאשר נרצה להתגבר על בעיית הערכים הקיצוניים הקיימת לפעמים בשימוש בפונקציה ליניארית. היתרון שלה זה שהערך נע בין 0 ל1, ולכן נוכל להגיד שערכים שהם מעל חצי הם 1 ומה שקטן מחצי הוא 0. גם שאנחנו מסווגים נכון את המודל עדיין יש לפונקציה ערך ונוכל להמשיך לתקן את המשקולות למרות שהאלגוריתם צודק, ולקבל כך מודל יותר טוב.
בחרנו את פונקציית השגיאה כ $Cost(h_\theta(x), y) = -ylog(h_\theta(x)) - (1-y)log(h_\theta(x))$ ואת מדד להצלחת המודל כAccuracy, שהוא אחוז התצפיות אשר מסווגות נכון.

בשביל תהליך הלמידה הראשוני של המודל בחרנו את ההיפרפרמטרים הבאים:

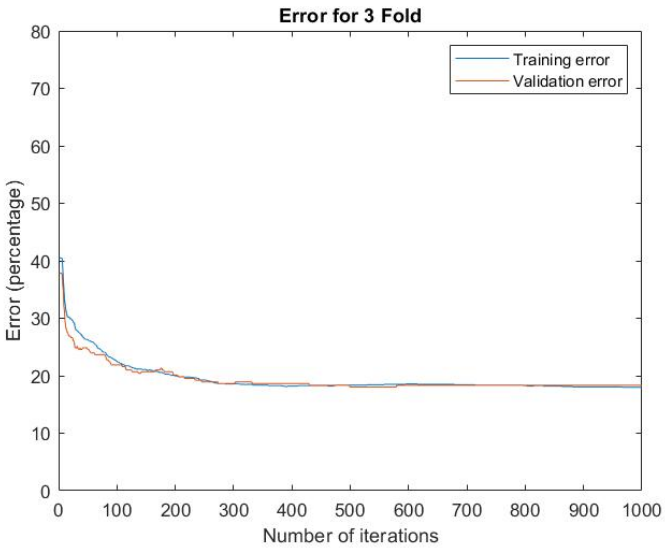
- **איטרציות** – הגדרנו כ1000. כלומר המודל יבצע אלף חזרות כדי להגיע לרמת למידה מספקת.
- **אלפא (קבוע למידה)** – הגדרנו כ0.1. בחרנו אותו כך שהמודל מצד אחד יוכל לבצע למידה טובה של המשקולות ב"צעדים" יחסית קטנים, אבלמצד שני לא קטן מידי אשר גורם לזמן למידה ארוך מידי.
- **למדה (קבוע רגולריזציה)** – בשביל סעיף זה לא הכנסנו את למדה לפונקציית הcost של למידת המודל. אולם כאשר הוספנו רגולריזציה בסעיף 5, קבענו כי קבוע הרגולריזציה יהיה 0.5. בחרנו מקדם הענשה זה על מנת שלא ייצור הענשה גדולה מידי על המשקולות שתפריע ללמידה, אבל מספיק גדול על מנת לייצר הענשה משמעותית אשר תשפר את למידת המודל.

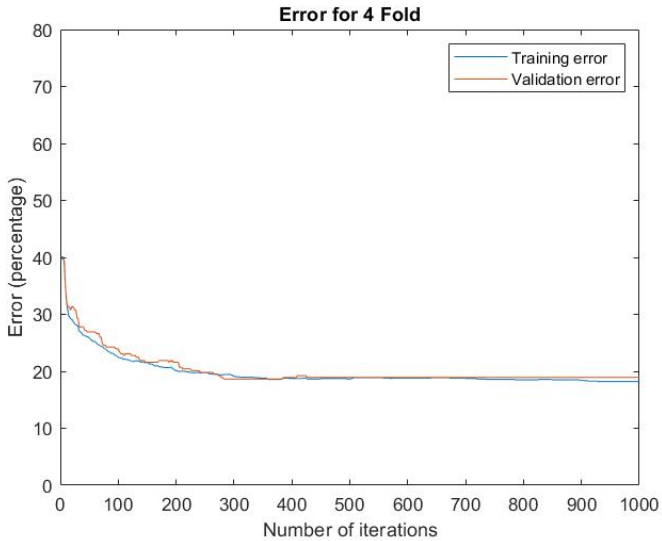
על פי הגרפים של שגיאות הווליידציה והאימון, ניתן לראות כי אכן יש הבדל בין הצלחת חמשת המודלים (5 fold). לכן ברגרסיה הלוגיסטית הצגנו את המודלים בגרפים נפרדים בשונה מעצי החלטה וSVM. בכל המודלים ניתן לראות כי הירידה הגדולה ביותר בשגיאת האימון והווליידציה נעשית בין 0 ל200 איטרציות, לאחר מכן השגיאות נשארות יחסית קבועות. בכל המודלים שגיאת האימון מגיע ב1000 איטרציות ל20%. שגיאת הווליידציה לעומת זאת מגיעה ל15% במודל הראשון, ל22% במודל השני, ל20% במודל השלישי, הרביעי והחמישי.
במודל הראשון שגיאת הווליידציה קטנה יותר משגיאת האימון, זה יכול לנבוע מכמה סיבות. ייתכן כי בסט האימון הספציפי לfold זה היו הרבה דוגמאות קשות לסיווג, או שבסט הווליידציה הזה היו דוגמאות קלות לסיווג.
במודל השני שגיאת האימון קטנה יותר משגיאת הווליידציה, מה שאומר שהמודל התאמן יותר מידי טוב על סט האימון. במודל השלישי, הרביעי והחמישי שגיאות האימון והווליידציה חופפות מה שיכול להצביע על כך שהמודל למד בצורה המיטבית על סט האימון והצליח להכליל את הלמידה על סט הווליידציה.

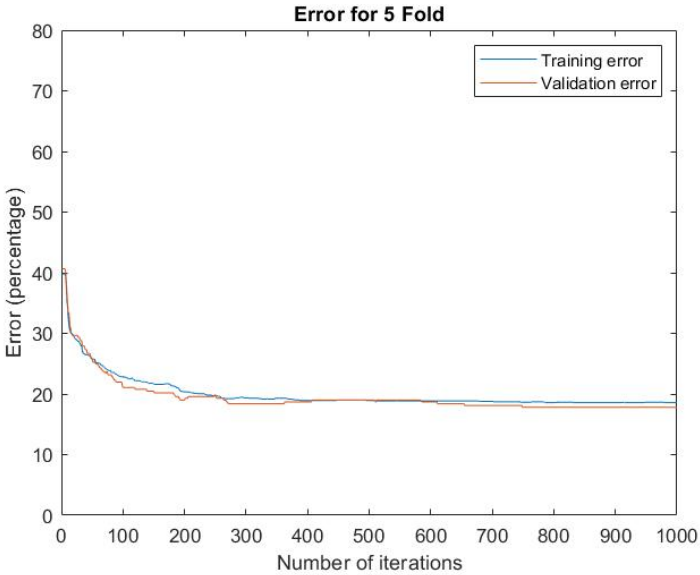




רננה מובאס 206698441
מור דניאלי-זולוטוב 205612484
שחף קובל 203962717





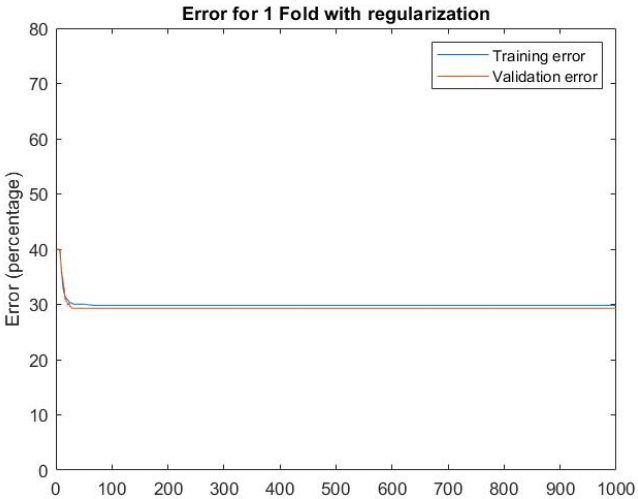


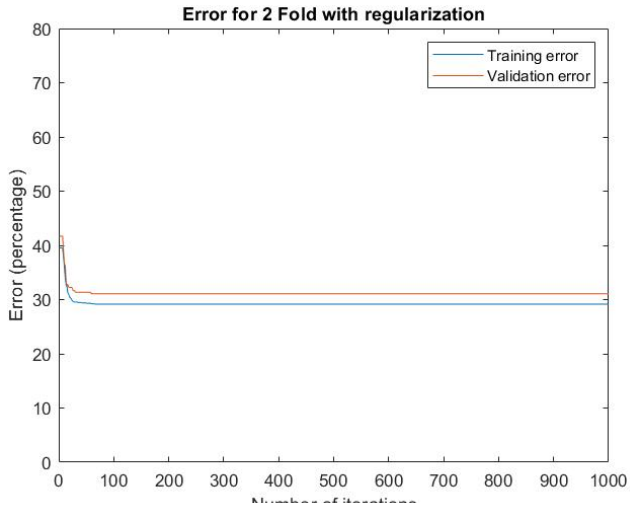
5. כעת צירפנו את קבוע הרגולריזציה (למדה) לפונקציית הcost. מטרת הרגולריזציה היא להעניש משקולות גדולות ולגרום לכך שהרשת תהיה פשוטה יותר. נקבע קבוע רגולריזציה המוכפל בפונקציית השגיאה- אם נבחר קבוע קטן מידי אז לא תהיה מספיק הענשה של המשקולות ונשאר במצב של overfitting, אך אם נבחר קבוע גדול מידי הענשת המשקולות תהיה כבדה ונגרום למצב של underfitting.

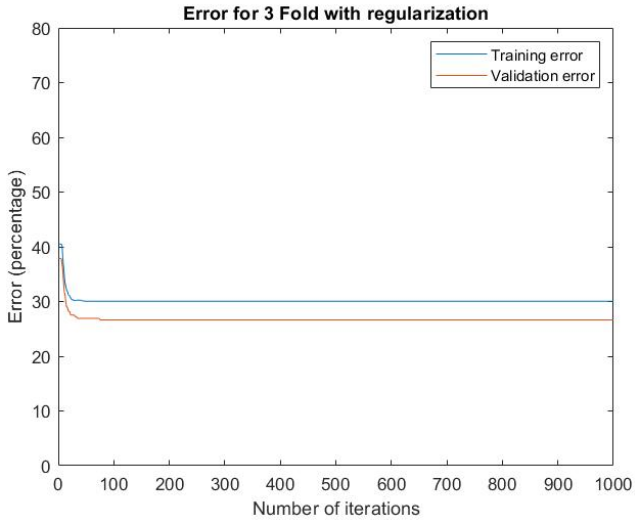
בכל המודלים ניתן לראות כי הוספת רגולריזציה הפריעה ללמידה של המודל. ניתן לראות ששגיאות האימון ושגיאות הוולידציה עלו יחד ל30%, בהשוואה ל20% שגיאה ללא רגולריזציה. אנו משערים כי ללמידה של המודלים, מה שיכול להיות גרם למודל לעשות Underfitting גם לאימון ולגם ולוולידציה.

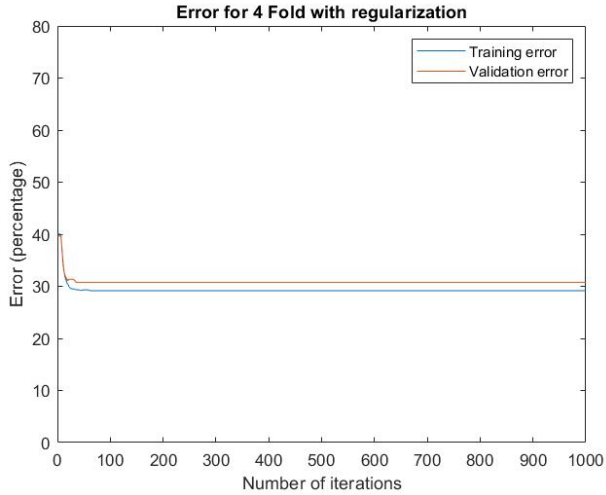
גרפים של פונקציית השגיאה עם רגולריזציה -

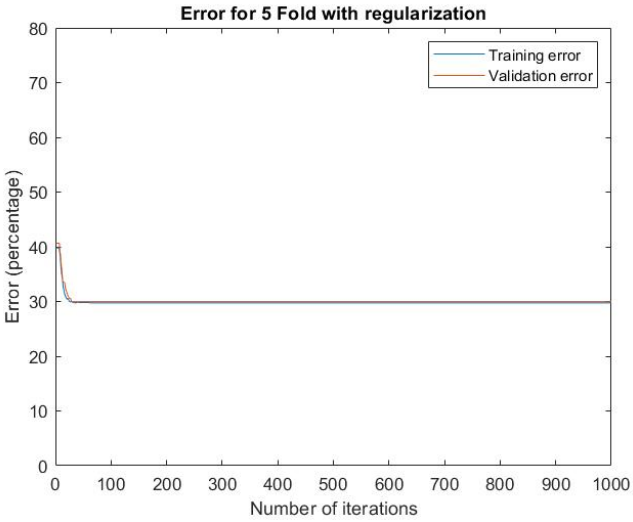
רננה מובאס 206698441
מור דניאלי-זולוטוב 205612484
שחף קובל 203962717











6. ביצענו grid search עבור כל אחד מחמשת המודלים. חיפשנו את ההיפרפרמטרים האופטימליים הבאים – מספר איטרציות, קבוע למידה (אלפא), קבוע רגורלריזציה (למדה). מסעיף זה לקחנו את המודל הטוב ביותר בעל ההיפרפרמטרים האופטימליים וטעות הוולידציה הקטנה ביותר והשתמשנו בו לשאלות 9 ו10.

רננה מובאס 206698441
מור דניאלי-זולוטוב 205612484
שחף קובל 203962717

טבלה של ההיפרפרמטרים האופטימליים שהתקבלו בשביל כל fold:

מספר מודל / fold	מספר איטרציות	קבוע למידה (אלפא)	קבוע רגולריזציה (למדה)
1	700	0.1	1.4
2	700	0.1	1.4
3	700	0.1	1.4
4	700	0.1	1.4
5	700	0.1	1.4

9) עשינו זאת בפונקציה **'ensembleModelandAccuracy.m'**. את הגרף של הדיוק נציג ונסביר בשאלה הבאה (10), זאת על מנת לצרף להשוואת הדיוק את המודל של הensemble אותו בנינו בשאלה 10.

10) עשינו זאת בפונקציה **'ensembleModelandAccuracy.m'**. חשוב לציין כי במודל הensemble לא הייתה אפשרות של בחירה רנדומלית בין הפרדיקציה של המודלים השונים. זאת מכיוון שישנם שלושה מודלים ואילו הסיווג הינו בינארי, לכן תמיד יהיו שני מודלים שיעשו את אותה הפרדיקציה. לדוגמה מודל עץ – 0, מודל SVM – 1 ומודל רגרסיה לוגיסטית – 1 ← כך הensmble יעשה פרדיקציה של 1.

ניתן לראות מגרף הדיוק כי המודל בעל אחוז הדיוק הגבוהה ביותר הינו מודל הSVM (80%), לאחר מכן למודל עץ ההחלטה יש את הדיוק הבינוני (75%) ולבסוף מודל הרגרסיה הלוגיסטית בעל אחוז הדיוק הנמוך ביותר (70%). ניתן לראות כי מודל הensemble הינו בעל אחוז דיוק בין SVM לעץ החלטה (77%), תוצאה זו הגיונית מכיוון שהשתמשנו בכלל של "הרוב קובע" על מנת לייצור את מודל זה.