פרויקט בקורס מערכות לומדות

נושא הפרויקט - Breast Cancer Classification

חלק ראשון

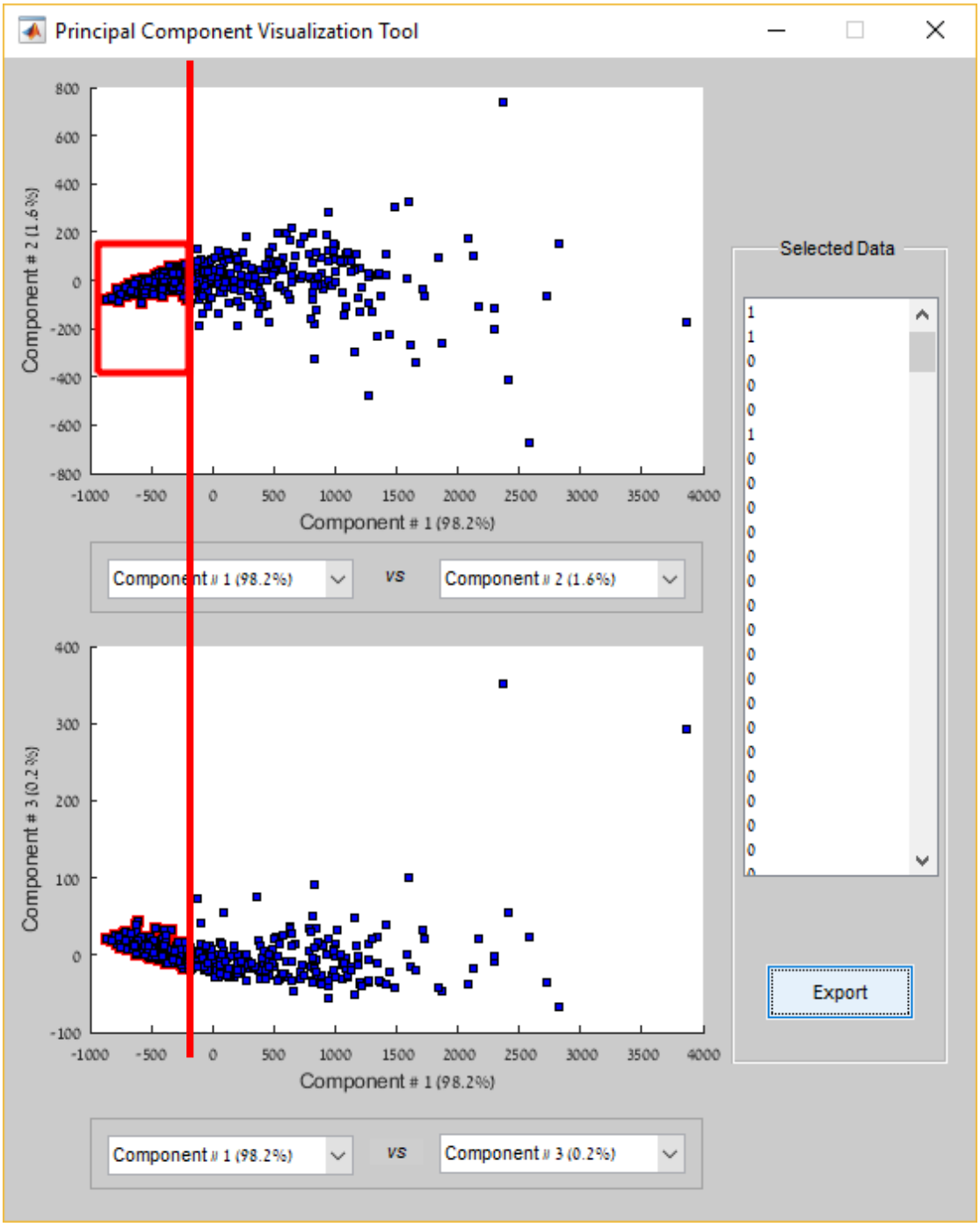
<u>מגישים:</u>

דולב עפרי – 304843659

אורי שטיגליץ - 201110830

מטלה 0 – PCA

לאחר שימוש בפונקציה mapcaplot, התבוננו בשלושת הרכיבים העיקריים, והשווינו את הדומיננטי ביותר לשניים האחרים. להלן מישור ההפרדה שבחרנו, כאשר משמאלו מדובר בדוגמאות המתוייגות כאפס(גידול שפיר) ומימינו דוגמאות המתוייגות כ-1 כלומר גידולים ממאירים.



4. בחירת מודל והערכת ביצועים

נשים לב כי ה data מורכב מ569 דוגמאות כאשר כמעט 40% מהם חולים ומעט יותר מ60% בריאים לכן נחלק את הdata כך שיחס זה יישמר בסדרת האימון וסדרת הבוחן כלומר חלוקה של כ-40% ו 60%.

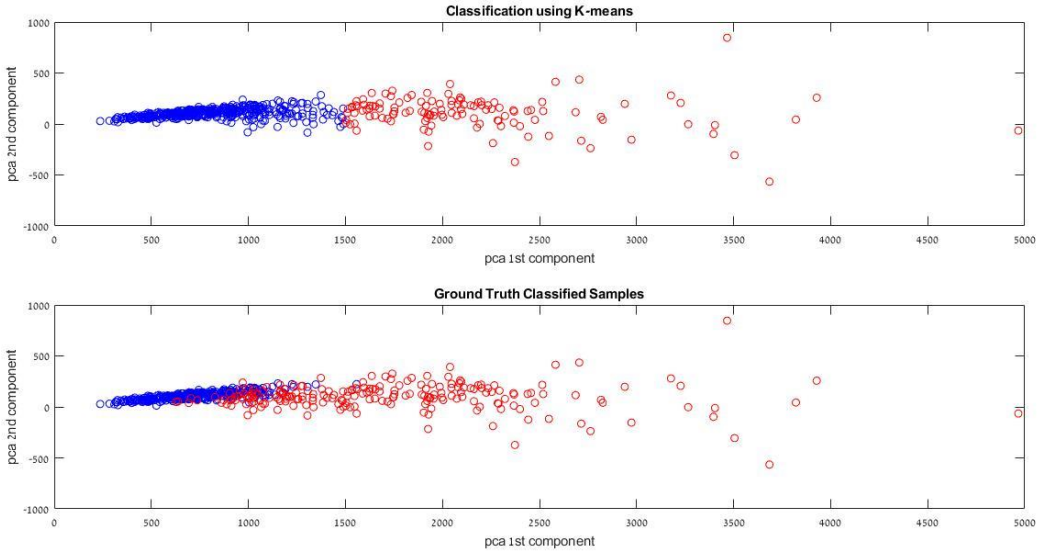
סט הבוחן יכיל 113 דוגמאות(20% מסך הדוגמאות בהתאם להנחיות התרגיל), כאשר נבחר 42 חולים ו- 71 בריאים.

5. Learning / Clustering Unsupervised

מטלה 1 K-Means

2) כמובן כי כמות האשכולות המתאימה ל-data set שלנו היא 2 משום שה-data הנתון לנו מתויג במקור ל2 קבוצות בלבד.

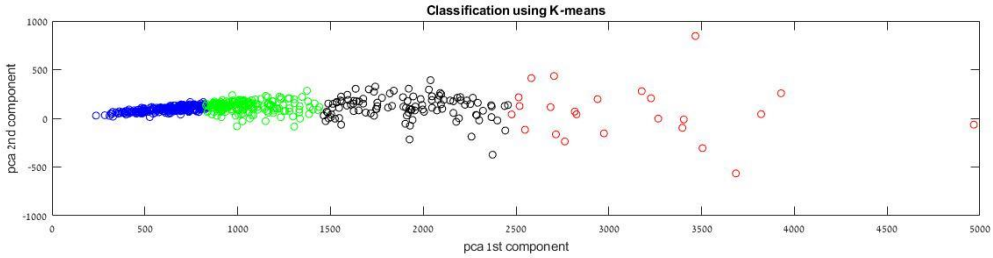
3)

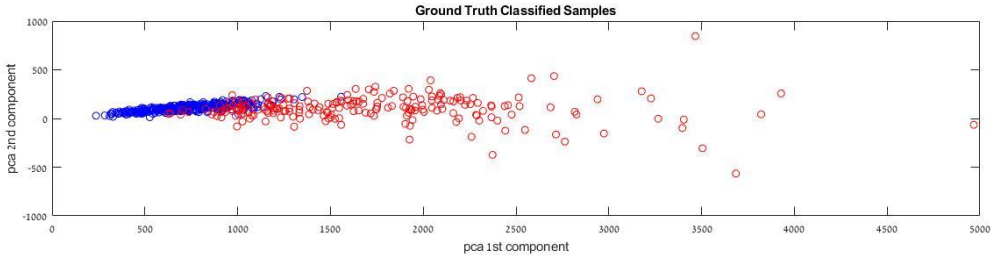


אכן ניתן לראות כי יש קשר בין התוצאות כאשר בצד שמאל בכחול בשני הגרפים מדובר באנשים בעלי גידול שפיר ובאדום אנשים בעלי גידול ממאיר. ניתן לראות כי בגרף הסיווג לפי K-MEANS(למעלה) ישנם סיווגים שגויים (אנשים עם גידול ממאיר שסווגו כשפיר).

הקורלציה אינה מבוטחת משום שה K-MEANS אינו מספק פתרון יחיד שכן לעיתים(כפי שראינו בתרגול) אתחול מרכזי המסה משפיעים על התוצאה הסופית.

(4





עבור הdata-set הנתון לנו המחולק ל-2 קבוצות, ועבור k שונה מ2 מאבדים את המשמעות עבור בעיה זו. בבעיה הנתונה יש צורך להפריד בין חולים לבריאים(יחסית לקבוצה הראשונה). הk-means עובד על המרחב הגיאומטרי של הנתונים בלי התחשבות בתיוג האמיתי. לכן עבור k=2 נצפה לקבל פלט שייתן התאמה כלשהיא לבעיה אותה אנו מנסים לפתור אבל עבור k שונה מ-2 נקבל פלט שאין לו כל משמעות בהקשר לתיוגים האמיתיים.

6. Supervised Binary Classification

מטלה 2 מסווג בייסיאני נאיבי:

בהתאם לנלמד בהרצאות עבור סיווג בייס נאיבי מימשנו את הנוסחאות הבאות על מנת לקבל את התוחלת והשונות של כל פיצ'ר (מתוך ה-30) עבור כלל הדגימות **בסט הלימוד**.

לאחר מכך עבור כל דגימה **בסט הבוחן** התקבל סיווג ע"פ חוק ההחלטה.

אנו מניחים אי תלות בין הפיצ'רים השונים לכן נחשב בצורה הבאה:

$$p_{X|Y}(x|C) \approx \prod_{j=1}^{d} p_j(x(j)|C)$$

בנוסף מניחים כי הפילוג המותנה של כל פיצ'ר מפולג נורמלית :

$$p(x(j)|C_k) \sim N(\mu_{j|k}, \Sigma_{j|k})$$

כאשר הפרמטרים של התוחלת וסטיית התקן מחושבים כך:

$$\hat{\mu}_{j|k} = \frac{1}{n_k}\sum_{i=1}^{n} x_i(j)\,\mathbb{I}\{y_i = C_k\}$$

$$\hat{\Sigma}_{j|k} = \frac{1}{n_k}\sum_{i=1}^{n}\left(x_i(j) - \hat{\mu}_{j|k}\right)\left(x_i(j) - \hat{\mu}_{j|k}\right)^T \mathbb{I}\{y_i = C_k\}$$

מבחינת **הנחות המודל** באופן כללי הנחות המודל, אי תלות בין מרכיבי ווקטור המאפיינים של כל דוגמא i והתפלגות נורמלית, אינן נכונות. לדוגמא ברור כי עבור אדם עם עודף משקל גדול והחולה בסוכרת יש קורלציה בין מאפיינים אלו אך אנו בכל זאת נניח אי תלות(מהגדרת מסווג בייסיאני נאיבי) עם זאת לאור אחוז השגיאה בסיווג סט הבוחן (שתוצג בהמשך) ניתן לראות במקרה ספציפי זה כי הנחה זו מתארת באופן לא רע את המודל.

הנחת הגאוסיות עד כמה שמסתבר שמתארת בצורה לא רעה את המודל (לאור השגיאה שתוצג בהמשך) אינה וודאית ולא בהכרח מתארת במדויק את התפלגות כל המאפיינים שניתנו לנו. דרך לשפר זאת היא להוסיף כמה שיותר דוגמאות לסט האימון(עשרות ומאות אלפים).

<u>הערכת ביצועים:</u>

זמן הריצה שהתקבל עומד על 0.0154 שניות.

שגיאת הסיווג המתקבלת עבור **סט הלימוד** היא **0.0592** כלומר כ6% שגיאה מתוך סט הלימוד.

שגיאת הסיווג המתקבלת עבור **סט הבוחן** היא **0.0796** כלומר כ7% שגיאה מתוך סט הבוחן.

אין צורך בקרוס ווליידציה משום שהפרמטרים היחידים לצורך תיאור המודל הינם פרמטרי ההתפלגות הנורמלית המותנית ואלו הם אינם פרמטרים העוברים "כיוונון/התאמה/עדכון" אלא מחושבים ישירות בצורה דטרמיניסטית מסט הלימוד.

מטלה 3: רגרסיה לוגיסטית

בחלק זה אימנו מסווג לינארי בעזרת רגרסיה לוגיסטית בעלת מודל לינארי עפ"י האלגוריתם הסדרתי ולאחר מכן עפ"י האלגוריתם בגרסת ה-batch.

פונקציית המחיר שברצוננו למזער הינה פונקציית הפרש ריבועי כפי שראינו בתרגול.

השתמשנו בפונקציית האקטיבציה הבאה, אותה ראינו בהרצאה ובתרגול:

$$\varphi(v) = \frac{1}{1 + e^{-v}}$$

בכדי ש-v לא יימצא בתחום הרוויה של פונקציית האקטיבציה, שם הנגזרת היא אפסית, נרמלנו את ה-data שלנו ע"י חיסור בממוצע וחילוק בערך המקסימלי (במקום בווריאנס, זאת לצורך החמרה) עבור כל קואורדינטה (מאפיין). הנירמול עוזר בכך שהמידע שלנו יימצא בחלק ה"מעניין" של פונקציית האקטיבציה, כלומר איפה שהנגזרת לא מתאפסת. בנוסף הוספנו bias למידע ע"י שרשור ווקטור של 1-ים בתחילת המידע(כלומר ווקטור המשקלים w גדל וגודלו כעת D+1 כאשר D מס' המאפיינים של הdata).

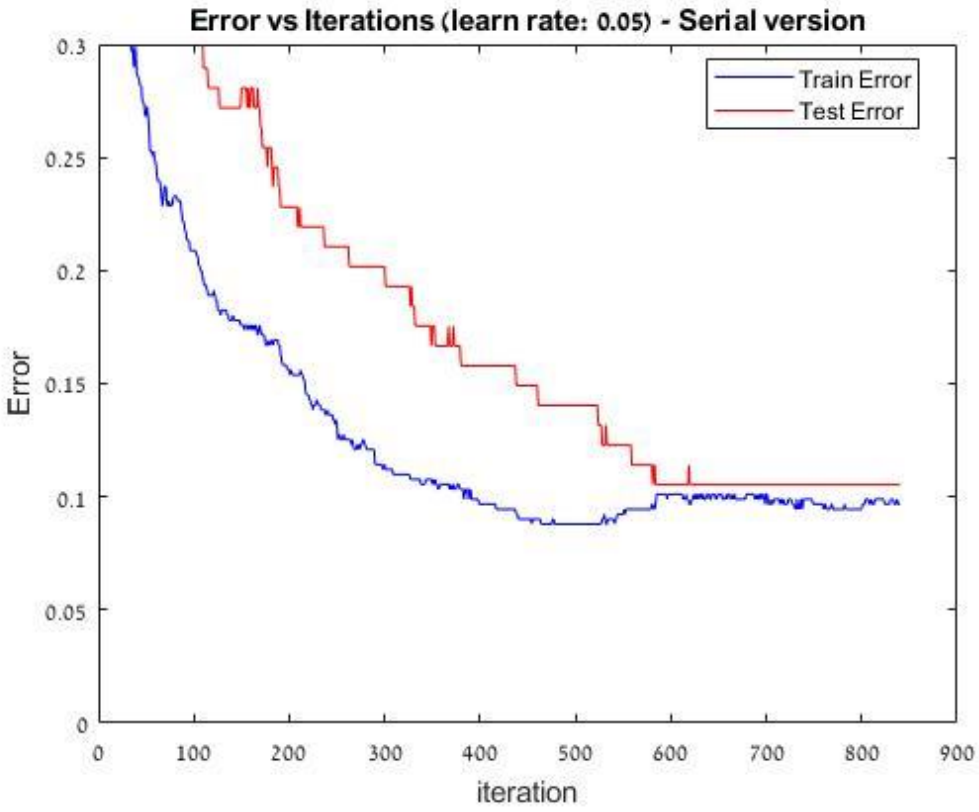
קבענו תנאי סף באופן הבא: ניסינו להריץ את האלגוריתם בעזרת מספר ספים כאשר המטרה הייתה לנסות להבין בעזרת הגרפים עבור איזה סף מגיעים לנקודת מינימום של פונקציית המחיר. בנוסף הגבלנו את מספר האיטרציות של האלגוריתם במטרה לא להגיע למצב של overfitting.

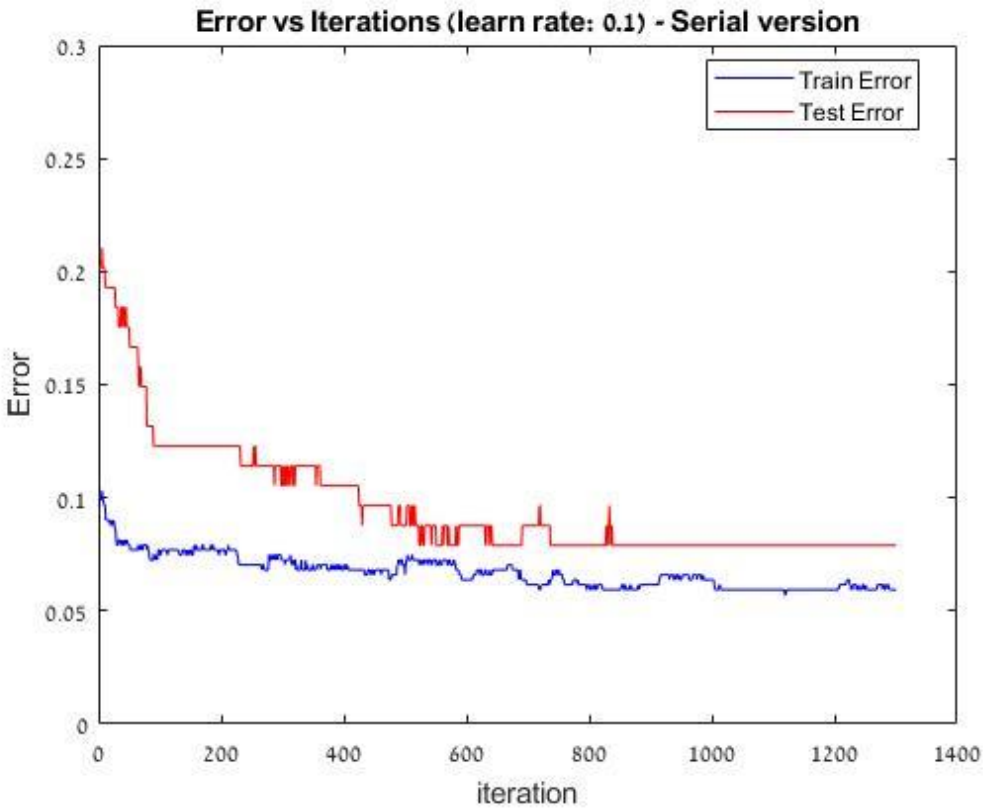
<u>גרסה סדרתית של האלגוריתם:</u>

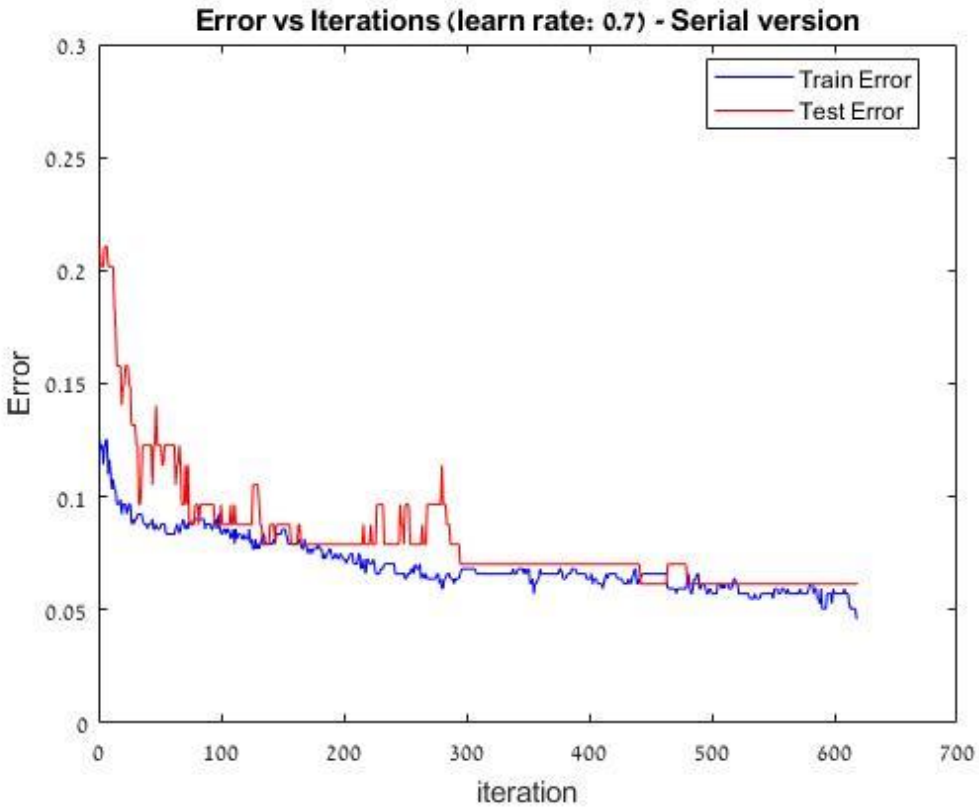
בגרסה זו אנו מעדכנים את Δw בכל צעד ע"י דוגמא אחת. אנו עוברים על כל הדוגמאות בצורה רנדומלית, ורק לאחר שעברנו על כל הדוגמאות פעם אחת אנו מגרילים מחדש את סדר הדוגמאות ורצים שוב על כולן אחת-אחת.

להלן התוצאות עבור שלושה ערכי קצבי לימוד שונים (בהם לא התרחשה התבדרות):

נציין כי עבור כל הרצה קיבלנו תוצאות שונות (גם בגלל רינדום סט האימון והבוחן שנבחרים וגם בגלל שלומדים את סט האימון בסדר שונה).


Error vs Iterations (learn rate: 0.05) - Serial version
Train Error
Test Error
Error
iteration
0
0.05
0.1
0.15
0.2
0.25
0.3
0
100
200
300
400
500
600
700
800
900


Error vs Iterations (learn rate: 0.1) - Serial version
Train Error
Test Error
Error
iteration
0
0.05
0.1
0.15
0.2
0.25
0.3
0
200
400
600
800
1000
1200
1400

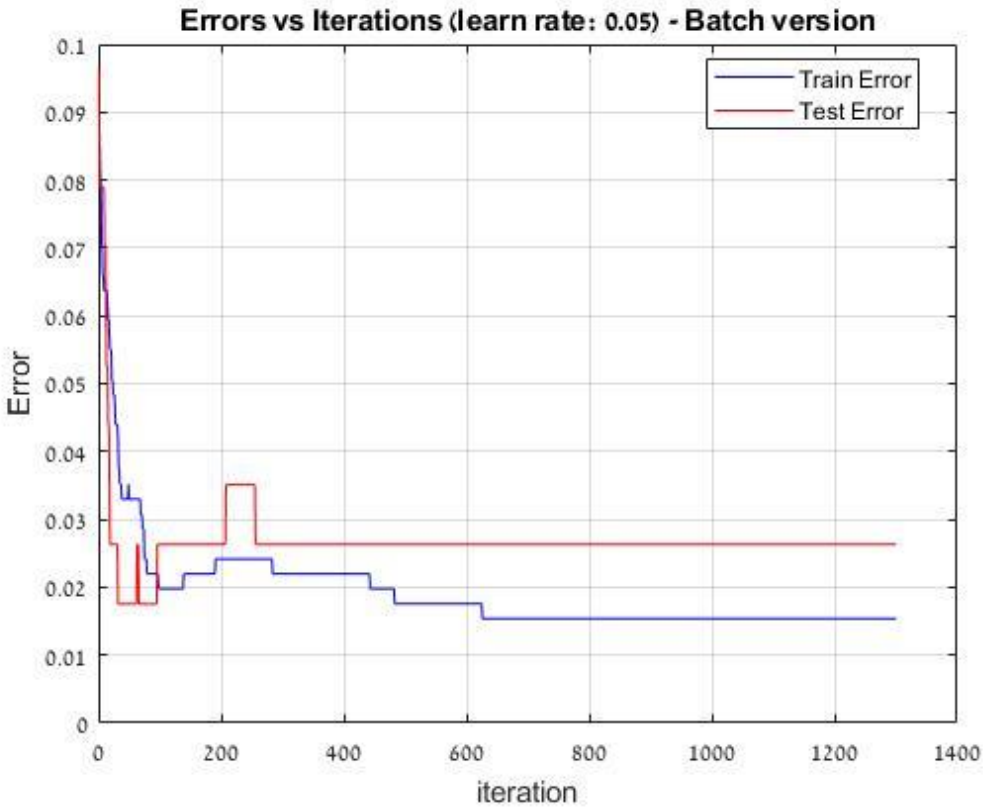


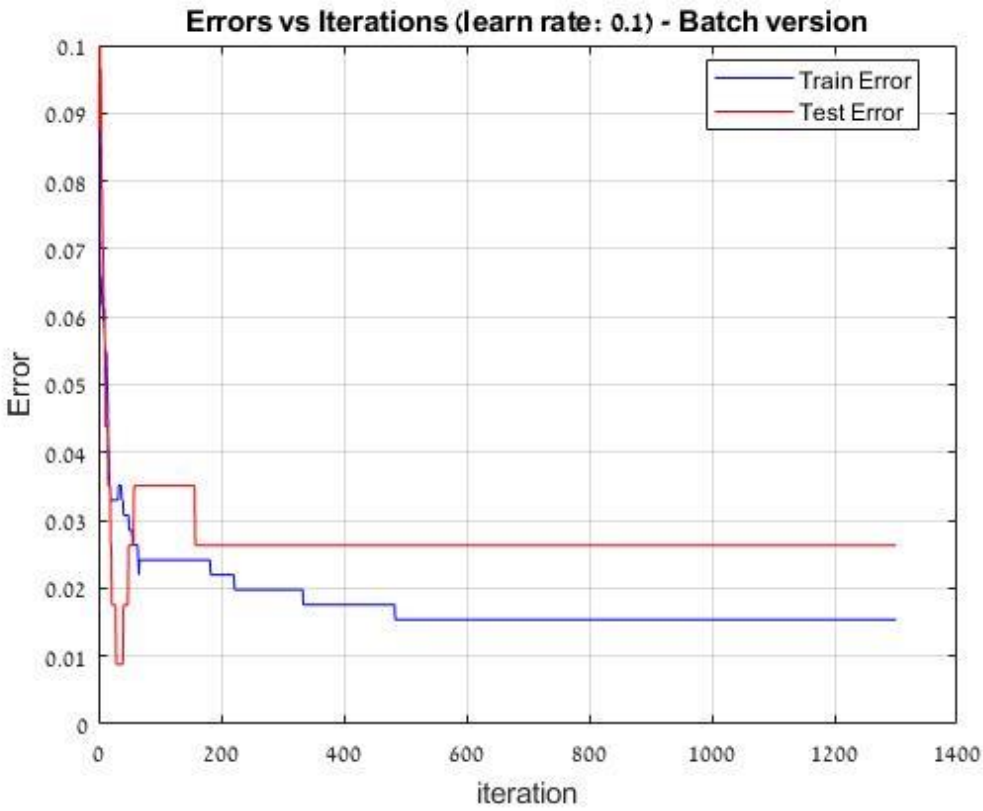
<u>גרסת ה-Batch של האלגוריתם:</u>

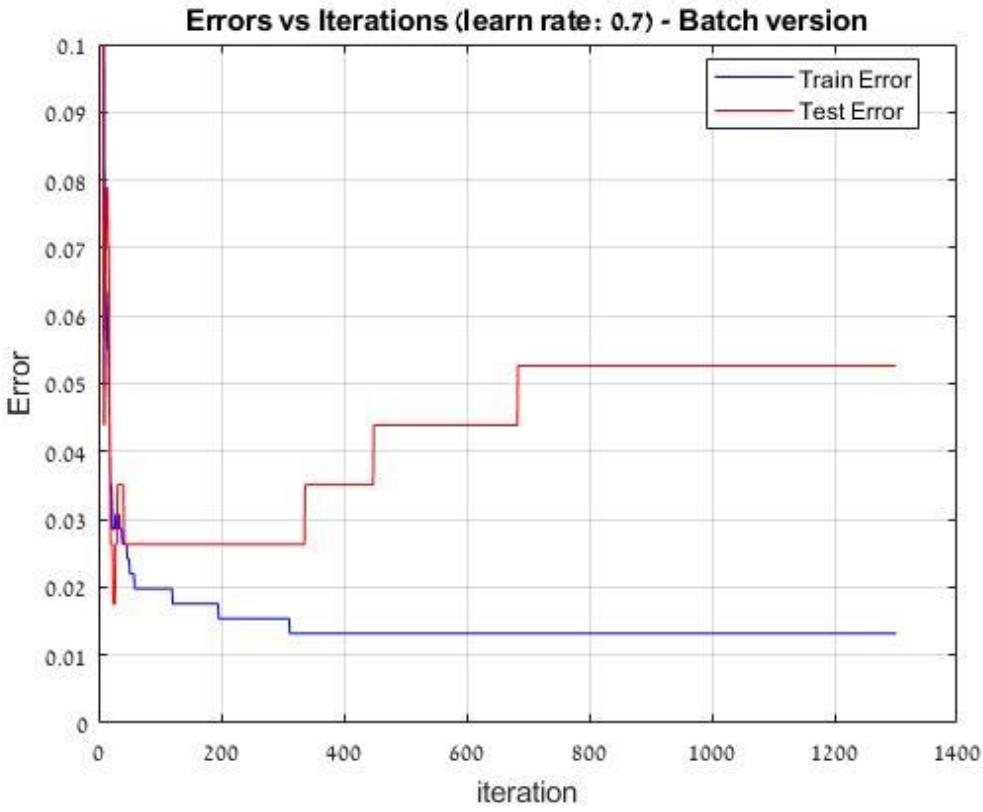
בגרסה זו אנו מעדכנים את Δw בכל צעד ע"י כל הדוגמאות של סדרת הלימוד.

להלן התוצאות עבור שלושה ערכי קצבי לימוד שונים (בהם לא התרחשה התבדרות):

נציין כי עבור כל הרצה קיבלנו תוצאות שונות (בגלל רינדום סט האימון והבוחן שנבחרים).


Errors vs Iterations (learn rate: 0.05) - Batch version
Train Error
Test Error
Error
iteration
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
0
200
400
600
800
1000
1200
1400


Errors vs Iterations (learn rate: 0.1) - Batch version
Train Error
Test Error
Error
iteration
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
0
200
400
600
800
1000
1200
1400



כפי שניתן לראות, קיבלנו שגיאה קטנה יותר בגרסת ה-Batch מאשר בגרסה הסדרתית (2.5-5% לעומת 7-11%). הסיבה להבדל זה נעוצה בכך שבגרסת הbatch לומדים מכל הדוגמאות בכל צעד לעומת הגרסה הסדרתית בה אנו לומדים רק מדוגמא אחת בכל צעד.

כמובן שלימוד מכל הדוגמאות בכל צעד מביא לתוצאות טובות יותר אך הוא יותר "כבד" חישובית. לשם השוואה זמן הריצה שלו עבור הסט המדובר הוא 16.35 שניות בעוד של הגרסה הסדרתית הוא 7.09 כלומר יותר מפי 2 וזה עבור כמות data יחסית קטנה.

ניתן לראות כי עבור קצב לימוד של $\eta = 0.7$ אנו מקבלים לאחר כמעט 300 איטרציות עלייה באחוז השגיאה עבור סט הבוחן. ככל הנראה זהו מצב של overfitting, כלומר המשקלים יותר מדיי מותאמים לדוגמאות ולא הצלחנו לקבל הכללה לאחר הלמידה.

זמן הריצה של האלגוריתם: 26.332 שניות(בסה"כ, כולל פעולות מקדימות כגון חלוקת המידע, מרכוזו וכו...) עבור הגרסה הסדרתית ולאחר מכן גרסת ה-batch

דרך לשפר את האלגוריתם הינה להוסיף עוד דוגמאות ובכך לקבל למידה טובה יותר של המשקלים.

בנוסף ניתן לקבוע **קצב לימוד/גודל צעד אדפטיבי** כאשר הרעיון הוא ללמוד מהר בהתחלה ע"מ לחסוך זמנים וכשאר מתקרבים לנק' המינימום והגרדיאנט קטן בהתאמה להקטין גם את גודל הצעד ובכך נשפר את הסיכוי להגיע למינימום(ולא לצאת ממנו במידה ואנו קרובים אליו)

מטלה 4 עץ החלטה:

1. להלן תוצאות האלגוריתם מרוכזים בטבלה.

	אנטרופיה	אינדקס ג'יני	שגיאת תיוג
שגיאה ממוצעת	0.0723	0.1078	0.0859
סטיית תקן	0.0293	0.0564	0.0532
עומק ממוצע	7.9	7.6	5.8
שגיאת סט הבוחן	0.0442	0.115	0.0531

ביצענו קרוס וולידציה לצורך מציאת העץ האופטימלי על סמך שלושת המדדים שנלמדו בתרגול. העץ האופטימלי על סמך מדד אופטימלי מסוים נבחר על סמך שגיאה ממוצעת מינימלית של הValidation Set. העץ האופטימלי שהתקבל מהאלגוריתם חושב ע"י מדד האנטרופיה ועומקו 8.

בעיה זו אכן יכולה להתאים לעץ החלטה שכן התקבל עץ החלטה בעומק לא גדול במיוחד כאשר התקבלה שגיאת test set של 4.4% בלבד.

2. עבור העץ האופטימלי הנבחר התקבלה שגיאת אימון של 4.44% (שגיאת סט הוולידציה) ושגיאה של 4.42% עבור סט הבוחן(כפי שנראה בטבלה).
3. נוכל להוסיף עוד פיצ'רים, ולהגדיל את מספר הדוגמאות. בנוסף במידה וקיים over fitting בעץ שהתקבל נוכל להגביל את מספר הצמתים של העץ ובזמן ש"נשלם" יותר בשגיאה עבור סט הבוחן וסט הוולידציה נקבל מודל יותר כללי שיתאים ליותר מקרים.

מטלת סיכום

להלן ריכוז התוצאות של המסווגים השונים:

Runtime	**שגיאת סיווג**	**זמן ריצה**
K-Means	15.64%	0.79
מסווג בייס נאיבי	7.96%	0.0154
רגרסיה לוגיסטית	2.6% (גרסת Batch)	16.35
עץ החלטה	4.4%	16.18

כפי שניתן לראות, סיווג בעזרת רגרסיה לוגיסטית נותנת את תוצאות הדיוק הכי טובות.

לא נרצה לבחור במסווג K-Means, כיוון שמדובר במסווג שלא מתייחס לתיוגים. הוא אלגוריתם למידה לא מפוקחת שלא משתמש במידע התיוג של הדוגמאות שברשותו ועובד רק על גיאומטריה.

ניתן להסיק כי עבור בעיה זו נרצה לבחור בפתרון המחזיר סיווג ולא פתרון המחזיר פונקציה המקשרת בין הקלט לבין הסיווג (כמו בייס). מבחינת השוואת עץ החלטה לרגרסיה הלוגיסטית – בבניית עץ החלטה ישנם מקרים רבים בהם נקבל overfitting או עצים עמוקים

במיוחד שלא לצורך. כלומר ע"י גיזום העץ נוכל לשפר בהרבה את התוצאות, אך נכון לעכשיו הן הרבה פחות טובות מהרגרסיה הלוגיסטית.

הקשיים בהם נתקלנו:

מימוש ארכיטקטורת עץ ההחלטה היה סבוך ולא פשוט כלל – היה צורך בתכנון מודולרי נרחב ומעקב אחרי רקורסיית בניית העץ.

בנוסף, בזמן מימוש הרגרסיה הלוגיסטית היה צורך בניסוי וטעייה מבחינת בחירת פרמטרים מתאימים (threshold וכן קצבי לימוד), וכן היה מאוד קריטי לנרמל את ה-data לפני עיבודו בכדי לקבל תוצאות טובות.