**Introduction to Machine Learning - 236756**

Major HW3 - Regression

Itzik Vaknin – 208223099

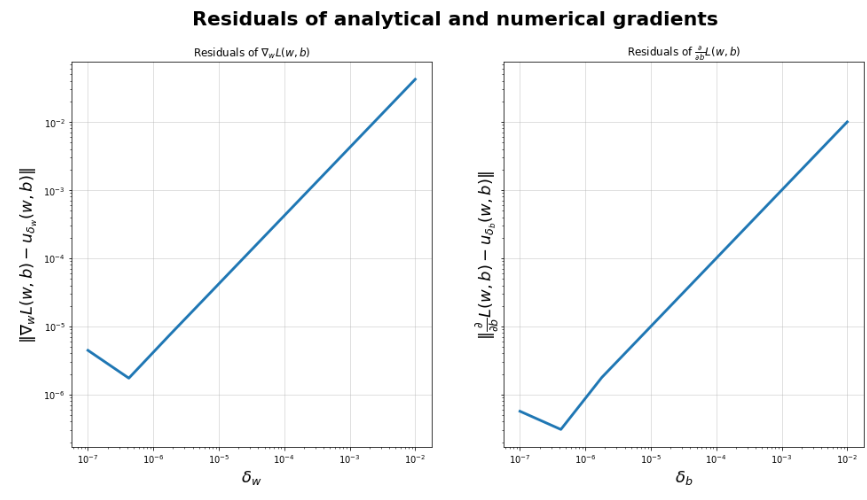
Shahar Shair – 209045921

23/01/23

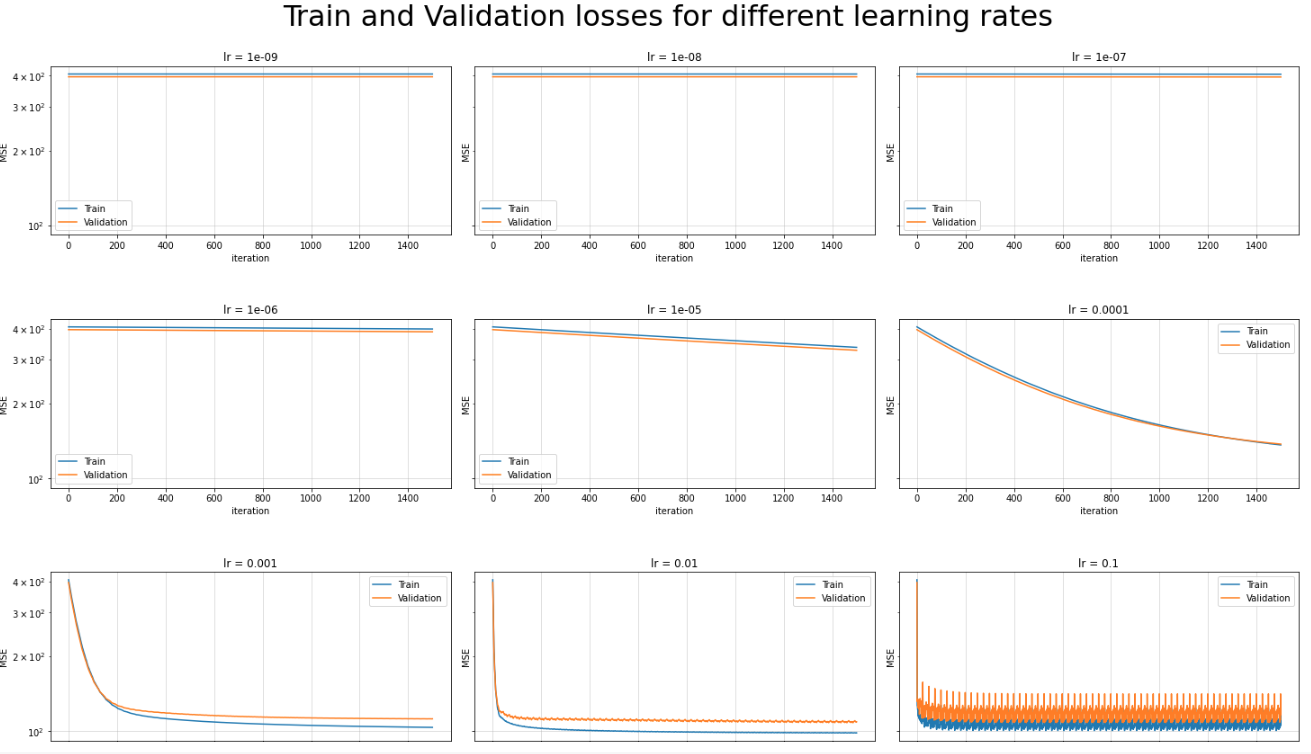
**Section 1: Linear regression implementation**

**(Q1)**

**(Q2)**



**(Q3)**

****

עבור ערכי learning rate הקטנים שווים ל- 10-6 נקבל שכמעט ואין ירידה בערך הMSE ככל שמספר האיטרציות גדל, זוהי התנהגות אופיינית עבור learning rate קטן מידי. צעדי הגרדיאנט קטנים מאוד ולכן כמעט ולא משפיעים על הMSE, וידרשו מספר רב של צעדים על מנת להגיע להתכנסות.

עבור ערכי learning rate  הגבוהים מ10-5 ונמוכים מ 10-3 נקבל ירידה בערך הMSE, ככל שמספר האיטרציות גדל, אך לא נראית התכנסות – סימן שהlearning rate עדיין קטן מידי.

עבור ערכי learning rate  הגבוהים מ10-3 ונמוכים מ 10-1 נקבל ירידה בערך הMSE, ככל שמספר האיטרציות גדל, והתכנסות סביב trainMSE =100, validationMSE=110 (ערכים מדויקים יצורפו בטבלה שבהמשך). ניתן להסיק מתוצאות אלו שlearning rate בטווח ערכים זה מתאים להגעה להתכנסות ע"י ביצוע 1500 צעדי גרדיאנט.

עבור learning rate = 0.1 ניתן לראות שהגרף "מזגזג", זהו סימן לlearning rate גבוה מידי, (אך לא מספיק גבוה על מנת שנכנה את המצב כהתבדרות). בפועל מה שקורה הוא שהתהליך מגיע לנקודה בקרבת המינימה, וממשיך לבצע צעדים נוספים, הצעד שלאחר מכן יהיה בכיוון המינימה אך מכיוון שהוא יהיה צעד גדול מידי, הנקודה הבאה בתהליך תהיה רחוקה יותר מהמינימה מזאת שקדמה לה. במקרה שלנו תהליך זה קורה באופן מחזורי (התקרבות – התרחקות) וזה מסביר את הקפיצות בגרף.

קצב הלמידה האופטימלי מבין אלו שנמדדו הוא 10-2 , מכיוון שהתוצאות להן הוא הביא היו הטובות ביותר מבין התוצאות. בנוסף, האלגוריתם התכנס מהר יותר עם learning rate = 10-2 מאשר עם 10-3.

**Section 2: Evaluation and Baseline**

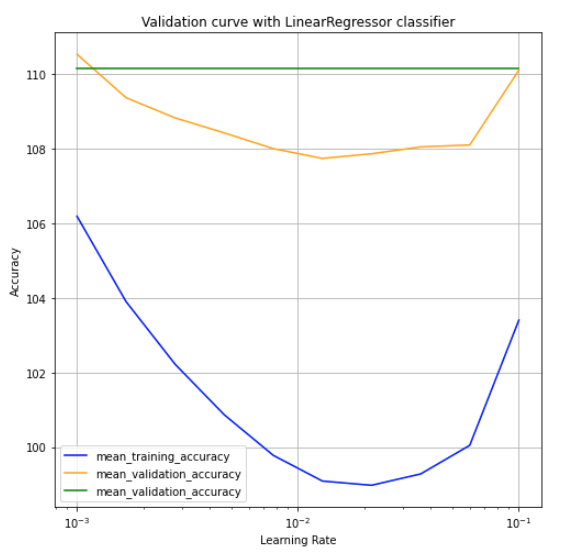
**(Q4)**

|  |  |  |  |
| --- | --- | --- | --- |
| Valid MSE | Train MSE | Section | Model |
| Cross validated | |  |  |
| 109.673 | 110.144 | 2 | Dummy |

**(Q5)**

np.logspace(-3, -1, 10)*הרצנו קרוס ולידציה עם טווח הערכים*

*זאת מכיוון שראינו כי ב10-1 מתחילים לקבל תוצאות המאפיינות* ל learning rare *גבוה מידי, וב10-4* קצב ההתכנסות נמוך מידי.  
התוצאות שקיבלנו

****

ערך ה- learning rare העבורו ה-MSE על סט הולידציה היה הנמוך ביותר הוא

The best lr is 0.012

The mean test accuracy computed for lr=0.012 is 107.739

|  |  |  |  |
| --- | --- | --- | --- |
| Valid MSE | Train MSE | Section | Model |
| Cross validated | |  |  |
| 109.673 | 110.144 | 2 | Dummy |
| 107.739 | 98.976 | 2 | Linear |

The mean training accuracy computed for lr=0.012 is 98.976

**(Q6)**

אילו לא היינו מנרמלים את הפיצרים קודם האימון, לא היינו מקבלים שינוי בפרדיקציות של המודל, זאת מכיוון שבנירמול אנחנו מפעילים מנרמל מסוג min-max, שהוא בסה"כ מבצע פעולה של כפל סקלר והיסט של הוקטור- כל אלו הן פעולות ליניאריות, והמודל שלנו יכול להפוך אותם על ידי הסטה במינוס ההיסט ה-min-max והכפלה בהופכי של המכפיל המנרמל.

לעומת זאת, היינו מקבלים שינוי בערכי הוקטורים w ו-b, שכן הם היו מביאים לידי ביטוי את ההופכי לנירמול כפי שציינו לעיל.

למרות חוסר ההשפעה של נירמול על הפרדיקציות, כדאי בכל זאת לנרמל, וזאת משום שנרצה לייחס חשיבות לערכי הוקטורים w ו-b, שכן אלו אמורים לרמז לנו על החשיבות שהמודל המאומן שלנו מעניק לכל פיצר. על ידי נירמול קבענו סקאלה אחידה, ואפשר להסיק בקווים כללים שפיצר בעל מקדם גדול יותר בערך מוחלט הוא חשוב יותר.

כמו כן, DUMMY REGRESSOR מחזיר תמיד את ממוצע ערך הלייבל בסט האימון, אז הוא לא ישתנה כתוצאה מנירמול, שכן הוא אינו מסתמך על הערכים אותם אנו משנים בפעולת הנרמול.

**Section 3: Linear regression with Lassi**

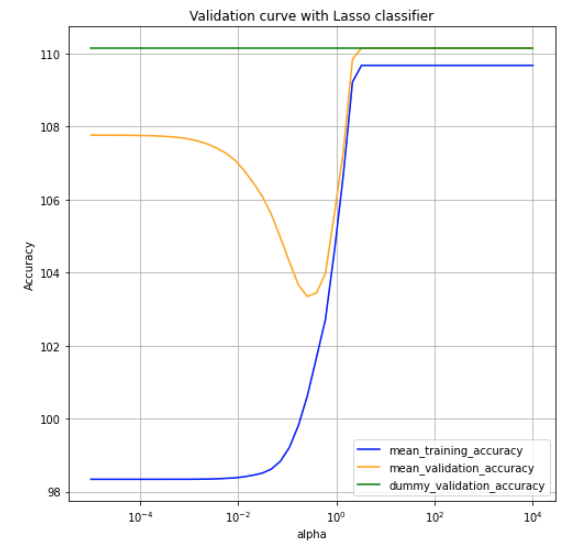
**(Q7)**

הרצנו קרוס ולידיציה עם טווח הערכים הבא עבור קבוע הרגולריזציה:

np.logspace(-5, 4, 50)

בחרנו בסקאלה לוגריתמית כי ניתן באמצעותה לבדוק טווח רחב של ערכים ולקבל אינטואיציה יחסית טובה ע"י מספר מועט של דגימות.

התוצאות שקיבלנו:



The best alpha is 0.244

The mean training accuracy computed for alpha=0.244 is 98.340

The mean test accuracy computed for alpha=0.244 is 103.345

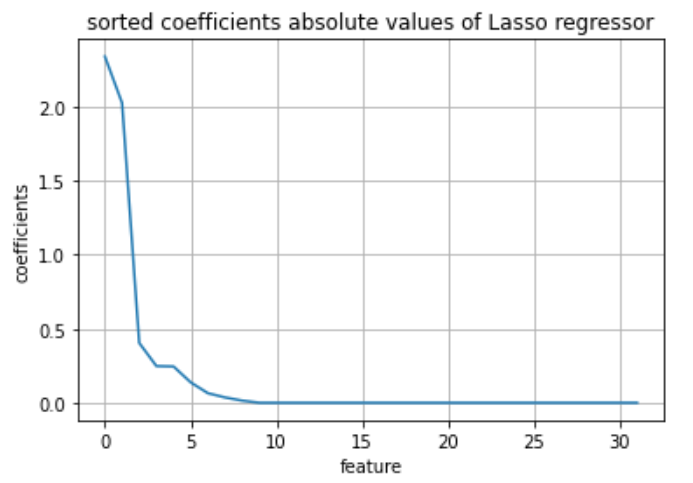
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Valid MSE | | Train MSE | Section | | Model | |
| Cross validated | | |  | |  | |
| 109.673 | | 110.144 | 2 | | Dummy | |
| 107.739 | | 98.976 | 2 | | Linear | |
| 95.4826 | | 99.384 | 3 | | Lasso Linear | |

**(Q8)**

**(Q9)**

|  |  |
| --- | --- |
| Coefficient/Weight | Feature |
| 2.341 | PCR\_01 |
| 2.027 | Sugar levels |
| 0.404 | PCR date |
| 0.248 | O+/O (blood type) |
| 0.245 | Low appetite |

**(Q10)**



**(Q11)**

הערך המוחלט של המשקלים מעניין, כי הוא מראה את היחס בין שינוי בערך הפיצ'ר לבין שינוי בערך פונקציית המטרה. כלומר, עבור פיצ'ר f עם מקדם wf , שינוי Δf בערך הפיצ'ר (עבור ערכים קבועים של שאר הפיצ'רים) יגרום לשינוי של wf\*Δf בערך פונקצית המטרה. לכן, באינטרפטציה על חשיבות הפיצ'רים, פיצ'ר עם מקדם גבוהה (בערך מוחלט) יהיה משמעותי יותר מאשר כזה עם מקדם נמוך.

**(Q12)**

פונקציית המטרה (Objective) אותה מנסים למזער במודל מסוג Lasso מכילה סכום של שני חלקים. החלק הראשון הוא פונקציית המטרה של מודל Linear Regressor והחלק השני הוא רגולריזציה בנורמת L1.

בעוד שפתרון המינימיזציה של החלק הראשון אינווריאנטי לטרנספורמציות לינאריות, החלק השני לא, ולכן אילו לא היינו מנרמלים היינו ככל הנראה מקבלים תוצאות אחרות.

לדוגמה, עבור פיצ'רים שהנרמול "מותח" אותם (scale up), המשקל שלהם עבור אותו מסווג ליניארי היה גבוה יותר במסווג על הדאטא הלא מנורמל מאשר על הדאטא המנורמל. כלומר מסווג שמתנהג זהה לחלוטין במודל LR היה "מקבל ציון" הרבה פחות טוב בחלק הרגולריזציה – ולכן סביר שיהיו תוצאות שונות.

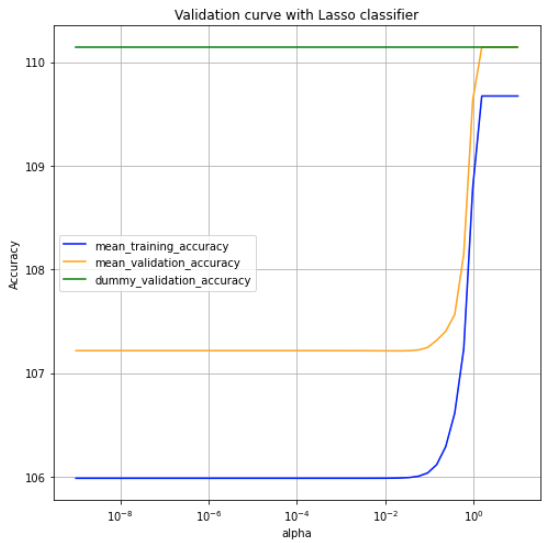
**Section 4: Polynomial fitting (visualization)**

**(Q13)  
Chart, scatter chart

Description automatically generated**

כפי שניתן לראות בתרשים, ההתפלגות של contamination level כפונקציה של PCR\_01 ו-PCR\_05 היא לא לינארית שכן במרחב האוקלידי התלת מימדי, התפלגות לינארית נראית כצורת מישור. לכן, linear regressor, dummy regressor ו- lasso regressor לא יהיו מאוד יעילים, שכן הם כולם מסווגים ליניאריים.

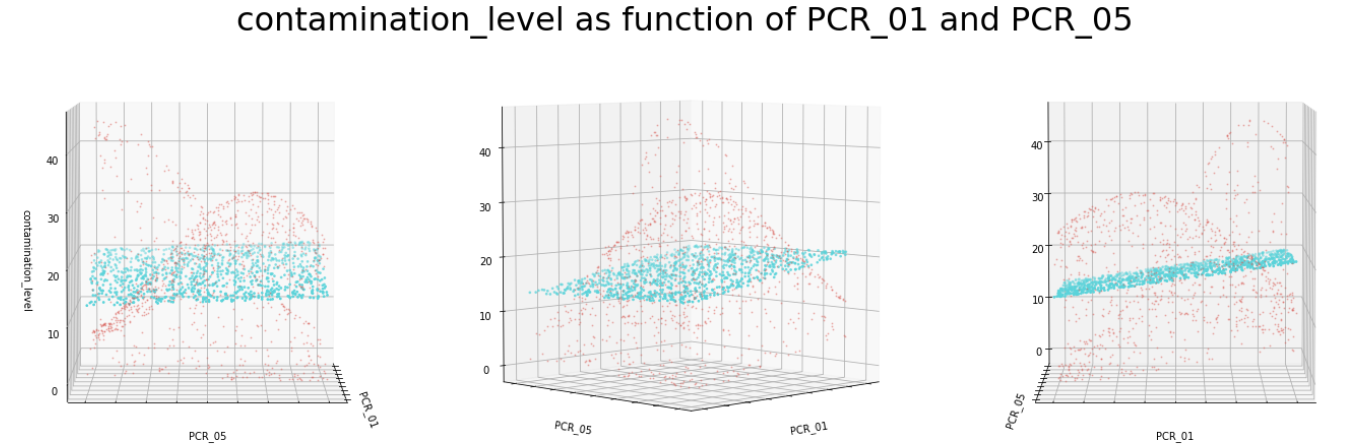
**(Q14)**

****

The best alpha is 0.022

The mean train accuracy computed for alpha=0.022 is 105.985  
The mean test accuracy computed for alpha=0.022 is 107.214

**(Q15)**

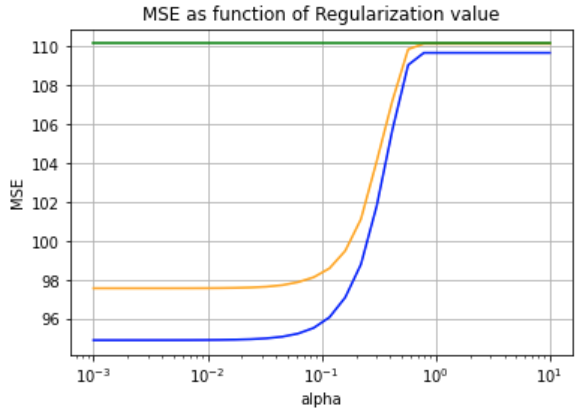
****

כאשר הנקודות הכחולות הן החיזוי של ה-lasso regressor, והנקודות האדומות הן הערכים האמיתיים.

**(Q16)**

נרמול מחדש לאחר ביצוע מיפוי פולינומי חשוב מפני שהפיצ'רים כבר מנורמלים, כלומר הערכים שלהם בטווח של [-1,1].לכן, בהשוואה למונומים ממעלה שני ערכים אלו יקטנו באופן משמעותי, ובכך המשקל של אותם פיצ'רים יגדל. הדבר משפיע על הרגולריזציה משום שלאחרי המיפוי הפולנימי, לכל פיצ'ר יש את הסקאלה שלו ונרמול מחדש של הפיצ'רים מאפשר לרגולריזציה לעבוד כצפוי שכן לכל הפיצ'רים אותה הסקאלה. לכן, החשיבות של כל ערך לא יושפע מהscaling.

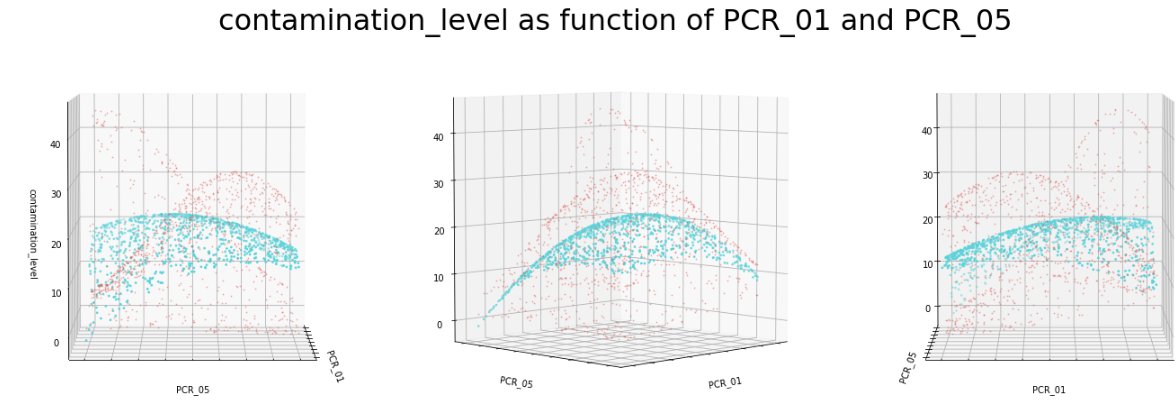
**(Q17)**

****

The best alpha is 0.002

The mean train accuracy computed for alpha=0.002 is 94.898  
The mean test accuracy computed for alpha=0.002 is 97.567

**(Q18)**

****

**(Q19)**

ניתן לראות שהמודל משיג ציון טוב יותר עם מיפוי פולינומי מדרגה 2 מאשר ללא מיפוי כלל.  
ה-Lasso linear regressor לא מדוייק שכן הוא יוצר היפר-מישור בעוד הצורה האמיתית של ה-contamination\_ level כפו' של ו- אינה לינארית אלא דומה לאיחוד של פרבולות (בהערכה גסה).   
יש לציין שחוסר דיוק זה גם בא לידי ביטוי בערך ה- MSE שקיבלנו.

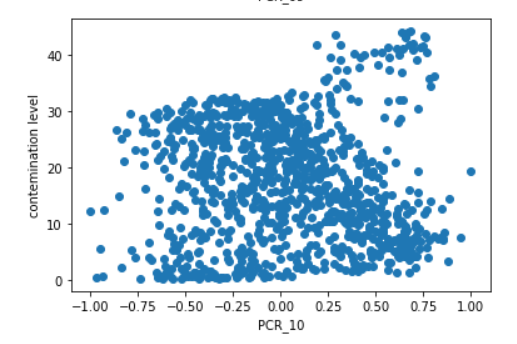
מצד שני ה-Lasso regressor עם שימוש במיפוי פולינומי מדרגה 2 יכול ליצור מישור פרדיקציה בצורת פרבולואיד, או מישור, או ביניהם (ציר אחד ליניארי והשני פרבולי), הרחבה כזאת של הcapacity של מחלקת המודלים, בנוסף לכך שהיא מכילה את המחלקה אליה אנו משווים (רגרסורים ליניאריים) גורמת לנו לצפות בסבירות גבוהה שהדיוק על סט האימון יהיה גבוה יותר. מהצורה הויזואלית של הדאטא שלנו, ניתן היה לצפות שפרבולה תביא לתוצאות טובות יותר ממישור.

אך עם זאת, השינוי בMSE לא היה גדול בצורה משמעותית.

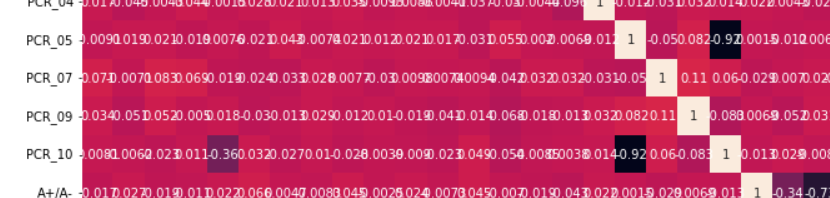
**Section 5: RandomForest fitting of the CovidScore**

**(Q20)**

תחילה הדפסנו גרף contamination level כפונקציה של כל פיצ'ר בנפרד, זאת על מנת לנסות ולזהות תבניות בצורת הגרף. מלבד PCR\_01 ו-PCR\_05 אותם ראינו שניתן לשפר באמצעות מיפוי פולינומי, גם הפיצ'ר PCR\_10 נראה לנו כפיצ'ר שניתן לשיפור על ידי מיפוי שכזה.



בנוסף, ראינו במטריצת הקורלציות בין הפיצ'רים שהוא קורלטיבי מאוד ((-0.92 ל- PCR\_05 וזה יכול לרמז שהוא גם מפוזר בצורה שניתן להבין טוב יותר באמצעות מיפוי פולינומי.



לאחר מכן יצרנו דאטא-סט מצומצם שמכיל רק את הפיצ'רים הבאים :

['PCR\_01', 'sugar\_levels', 'pcr\_date', 'PCR\_05', 'PCR\_10']

הרעיון מאחוריי בחירה זו היה לקיחת 5 הפיצ'רים בעלי המשקל הגבוה ביותר מQ9 , להסיר את הפיצ'רים הבינאריים ולהוסיף את PCR\_10.

לאחר מכן, על מנת לקבל אינטואיציה להתאמת מיפוי לפיצ'ר, ביצענו עבור מיפוי פולינומי בדרגות 2-5 ומיפוי RBF עם גמא בטווח np.logspace(-10,1,12) את התהליך הבא:

* לכל סוג מיפוי p:
  + לכל פיצ'ר f מבין החמישה:

אמן מודל Lasso (עם אלפא מהסעיף הקודם) כשרק הפיצ'ר f ממופה על ידי p.

הרץ Cross\_Validation עם 5 folds על סט האימון מצומצם ל5 הפיצ'רים שנבחרו.

להלן התוצאות הטובות ביותר שהושגו:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Degree | Polynomial mapping MSE test | gamma | RBF MSE test | feature |
| 3 | 104.677 | 1 | 99.667 | PCR\_01 |
| 2 | 108.751 | 0.01 | 101.364 | PCR\_10 |
| 3 | 105.838 | 0.1 | 98.244 | PCR\_05 |
| 4 | 105.422 | 1e-06 | 102.705 | sugar\_levels |
| 5 | 110.104 | 1e-08 | 102.712 | pcr\_date |

מכאן ניתן להניח הנחה פשטנית שRBF מסביר טוב יותר כל אחד מהפיצ'רים הנ"ל. למשך יתרת החקירה נשתמש בגמא 0.01.

כעת עברנו לעבוד עם מודל מסוג Random Forest Regressor

לצורך תרחיש ייחוס יצרנו מודל עם פרמטרים דיפולטיים שאומן על הדאטא-סט המצומצם שמכיל את חמשת הפיצ'רים הנבחרים. הוא השיג דיוק של 1.064 במדד MSE.

כפי שלמדנו בהרצאות ומופיע בתיעוד של SKLEARN , מודל זה מערב בתוכו בחירה מצומצמת של פיצ'רים עליהם יאומן כל עץ. **(max\_features=1 by default in sklearn)**

לכן חששנו ששימוש בRBF עם n\_components=100 על פיצ'רים מסוימים בעוד על האחרים נבצע מיפוי פולינומי או ללא מיפוי כלל (לכל היותר פיצ'ר יהפוך ל6 פיצרים) – פרקטית יגרום לfeature selection של הפיצ'רים שמופו עם RBF.

על מנת לבדוק השערה זו, הרצנו RF עם מיפוי RBF לפיצר בודד ובדקנו את תוצאותיו על שני דאטא-סטים: אחד המכיל את חמשת הפיצרים ואלו החדשים שנוספו עם המיפוי, ואחד המכיל רק את הפיצר הממופה ואלו החדשים שנוספו. על שניהם הרצנו cross-validation וקיבלנו הפרש אפסי בשיעור דיוק. לכן הנחנו שההנחה נכונה וזוהי אכן ההתנהגות.

לכשהבנו טוב יותר את היחסים בין מיפוי RBF לבין מודל RF, החלטנו לבחור פיצרים (כלומר למפות RBF) בסביבה ניטרלית יותר. לכן יצרנו את התהליך הבא:

לכל תת קבוצה של חמשת הפיצרים הנבחרים מגודל 3, אמן מודל RF על דאטא-סט המצומצם לתת הקבוצה שממופה RBF, והרץ cross-validation.

|  |  |  |
| --- | --- | --- |
| No mapping test MSE | RBF test MSE | 3 features |
| 62.3528433849704 | 80.04773956693036 | ('PCR\_01', 'sugar\_levels', 'pcr\_date') |
| 0.8274886013181202 | 30.139866778801945 | ('PCR\_01', 'sugar\_levels', 'PCR\_05') |
| 11.401216838966047 | 35.994933672530806 | ('PCR\_01', 'sugar\_levels', 'PCR\_10') |
| 107.91143127967098 | 104.3198779396389 | ('PCR\_01', 'pcr\_date', 'PCR\_05') |
| 110.84936635372142 | 106.63743379377324 | ('PCR\_01', 'pcr\_date', 'PCR\_10') |
| 109.52208042002397 | 105.94698662062808 | ('PCR\_01', 'PCR\_05', 'PCR\_10') |
| 63.77732537500649 | 75.1190107352479 | ('sugar\_levels', 'pcr\_date', 'PCR\_05') |
| 70.3828410463046 | 79.04662288217446 | ('sugar\_levels', 'pcr\_date', 'PCR\_10') |
| 59.77387747374438 | 62.44661552462255 | ('sugar\_levels', 'PCR\_05', 'PCR\_10') |
| 123.44622540251969 | 117.58825100817235 | ('pcr\_date', 'PCR\_05', 'PCR\_10') |

התוצאות:

מכאן הסקנו שתי מסקנות:

* הקבוצה הטובה ביותר היא ('PCR\_01', 'sugar\_levels', 'PCR\_05'), וצמצום אליה משפר את הדיוק.
* מיפוי RBF לא מסייע לנו.

לכן בסופו של דבר בחרנו להישאר ללא feature mapping ועם שלושת הפיצ'רים 'PCR\_01', 'sugar\_levels', 'PCR\_05'

**(Q21)**

הרעיון מאחורי מיפוי RBF הוא להעביר את הדאטא המקורי לדאטא ממרחב במימד גבוה יותר, כאשר בו הדאטא יוכל להיות ספרבילי. הדבר עשוי לשפר את ההתנהגות של random forest בכך שהוא יוכל להתמודד עם סיבוכיות גבוהה יותר. בשימוש במיפוי RBF אנו נצפה מהשגיאת האימון לקטון שכן הrandom forest יוכל לעבוד על הדאטא החדש בצורה טובה יותר. עם זאת, שגיאה הואלידציה לא בהכרח תקטן, שכן המידע החדש עשוי לא לייצג באופן לא מדוייק את ההתפלגות של הדאטא בגלל overfitting.

בנוסף, ישנם גורמים במיפוי RBF שעלולים לפגוע בתוצאות של RF, למשל, כפי שנידון בשאלה 20, הבחירה האקראית של דוגמאות עלולה לגרום לתופעה שמזכירה feature selection, מכיוון שהפכנו פיצ'ר אחד ל100 פיצרים אחרים.

לדוגמה, עבור 2 פיצרים A ו-B , עם מיפוי RBF לA, ואימון מודל RF המקבל החלטה על סמך 2 דוגמאות , ההסתברות לבחור קבוצה שמכילה את A וB ללא המיפוי שווה לחצי, בעוד שלאחר המיפוי שווה ל: 1-(100/101)2 = 0.02

**(Q22)**

אחד ההבדלים המרכזיים בין שני המודלים הללו הוא היכולת לזהות תבניות לא ליניאריות. בעוד שמודל ליניארי יכול לזהות ביעילות רק התנהגויות הקרובות לליניאריות, ועל ידי שימוש במיפוי פולינומי להגדיל את טווח הזיהוי להתנהגויות פולינומיות עד לדרגה המקסימלית של המיפוי, מבניות נוקשה זו מגבילה אותנו בגלל התכונות של פולינומים – עד דרגה 5-10 הם "רגועים" ולא משתוללים, אך בדרגות גבוהות יותר הסיכוי לoverfitting גדל וקשה לזקק מהם מגמה ברורה.  
 RF מסוגל לזהות תבניות שאינן מתנהגות בצורה ליניארי או פולינומית. היכולת הזאת היא פועל יוצא של מבנה המודל – המון עצי החלטה אקראיים, ושיקלול/מיצוע כלשהו שלהם מוביל ליכולת זיהוי גבוהה של תבניות ללא קשר לצורתם. יתרה מכך, יתרון נוסף של RF הוא שאין צורך לבצע את המיפוי הפולינומי בעצמנו לפני כן, מה שגורם להשגת תוצאה דומה גם ללא עיבוד מקדים של הדאטא וידע/הנחות נוספות עליו.

**(Q23)**

|  |  |
| --- | --- |
| תמונה שמכילה שולחן  התיאור נוצר באופן אוטומטי | תמונה שמכילה שולחן  התיאור נוצר באופן אוטומטי |

The optimal combination is {'min\_samples\_leaf': 1, 'n\_estimators': 136}  
Mean train MSE for optimal parameters: 0.168  
Mean valid MSE for optimal parameters: 0.796

**(Q24)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Valid MSE | | Train MSE | Section | | Model | |
| Cross validated | | |  | |  | |
| 109.673 | | 110.144 | 2 | | Dummy | |
| 107.739 | | 98.976 | 2 | | Linear | |
| 95.4826 | | 99.384 | 3 | | Lasso Linear | |
| 0.796 | | 0.168 | 5 | | RF Regressor | |

**Section 6: Testing your models**

**(Q25)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test MSE | | Valid MSE | | Train MSE | Section | | Model | |
| Retrained | | Cross validated | | |  | |  | |
| 100.102 | | 109.673 | | 110.144 | 2 | | Dummy | |
| 90.224 | | 107.739 | | 98.976 | 2 | | Linear | |
| 89.314 | | 95.4826 | | 99.384 | 3 | | Lasso Linear | |
| 0.508 | | 0.796 | | 0.168 | 5 | | RF Regressor | |

ניתן לראות שהמודל הטוב ביותר הוא ה-RF Regressor. ראינו שהדאטא לא לינארי שכן עבור ה-dummy והlasso linear זה מוצא את אותן תוצאות. בנוסף, ראינו ביטוי חזק מאוד ליכולות של RF כמודל. הביצועים שלו על הדאטא הנ"ל היו מעולים. מסקנה נוספת שהסקנו היא שהכוח של feature mapping סטנדרטי הוא מוגבל, ושלא תמיד הוא נחוץ.