תרגיל בית 3 רטוב

מבוא למערכות לומדות

תאריך הגשה – 28.06.2022

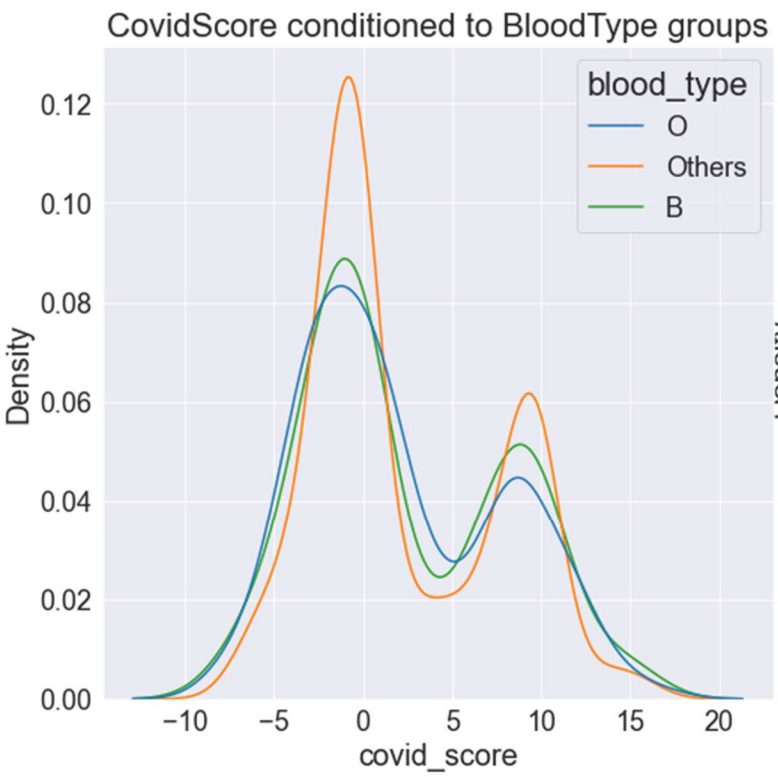
מגישים:

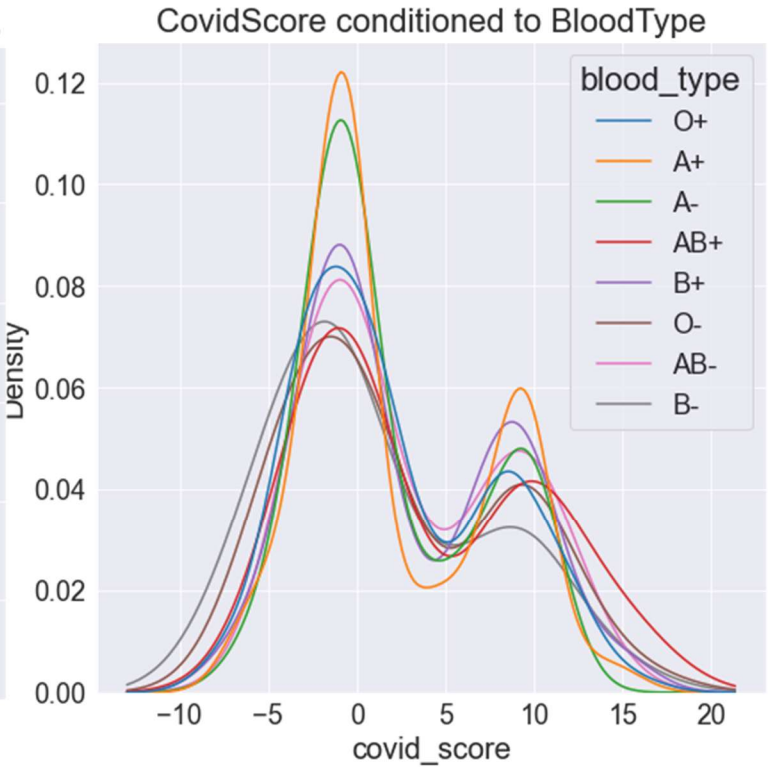
יואב יעבץ, תעודת זהות 212617864.

מור לוי, תעודת זהות 211810452.

חלק 1

שאלה 1:





שאלה 2: שני הגרפים המצורפים למעלה מתארים את חלוקת ערך מאפיין המטרה CovidScore בהתאם למאפיין BloodType, כאשר הגרף מימין מבצע חלוקה זו לפי כל סוג אפשרי של המאפיין BloodType והגרף משמאל עושה זאת לפי חלוקה לכמה קבוצות של סוגי המאפיינים.

נשים לב כי הגרף מצד ימין מראה לנו כמה תכונות אשר מתקיימות:

- גרף הצפיפות של מאפיין המטרה CovidScore של מטופלים עם דם A+ או עם דם A- דומה אחד לשני.
- גרף הצפיפות של מאפיין המטרה CovidScore של מטופלים עם דם O+ או עם דם O- דומה אחד לשני.
- גרף הצפיפות של מאפיין המטרה CovidScore של מטופלים עם דם B+ או עם דם B- או עם דם AB+ או עם דם AB- דומה אחד לשני.

לכן הגיוני להוסיף לחלק את סוג הדם לשלוש קבוצות, כפי שתואר בשאלה 1, כי בכל קבוצה גרף הצפיפות המתקבל של מאפיין המטרה CovidScore דומה לגרף הצפיפות המתקבל של מאפיין מטרה זה ללא איחוד סוגי דם אילו לקבוצה אחת. משמע לא איבדנו מידע באיחוד קבוצות זה והורדנו את מספר המאפיינים – מה שמוריד את סיבוכיות מחלקת המודולים שלנו, ויכול להביא לתוצאות יותר טובות בעתיד (מניעת overfitting).

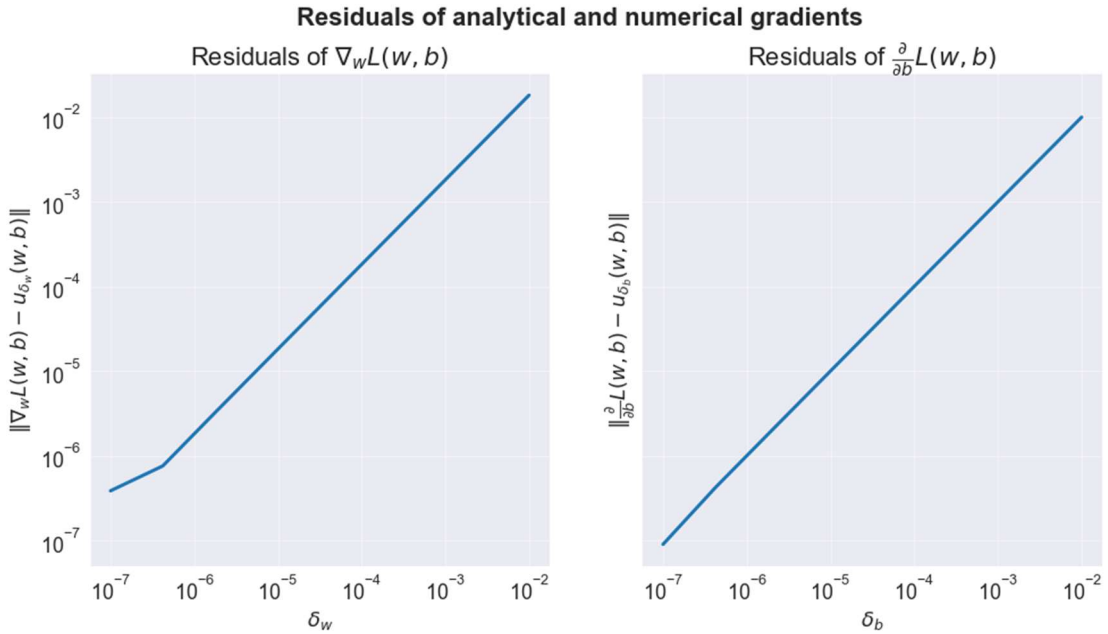
חלק 2

<u>שאלה 3</u>: מתקיים $L(w,b)=\frac{1}{m}\sum_{i=1}^{m}\left(w^T x_i+b-y_i\right)^2=\frac{1}{m}\left|\left|Xw+1_m*b-y\right|\right|_2^2$. כעת:

$$\frac{\partial}{\partial b}L(w,b)=\frac{\partial}{\partial b}\left(\frac{1}{m}\sum_{i=1}^{m}(w^T x_i+b-y_i)^2\right)=\frac{1}{m}\left(\sum_{i=1}^{m}\frac{\partial}{\partial b}((w^T x_i+b-y_i)^2)\right)$$

$$=\frac{2}{m}\sum_{i=1}^{m}(w^T x_i+b-y_i)=\frac{2}{m}\sum_{i=1}^{m}(w^T x_i-y_i)+\frac{2}{m}\sum_{i=1}^{m}b=\frac{2}{m}\sum_{i=1}^{m}(w^T x_i-y_i)+2b$$

<u>שאלה 4</u>:



<u>שאלה 5</u>: גרף מצורף למטה. ניתן לראות כי מתקיים:

- עבור ערכי learning rate קטנים מ-e^{-5} מתקבל כי ה-MSE Loss על ה-train data ו - Validation data קבוע במקומו (עד כדי ירידה זניחה). התנהגות זו הגיונית, כי ניזכר שנוסחת העדכון של SGD היא $\theta-\eta*\frac{\partial}{\partial\theta}L\to\theta$, לכן אם ה-learning rate קטן מאוד לא נעדכן את הפרמטרים, לכן ה-MSE Loss לא ישתנה.
- עבור ערכי learning rate גדולים מ-e^{-5} וקטנים מ-0.01 ניתן לראות כי ה-MSE Loss על ה-train וה-validation כתלות במספר ה-iteration יורד באופן עקבי. זו התנהגות הגיונית כי אנחנו מצפים שעבור ערכי למידה טובים אלגוריתם SGD יעדכן את סט הפרמטרים כך שה-MSE Loss ירד עם הזמן.

נוסף על כך, עם גדילת ערך ה-learning rate לכיוון 0.01 ניתן לראות כי ערך ה-MSE Loss עבור ה-train וה-validation אליו מגיעים לאחר 1500 איטרציות נמוך יותר. זו גם כן התנהגות הגיונית, הרי ברור כי ערכי למידה שונים יגררו קצב למידה שונה. עבור כל ערכי הלמידה ניתן לראות ירידה ב-MSE Loss אבל קצב הירידה שונה לכל אחד, כי עדכון סט הפרמטרים באלגוריתם שונה לכל אחד. יכול להיות כי עבור כל ערכי למידה אלו עבור

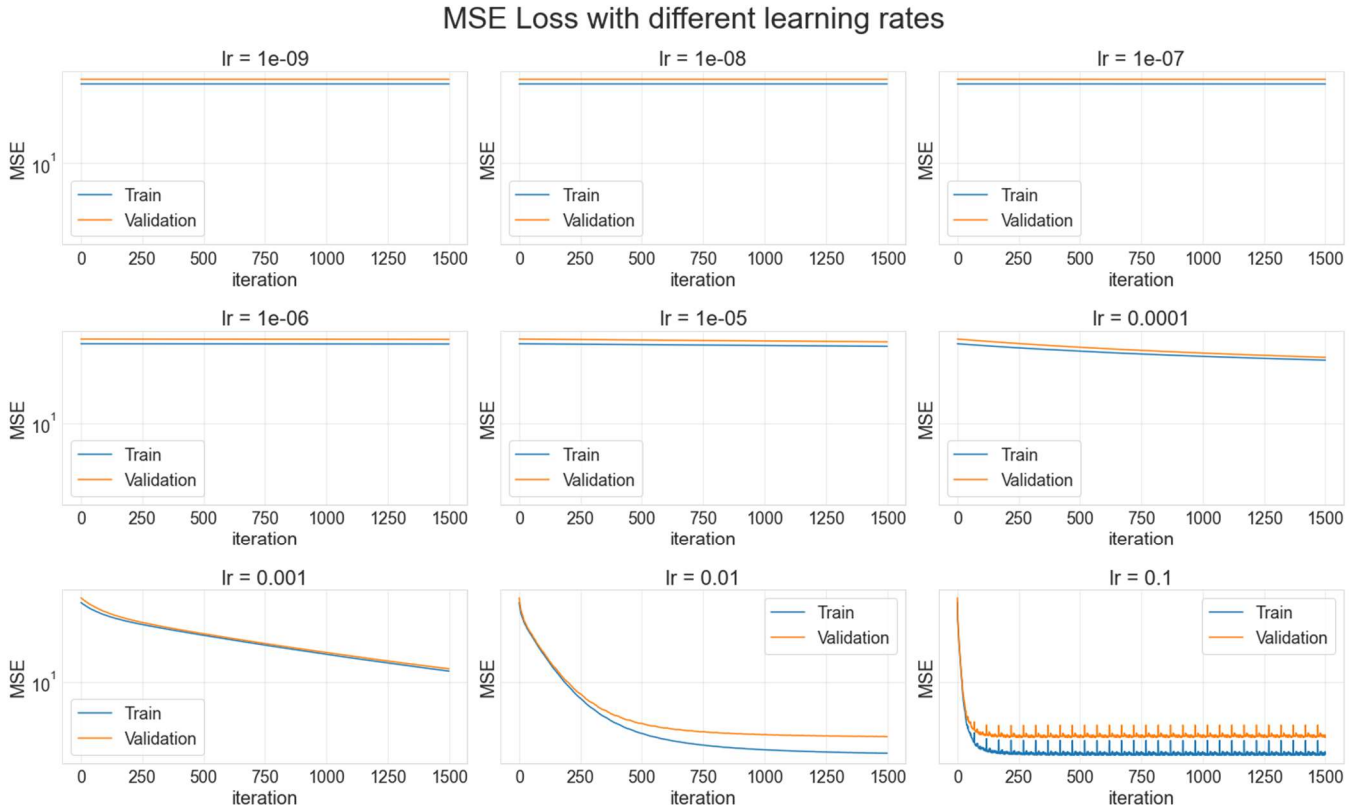
מספר איטרציות שואף לאינסוף נתכנס לאותו ה-loss, אך כל עוד זה לא מתקיים – קצב הלמידה שונה ולכן גם ה-loss אליו מגיעים לאחר 1500 איטרציות שונה.

- עבור ערך learning rate השווה ל-0.1 ניתן לראות כי ה-MSE Loss על ה-train וה-validation יורד עם הזמן, אך אינו יורד בצורה מונוטונית - קיימות קפיצות בערך ה-loss עבור שני הסטים של המידע באיטרציות שונות לאורך למידת האלגוריתם. דבר זה קורה כתוצאה מבחירת learning rate גבוה מדי, מה שמוביל לעדכון סט הפרמטרים בצורה גדולה מדי כאשר מתקרבים אל נקודת המינימום בתחתית. לכן אנחנו לא מצליחים להתכנס אל סט הפרמטרים שיביא את ה-loss נקודת המינימום, כי מדלגים מעליה כל פעם (הערה - כל פעם הדילוג מקרב אותה אל נקודת המינימום של ה-loss עוד טיפה).

קצב הלמידה הטוב ביותר הוא בין 0.01 ל-0.1 לפי הגרף שנראה מטה. מכיוון שעבור learning rate השווה ל-0.1 נקבל הרבה קפיצות, נבחר את ה-learning rate האופטימלי בסעיף זה להיות 0.01.

הגדלת מספר ה-gradient steps עבור קצב הלמידה הטוב ביותר לא ישפר את שגיאת ה-MSE, זאת מכיוון שעבור קצב הלמידה הטוב ביותר הספקנו להתכנס לאחר 1500 איטרציות .

אך, הגדלת מספר ה-gradient steps עבור learning rates קטנים יותר יכולה להוביל אותנו אל התכנסות לנקודת המינימום של ה-loss, והתכנסות זו היא באופן מונוטוני ללא קפיצות. לכן הגדלת מספר איטרציות העדכון יוביל אותנו לעדכון וכוונון סט הפרמטרים כך שנגיע לטוב ביותר המביא לנו את ה-loss המינימלי.

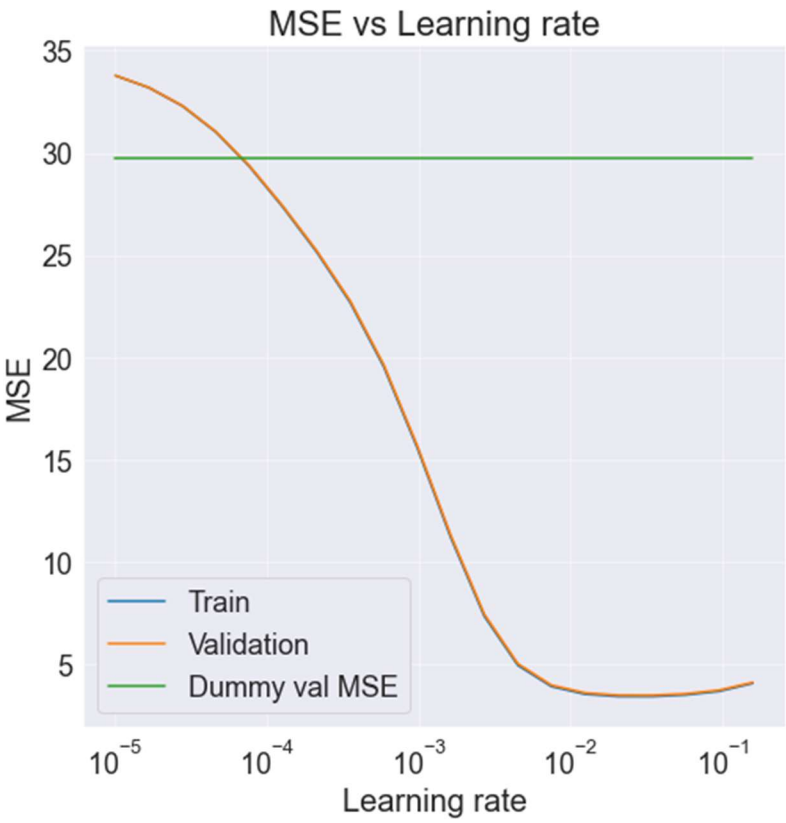


חלק 3

שאלה 6 :

Model	Section	Train MSE	Valid MSE
		Cross Validate	
Dummy	2	29.669	29.737

שאלה 7 :




```
LinearRegressor best lr: 0.034
LinearRegressor best lr train MSE: 3.449
LinearRegressor best lr val MSE: 3.506
```

Model	Section	Train MSE	Valid MSE
		Cross Validate	
Dummy	2	29.669	29.737
Linear	3	3.449	3.506

שאלה 8 : נחלק את התשובה שלנו עבור כל מודול :

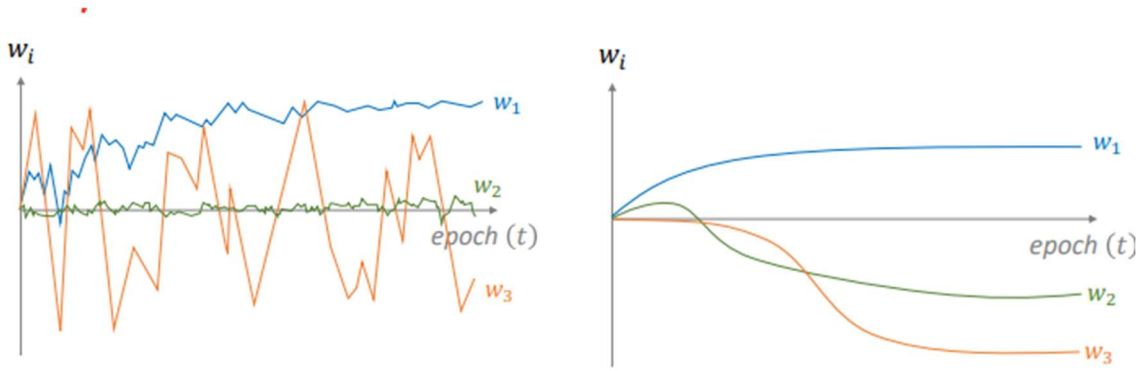
1. עבור ה-Dummy Regressor ה-training performance לא היה משתנה במידה ולא היינו מנרמלים את המאפיינים לפי הלמידה. בהינתן ה-training data מודול זה מחשב את ממוצע ערך מאפיין המטרה וחוזה לכל נקודה חדשה את ממוצע זה בתור ערך מאפיין המטרה שלה.

בהתחשב בכך שאנחנו לא מנרמלים את ערך מאפיין המטרה, נרמול לא היה משנה את תוצאת החיזוי של מודול זה ולכן לא היה משנה את ה-training performance – החיזוי שלו לכל נקודה ב-train היה נשאר זהה מכאן גם השגיאה (כי השגיאה מוגדרת על ידי $\frac{1}{m}\sum_i(h(x_i)-y_i)^2$ לכן חיזוי זהה גורר שגיאה זהה). כל זה כמובן בהנחה שאנחנו לא מנרמלים את מאפיין המטרה!

2. <u>עבור ה-Linear Regressor</u> ה-training performance היה משתנה במידה ולא היינו מנרמלים את המאפיינים לפני הלמידה, נסביר מדוע.

ללא נרמול כל מאפיין היה עלול להיות באזור אחר ולכן בעל סקאלה אחרת משל שאר המאפיינים. מטרת המודול היא למזער את ה-loss אשר מוגדר על ידי $L(w,b)=\frac{1}{m}\sum_{i=1}^{m}(w^Tx_i+b-y_i)^2$, לכן מאפיינים בעלי סקאלה גבוהה יותר יהיו בעלי חשיבות גדולה יותר מכיוון שיתרמו ערך גבוה יותר ל-loss. אנחנו משתמשים ב-learning rate קבוע עבור כל המאפיינים בעת ה-gradient step לעדכון ווקטור המשקולות w ביחס ל-loss מה שעלול לגרום לווקטור המשקולות w להיות מעודכן בצורה לא טובה – נגזרת ה-loss לפי מאפיינים בעלי סקאלה גבוהה יותר תהיה בעלת ערך יותר גבוה, מה שיגרום לקפיצות בעדכון המקדם המתאים להם בווקטור המשקולות w בעת צעד ה-gradient, ולהיפך עבור מאפיינים בעלי סקאלה נמוכה יותר. כלומר הבעיה היא כי קצב הלמידה קבוע לכל אחד מהמאפיינים, ומכיוון שהם עלולים להיות בסקאלות שונות – קצב למידה זהה עבור כולם לא יביא אותנו לתוצאה הרצויה (אם היינו יכולים להביא קצב למידה אחר עבור כל מאפיין, יכול להיות כי זה היה פותר את הבעיה).

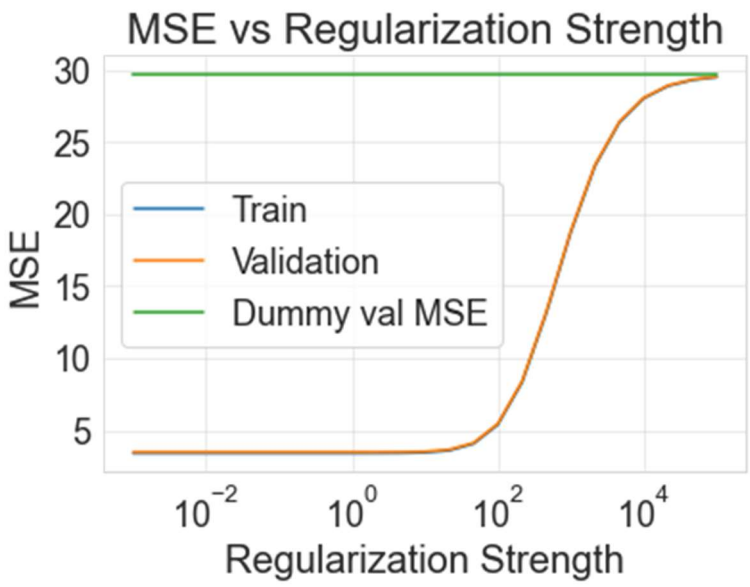
במידה וננרמל את המאפיינים נקרב אותם להיות באותה הסקאלה מה שיעזור לנו לעדכן את ווקטור המשקולות w בצורה טובה יותר. כזכור הוא משפיע על ה-prediction , ולכן ווקטור משקולות w שונה מביא גם ל-training performance שונה. בתמונה מצד שמאל מופיעה דוגמה לעדכון מקדם בווקטור המשקולות w לאורך איטרציות רבות של gradient steps לפני נרמול הנתונים, ובתמונה מצד ימין מופיע אותו עדכון אך לאחר נרמול הנתונים. ניתן לראות שהמקרה השני הוא הרצוי.



סיבה נוספת לכך ש-training performance ישתנה אם לא היינו מנרמלים את המאפיינים היא נקודות חריגות. במידה ולא היינו מנרמלים את המאפיינים עלולות היו להיות נקודות חריגות רחוקות, אשר השפעתן על למידת המודול גבוהה מאוד למרות שאינן מייצגות את הרוב. נרמול מקרב את כל הנקודות לאותו האזור ומונע מאותן נקודות חריגות כאלו להתקיים ולהשפיע לרעה על למידת המודול.

חלק 4

שאלה 9:




```
Best alpha: 0.46
Best alpha train MSE: 3.392
Best alpha val MSE: 3.467
```

שאלה 10:

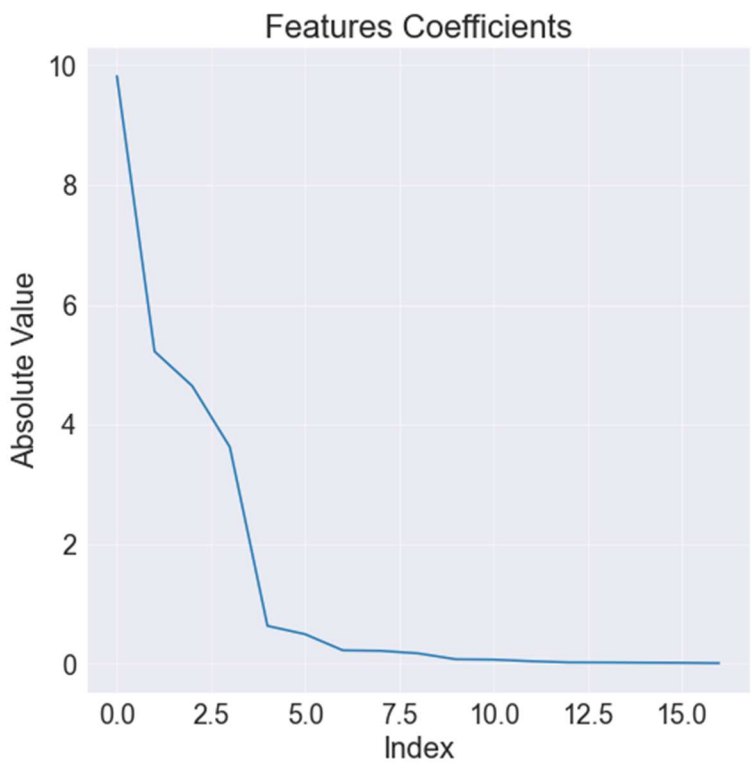
Model	Section	Train MSE	Valid MSE
		Cross Validate	
Dummy	2	29.669	29.737
Linear	3	3.449	3.506
Ridge Linear	4	3.392	3.467

שאלה 11:

```
symptom_sore_throat : 9.819880410829912
sport_activity : 5.216490305730478
symptom_shortness_of_breath : 4.640252991193262
PCR_01 : 3.618601592736039
symptom_cough : 0.6289507061706399
```

הערה – ערכי המקדמים המוצגים הם בערך מוחלט!

שאלה 12:



שאלה 13: ראשית נסביר מה מייצג כל מקדם מאפיין בווקטור w. אנחנו עוסקים במודל Ridge Regressor, זהו מודול לינארי שנותן פרדיקציה לפי $h(x) = w^T + b$ ומנסה למזער את ה-loss המוגדר על ידי $L(w,b) = \frac{1}{m}\sum_{i=1}^{m}(w^T x_i + b - y_i)^2 + \lambda\left|\left|w\right|\right|_2^2$ בעזרת שיטת SGD. המטרה שלנו היא למזער את פונקציית ה-loss על ידי מציאת וקטור מקדמי מאפיינים w וסקלר b שיביאו אותנו אל השגיאה המינימלית. ניתן לראות לפי ביטוי ה-loss כי מאפיין עם מקדם גבוה יותר יהיה בעל השפעה יותר גבוהה על ה-loss, ולהיפך – מאפיין עם מקדם נמוך יותר יהיה בעל השפעה נמוכה יותר על ה-loss. מכאן ניתן להסיק כי מקדם מאפיין מעיד על חשיבות אותו המאפיין בחיזוי מאפיין המטרה המוגדר. כלומר כי אם למאפיין ערך המקדם שלו גבוה יותר – הוא חשוב יותר לחיזוי, ולהיפך.

משאלות 11,12 ניתן לראות כי קיימים ארבעה מאפיינים להם ערך מקדם גבוה וכי לכל שאר המאפיינים במדגם ערך המקדם נמוך (פי 10 ויותר). דבר זה מעיד על כך שחיזוי מאפיין המטרה רק לפי אותם ארבעת המאפיינים בעלי המקדם הגבוה ביותר בערך מוחלט יהיה באופן יחסית זהה לדיוק של חיזוי מאפיין המטרה בעזרת כל המאפיינים. חשוב לציין כי אף יכול להיות שחיזוי בעזרת ארבעת מאפיינים אלו יניב דיוק גבוה יותר – שכן צמצום מספר המאפיינים מקטין את סיבוכיות המודול, מה שיכול למנוע overfitting על ה-training data (וכן כי ארבעת מאפיינים אלו חשובים בהרבה לחיזוי מאשר אחרים).

כלומר ה-magnitude של מקדמי המאפיינים מעיד על איזשהו feature selection – בחירת המאפיינים החשובים ביותר לחיזוי מאפיין המטרה לפי גודל המקדם שלהם.

הערה – magnitude זה יכול להעיד על חשיבות כפי שתוארה לעיל רק במידה והמאפיינים מנורמלים, אחרת יכול להיות כי מקדם מאפיין יהיה בעל ערך גבוה או נמוך ללא קשר לחשיבות שלו, אלא בגלל הסקאלה של אותו מאפיין.

דבר נוסף אשר חשוב לשים לב בקשר ל-magnitude של מקדמי המאפיינים הוא הניסיון של מודול ה-Ridge Regressor לתת ערך כמה שיותר נמוך למרבית מקדמי המאפיינים. זה נובע בשל הכנסת הרגוליזציה למודול וניסיונו למזער את ה-loss, שבין היתר נקבע לפי גודל המקדמים.

<u>שאלה 14</u>: אם היינו בוחרים לא לנרמל את המאפיינים לפני למידת מודול ה-Ridge Regressor ה-training performance של המודול היה משתנה. נסביר מדוע.

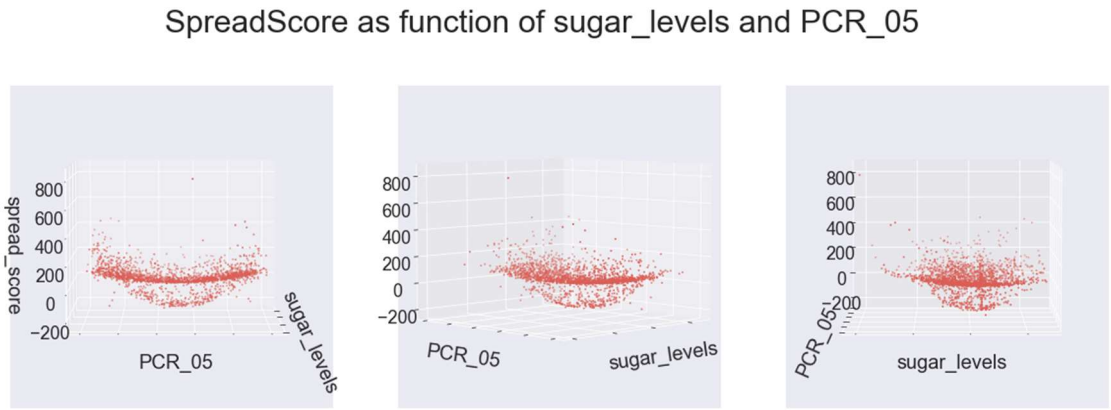
מאפיינים אשר לא מנורמלים יכולים לחיות באזורים שונים, ומכאן להיות בעלי סקאלה שונה. מאפיינים בעלי סקאלה גבוהה יותר משפיעים במידה גדולה יותר על ערך ביטוי ה-loss של המודול, ולכן סקאלה גדולה של מאפיין עלולה לגרום למקדם המתאים לאותו המאפיין בוקטור w להיות נמוך. לכן חוסר נרמול המאפיינים והבאתם לאותה סקאלה עלול לגרום לנו לחיזוי פחות מדויק של מאפיין המטרה, בכך שהמודול כעת עשוי לפרש מאפיינים חשובים יותר כחשובים פחות, ולהיפך.

נסתכל על הדוגמה הבאה, שממחישה את מה שכתוב לעיל: נניח כי נתון לנו מאפיין בעל סקאלה גדולה לפני נרמול, אשר חשיבותו לחיזוי מאפיין המטרה גבוהה. היינו רוצים כי אותו המקדם המתאים לאותו מאפיין יהיה גדול – בשאלה קודמת הבנו שיש קורלציה בין חשיבות המאפיין לחיזוי לבין המקדם המתאים לו. אך אם אותו המקדם יהיה גדול לא נצליח למזער את ה-loss באופן מיטבי. לכן אנחנו מקבלים עבור אותו המאפיין מקדם קטן, בסתירה לקורלציה שהראנו כי קיימת מסעיף קודם.

נוסף על כך חוסר נרמול המאפיינים לפי הלמידה יכול להביא לשגיאה לא רצויה של הרגוליזציה שאותה הכנסו אל ה-objective של מודול ה-Ridge Regressor. כל מאפיין עשוי להיות בעל סקאלה שונה, ולמשל מאפיינים עם סקאלה קטנה ובעלי חשיבות גדולה לחיזוי – נרצה כי המקדם המתאים להם בוקטור w יהיה גדול (על מנת להדגיש את החשיבות שלהם). אך מקדם גדול בוקטור w יביא לעונש רגוליזציה. לכן התוצאות עלולות להשתנות, ולא בהכרח לטובה.

חלק 5

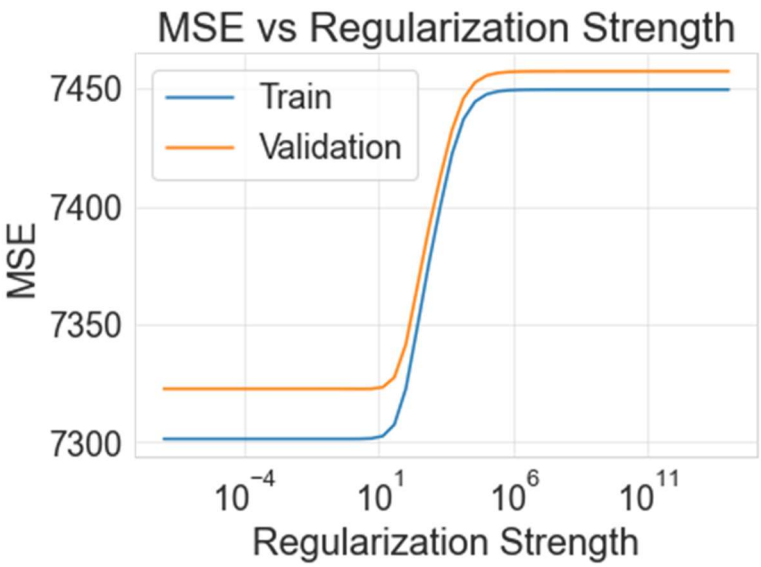
שאלה 15:



ניתן לראות כי לא קיימת תלות לינארית בין התפלגות מאפיין המטרה SpreadScore לבין המאפיינים sugar_levels ו-PCR_05 - אין מישור שיכול להסביר את התפלגות מאפיין המטרה ביחס לשני המאפיינים. המבניות הקיימת היא טבעת הנחה על משטח לינארי אשר ביחד יוצרים סוג של פרבולה. מבניות זו מובילה אותנו למסקנה כי במידה ונפעיל רגרסיה לינארית נקבל אחוז דיוק נמוך שכן המידע לא ניתן להפרדה לינארית. פתרון לבעיה זו יכול להיות העברת המידע בעזרת feature mapping למרחב אחר, בו המידע כן יהיה מופרד לינארית.

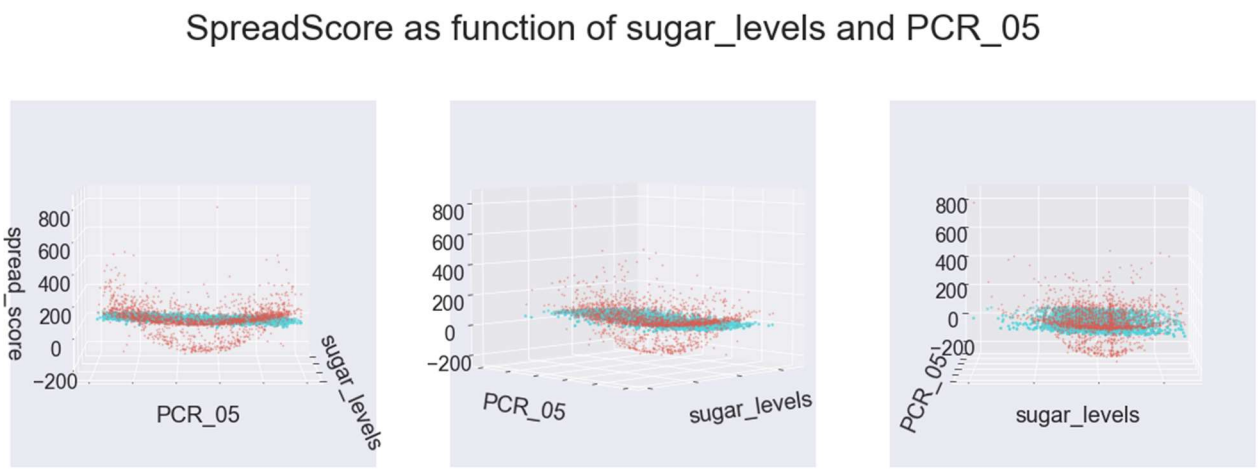
דבר נוסף שחשוב לציין הוא שנראה כי ההתפלגות מגיעה מצורה מוגדרת מסוימת, שהיא לא לינארית. לכן עדיף להבין את מבניות הבעיה ולהמיר אותה למרחב מתאים לה, בה נוכל להשיג את דיוק החיזוי הטוב ביותר.

שאלה 16:




```
Best alpha: 1.93
Best alpha train MSE: 7301.353
Best alpha val MSE: 7322.539
```

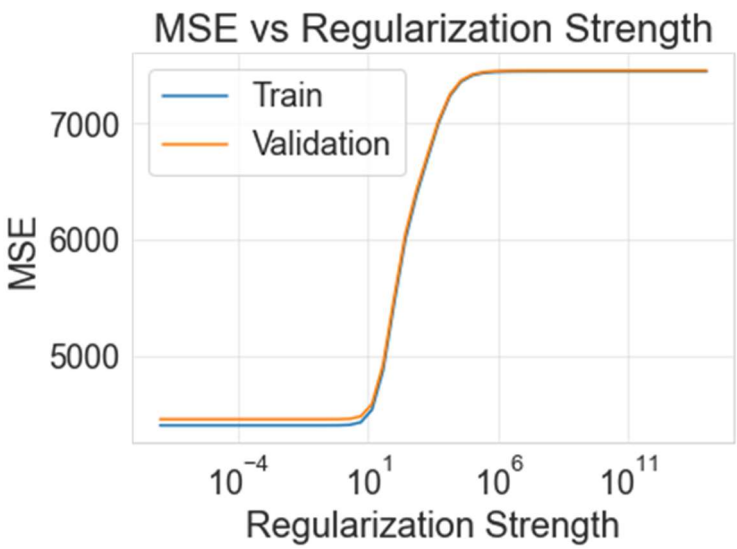
שאלה 17:



שאלה 18: נסביר מדוע כדאי לעשות נרמול מחדש לאחר הפעלת ה-polynomial mapping. המאפיינים אותם אנחנו ממפים לפולינום מדרגה 2 כבר מנורמלים לפני המיפוי בעזרת השיטה Min-Max או בעזרת השיטה Standardization. בשיטת הנרמול הראשונה ערכי המאפיינים נמצאים בתחום $[0,1]$ לכן המון מהערכים של המאפיינים קטנים. גם בשיטת הנרמול השנייה מכיוון שאנחנו "ממרכזים" את הערכים (מחסירים מהם את התוחלת) המון מערכי המאפיינים שנורמלו בצורה זו יהיו קטנים.

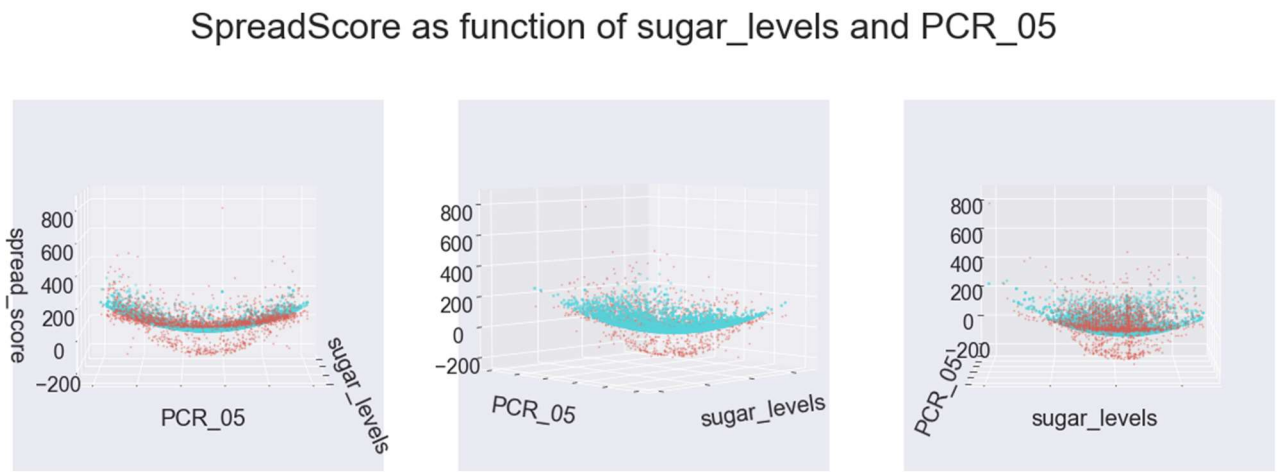
הפעלת polynomial mapping לפולינום מדרגה 2 על וקטור שמכיל ערכים קטנים, ספציפית ערכים בתחום $[-1,1]$ יוביל להקטנת ערכים אילו עוד יותר (העלאה בריבוע של ערך בין 1- ל-1 מקטינה את אותו הערך). לכן לאחר הפעלת המיפוי יהיו המון וקטורים שערכם יהיה קטן מאוד. דבר זה גורם לכך שכעת המאפיינים לא משקפים בצורה הטובה ביותר את הגודל ואת המידע שהיו להם לפני הפעלת המיפוי הפולינומי. לכן על מנת לשחזר את מידע זה כדאי להפעיל נרמול מחדש שיחזיר את אותם המאפיינים לטווח הנכון ולסקאלה בה הם אמורים להיות.

שאלה 19:




```
Best alpha: 0.1
Best alpha train MSE: 4395.618
Best alpha val MSE: 4448.241
```

שאלה 20:



שאלה 21: ניתן לראות כי המודול השני שהפעלנו – עם המיפוי הפולינומי, עדיף בהרבה על המודול הראשון – ללא המיפוי הפולינומי. נשווה בין התוצאות השונות שקיבלנו ונסביר את ההיגיון הנח מאחורי התוצאה שהמודול השני טוב יותר בחיזוי מאשר המודול הראשון.

נשים לב כי עבור מקדם הרגוליזציה הטוב ביותר למודול הראשון אנחנו מקבלים שגיאת MSE של 7,322 בעוד עבור מקדם הרגוליזציה הטוב ביותר למודול השני אנחנו מקבלים שגיאת MSE של 4,448, שתי השגיאות מתייחסות לשגיאה על ה-validation data. כלומר המודול השני טוב כמעט פי שתיים מהמודול הראשון לפי מדד ה-MSE.

ראינו בייצוג הוויזואלי משאלה 15 כי ההתפלגות של מאפיין המטרה כתלות בשני המאפיינים האחרים היא לא לינארית, אלא בעלת צורה מוגדרת אחרת כשל פרבולה. מודול רגרסיה לינארי מנסה למצוא מישור שיחזה בצורה טובה את ההתפלגות של מאפיין המטרה כתלות במאפיינים האחרים, ומכיוון שלא קיים מישור כזה – הגיוני שהוא לא יעבוד.

למדנו בהרצאה כי אם מידע לא ניתן להפרדה לינארית אפשר להפעיל עליו מיפוי ולהעבירו למרחב אחר שם הוא יהיה פריד לינארית. אם נמצא מרחב בו הוא יהיה פריד לינארית נוכל להפעיל באותו המרחב את מודול הרגרסיה שלנו ולמצוא את המישור המפריד המתאים במרחב זה, וזה יאפשר לנו לחזות בצורה טובה יותר את המודול. מכיוון שמבניות ההתפלגות שנוצרה היא פרבולה נכון להעביר את המאפיינים מיפוי לפולינום מדרגה 2, שם כנראה המידע יהיה פריד לינארית. ואכן ניתן לראות שאם מפעילים את מיפוי זה אנחנו מקבלים חיזוי הרבה יותר טוב למידע. אנחנו יצרנו סוג של פרבולה שמקרבת את נקודות החיזוי לנקודות האמת.

חלק 6

<u>שאלה 22</u>: נסביר כיצד ה-training error וה-validation error ישתנו במידה ונשתמש במיפוי ולא במידע המקורי ללא מיפוי. נזכיר כי המידע המקורי שלנו הוא פולינומים מדרגה 1 בעוד כי המיפוי הוא לפולינומים מדרגה 2. מכיוון שפולינומים מדרגה 1 נכללים בפולינומים מדרגה 2 נקבל כי המיפוי מגדיר מחלקת מודולים בעל סיבוכיות יותר גדולה מהמחלקה הראשונה, שכן מחלקת המודולים הראשונה מוכלת בתוך מחלקת המודולים השנייה.

ה-training error כתוצאה מהמיפוי הפולינומי יישאר זהה או יקטן, נסביר מדוע. ראינו בכיתה כי במקרה הכללי הגדלת סיבוכיות מחלקת המודולים גוררת הקטנה של ה-training error. זה קורה מכיוון שהגדלת סיבוכיות מחלקת המודולים מאפשרת ייצוג עשיר יותר ומרחב חיפוש מגוון יותר למציאת מודול החיזוי שלנו, לכן תקטין או תשאיר את ה-training error.

לא ניתן לדעת מה יקרה ל-validation error כתוצאה מהמיפוי הפולינומי. כפי שהסברנו לעיל המיפוי הפולינומי גורם להגדלת סיבוכיות מחלקת המודולים, מה שגורר שתי אפשרויות שונות לשינוי ב-validation error:

1. סיבוכיות מחלקת מודולים גדולה יותר תגרום ל-overfitting, שכן בזמן האימון נלמד את ה-training data יתר על המידה ויהיה לנו חוסר בהכללה. זה יגרום להגדלת השגיאה.
2. סיבוכיות מחלקת מודולים גדולה יותר תאפשר לנו למצוא שתופס את המידע בצורה טובה יותר ולכן נותן פרדיקציה מדויקת יותר. לכן זה יגרם להקטנת השגיאה.

<u>שאלה 23</u>: נראה כי מתקיים $H_{multi} \subset H_{poly}$. יהי $h_{multi} \in H_{multi}$, לכן מתקיים:

$$h_{multi}(x) = \begin{cases} w_1^T x + b_1 & x \text{ is in the O blood type group} \\ w_2^T x + b_2 & x \text{ is in the B blood type group} \\ w_3^T x + b_3 & x \text{ is in the other blood type group} \end{cases}$$

נסמן x_O משתנה מקרי המקבל את הערך 1 במידה ו-x נמצא ב-O blood type אחרת אפס, x_B משתנה מקרי המקבל את הערך 1 במידה ו-x נמצא ב-B blood type אחרת אפס ו- x_{othe} משתנה מקרי המקבל את הערך 1 במידה ו-x נמצא ב-other blood type אחרת אפס. כעת מתקיים:

$$h_{multi}(x) = x_O * (w_1^T x + b_1) + x_B * (w_2^T x + b_2) + x_{other} * (w_3^T x + b_3)$$

נראה כי קיים מיפוי θ לפולינום מדרגה 2 כך שיתקיים כי $h_{multi}(x) = \theta(x)$. נגדיר את המיפוי הבא:

$$x = [x_1 \dots x_n]^T, \quad w = \left[w_{1,1}, w_{1,2}, \dots, w_{1,n}, w_{2,n}, \dots., w_{3,n}, b_1, b_2, b_3\right]^T$$

$$\theta(x) = [x_1 * x_O, \dots, x_O^2, \dots, x_n * x_O, x_1 * x_B, \dots, x_B^2, \dots, x_B * x_n, \dots, x_1 * x_{ot} \quad , \dots, x_{othe}^2 \ , \dots, x_n * x_{other}]^T$$

(הערה – מתקיים כי ערך $w_{i,j}$ זה ערך האיבר j בוקטור w_i.

נגדיר את המודול הפולינומי - $h_{poly(x)} = w^T\theta(x)$. מתקיים כי h_{poly} מדרגה 2 ולכן לפי הגדרה הוא נמצא בקבוצה H_{poly}. כעת נראה שמתקיים $h_{poly(x)} = h_{multi}(x)$:

$$h_{poly(x)} = w^T\theta(x) =$$

$$= \sum_{i=1}^{n} w_{1,i} * \theta(x)_i + \sum_{i=1}^{n} w_{2,i} * \theta(x)_i + \sum_{i=1}^{n} w_{3,i} * \theta(x)_i + x_O * b_1 + x_B * b_2 + x_{other} * b_3 =$$

$$= \sum_{i=1}^{n} w_{1,i} * x_i * x_O + \sum_{i=1}^{n} w_{2,i} * x_i * x_B + \sum_{i=1}^{n} w_{3,i} * x_i * x_{othe} + x_O * b_1 + x_B * b_2 + x_{oth} * b_3 =$$

$$= x_O * \sum_{i=1}^{n} w_{1,i} * x_i + x_B * \sum_{i=1}^{n} w_{2,i} * x_i + x_{other} * \sum_{i=1}^{n} w_{3,i} * x_i + x_O * b_1 + x_B * b_2 + x_{other} * b_3 =$$

$$= x_O * (\sum_{i=1}^{n} w_{1,i} * x_i + b_1) + x_B * \left(\sum_{i=1}^{n} w_{2,i} * x_i + b_2\right) + x_{other} * (\sum_{i=1}^{n} w_{3,i} * x_i + b_3) =$$

$$= x_O * (w_1^T x + b_1) + x_B * (w_2^T x + b_2) + x_{other} * (w_3^T x + b_3) = h_{multi}(x)$$

הוכחנו כי מתקיים $h_{poly(x)} = h_{multi}(x)$ וגם כי $h_{poly} \in H_{poly}$ לכן מתקיים $h_{multi} \in H_{poly}$ כנדרש. מכאן לפי הגדרת הכלה נובע כי $H_{multi} \subset H_{poly}$.

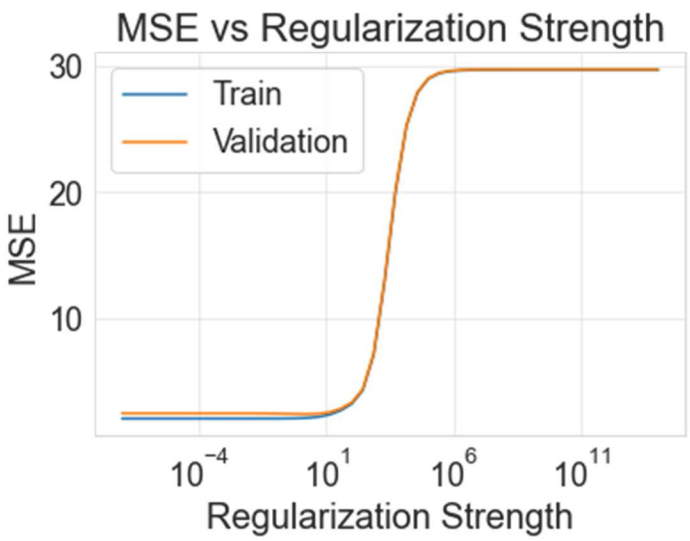
<u>שאלה 24</u>: נסביר על יתרונות וחסרונות בשימוש ב-H_{multi} מול שימוש ב-H_{poly} בתהליך הלמידה. מחלקת המודלים H_{multi} מוכלת בתוך מחלקת המודולים H_{poly} לכן הסיבוכיות שלה יותר נמוכה משל האחרת. נשים לב כי בעוד שסיבוכיות מחלקת המודולים H_{multi} קטנה יותר אנחנו לא בהכרח מאבדים ייצוג של המידע שלנו, כי מחלקת מודולים זו נבנתה מידע קודם שהפקנו כתוצאה מחקירת ה-training data. לכן סביר להניח כי סיבוכיות מחלקת המודולים H_{multi} נמוכה יותר מ-H_{poly} אבל היא לא מאבדת ייצוג של המידע. שימוש במחלקה זו בתהליך הלמידה יכול לגרום לתוצאות טובות יותר על ה-test data, שכן הסבירות שהוא יכליל את החיזוי באופן טוב גבוה יותר משימוש במחלקה השנייה – סיבוכיות מחלקת המודולים קטנה יותר מה שמוריד את הסיכוי ל-overfitting ולא איבדנו מידע לכן אנחנו עדיין לומדים כפי שצריך.

חסרון שיכול לנבוע משימוש ב-H_{multi} ולא ב-H_{poly} הוא כי לא בהכרח חלוקה לפי blood type group היא הטובה ביותר עבור המידע הקיים. יכול הליות כי היה עדיף לעשות חלוקה אחרת,

שהייתה מתבטאת בפולינומים אחרים הנמצאים במחלקת המודולים H_{poly} ולא ב-H_{multi}. לא נוכל ללמוד את אותם הפולינומים כי הם לא יהיו חלק ממחלקת המודולים שלנו ייתכן כי נגדיל את השגיאה בהתאם.

סה"כ: היתרון – הסיבוכיות יותר קטנה ולכן הסיכוי שנגיע למצב של overfitting נמוך יותר.
חסרון על אותו מטבע הוא כי יכול להיות שסיבוכיות זו לא תספיק להבין את מורכבת המבנה ולכן לא תצליח ללמוד מודול טוב, מה שיוביל להגדלת השגיאה.

שאלה 25:




```
Best alpha: 1.93
Best alpha train MSE: 2.085
Best alpha val MSE: 2.383
```

שאלה 26:

Model	Section	Train MSE	Valid MSE
		Cross Validate	
Dummy	2	29.669	29.737
Linear	3	3.449	3.506
Ridge Linear	4	3.392	3.467
Ridge Polynomial	6	2.091	2.384

חלק 7

שאלה 27:

Model	Section	Train MSE	Valid MSE	Test MSE
		Cross Validate		Retrained
Dummy	2	29.669	29.737	28.936
Linear	3	3.449	3.506	2.965
Ridge Linear	4	3.392	3.467	2.958
Ridge Polynomial	6	2.091	2.384	2.083

המודול הטוב ביותר שמצאנו הוא Ridge Polynomial. נסביר את תוצאות הטבלה:

ראשית נשים לב כי ההפרש בין שגיאת ה-MSE עבור כל מודול בין הסטים השונים (Train, Validation, Test) לא גדול – מה שמעיד על כך שלא היה overfitting של המודול (אם היה מתקיים overfitting הייתה ירידה ב-Valid MSE וב-Test MSE ביחס ל-Train MSE).

ניתן לראות כי השגיאה של המודול הראשון – Dummy גבוהה ביותר. מודול זה הוא פשוט ביותר ונותן את ערך התיוג הממוצע לכל אחד מהנקודות במידע. ברור כי מאפיינים שונים משפיעים על ערך מאפיין המטרה לכן הגיוני שהערך הממוצע לא ייתן שגיאה קטנה.

שאר המודולים, מלבד הראשון, מביאים תוצאה טובה – והתוצאות שלהן קרובות באופן יחסי אחת לשנייה. תוצאות מודול Linear Regression ומודול Ridge Linear Regression קרובות מאוד, וזאת מכיוון שההבדל בין המודול הראשון לבין השני הוא השאלה האם משתמשים ברגוליזציה או שלא. מסתבר כי שימוש ברגוליזציה לא משנה הרבה.

המודול השלישי – מודול Ridge Regression עם מיפוי פולינומי, מביא את השגיאה הטובה ביותר. זה מעיד על כך שכנראה אין תלות לינארית מובהקת בין התפלגות מאפיין המטרה לשאר המאפיינים, אלא כנראה התלות יותר מורכבת מלינארית. לכן מעבר למרחב אחר אשר הסיבוכיות שלו גדולה יותר יכול להסביר טוב יותר את מבניות המידע ולכן להביא לשגיאה קטנה יותר.

סה"כ ניתן להבין ששלושת המודולים הראשונים היו ב-underfitting מכיוון שלא הצליחו לתפוס את מבניות המידע בצורה מספקת או שסיבוכיות מחלקת המודולים שלה לא הייתה מספיק גדולה, על מנת למצוא מודול שיתאר את התפלגות מאפיין המטרה כתלות באחרים באופן מיטבי. המודול הרביעי והטוב ביותר הצליח לייצג את המידע בצורה טובה ולכן לא היה ב-underfitting. נוסף על כך, כפי שהסברנו קודם אף מודול לא היה במצב של overfitting.