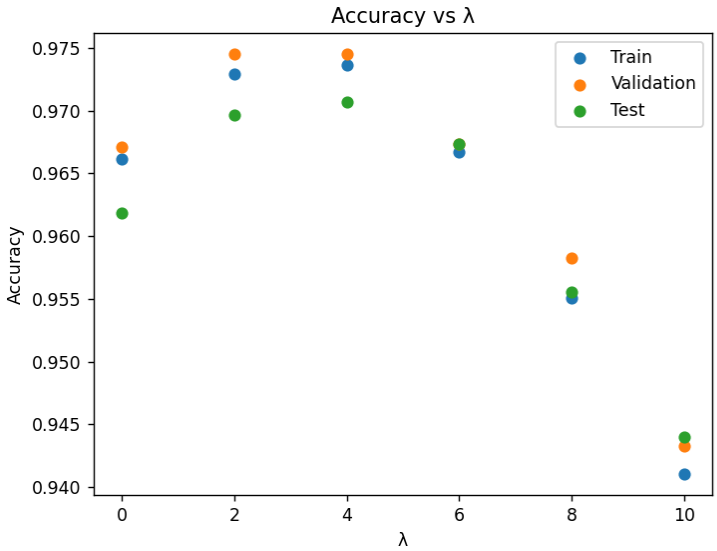
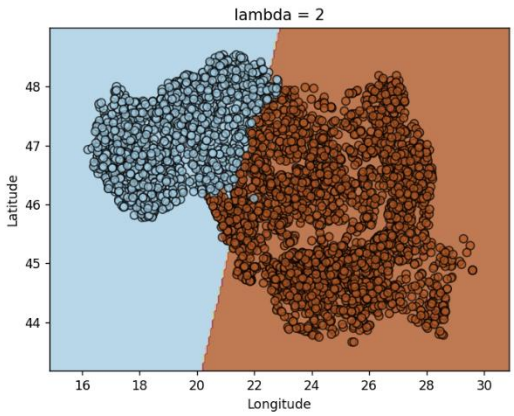
תרגיל 3

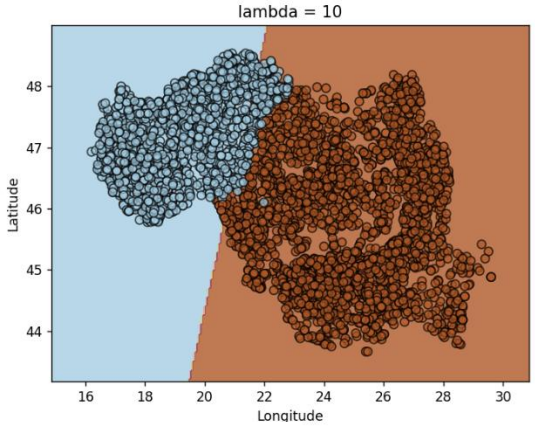
ניק גלבוב תז 212616031

6.1



1. הtest accuracy של המודל הכי טוב על הvalidation הוא 96.9%.
2.





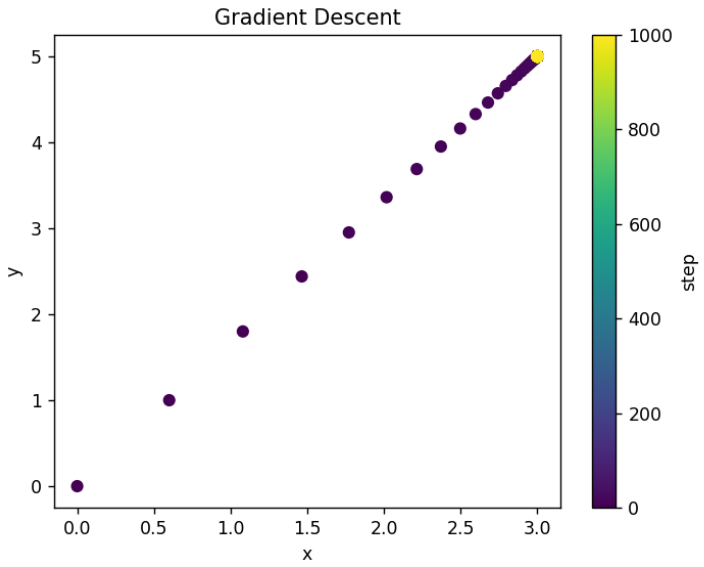
משוואת העלות : $\underset{W}{\operatorname{argmin}}\left[||XW-Y||_2^2+\lambda||W||_2^2\right]$

ניתן לראות משמאל את החלוקה (המודל נבחר עם דיוק הכי גבוה על הvalidation set) על הtest כאשר למבדה שווה 2 , בדיוק של 96.9% על הtest. ומימין את החלוקה (המודל נבחר עם הדיוק הכי נמוך על הvalidation set) על הtest כאשר למבדה שווה 10, בדיוק של 94.3% על הtest. פרמטר למבדה מתאים אחראי על כמה יהיה המיתון של תכונות שפחות משפיעות לנו על חיזוי הדברים. אם באמת יש לנו תכונות כאלו אז הלמבדה הכי טובה (כזאת שמניבה את הדיוק הכי גבוה) תהיה יותר גדולה מ0. מאחר ואנחנו מתאמנים על חלוקה שמאוד מזכירה את המפות למעלה ניתן לראות שפחות יש משמעות לתכונת הקו רוחב (צריך y) ויהיה לי יותר חשוב לדעת איפה הנקודה נמצאת על הקו אורך (ציר x) בשביל לקטלג אותה. ולכן אכן נצפה שהלמבדה תהיה גדולה מ0 כי יש פה תכונה שצריך למתן מבחינת ההשפעה שלה. נסביר את העניין הזה לפי משוואת העלות. אם למבדה שווה 0 אין בכלל מיתון של תכונות פחות משפיעות מה שיכול להוביל לoverfitting. זאת מאחר ואנחנו מתרכזים בשיפור הw רק על החלק $[||XW-Y||_2^2]$ במשוואת העלות . דבר זה יכול דווקא לפגוע בחיזוי על הtest ולהוריד מהדיוק שלו. בגלל זה , כמו גם בתוצאות שלנו , עדיף שיהיה למבדה שגדולה מ0 אך גם לא יותר מדי גדולה. ככל

שהלמבדה גדולה יותר יש יותר סיכוי לunderfitting. וזאת מכיוון שאנחנו עסוקים בלמצוא פרמטרים w שישפרו את החלק $\lambda||W||_2^2]$ במשוואת העלות, ובשביל לעשות זאת צריך לצמצם את הערכים של w . וכאשר הלמבדה שלנו גבוהה מאוד אז כל תיקון בחלק $||XW-Y||_2^2]$ במשוואת העלות יכול ממש לגרוע לחלק $\lambda||W||_2^2]$, למשל אם התיקון הוא לעלות ערך w מסוים, ואז התיקון הזה לא יהיה יעיל וכנראה לא יתממש. נוכל להסיק מזה שהתייחסות שלנו למציאת w מתאימים לסט האימון שלנו קטנה ככל שהלמבדה גדלה ובכך נוצר underfitting. התוצאות שלנו תואמות את אותם דברים, הלמבדות הכי טובות הן 2 ו4 (השיגו את הדיוק הכי גבוה בכל set) ניתן לראות זאת בגרף בשאלה הקודמת.

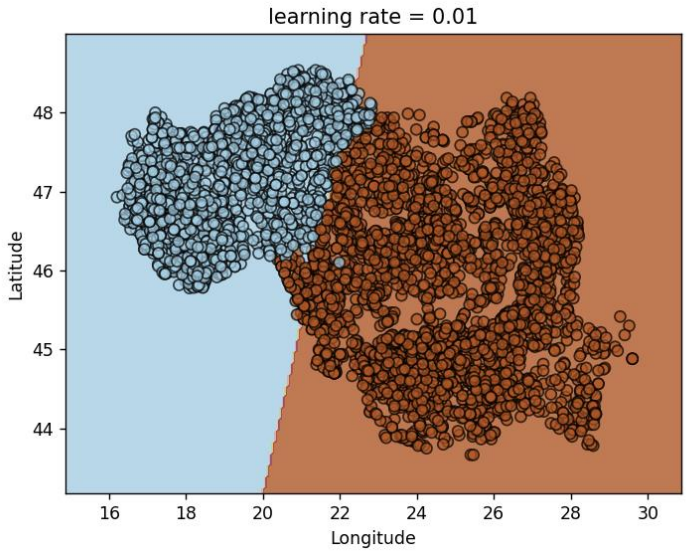
.7

.1

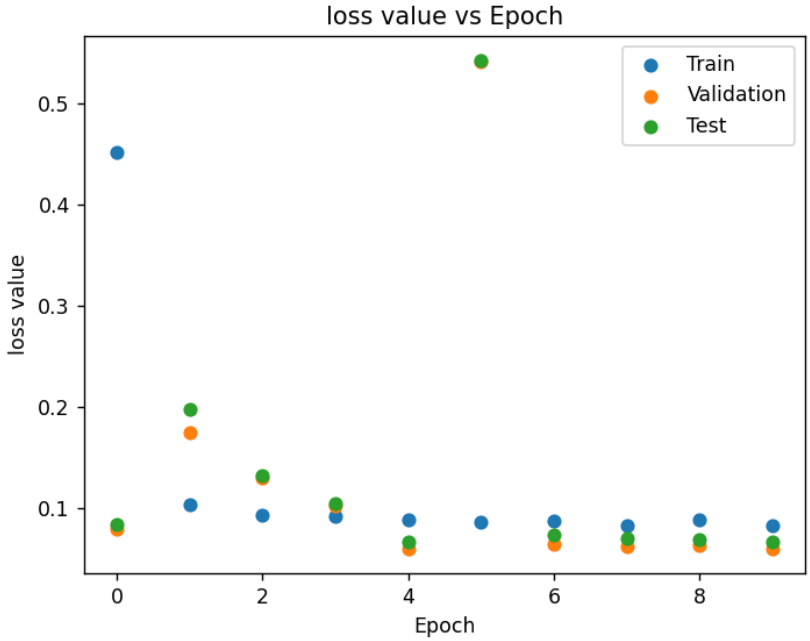


הנקודות שאליהם הגענו זה 2.999 בx ו4.999 בy . ערכים אלו כמעט מניבים לנו 0 על הפונקציה הנתונה בתרגיל. ו0 הוא הערך האפשרי הכי נמוך לפונקציה, מאחר ואנחנו מחברים שני ערכים בריבוע.

.9

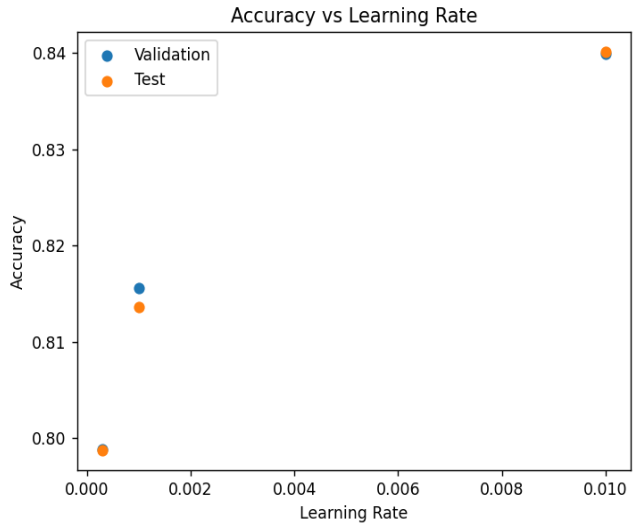


1. ניתן לראות במפה למעלה כיצד המודל הכי טוב (עם הvalidation accuracy הכי גבוה, עם 97.5%) ניבא על הtest set (עם דיוק של 97%) . מודל זה משתמש ב logistic regression ובעל learning rate של 0.01.

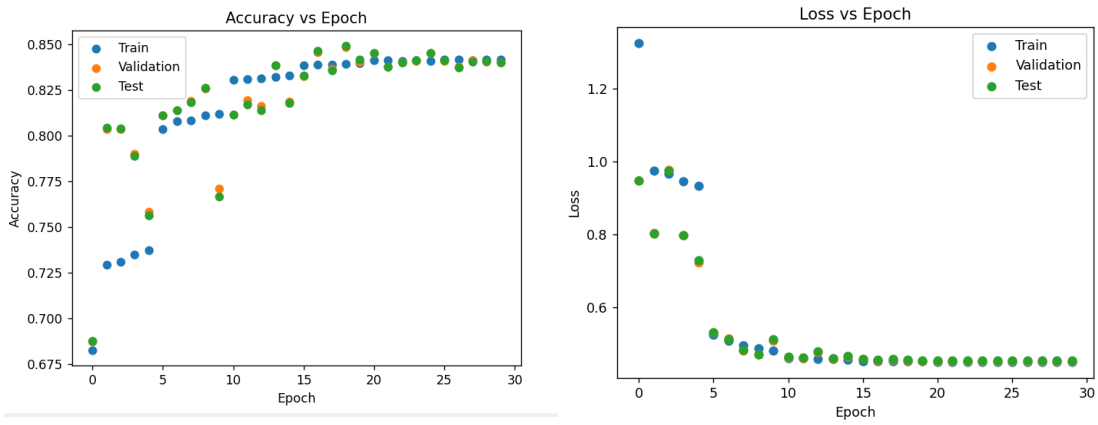


2. בגרף למעלה ניתן לראות את הloss values של המודל הכי טוב (זה שנבחר בשאלה הקודמת) לאורך הepoch על כל 3 הסטים . כלומר, כאשר אנו מאמנים את המודל הכי טוב שלנו על הtrain set לכל batch בגודל 32, אנו נקבל loss value מהloss function. הערך הזה בעצם מתקבל ממוצע הערכים בנוסחה $\ell(w,x,y) = -\log(Soft(y_k|z))$. בסופו של דבר אחרי שהתאמנו על כל הtrain set וסכמנו את ערכי הloss עבור כל batch נעשה ממוצע שלהם וזה יהיה הloss value הסופי לאותו epoch. בגרף ניתן לראות זאת כנקודות הכחולות. לאחר מכן, בסוף כל epoch ואחרי שסיימנו את האימון של המודל ניקח את הtest set ואת הvalidation set (פרמטרי ה-W הם של המודל המאומן וערכי ה-X וה-Y משתנים בהתאם לset) ולכל אחד נחשב את הloss value , ששוב מתקבל ממוצע הערכים בנוסחה $\ell(w,x,y) = -\log(Soft(y_k|z))$ ואלו יהיו ערכי הloss עבור הtest והvalidation עבור אותו epoch, הנקודות הירוקות והכתומות בהתאמה. כעת ננתח את הנראה בגרף והאם באמת המודל שהתאמן על הtrain set היה טוב לדוגמאות חדשות. ראשית, ניתן לראות עבור epoch 0 שהנקודה הכחולה גבוהה. וזאת מכיוון שבהתחלה המודל לא מאומן כל כך טוב ועל כל batch בגודל 32 נקבל loss יחסית גבוה, אך בסוף הepoch הזה המודל כבר יחסית מאומן ועל כך נקבל loss נמוך לtest ולvalidation. לאחר מכן אפשר לראות שהנקודות הכחולות מתייצבות לאורך קו מסוים (באזור 0.1) וכל פעם יש מן עלייה וירידה בין כל נקודה. סיבה אפשרית לכך היא שlearning rate גבוה מדי וכל פעם אנו מפספסים את המינימום בפונקציית העלות. לעומת זאת, אפשר לראות שערכי הloss עבור דוגמאות חדשות (ה train set והvalidation set) יורדות בהדרגה עד שבשלב מסוים אפילו יורדת במעט מהערכי הloss של הtrain set. מכך נסיק שהמודל שלנו מתוך הtrain היה מוצלח גם על דוגמאות חדשות ולא רק אלו שהתאמן עליו.
3. גם מבחינת המפות וגם מבחינת נתוני הדיוק (97% על המודל בשאלה הקודמת מול 96.9% בשאלה הראשונה) אפשר להסיק ששתי השיטות טובות כמעט באותה מידה. המודל בשאלה הקודמת מאוד מותאם לבעיות של קטלוג לעומת המודל בשאלה הראשונה שמותאם לבעיות כלליות יותר של חיזוי . ומאחר שהבעיה שאנו מתמודדים איתה פה יותר מותאמת לקטלוג (מדינה מסוימת היא או 0 או 1) הגיוני שהמודל בשאלה הקודמת עדיף במעט.

9.4



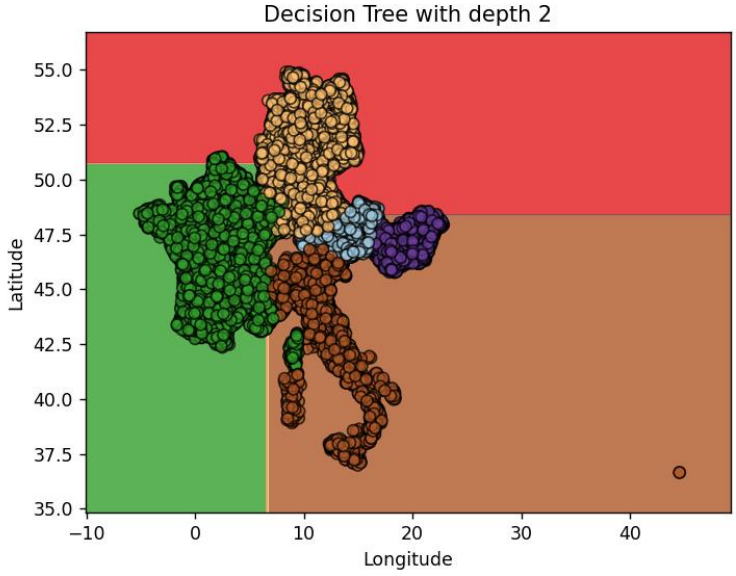
1. הדיוק על הtest set של המודל הכי טוב שנבחר לפי הvalidation set הוא 84% . ולו learning rate של 0.01.
2.



הגרפים מוצגים על המודל שנבחר בשאלה הקודמת. הגרף הימני מראה את הloss value לכל סט של נתונים. זה עובד בדיוק באותה צורה כמו שתואר בהרחבה בשאלה הקודמת, רק שהפעם יש לנו ערכי learning rate קטנים יותר ובנוסף יש הכפלה של הlearning rate ב0.3 כל 5 epochs, זה בעצם אומר שאנחנו מצמצמים אותו כל פעם. ובאמת ניתן לראות שערכי הloss value על הדוגמאות החדשות, שהן הtest set והvalidation set, קטנות בכל epoch. גם בגרף השמאלי ניתן לראות אותה מסקנה רק שהפעם אנו מדברים על הדיוק של כל סט נתונים לכל epoch. אפשר לראות עלייה על כל סט של נתונים (מגיע כמעט ל85% לכל אחד מהם). דבר זה הגיוני כי שני הגרפים בסופו של דבר שקולים אחד לשני, ככל שיש loss value נמוך יותר זה אומר שאנחנו יותר מתקרבים לאמת ועל כך הדיוק אמור לעלות. מה שאומר שהמודל שלנו , שמאומן לקטלג ל5 מדינות שונות, עושה עבודה יחסית טובה על דוגמאות חדשות (validation וtest) ומצליח , בנוסף לloss value, לעלות את אחוזי

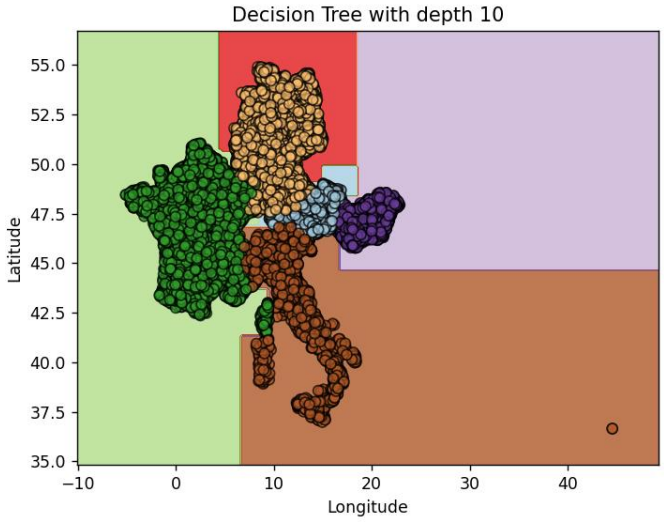
הדיוק שלהם עד לכמעט 85%. אך אפשר לשים לב שמודל שמקטלג ל2 מדינות קיבלנו אחוזי דיוק גבוהים יותר (הכי טוב יצא 97% על הtest) וערכי loss נמוכים יותר. לזה כמה סיבות אפשריות. אחת מהם היא שהגדלנו את כמות המדינות שאנחנו יכולים לקטלג אליהם, דבר זה מעלה את הסיכוי שנקטלג למדינה לא נכונה, ועל כך דווקא נרצה להוסיף עוד דוגמאות לtraining set בשביל לשפר את הקטלוג שלנו. בנוסף, בגלל ההכפלה של הlearning rate ב0.3 לכל 5 epochs (יש 30 epochs) הקצב שלו קטן באופן משמעותי ביחס למודל שמקטלג ל2 מדינות , ובכך אנחנו מאטים את הצעד שאחראי להתקדם לכיוון של המינימום בפונקציית העלות . מזה אפשרי שנקבל דיוק יותר נמוך וloss יותר גבוה. המסקנה שלנו היא שהמודל הצליח לנבא בצורה יחסית טובה על הדוגמאות החדשות אבל לא כמו במודל שמקטלג ל2 מדינות.

.3



הדיוק של מודל העץ עם רמה מרבית של 2 הוא 75% על הtest set. לפי אחוזי הדיוק ניתן לראות שהמודל של העץ עם רמה מרבית של 2 פחות טוב מהמודל בשאלה הקודמת . נראה שמי שמותאם יותר למשימה הוא המודל שמקטלג ל5 מדינות. אך זה לא נכון בוודאות. וזאת מכיוון שאנו לא באמת יודעים מה קורה אם היו יותר רמות בעץ. דווקא היינו מצפים שמודל העץ יהיה מתאים יותר כי מדובר בשטחים של מדינות. יהיה מאוד נוח לחלק את אותם שטחים לצורות מלבניות בשביל לקבל מידור .

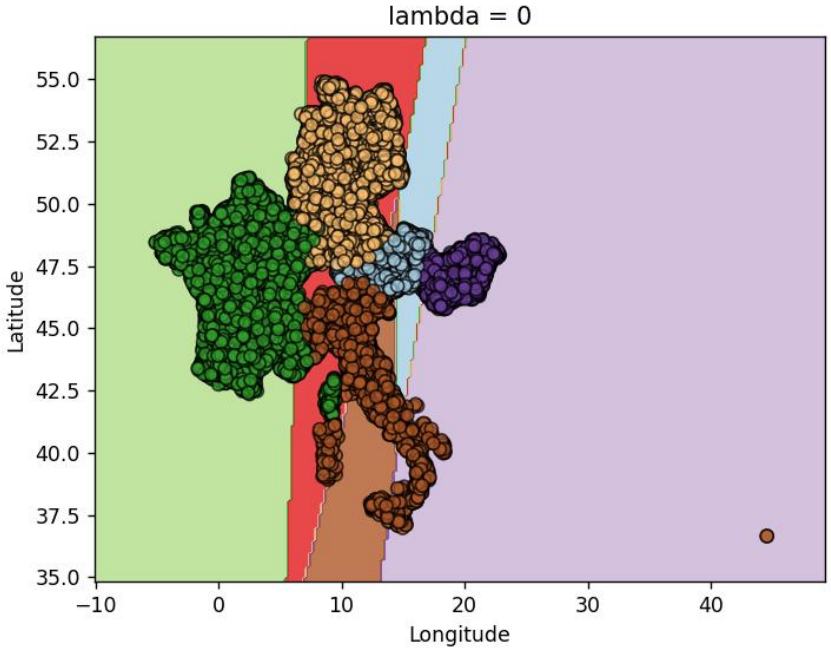
.4



הדיוק של מודל העץ עם רמה מרבית של 10 הוא 99.6% על הtest set. לפי אחוזי הדיוק ניתן לראות שהמודל של העץ עם רמה מרבית 10 יותר טוב מהמודל בשאלה הראשונה. נראה שמי שמותאם

יותר למשימה הוא מודל העץ, וזאת לא מה שהיה ניתן לראות בשאלה הקודמת. הסיבה לכך היא בדיוק כמו הצפייה שלנו מהשאלה הקודמת. העלנו את כמות הרמות וקיבלנו מידור מלבני איכותי יותר שמתאים ספציפית לבעיה הנוכחית שלנו, שזה להפריד בין שטחי המדינות.

בונוס:



קיבלנו שהלמבדה הכי טובה היא 0 עם דיוק של 84.7% על הtest. כלומר המודל פה והמודל בשאלה 2 הם בדיוק אותו מודל. הם אותו מודל כי כאשר למבדה שווה 0 אין בכלל הוספה של חלק אשר מבצע L2 regularization (כמו המודל בשאלה 2) וגם בגלל שאנחנו מתחילים מ0.01 learning rate (כמו המודל בשאלה 2). נסיק מכך שאין פה תכונות שצריך למתן את ההשפעה שלהם על התוצאה בשביל לקבל את הדיוק הכי גבוה. וזה מאוד הגיוני מאחר והמודל מתאמן על חלוקה מאוד דומה לחלוקה שאנחנו רואים במפה למעלה (למעלה רואים את התוצאות על הtest set) ובחלוקה הזו גם תכונת קו הרוחב וגם תכונת קו האורך חשובות באותה מידה בשביל להצליח לנבא לאיזה מדינה שייכת הנקודה (יהיה קשה רק עם תכונה אחת להצליח לנבא באיזה מדינה אנחנו). לעומת זאת בשאלה 2 ב6.1 ראינו שפחות חשוב לנו לדעת איפה הנקודה מבחינת קו הרוחב אלא יותר מבחינת קו האורך שלה בשביל לנבא אם היא במדינה 0 או 1.