**למידת מכונה – פרויקט מסכם**

**קבוצה 2: דורון שפירא** 316208032  **בן מלי** 204148225  **אריאל רגב** 321776627

**תקציר מנהלים:**

הקוד מחולק כך שיהיה קל לשנות ולהגדיר קונפיגורציות שונות על מנת לנתח את סט הנתונים בצורות מגוונות, בתקווה כי הדבר יניב תוצאות טובות יותר במטריקות הנבדקות. מפאת מחסור במשאבי

**חלק ראשון – אקספלורציה:**

בשלב זה ביצענו חיתוכים וניתוחים ויזואליים של הדאטה, וזאת על מנת לאפיין את הנתונים ולהבין כיצד להתמודד איתם.

ראשית, טענו את סט הנתונים לאובייקט מסוג data frame של pandas, ונוכחנו לדעת כי יש בידינו 21 פיצ'רים ו-21,931 דגימות. בנוסף, נבחין כי קיימים לנו שני תיוגים אפשריים לכל דגימה – 0 או 1. לאחר מכן, אנו בודקים כמה נתונים חסרים יש לנו בסט הנתונים, ובנוסף מחשבים עבור כל פיצ'ר מהו אחוז הדגימות החסרות עבורו. נבחין כי בפיצ'רים 14 ו-19 יש אחוז ניכר של ערכים חסרים. נרצה לאפיין את הפיצ'רים הקיימים ולסווגם בתור נומרים, בינאריים או קטגוריאליים, וזאת מאחר וכל סוג פיצ'ר יצריך ביצוע אקספלורציה מעט שונה מהאחרים. ראשית, בדקנו כמה פיצ'רים יש בידינו מכל סוג, וקיבלנו כי ישנם X פיצ'רים נומריים, X פיצ'רים בינאריים, ו-X קטגוריאליים. ראינו לנכון להציג בתצוגה מקדימה ודינאמית את ההיסטוגרמות של כלל הפיצ'רים, טרם החלוקה וההתייחסות הפרטנית לכל סוג, וזאת בכדי להפיק תובנות מהירות וחשובות שיכווינו אותנו בהמשך הניתוח. את התצוגה הדינאמית ביצענו בעזרת כלי ה-Tableau[[1]](#footnote-1).

בתום שלב חלוקת הפיצ'רים לסוגם, התחלנו את הניתוח מהפיצ'רים הנומריים. בשלב הראשון, הצגנו עבור כל פיצ'ר נומרי שכזה טבלה סטטיסטית המכילה נתונים של ממוצע, סטיית תקן, מינימום, מקסימום ואחוזוני 25%/50%/75%. מהתבוננות בסטיות התקן של הפיצ'רים, ניתן להבחין בכך שקיים הבדל משמעותי בהן בפיצ'רים השונים, ויתרה מכך טווח הערכים גדול למדי, מה שמעיד על חשיבות העיבוד המקדים שעתיד לבוא בשלב הבא, בפרט השלבים של הטיפול בחריגים, הטיפול בחסרים וכמו כן הנרמול. לאחר מכן, עבור כל פיצ'ר נומרי, בנוסף להצגה הכללית של ההיסטוגרמות שנעשתה באמצעות פלטפורמת ה-Tableau, הצגנו היסטוגרמות ספציפיות באמצעות שימוש ב-pandas, כל שכן הצגת גרף boxplot עבור כל פיצ'ר, וזאת בכדי ללמד האם השונות אחידה על פני הרמות השונות של הפיצ'ר. למדנו כי ברוב הפיצ'רים השונות כן מתפלגת בצורה אחידה, אך יחד עם זאת בחלק לא מבוטל מהם קיימת שונות גבוהה למדי בין הרמות השונות של אותו פיצ'ר (למשל פיצ'רים 0,4,10,14).

השלב הבא בניתוח הפיצ'רים הנומריים היה הצגת מטריצת הקורלציה, הן באמצעות ה-pandas והן באמצעות כלי ה-Tableau. מניתוח מטריצת הקורלציה הסקנו כי פיצ'רים 7,9 ו-5,17 קורלטיביים אחד עם השני, ולכן ייתכן כי נוכל בהמשך לצמצם בעזרת נתון זה את מספר המימדים.

בתום ניתוח הפיצ'רים הנומריים, עברנו לנתח את אלה הקטגוריאליים והבינאריים באמצעות היסטוגרמות לכל פיצ'ר. מהניתוח הזה הבחנו כי בחלק מהפיצ'רים הקטגוריאליים ישנן דגימות רבות שהן ריקות ואינן משויכות לאף קטגוריה, ולכן נצטרך לתת על כך את הדעת בשלב הבא של העיבוד המקדים (למשל פיצ'רים 1,6,19), בין אם להורידם או בין אם להשלימם בדרך כזו או אחרת.

**חלק שני – עיבוד מקדים:**

ראשית, נרצה להתייחס לסוגיית **הנתונים החריגים (Outliers)**. כפי שעולה מהאקספלורציה שבוצעה בחלק הראשון, ניתן להבחין כי אכן קיימים לנו נתונים חריגים בסט הנתונים, כאלו אשר עלולים ליצור הטיות משמעותיות במדדים הנבדקים, ועל כן נרצה לקבל החלטה כיצד לנהוג עמם. מאחר ובידינו קיים מספר רב למדי של תצפיות, נוכל לנהוג בגישת הסרה אגרסיבית של תצפיות חריגות עבור הפיצ'רים הנומריים. המתודולוגיה לפיה תתבצע ההסרה תתבסס על אחת מהשתיים המקובלות – שימוש ב-IQR או שימוש בסטיית תקן. בהתאם למה שלמדנו בתרגול, השימוש בשיטת IQR, אשר משמיטה תצפיות הגדולות מ-2.7 סטיות תקן (הן לצד החיובי והן לצד השלילי), גרם לכך שהרבה תצפיות נפלו מחוץ לתחום הבטוח ולכן נפסלו, מה שהוביל אותנו להעדיף להשתמש בשיטת סטיית התקן, זו שמרחיבה במעט את תחום הקטימה ומשאירה למעשה יותר תצפיות (מרווח של 3 סטיות תקן). בנוסף, הבחנו כי בפיצ'ר 19 יש כ-80% ערכים ריקים, ומאחר ופיצ'ר זה קטגוריאלי, לא ניתן היה להשלימו בצורה נכונה, ולכן החלטנו להוריד אותו במלואו.

כעת, נרצה לתת את הדעת אודות נוגעת **הפיצ'רים הקטגוריאליים** בסט הנתונים שלנו. לטובת השלבים הבאים בניתוח, אנו נדרשים להחליט אודות אחד משני תהליכי המרה אפשריים עבור פיצ'רים אלו – One Hot Encoding או Label Encoding. שימוש בשיטת One Hot Encoding טומן בחובו יצירת מימדים רבים אך נטולי הטייה, ואילו עבור Label Encoding בדיוק ההיפך – מימד בעיה נמוך יותר אך מוטה. אנו בחרנו להשתמש בשיטת One Hot Encoding כדי להימנע מהטעיות כאלה ואחרות לסט הנתונים שלנו.

יישמנו את שיטת One Hot Encoding בשני אופנים שונים. לקחנו את סט הנתונים שהתקבל לאחר הסרת החריגים, ופיצלנו לשתי אפשרויות. הראשונה הייתה הפעלת One Hot Encoding על סט הנתונים כפי שהתקבל – as is. השנייה הייתה לבצע Feature Selection ולאחר מכן ליישם את ה- One Hot Encoding. בשלב ה- Feature Selectionהבחנו במספר פיצ'רים אשר היו זקוקים ל"תיקון". בפיצ'ר 2 כל דגימה הייתה נומרית אך הכילה את האות "d" בסופה, ואילו בפיצ'ר 18 גם כן היו דגימות נומריות אך עם האות "a" בתחילתן. בעבור שני הפיצ'רים הללו, ביצענו תיקון והורדנו את האותיות המיותרות, לקבלת דגימות נומריות טהורות בלבד. כמו כן, בפיצ'ר 12 היו שני ערכים קטגוריאליים של "y" ו-"n", אשר בחרנו להמיר אותם ל-1 ול-0, בהתאמה. בנוסף, החלטנו להוריד את פיצ'ר 17 אשר היה קורלקטיבי עם פיצ'ר 5 (קורלציה של 90%). פיצ'רים נוספים אשר הורדנו היו 15 ו-0, וזאת מאחר והכילו כמות גבוהה של אפסים, כלומר לא תרמו מידע נוסף. לאחר סינון הפיצ'רים הללו ביצענו כאמור את ה-One Hot Encoding וקיבלנו את סט הנתונים של האפשרות השנייה.

בתום הסרת החריגים נותרו בידינו 19,855 דגימות לעומת 21,931 שהיו בהתחלה.

כעת נעבור להתייחס **לטיפול בחוסרים**. בנתונים החסרים בסט הנתונים החלטנו לטפל באמצעות ב-4 דרכים שונים, כלומר 4 Imputers שונים. הראשון שבהם, Simple Imputer, התבסס על השלמת הערכים החסרים באמצעות שימוש בחציון של כל פיצ'ר. השני, Distribution Imputer, התבסס על השלמת ערכים לפי הגרלת ערך רנדומי מההתפלגות המשוערכת של הפיצ'ר. כלומר, עבור כל פיצ'ר בוצע ניסיון השוואה להתפלגויות מוכרות שמזכירות בצורתן את זו של הפיצ'ר. ע"י שימוש במדד SSE, בחרנו את ההתפלגות המוכרת שמתאימה בצורה הטובה ביותר לנתוני הפיצ'ר, כלומר בעלת ערך ה-SSE הקטן ביותר (ניתן להשתמש ב-SSE מאחר וכמות הדגימות עבור כל התפלגות נבחנת הייתה זהה). ההתפלגויות המוכרות שנבחנו הינן: נורמלית, F, חי בריבוע, אקספוננציאלית. הדרך השלישית שעשינו בה שימוש הייתה Choice Imputer. בגישה זו בחרנו לבצע את ההשלמות עבור כל פיצ'ר ע"י הגרלת דגימה מסט הנתונים של אותו פיצ'ר, כאשר כל דגימה הינה בעלת הסתברות זהה, וזאת מאחר ובידינו יש מעל 21 אלף דגימות, והנחנו כי דבר זה מייצג כהלכה את כלל הדגימות האפשריות עבור אותו פיצ'ר. הגישה הרביעית והאחרונה הייתה Iterative Imputer. בגישה זו ההשלמה נעשית באמצעות חיזוי שנעשה ע"י סוגי רגרסיות שונות, וזאת על מנת למצוא את הערך המסתבר ביותר מהרגרסיות השונות, ולבסוף איתו תתבצע ההשלמה[[2]](#footnote-2).

השלב הבא בתהליך העיבוד המקדים הינו **נרמול הנתונים**. לאחר בדיקת הנתונים ניתן היה לראות כי הנתונים אינם מנורמלים, ולכן נרצה לנרמל אותם כדי שלכל הפיצ'רים יהיה אותו טווח ערכים, כלומר שכל הפיצ'רים "ידברו" באותה שפה. לטובת הנרמול השתמשנו בשיטת MinMax, וזאת משום שרצינו שכל הדגימות בכל הפיצ'רים יהיו בטווח שבין 0 ל-1, מה שיעלה בקנה אחד עם הדגימות הבינאריות של הפיצ'רים הקטגוריאליים, שבלאו הכי נמצאים בטווח זה (לאחר ביצוע One Hot Encoding).

השלב האחרון בעיבוד המקדים היה נושא **הקטנת המימדיות**. טרם ביצוע השלב הזה אנו עדיין נותרים עם כמות גבוהה של מימדים, 512 בסטי הנתונים ללא ה-Feature Selection, ו-30 בסטים שעברו Feature Selection. נבחין כי 512 המימדים עתידים ליצור עומס ומורכבות גבוהה מאוד שתצריך כוח עיבוד משמעותי בשלבים הבאים של יישום המודלים. בהמשך ישיר למה שלמדנו בהרצאה לגבי כלל האצבע ש-n מימדים מצריכים לפחות n^2 תצפיות, במקרה שלנו ישנן כאמור כ-20,000 תצפיות בלבד עבור מעל ל-500 מימדים, ולכן אנו ניעזר בשיטת PCA לטובת הורדת המימדיות לסטי הנתונים הללו. בנוסף, למרות שבסטי הנתונים האחרים ישנם 30 מימדים, אנו בחרנו להחיל את שיטת ה-PCA גם עליהם, וזאת מאחר ואם יתאפשר להוריד את המימדיות ולשמור על הסבר מספק של השונות, אז בהחלט זה יפחית, גם אם במעט, את הסיבוכיות החישובית של המודלים. בכל ניתוח PCA שכזה, אנו נבחן 3 אפשרויות של אחוז השונות המוסברת - בראשונה נדרוש כמות פיצ'רים שתסביר 90% מהשונות, בשנייה 95% ובשלישית 98%. עבור כל אפשרות כזו בחרנו להריץ, כמעין מודל מייצג, את מודל ה-ANN, ובחרנו לקחת את אפשרות ה-PCA בעלת ערך ה-AUC הגבוה ביותר. נציין כי כל אפשרות אנו בדקנו הן על סט הנתונים לאחר ה- One Hot Encoding ללא התיקונים, ועל הסט השני שכן כלל תיקונים.

נבחין כי בשלב זה יש בידינו 24 סטי נתונים (3\*4\*2). על מנת לבחור את סט האימון שעליו נריץ בהמשך את המודלים, נדרש מאיתנו לבצע הערכה כלשהי ולהכריע בסוגייה זו. לטובת ההערכה, החלטנו לבצע פרידיקציה באמצעות ANN, בחירה שרירותית. עבור כל סט אימון פוטנציאלי בוצעו 5 פרידיקציות בשיטת K-Fold (עם 5 פולדים), וכמו כן חושב מדד ה-AUC הממוצע עבור כל סט. בסופו של דבר, נבחר סט האימון בעל ערך ה-AUC הממוצע הגבוה ביותר. סט האימון הטוב ביותר היה תוצר של השלבים הבאים: One Hot Encoding ללא Feature Selection, שימוש ב- Iterative Imputer ו-PCA של 95%. נשים לב כי אמנם בחרנו במודל שכלל PCA ברמת שונות מוסברת של 95% על פני ה-98%, וזאת על אף שערך ה-AUC היה נמוך במעט מזה של השני, מאחר ורצינו ליצור מודל בעל רמת הכללה גבוהה יותר שתתאים בצורה טובה לסט נתונים חדש שהמודל לא ראה לפני כן, מה שלא היה נעשה בצורה טובה תחת PCA של 98%.

לאחר שמצאנו את סט האימון הטוב ביותר, ביצענו את אותם שלבי העיבוד המקדים גם על סט המבחן. נבחין כי בשלב ה- One Hot Encoding, הרמות השונות של הפיצ'רים הופכות לפיצ'רים בפני עצמן, ולכן במידה ובסט המבחן יש פיצ'רים קטגוריאליים עם רמות שלא הופיעו בסט האימון, הן יהפכו לעמודות חדשות ויושמטו ע"י שימוש בפונקציית transform של אובייקט ה- One Hot Encoding. בנוסף, במידה וקיבלנו בשלב הנרמול עבור הפיצ'רים הנומריים ערכים הנמוכים או הגבוהים מ-0 או מ-1 בהתאמה, הם יקבלו את ערך הסף המתאים עבורם, כלומר ערך 0 עבור אחרים קטנים מערך הסף 0, וערך 1 עבור ערכים הגדולים מ-1.

**חלק שלישי – הרצת המודלים:**

אנו בחרנו את המודלים הבאים:

* KNN
* Logistic Regression
* Multi-Layer Perceptron (ANN)
* Support Vectors Machine (SVM)

עבור כל מודל הרצנו GridSearch עם סט האימון שנבחר בתום שלב העיבוד המקדים, ובאמצעות זאת מצאנו את ההיפר-פרמטרים הטובים ביותר מכל אלו שנבדקו, כלומר אלו שהפיקו ערך AUC הגבוה ביותר.

הערה: ה-GridSearch שבוצע נמצא בקובץ פייתון נפרד, וזאת בכדי למנוע ריצה נוספת שלו שלא צורך, שכן כבר מצאנו את אותם ההיפר-פרמטרים.

KNN:

הפרמטר היחיד של מודל זה הינו ערך ה-K, כלומר מספר השכנים. אנו ביצענו 5 בדיקות עם הערכים הבאים: K=3, 15, 25, 75, 100. מכלל הבדיקות בחרנו את 100K= מאחר וערך ה-AUC שלו היה הגבוה ביותר.

Logistic Regression:

מאחר וה-label שלנו הוא בינארי, ניתן להשתמש במודל הרגרסיה הלוגיסטית אשר מבצעת חיזוי לערכים בינאריים. ההיפר-פרמטרים של המודל הזה מייצגים למעשה את מידת ההשפעה של כל פיצ'ר על ה-label, כלומר כמה כל פיצ'ר מסביר את המשתנה המוסבר. ההיפר-פרמטרים נבחנו ע"י שילוב קומבינציות שונות של פונקציית ה- Penalty הריבועית (L2), מס' איטרציות מקסימלי וכן ערכי C שונים. בסופו של דבר בעזרת פונקציית ה-GridSearch נבחרו ההיפר-פרמטרים הבאים:, max\_iter=150, Penalty=L2, C=0.1. שאר הפרמטרים[[3]](#footnote-3) של מודל הרגרסיה הלוגיסטית נשארו ללא שינוי.

Multi-Layer Perceptron (ANN):

זהו מודל שבו מתבצעת למידה של הרשת והיא תלוי במספר רחב של היפר-פרמטרים. אנו ביצענו את ההרצה תוך בחינת ההיפר-פרמטרים הבאים: פונקציית האקטיבציה, כמות וגודל השכבות החבויות (hidden layers), שיעור הלמידה (learning rate), ה-solver (Stochastic Gradient Descent(SGD), ADAM או L-BFGS) ומספר החזרות המקסימלי (max iter). בתום ה-GridSearch נבחרו הערכים הבאים להיפר-פרמטרים:

Support Vectors Machine (SVM):

עבור מודל זה קיימים היפר-פרמטרים רבים[[4]](#footnote-4), כאשר אנו בחנו את שלושת ההיפר-פרמטרים המרכזיים: פרמטר C – ערך הרגולריזציה, פונקציית הקרנל, הדרגה (רלוונטית לקרנל פולינומיאלי). לאחר ביצוע ה-GridSearch נבחרו הערכים הבאים להיפר-פרמטרים:

C=0.01, kernel=rbf, degree=n/a. שאר הפרמטרים נותרו הדיפולטיבים של מודל זה.

**חלק רביעי – הערכת המודלים:**

בניית Confusion Matrix:

אנו בנינו את ה- Confusion Matrix עבור מודל ה-ANN שנבחר מדגמית. בהקשר למודל זה, תא 0 במטריצה שמופיעה בנספח ? מסמן תיוג 0, ואילו תא 1 מייצג תיוג 1. הציר האנכי מסמן את התיוגים האמיתיים ואילו הציר האופקי את התחזיות.

הערכת המודל באמצעות K-Fold Cross Validation:

עבור כל מודל עם סט האימון שנבחר בתום שלב העיבוד המקדים, בתוספת ערכי ההיפר-פרמטרים הטובים ביותר שנמצאו עבורו, ביצענו הערכה באמצעות שימוש ב-K-Fold ובניית פלט ה-ROC המתקבל. ב-K-Fold עשינו שימוש ב-5 פולדים שונים ושמרנו על ההגדרות הדיפולטיביות של חלוקת 80% מהדגימות לסט האימון ו-20% לסט הוולידאציה. עבור כל מודל התקבל כאמור גרף ROC וערך AUC, ובנוסף מופיע ערך ה-AUC הממוצע של אותו מודל על גבי 5 הפולדים. מדד ה-AUC הממוצע הטוב ביותר התקבל עבור מודל ה-ANN וערכו היה 90%.

Overfitting:

לטובת בחינת סוגיית ה- Overfitting נעזרנו בשני מדדים. ראשית, עבור כל מודל התמקדנו במדד העיקרי של הפרויקט, הלוא הוא מדד ה-AUC. שמנו לב שהפער המרבי בערכי ה-AUC נמוך מ- 1.2%. בחרנו להגדיר מודל שהוא Overfitted תוך שימוש במדד ה-AUC, אך ורק אם ההפרשים בין ה-AUC Scores של סט האימון וסט הוולידציה גבוהים מ-2.5%. אי לכך, ניתן לומר כי כל המודלים שלנו אינם Overfitted, היות והפער המקסימלי היה כאמור 1.2%. יתרה מכך, רצינו לוודא באמצעות מדד נוסף כי המודלים אינם Overfitted, ולכן בחרנו להיעזר בפונקציית ה-Log Loss. עבור כל מודל הרצנו את פונקציית ה- Log Loss וקיבלנו ערך מספרי. הפער המקסימלי שהתקבל בעבור פונקציות ה- Log Loss בהרצות בין סט האימון לסט הוולידציה, בכל המודלים, עמד על X. אנו בחרנו את X להיות ערך הסף עבור הקביעה שמדובר ב-Overfitting, ולכן נוכל להסיק גם ממדד זה כי המודלים אינם Overfitted.

סיכום:

בפרויקט זה עסקנו בבעיית קלסיפיקציה, כלומר חיוזי תיוג בינארי על סמך סט דגימות נתון של 21 פיצ'רים. לאורך שלביו השונים של הפרויקט יישמנו את אשר למדנו לאורך הקורס, לרבות היישום בפועל של מודלים ראשוניים ומתקדמים. בתחילתו של הפרויקט ביצענו אקספלורציה מעמיקה על סט הנתונים, אשר ממנה הופקו מסקנות מכריעות אודות הצעדים שאנו היינו צריכים לבצע על סט הנתונים על מנת להכינו ולהכשירו לטובת הרצת המודלים. חילקנו את הפיצ'רים הקיימים של סט הנתונים לפיצ'רים נומריים, קטגוריאליים ובינאריים, וביצענו אקספלורציה ייעודית בנפרד על כל סוג פיצ'רים, לרבות שימוש בכלי ה- Tableau. לאחר מכן, התחלנו בשלב העיבוד המקדים. שלב זה כלל בתוכו מספר תתי שלבים. ראשית, השתמשנו לטובת הנתונים החריגים בשיטת הסטיית תקן כדי להחליט אילו דגימות לשמור ואילו להסיר, זאת בנוסף לפיצ'ר 19 שכלל כ-80% ערכים ריקים ולכן הוסר במלואו. לאחר מכן, ביצענו One Hot Encoding עבור הפיצ'רים הקטגוריאליים, כאשר כאן נוצרו לנו למעשה שני תתי סטים, הראשון עם Feature Selection ו- One Hot Encoding והשני רק עם One Hot Encoding. לאחר מכן, ביצענו השלמת חוסרים ב-4 דרכים שונות, כאשר כל דרך שכזו יצרה למעשה תת סט שהיווה עבורנו אופציה נוספת בהרצות של המודלים בהמשך. השלב הבא היה נרמול נתונים באמצעות MinMax לכל סט נתונים שכזה, ולבסוף הקטנו מימדיות באמצעות שימוש ב-PCA, וגם כאן בחנו 3 אפשרויות שונות – כזו עם שונות מוסברת של 90%, 95% ו-98%. עבור כל סט נתונים הרצנו פרידיקציות באמצעות ANN, על מנת להכריע באיזה סט נתונים להשתמש במודלים בשלב הבא. כאשר מצאנו את תצורת סט הנתונים הטובה ביותר מבחינתנו, הרצנו 2 מודלים ראשוניים – KNN ו- Logistic Regression, ו-2 מודלים מתקדמים – ANN ו- SVM. בעזרת שימוש ב- GridSearch מצאנו עבור כל מודל את ההיפר-פרמטרים הטובים ביותר מאלו שנבדקו, אלו אשר עבורם התקבל ערך ה-AUC הגבוה ביותר. בתום הרצות המודלים השונים, השתמשנו ב- K-Fold עם K=5 ובנינו פלט ROC, אשר דרכו חישבנו את מדד המבחן שנקבע עבורנו – מדד ה-AUC, וקיבלנו כי הערך הגבוה ביותר התקבל הוא עבור מודל ה-ANN. בשלב האחרון והמכריע ביצענו בדיקת ווידוא האם המודל שלנו היה Overfitted, ובדקנו זאת באמצעות ההפרשים בין ה- AUC Score של סט האימון וסט הוולידציה, וכמו כן בהפרשים של פונקציית ה- Log Loss, כאשר בשני המקרים הוגדר ערך סף שמעליו המודל הוגדר בתור Overfitted, ומתחתיו כתקין. עבור כל ארבעת המודלים, הן בהפרש של ה- AUC Score והן בערך פונקציית ה- Log Loss, התקבלו ערכים הקטנים מערך הסף ולפיכך הסקנו כי המודלים אינם Overfitted, ובפרט מודל ה-ANN שאיתו חישבנו את ההסתברויות של התיוג הסופי בסט המבחן. כפי שניתן היה לצפות, המודלים הראשוניים, מעצם היותם פשוטים ובסיסיים למדי, הניבו תוצאות פחות טובות מאשר המודלים המתקדמים יותר.

1. <https://public.tableau.com/app/profile/doron.shapira8537/viz/MLProjectG2/homepage> (יש לבחור בתצוגה מלאה בתחתית המסך בצד ימין) [↑](#footnote-ref-1)
2. <https://scikit-learn.org/stable/auto_examples/impute/plot_iterative_imputer_variants_comparison.html> [↑](#footnote-ref-2)
3. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html> [↑](#footnote-ref-3)
4. <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html> [↑](#footnote-ref-4)