**תקציר מנהלים:**

בפרויקט זה נעסוק בבעיית סיווג בינארית על סמך סט נתונים בעל 21 מאפיינים (פיצ'רים). אנו נבדוק 4 מודלים – KNN, Logistic Regression, ANN, SVM. מטרת הפרויקט הינה למצוא את המודל אשר ממקסם את מדד ה-AUC, ובכך בתקווה יניב לנו חיזוי מדויק ככל הניתן. בשלב הראשון של הפרויקט ביצענו אקספלורציה של הנתונים וממנה הפקנו מידע סטטיסטי וויזואלי, אשר הכין עבורנו את הקרקע לטובת שלב העיבוד המקדים. בשלב העיבוד המקדים טיפלנו בדגימות חריגות ובדגימות חסרות, נרמלנו את הנתונים והקטנו מימדיות. אנו השתמשנו בסטי נתונים שונים ומגוונים, כאשר בסופו של דבר בחרנו את סט הנתונים שהניב את ה-AUC הגבוה ביותר במודל שנבחר שרירותית, מודל ה-Logistic Regression. בתום גיבוש סט הנתונים, הרצנו את המודלים לעיל ובאמצעות שימוש בשיטת K-Fold חילקנו את אותו סט נתונים לשניים – Train ו- Validation. עבור כל פולד שכזה ביצענו הערכה בעזרת Confusion Matrix ועקומת ROC (שממנה התקבל ערך ה-AUC), וזאת לטובת הכרעה איזה מודל הוא הטוב ביותר מבין אלו שנבדקו. לטובת בדיקת ה-Overfitting השתמשנו בפונקציית Log-Loss וכן בהפרשים מקסימליים של מדד ה-AUC. בסופו של דבר, קיבלנו כי המודל הטוב ביותר היה מודל ה-ANN, שאינו חטא ב-Overfitting והניב את ערך ה-AUC הגבוה ביותר – 89%. נציין כי הקוד חולק כך שיהיה קל להגדיר באמצעותו קונפיגורציות שונות עבור ניתוח של סט הנתונים בצורות מגוונות, בתקווה כי הדבר יניב תוצאות טובות יותר במטריקות הנבדקות. כמו בכל מודל למידת מכונה, במידה והיינו ממשיכים להריץ ולבחון סטי נתונים או היפר-פרמטרים נוספים, קרוב לוודאי שהיינו מצליחים לשפר את הקיים אף יותר.

**חלק ראשון – אקספלורציה (כלל הוויזואליזציות של חלק זה נמצאות בנספחים 1 ו-2):**

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. לאחר מכן, חקרנו כיצד התיוג שלנו נראה. מהפלט שהתקבל (איור 4) ניתן לראות כי 82% מהתיוג הוא 0, והיתר 1. כעת, נרצה לאפיין את הפיצ'רים הקיימים ולסווגם בתור נומריים, בינאריים או קטגוריאליים, וזאת מאחר וכל סוג פיצ'ר יצריך ביצוע אקספלורציה מעט שונה מהאחרים. בדקנו כמה פיצ'רים יש בידינו מכל סוג, וקיבלנו כי ישנם 12 פיצ'רים נומריים, 2 פיצ'רים בינאריים, ו-7 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) (הפלטים מצורפים בנספח 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 שנעשתה באמצעות ה-Tableau, הצגנו את ההיסטוגרמה ואת גרף הצפיפות שלו באמצעות שימוש בחבילת Seaborn. בנוסף, עבור כל פיצ'ר נומרי הצגנו גרף התפלגות המראה כיצד ערך ה-target (label) מפולג על פני אותו פיצ'ר. לבסוף, הצגנו גרף boxplot, וזאת בכדי ללמוד האם השונות אחידה על פני הרמות השונות של כל פיצ'ר נומרי. למדנו כי ברוב הפיצ'רים הנומריים 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 Seaborn והן באמצעות 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 (למשל בפיצ'רים 1,6,19), כאשר עבור פיצ'ר 2 התקבל גרף לא תקין, ולכן נצטרך להתייחס לכך בשלב העיבוד המקדים ולהחליט כיצד לפעול עם פיצ'רים אלו.

**חלק שני – עיבוד מקדים:**

ראשית, כהמשך ישיר לאקספלורציה שביצענו, נרצה להתחיל את שלב העיבוד המקדים דווקא משיטת ה- **Feature Selection** שמשמשת להקטנת מימדיות, נושא שנתייחס אליו בהמשך. זהו אינו מהלך טריוויאלי, אך ראינו לנכון לבצע זאת לאור תוצאות מודל טובות יותר, כל שכן לאחר התעמקות[[2]](#footnote-2) בקשר שבין ויזואליזציות וקבלת החלטות ב-Feature Selection. 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. כמו כן, שמנו לב כי בפיצ'רים 15 ו-0 ישנה כמות גבוהה של אפסים אשר אינה מוסיפה מידע משמעותי אודות התיוג (איור 7), ולכן החלטנו להורידם גם כן. כעת, נרצה להתייחס לסוגיית **הנתונים החריגים (Outliers)**. כפי שעולה מהאקספלורציה שבוצעה בחלק הראשון, ניתן להבחין כי אכן קיימים נתונים חריגים בסט הנתונים, כאלו העלולים לייצר הטיות משמעותיות במדדים הנבדקים, ועל כן נרצה לקבל החלטה כיצד לנהוג עמם. מאחר וקיים בידינו 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 לנלמד 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). בתום ההסרה נותרו בידינו כ-90% מכמות הדגימות המקורית.

כעת, נרצה לתת את הדעת אודות **הפיצ'רים הקטגוריאליים** בסט הנתונים שלנו. לטובת השלבים הבאים בניתוח, אנו נדרשים להחליט על 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. לנוכח הטרייד-אוף הזה, החלטנו לבחון את שתי הגישות על סט הנתונים שלנו. האפשרות הראשונה הייתה הפעלת One Hot Encoding על סט הנתונים כפי שהתקבל בתום שלב ה- Feature Selection, ואילו האפשרות השנייה הייתה לקיחת אותו סט נתונים אך עם ביצוע מספר שינויים. 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. בפיצ'ר 12 היו שני ערכים קטגוריאליים של "y" ו-"n", ובחרנו להמיר אותם ל-1 ול-0, בהתאמה. כמו כן, בפיצ'רים 1,6,16 המרנו את האותיות של כל קטגוריה לערך מספרי. כעת נעבור להתייחס **לטיפול בחסרים**, אשר נבצע טרם הנרמול. קיימת סברה שהדבר יכול לפגוע בהמשך בנרמול ולהוסיף רעשים, אך אנו מתבססים על ההנחה שכמות החסרים ביחס לכמות הנתונים אינה גבוהה, ולכן ההשפעה זניחה. בנתונים החסרים החלטנו לטפל ב-4 דרכים שונות – באמצעות 4 Imputers שונים. הראשון שבהם, **Simple Imputer**, התבסס על השלמה 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 להשלים ע"י הגרלת דגימה מסט הנתונים של אותו פיצ'ר, כאשר כל דגימה בעלת הסתברות זהה, וזאת מאחר שנשארו בידינו מעל 19 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 שהתבצע ע"י סוגי רגרסיות שונות, וזאת על מנת למצוא את הערך המסתבר ביותר מהרגרסיות השונות, ולבסוף איתו תתבצע ההשלמה[[3]](#footnote-3). בחרנו לבצע את ההשלמה בדרך זו מאחר והיא לוקחת בחשבון גם את האינטרקציות בין הפיצ'רים השונים ולא רק על סמך הפיצ'ר הבודד. 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. במידה ובהמשך נרצה להקטין מימדיות מבלי שביצענו נרמול, הקטנת המימדיות עלולה לתת חשיבות לפיצ'רים מסוימים על סמך האחרים שלא מטעמים רלוונטים. 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 נמצאות 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-18 בסטים שעברו Feature Selection. נבחין כי 512 המימדים עתידים ליצור עומס ומורכבות גבוהה מאוד שתצריך כוח עיבוד משמעותי וזמן חישוב ארוך בשלבים הבאים של יישום המודלים. בנוסף, קרוב לוודאי שיהיו בידינו מאפיינים חופפים אשר אינם מוסיפים מידע חדש, אלא אף מקשים על הסבר התוצאה. בהמשך ישיר למה שלמדנו בהרצאה אודות 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 ניעזר בשיטת **PCA** לטובת הורדת המימדיות לסטי הנתונים הללו. טרם ביצוע שיטת ה-PCA, נזכור בי בשלב האקספלורציה נוכחנו לדעת כי ישנם שני זוגות של פיצ'רים בעלי קורלציה גבוהה (7,9 ו-5,17). נבחנה אפשרות להוריד כל אחד מהם או את שניהם יחדיו, אך בסופו של דבר הוחלט לבצע את הקטנת המימדים אך ורק באמצעות ה-PCA, זאת לנוכח כמות הקומבינציות הגבוהה שהחלה להצטבר והצריכה זמני עיבוד ארוכים. נציין כי את שיטת ה-PCA החלטנו ליישם גם על סטי הנתונים שהיו בהם 18 מימדים, זאת מאחר ואם יתאפשר להוריד את המימדיות אך עדיין לשמור על אחוז גבוה של שונות מוסברת, אז בהחלט זה יפחית, גם אם במעט, את הסיבוכיות החישובית של המודלים. בכל ניתוח PCA שכזה, אנו נבחן 3 אפשרויות עבור אחוז השונות המוסברת - בראשונה נדרוש כמות פיצ'רים שתסביר 90% מהשונות, בשנייה 95% ובשלישית 98%. נבחין כי בשלב זה היו בידינו 24 סטים של נתונים (3\*4\*2). על מנת לבחור את סט האימון שעליו נריץ בהמשך את המודלים, נדרש מאיתנו לבצע הערכה כלשהי ולהכריע בסוגייה זו. לטובת זאת, החלטנו להחיל את מודל הרגרסיה הלוגיסטית (בחירה שרירותית של יעילות זמנים) על כל סטי הנתונים. 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 לאחר Feature Selection, שימוש ב- Iterative Imputer ו-PCA של 95%. נשים לב כי בחרנו במודל שכלל PCA ברמת שונות מוסברת של 95% על פני ה-98% גם מאחר והוא קיבל ערך AUC גבוה יותר, וגם כי הוא נמצא במקום מאוזן יותר מבחינת הטרייד-אוף שבין הסברת שונות גבוהה ככל הניתן לבין הוספת כמות מימדים. ניתן להתרשם מכך באיור 11.

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:**

בחרנו את המודלים הבאים: KNN Logistic Regression ANN 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 מוגש בקובץ פייתון נפרד, וזאת בכדי למנוע ריצה נוספת שלו שלא צורך, שכן כבר מצאנו ההיפר-פרמטרים. יתרה מכך, במחברת ה-Jupyter מופיע תיעוד שממחיש את רצף השלבים שביצענו במהלך ה-GridSearch.

**KNN:** הפרמטר היחיד של מודל זה שאיתו ביצענו בדיקות היה ערך ה-K, כלומר מספר השכנים. אנו ביצענו הרצות בעזרת פונקציית ה-Gridsearch בקפיצות של 10 כל פעם (החל מ-31 ועד 201), והערך הטוב ביותר של ערך ה-AUC התקבל עבור K=131. ההיפר-פרמטרים האחרים נותרו ללא שינוי, ובפרט אלו שהוזכרו בתרגול: weights=’uniform’, metric=’euclidean’.

**Logistic Regression:** מאחר וה-label שלנו בינארי, השתמשנו במודל הרגרסיה הלוגיסטית אשר מבצעת חיזוי לערכים בינאריים. ההיפר-פרמטרים של המודל הזה מייצגים 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), ערכי C שונים, מס' איטרציות מקסימלי, וכן שינוי ה-solver. בסופו של דבר בעזרת פונקציית ה-GridSearch נבחרו ההיפר-פרמטרים הבאים: max\_iter=150, solver=’sag’, tol=1e-4[default] penalty=l2, C=15,. שאר הפרמטרים[[4]](#footnote-4) של מודל הרגרסיה הלוגיסטית נשארו ללא שינוי.

**Multi-Layer Perceptron (ANN):** זהו מודל שבו מתבצעת למידה של הרשת והיא תלויה במספר רחב של היפר-פרמטרים. אנו ביצענו את ההרצה תוך שינוי ההיפר-פרמטרים הבאים: פונקציית האקטיבציה, כמות וגודל השכבות החבויות (hidden layers sizes), שיעור הלמידה ההתחלתי (learning rate init), ערך הלמידה (learning rate), ה-solver (Stochastic Gradient Descent(SGD), ADAM או L-BFGS) , גודל ה-alpha (L2 penalty), גודל המנה (batch size), מספר החזרות המקסימלי (max iter), ערך ה-seed (random state) וערך ה-beta\_1 (פרמטר ייחודי עבור solver מסוג adam). בתום ה-GridSearch נבחרו הערכים הבאים להיפר-פרמטרים:

activation=’relu’, hidden\_layer\_sizes=100, learning\_rate\_init=0.0015, learning\_rate=’invscaling’, solver=’adam’, alpha=0.035, batch\_size=10, max\_iter=500, random\_state=42, beta\_1=0.8, power\_t=0.5[default], early\_stopping=False[default], tol=1e-4[default], warm\_start=False[default]

**Support Vectors Machine (SVM):** עבור מודל זה קיימים היפר-פרמטרים רבים[[5]](#footnote-5), כאשר אנו בחנו את שלושת ההיפר-פרמטרים המרכזיים: פרמטר C – ערך הרגולריזציה, פונקציית הקרנל והדרגה (אשר רלוונטית לקרנל הפולינומיאלי בלבד). לאחר ביצוע ה-GridSearch נבחרו הערכים הבאים להיפר-פרמטרים: C=0.01, kernel=rbf, degree=n/a, gamma=scale[default], probability=True.

שאר הפרמטרים נותרו הדיפולטיבים של מודל זה (הפרמטרים שלא דיברנו עליהם בהרצאה או בתרגול).

**חלק רביעי – הערכת המודלים (הוויזואליזציות עבור חלק זה מופיעות בנספח 3):**

**בניית Confusion Matrix (איור 12):** אנו בנינו את ה- Confusion Matrix עבור מודל ה-ANN. הציר האנכי מסמן את התיוגים האמיתיים והציר האופקי את התחזיות. התא השמאלי העליון מייצג את כמות הפעמים שחזינו 0 והערך האמיתי אכן 0, ניתן לראות שב-79% מהפעמים אנו צודקים. התא הימני התחתון מראה כמה פעמים חזינו 1 והערך האמיתי היה 1 – 7%. שני התאים האחרים מייצגים את שיעור החיזוי של ערך 0 או 1, כאשר הערך האמיתי היה 1 או 0 בהתאמה, ובסה"כ ניתן לראות כי המודל לא מצליח לתייג נכונה ב-14% מהמקרים.

**הערכת המודל באמצעות K-Fold Cross Validation:** עבור כל מודל עם סט האימון שנבחר בתום שלב העיבוד המקדים, בהתחשב בהיפר-פרמטרים הטובים ביותר שנמצאו, ביצענו הערכה תוך שימוש ב-K-Fold ובניית פלט ה-ROC. ב-K-Fold עשינו שימוש ב-5 פולדים ושמרנו על חלוקה של 80% מהדגימות לסט האימון ו-20% לסט הוולידציה. לכל מודל התקבל גרף ROC וערך AUC (איורים 13-16), 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 89%.

**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- 1.8%. בחרנו להגדיר מודל שהוא Overfitted על סמך מדד ה-AUC, אך ורק אם ההפרשים בין ה-AUC Score של סט האימון לזה של סט הוולידציה גבוהים מ-4%. אי לכך, ניתן לומר כי כל המודלים שלנו אינם Overfitted, היות והפער המקסימלי היה כאמור 1.8%. 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 2%. אנו בחרנו את 3% להיות ערך הסף עבור הקביעה אם מדובר ב-Overfitting, ולכן נוכל להסיק גם ממדד זה כי המודלים אינם Overfitted. יתרה מכך, על מנת לבצע הכללה של המודל יישמנו עליו את סוגיית ה-Data Leakage, כל שכן במהלך הרצות ה-GridSearch, פרט לרצון העז להשיג ערך AUC גבוה ככל שניתן, הקפדנו לבצע בדיקה צמודה שמא אנו ב-Overfitting (באמצעות הקריטריונים שהוגדרו על ידינו לעיל עבור מודל שהוא Overfitted, בהתחשב באילוצי זמן עיבוד ומשאבי מערכת).

**הערכת מודל באמצעות K-Fold Cross Validation עבור הימנעות מ- Data Leakage[[6]](#footnote-6):** במהלך הפרויקט נוכחנו לגלות את התופעה[[7]](#footnote-7) שנקראת Data Leakage. כשמה כן היא, זוהי תופעה שבה בשלב אימון המודל משתמשים בנתונים אשר לא היו אמורים להיות זמינים או לא היו אמורים להיות חשופים למודל. נבחין כי בשלב העיבוד המקדים שביצענו, סט הנתונים עבר את שלבי העיבוד כיחידה אחת, ורק בתום שלב זה, כאשר היה בידינו סט הנתונים הטוב ביותר, חילקנו אותו לסט ה- Train ולסט ה- Validation. מכאן, כאשר רצינו לבצע את הווידוא בעזרת ה- Validation, למעשה המודל כן הכיר במידה מסוימת את הנתונים שנגלו לפניו, שכן "אחיו החורג", סט ה- Train, היה בעל מאפיינים דומים אם לא זהים, ולכן מלכתחילה נוצרה כאן הטייה שיטתית. לבסוף, החלטנו בנוסף להערכות החובה שנדרשו מאיתנו במסגרת הפרויקט, לבצע הערכה נוספת תוך שימוש ב-K-Fold וביצוע עיבוד מקדים בצורה נפרדת עבור ה-Train וה-Validation. כך קיבלנו אינדיקציה נוספת ליכולותיו של המודל שלנו להתמודד עם נתונים שלא ראה מימיו.

**סיכום:** בפרויקט זה עסקנו בבעיית קלסיפיקציה, כלומר חיוזי של תיוג בינארי על סמך סט דגימות נתון של 21 פיצ'רים. לאורך הפרויקט יישמנו את שלמדנו לאורך הקורס, לרבות ההרצה בפועל של מודלים שונים. בתחילת הפרויקט ביצענו אקספלורציה מעמיקה על סט הנתונים, וממנה הפקנו מסקנות מכריעות אודות הצעדים שנצטרך לבצע על סט הנתונים על מנת להכינו לשלב הרצת המודלים. חילקנו את הפיצ'רים של סט הנתונים לפיצ'רים נומריים, קטגוריאליים ובינאריים, וביצענו אקספלורציה ייעודית על כל סוג פיצ'ר, לרבות שימוש בכלי ה- Tableau. לאחר מכן, התחלנו בשלב העיבוד המקדים. שלב זה כלל בתוכו מספר תתי שלבים. ראשית, נעזרנו בשיטת Feature Selection כדי לסנן שלושה פיצ'רים. הראשון שבהם היה פיצ'ר 19 שכלל כ-80% ערכים ריקים, והשניים הנותרים היו פיצ'רים 0 ו-15 שכללו אפסים רבים שלא הוסיפו מידע ענייני. לאחר מכן, השתמשנו לטובת הטיפול בנתונים החריגים בשיטת סטיית התקן והחלטנו אילו דגימות לשמור ואילו להסיר. השלב הבא היה הטיפול בפיצ'רים הקטגוריאליים באמצעות One Hot Encoding, ו-Label Encoding לאחר ביצוע תיקונים עבור הפיצ'רים 1,2,6,12,16,18. לאחר מכן, ביצענו השלמת חסרים ב-4 דרכים שונות, כאשר כל דרך שכזו יצרה תת סט שהיווה עבורנו אופציה נוספת בהרצות המודלים בהמשך. השלב הבא היה נרמול נתונים באמצעות MinMax לכל סט נתונים, 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 הרצנו פרידיקציות באמצעות מודל ה-Logistic Regression, בכדי להכריע באיזה סט נתונים להשתמש במודלים בשלב הבא. כאשר מצאנו את תצורת הסט 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, אשר עבורם התקבל ערך ה-AUC הגבוה ביותר. בתום ההרצות, השתמשנו ב- K-Fold עם K=5 ובנינו פלט ROC, אשר דרכו חישבנו את מדד המבחן – מדד ה-AUC, וקיבלנו כי הערך הגבוה ביותר התקבל עבור מודל ה-ANN. בשלב האחרון והמכריע ביצענו ווידוא האם המודל שלנו הוא Overfitted, ובדקנו זאת באמצעות ההפרשים בין ה- AUC Score של סט האימון וסט הוולידציה, וכמו כן בהפרשים של פונקציית ה- Log Loss. בשני המקרים הוגדר ערך סף שמעליו המודל הוגדר בתור Overfitted, ומתחתיו כתקין. עבור כל ארבעת המודלים, הן בהפרשי ה- AUC Score והן בהפרשי פונקציית ה- Log Loss, התקבלו ערכים הקטנים מערך הסף. מכאן, הסקנו כי המודלים אינם Overfitted, ובפרט מודל ה-ANN שאיתו חישבנו את ההסתברויות של התיוג הסופי בסט המבחן.

**נספחים:**

**נספח מס' 1 – הוויזואליציות מכלי ה-Tableau:**

Chart

Description automatically generated

איור 1 – Tableau Dashboard

מסך features overview: מסך זה מציג את היסטוגרמות הפיצ'רים יחדיו. נשים לב שקיימים הבדלים בין ההיסטוגרמות השונות, כל שכן הבדלים בין הפיצ’רים עצמם ((Num - Numeric, Cat-Category, Bin-Binary. מההתבוננות הויזואליות ניתן להבחין במספר תובנות, למשל שפיצ'ר 15 אשר ברובו ככולו מסווג כ-0 או למשל פיצ'רים 3 ו-20 המזכירים התפלגות נורמלית.

Diagram

Description automatically generated

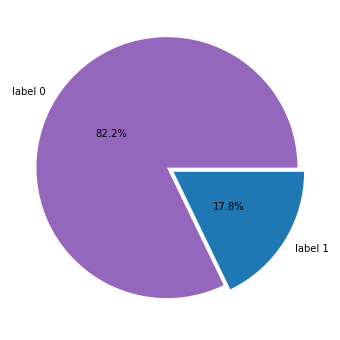
איור 2 – מסך Discover a Correlation

במסך זה ניתן לראות את ההתנהגות הקורלטיבית של כל הפיצ'רים. כפי שניתן לראות, פיצ'רים 5 ו-17 ואיתם גם 7 ו-9 הינם קורלטיביים. להבדיל מפיצ'רים אלו, שאר הפיצ'רים לא נראים קורלטיביים וכך גם ראינו במטריצת הקורלציה שבנינו במחברת ה-Jupyter.

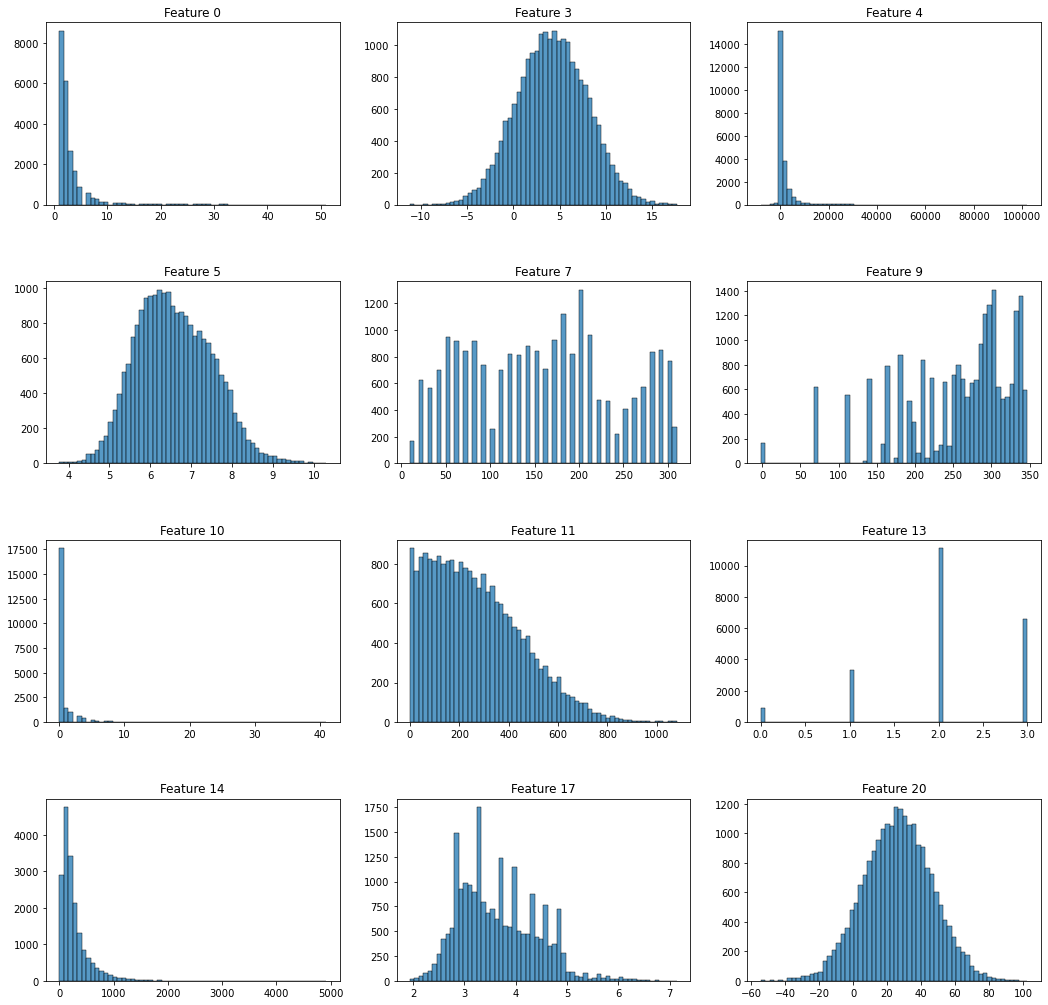
Chart, scatter chart

Description automatically generatedבנוסף, ניתן ללחוץ על כפתור Explore Correlative Features על מנת להתמקד בפיצ'רים הקורלטיביים בלבד.

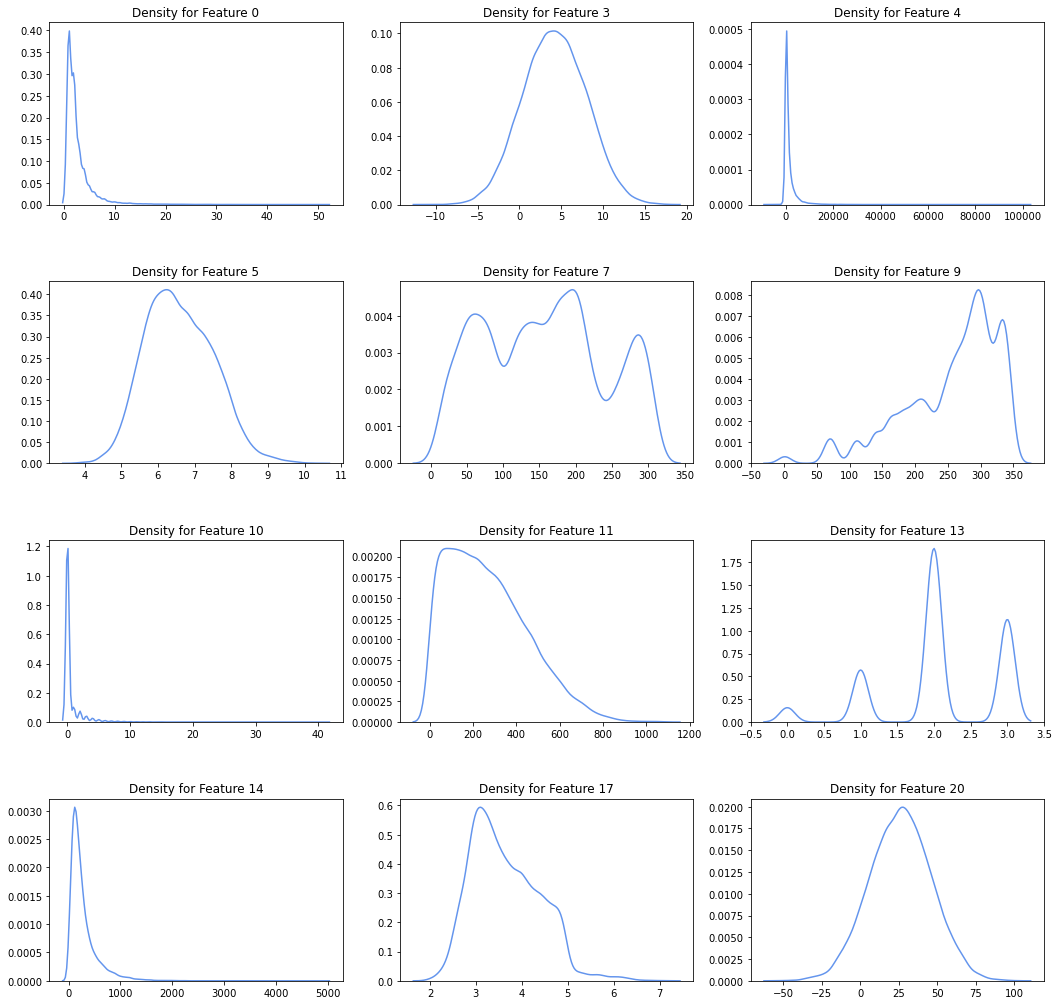
איור 3 – מסך Explore Correlative Features

**נספח מס' 2 – הוויזואליציות משלב האקספלורציה:**

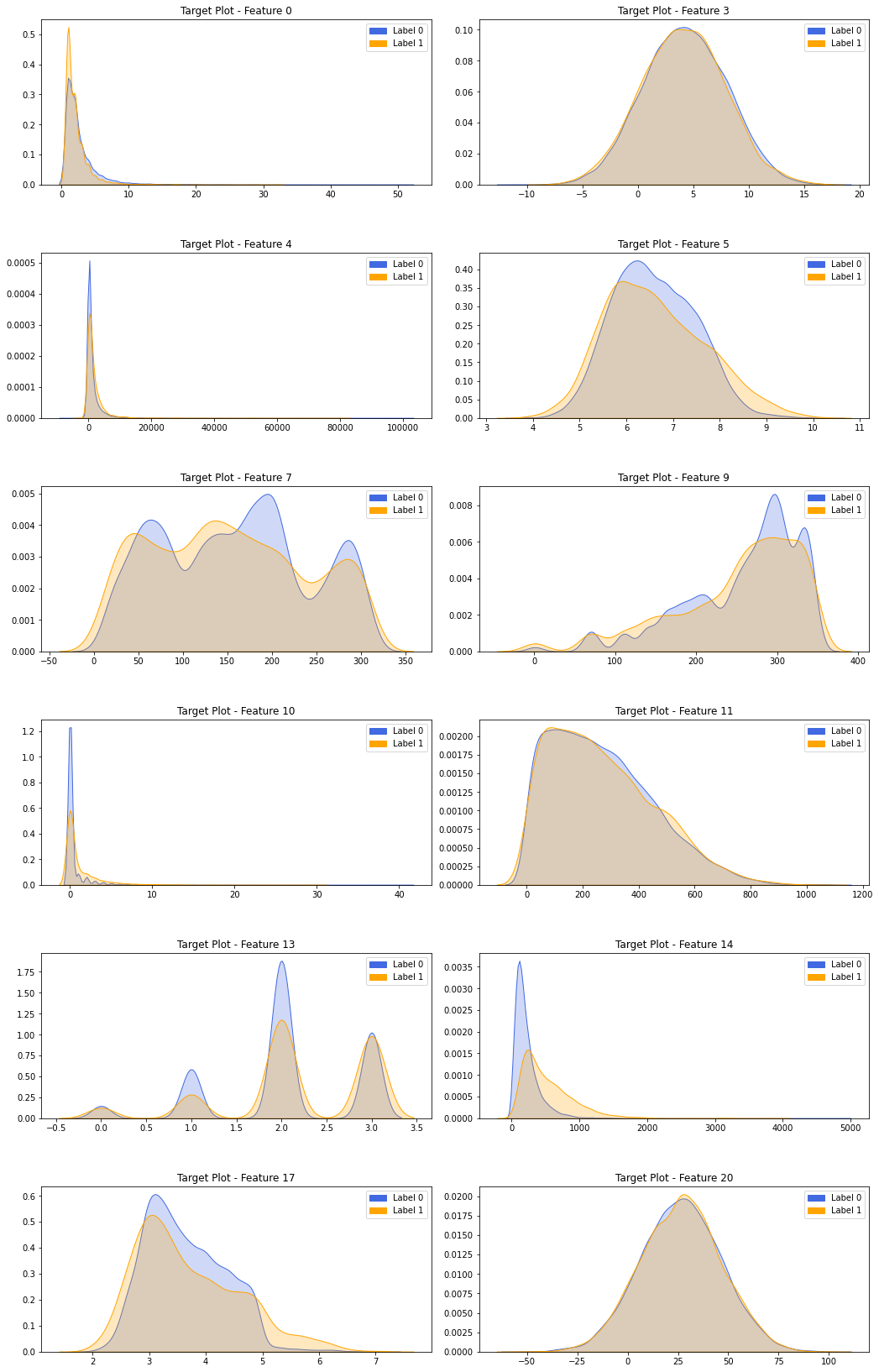
איור 4 – התפלגות ה-label



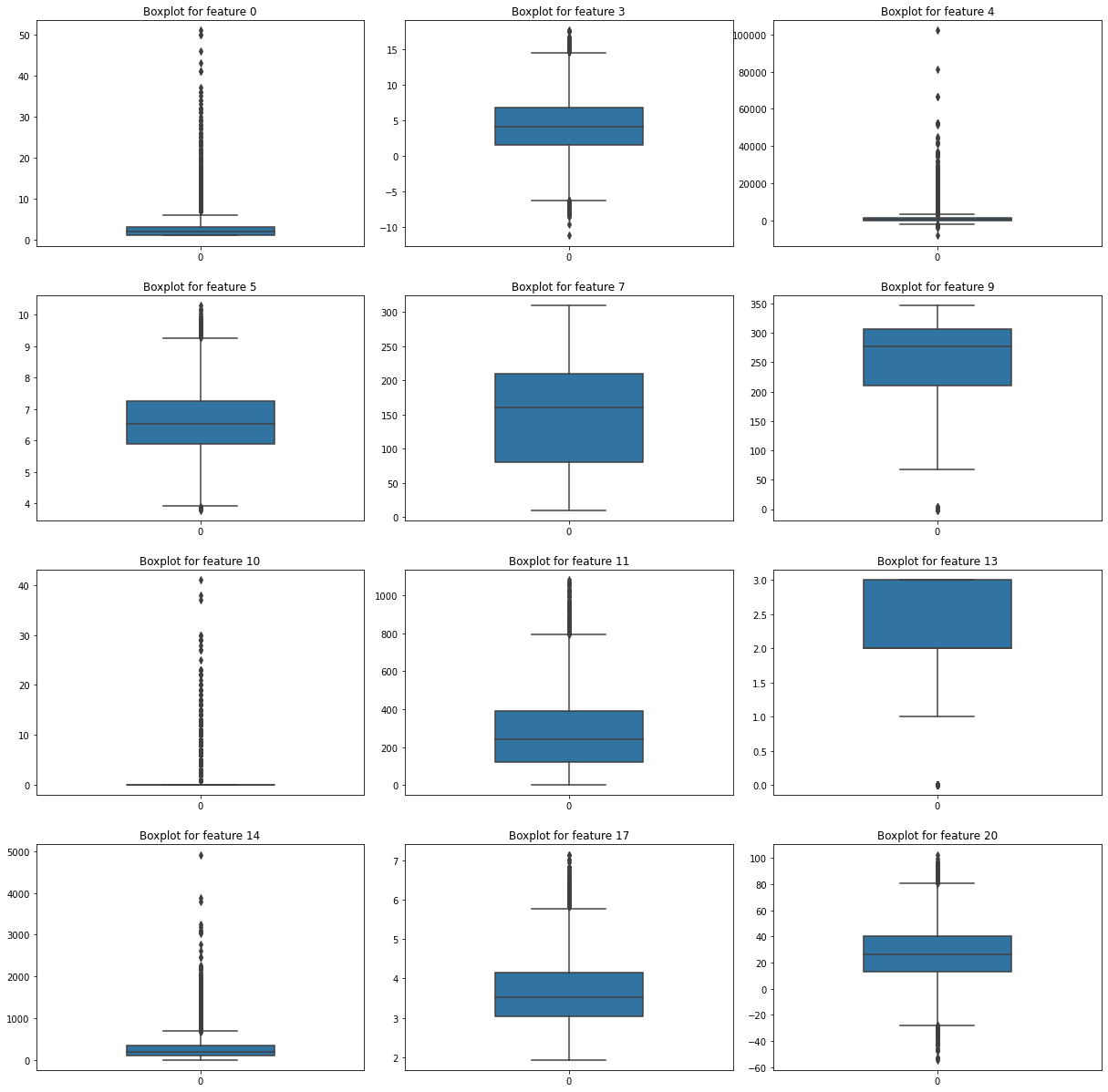
איור 5 – ההיסטוגרמות עבור הפיצ'רים הנומריים



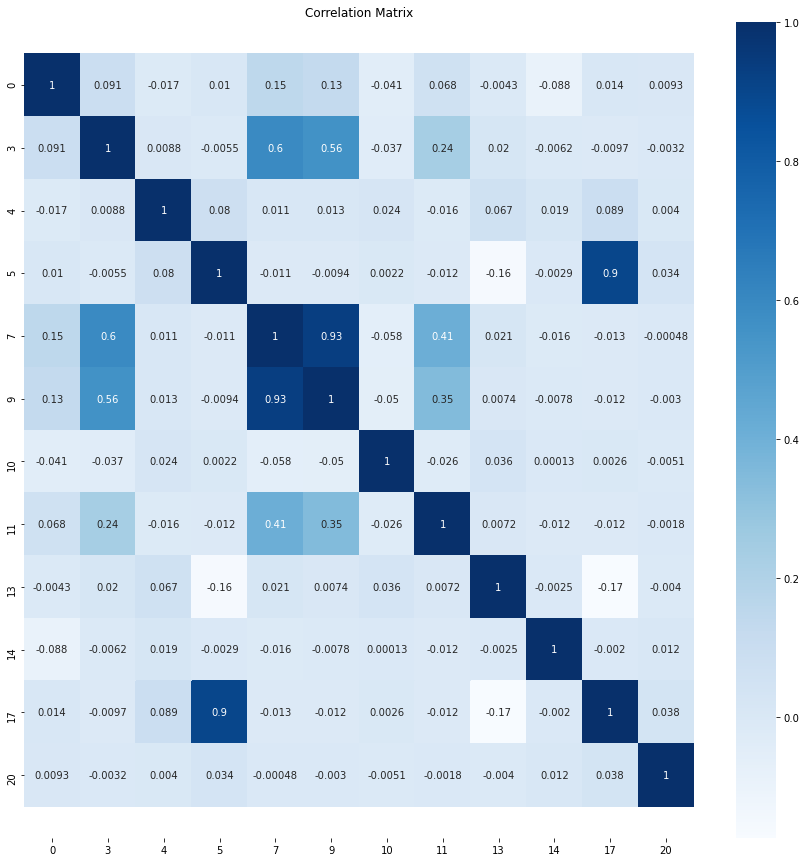
איור 6 – גרף הצפיפות של כל פיצ'ר נומרי



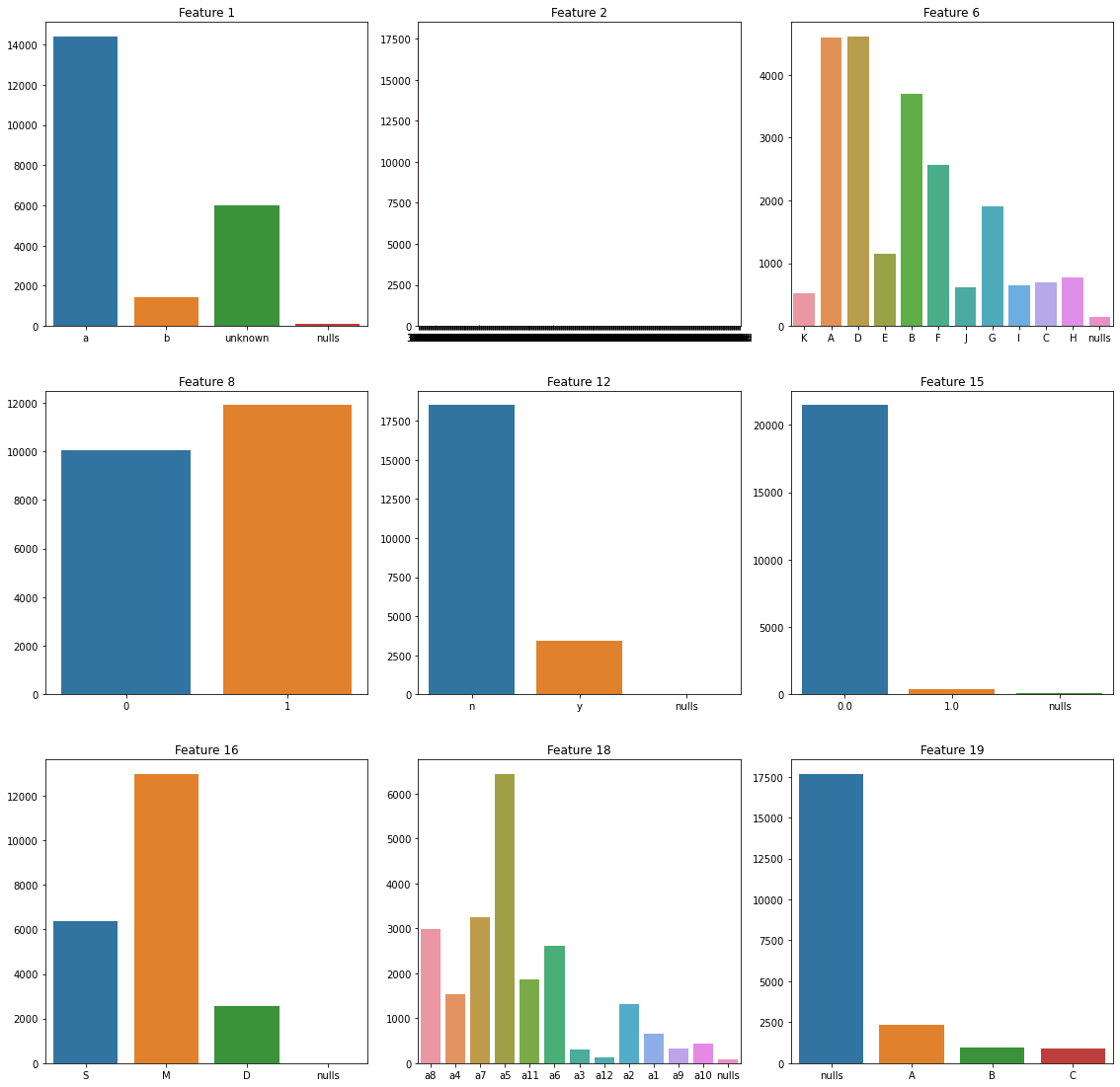
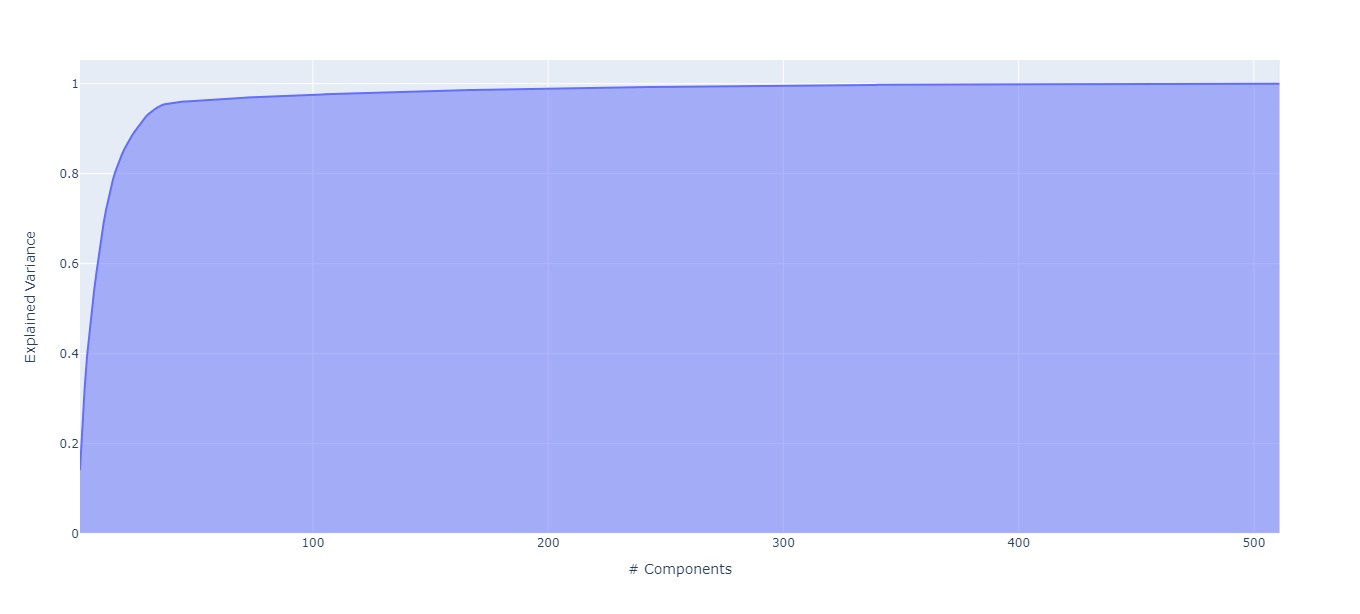
איור 7 – התפלגות ה-Target על פני כל פיצ'ר נומרי



איור 8 – תרשימי ה-boxplot עבור כל פיצ'ר נומרי



איור 9 – מטריצת הקורלציה

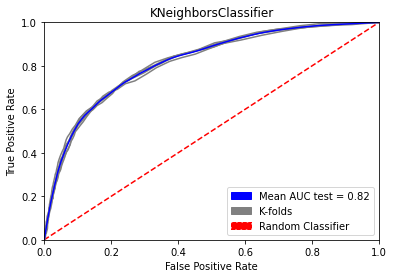
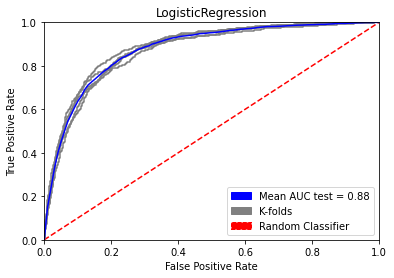


איור 11 – המחשת הטרייד-אוף של ביצוע PCA

איור 10 – היסטוגרמות עבור הפיצ'רים הבינאריים והקטגוריאליים

**נספח מס' 3 – שלב הערכת המודלים:**

איור 12 – Confusion Matrix

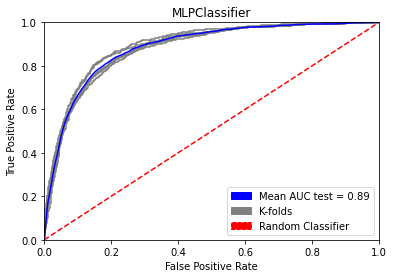
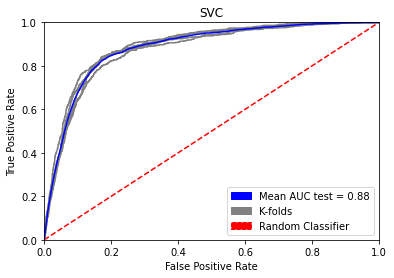


איור 15 – פלט ROC עבור ANN

איור 16 – פלט ROC עבור SVM

איור 14 – פלט ROC עבור Logistic Regression

איור 13 – פלט ROC עבור KNN



1. <https://public.tableau.com/app/profile/doron.shapira8537/viz/MLProjectG2/homepage> (יש לבחור בתצוגה מלאה בתחתית המסך) [↑](#footnote-ref-1)
2. <https://www.kaggle.com/kanncaa1/feature-selection-and-data-visualization> [↑](#footnote-ref-2)
3. <https://scikit-learn.org/stable/auto_examples/impute/plot_iterative_imputer_variants_comparison.html> [↑](#footnote-ref-3)
4. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html> [↑](#footnote-ref-4)
5. <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html> [↑](#footnote-ref-5)
6. <https://machinelearningmastery.com/data-preparation-without-data-leakage/> [↑](#footnote-ref-6)
7. <https://en.wikipedia.org/wiki/Leakage_(machine_learning)> [↑](#footnote-ref-7)