# פרויקט גמר - מבוא ללמידת מכונה

# קבוצה 19

הדו"ח

הקוד

# תוכן עניינים

חלק ראשון - אקספלורציה ..... עמוד 3

חלק שני - עיבוד מקדים ..... עמוד 3

חלק שלישי - הרצת מודלים ..... עמוד 4

חלק רביעי - הערכת המודלים ..... עמוד 6

חלק חמישי - ביצוע פרדיקציה ..... עמוד 7

חלק שישי - שימוש בכלים שלא נלמדו ..... עמוד 7

נספח א' - ויזואליזציה ..... עמוד 8

נספח ב' - חלוקת אחריות בצוות ..... עמוד 14

## חלק ראשון - אקספלורציה

חלקו הראשון של הפרויקט יעסוק בניתוח נתונים ובחינת הרלוונטיות שלהם. לטובת שלב זה בחרנו לבצע מספר בדיקות על כל אחת מהתוצאות שקיבלנו בדאטה. את הבדיקות החלטנו לבצע על סט הנתונים המלא לאחר הסרה של תוצאות (שורות) בהן חסרים יותר מ-3 ערכים שונים (3 ערכים בשורת התוצאה בהם יש N/A). ההנחה היא שעל משתנה עם יותר משלושה ערכים חסרים עלולה להטות את המודל ולא לסייע לו.

הבדיקות כללו בדיקת נתונים חסרים (נספח א' - חלק ראשון), אופן התפלגות המשתנים הנומריים (היסטוגרמה, נספח א' - חלק ראשון), מציאת הערכים המצויים בכל עמודה של משתנה קטגוריאלי (נספח א' - חלק ראשון) ובנוסף קורלציה בין המשתנים הנומריים (נספח א' - חלק ראשון) וקורלציה בין המשתנים הקטגוריאליים (נספח א' - חלק ראשון).

מבדיקת **כמות הנתונים החסרים בכל עמודה** (פיצ'רים) ניתן לראות כי אחוז הנתונים החסרים נע בין 0% עד 6.25% ולכן לא ניתן לסנן עמודות בצורה מובהקת. (הנחנו כי במידה והאחוז יעלה על 10% - הפיצ'ר יהיה לא רלוונטי להשפעה על ההחלטה הסופית ונוכל להסירו בהמשך).

הבדיקה הבאה הינה **בדיקת היסטוגרמה של המשתנים הנומריים,** בבדיקה זו הסרנו את כלל הפיצ'רים הקטגוריאליים ואת כלל הפיצ'רים הבינאריים. לאחר הסרה של הפיצ'רים אשר לא ניתנים לניתוח בעזרת היסטוגרמה, למדנו על שאר הפיצ'רים שהם אינם מתפלגים בצורה נורמלית ואף לרוב הפיצ'רים יש נתונים קיצוניים. נדרש לבחון בהמשך האם נכון להסיר את הנתונים הקיצוניים לטובת בניית המודל והסקת המסקנות. אמנם לא זיהינו התפלגות נורמלית אבל כן נוכל להסיק כי רוב תוצאות הפיצ'רים מצויות באזור מרכזי אחד וקיימות דוגמאות קיצון בודדות ברוב הפיצ'רים.

כעת נעבור לבדיקה נוספת והיא **בדיקת הערכים המצויים בכל עמודה במשתנים הקטגוריאליים**. בחנו את הפיצ'רים הרלוונטיים כדי להבין האם הערכים המצויים חוזרים על עצמם בצורה משמעותית/ניתן לייצר מהמשתנים הקטגוריאליים קבוצות של דוגמאות העולות ברוב הבדיקות, זאת לטובת בניית המודל בהמשך. מצאנו כי ברגע שהמשתנה הוא משתנה בינארי ניתן ללמוד אודות השפעתו על המודל, במידה ופיזור התוצאות (0 או 1) מחולק בצורה שוויונות בקירוב, כלומר כמחצית מהתוצאות הן 0 ומחצית הן 1, אזי ככל הנראה לא יהיה ניתן להסיק מסקנות ישירות מפיצ'ר זה (כמו לדוגמא משתנה has_relocations). בנוסף לכך, ניתן ללמוד במשתנים הקטגוריאליים הלא בינאריים האם ישנם גורמים משותפים המאחדים תוצאות רבות כמו לדוגמא פיצ'ר C המראה כי רוב התוצאות שייכות ל- vh או er.

לבסוף, החלטנו לבצע **קורלציות בין הפיצ'רים**. את המשמעות של התוצאות הסקנו אל מול ההבנה של משמעות הפיצ'רים. מהקורלציות בין המשתנים הנומריים למדנו כי יש קורלציה גבוהה בין המשתנה size למשתנה numstrings ובין המשתנה size למשתנה MZ. לכן, נוריד את משתנה ה-size מהנתונים כדי למנוע "רעש מיותר" בניתוח הנתונים בהמשך. יתר על כן, שאר הקורלציות אינן בעלות מתאם חזק ולכן לא ניתן להסיק משמעויות נוספות.
מהקורלציות בין המשתנים הקטגוריאליים הסקנו כי אין אף משתנים בעלי קורלציה חזקה ולכן על כלל המשתנים הקטגוריאליים לקבל מענה במודל בהמשך.

## חלק שני - עיבוד מקדים

במהלך חלק זה טייבנו את הדאטה ואפשרנו שימוש בדאטה בצורה מדויקת יותר בהמשך לבניית המודלים. החלק כלל התמודדות עם תוצאות חריגות (Outliers), התמודדות עם נתונים חסרים בעמודות השונות, נרמול הנתונים וניתוח מימדיות הבעיה. כל אלו בוצעו על נתוני ה-Train, ולבסוף מומשו על נתוני ה-Test.

מהתהליך עלו מספר מסקנות עיקריות שרצינו לציין בדו"ח זאת בנוסף להסברים המפורטים במחברת הקוד.
**התמודדות עם תוצאות חריגות -** בתחילה חשבנו כי נכון להוריד 5% של תוצאות קיצון מכל עמודה ובפועל הנ"ל גרם להשמטה של 15% מהדאטה הכולל. הבנו שזו טעות ולכן שינינו את השיטה להתמודד - השיטה

היא קביעת ערך עבור כל פיצ'ר המסמל אחוזון עליון או תחתון ע"פ התפלגות ולקרב את כל דוגמאות הקיצון לערך שמציין את האחוזון.
**מדיניות מילוי חוסרים -** מדיניות מילוי הנתונים החסרים בכל עמודה הותאם לפי מאפייני הפיצ'ר (התפלגות הפיצ'ר וניתוח היסטוגרמה וגרף הפיזור של המשתנה). ניתן לראות את ההסבר המפורט לכל פיצ'ר בהסברים בקוד.
**קידוד משתנים נומינליים (שמים) -** הבנו כי אין באפשרותנו לנרמל את הנתונים כאשר מופיעות דוגמאות שמיות ולכן קודדנו את העמודות השמיות (File Type Trid ,C) בעזרת קידוד בינארי כי שיטה זו מוסיפה פחות מימדים ביחס לשיטות מקבילות.
**נרמול הנתונים -** ראינו כי כל פיצ'ר נמדד בסקלה שונה ולכן כדי לנטרל את ההשפעה השונה של כל נתון בפיצ'ר על המסקנה הכוללת במודל, ביצענו נרמול של הנתונים בעזרת MinMaxScaler . בחרנו בשיטה זו כי בשיטה זו ניתן לשמור על היחס בין הדוגמאות השונות למרות השינוי לערכים בין 0 ל-1, למעשה שינוי הערך לא פוגע במשמעות היחסית. את הנ"ל ביצענו לאחר התמודדות עם ה- Outliers, מכיוון שערכן משפיע על תהליך הנרמול של שאר הנתונים המתייחס לדוגמא הגדולה והקטנה ביותר וכך מחשב את היחסים בין הדוגמאות. בנוסף לנ"ל ישנן עוד מסקנות המפורטות ב- MarkDown במחברת הקוד.

## חלק שלישי - הרצת מודלים

בחלק זה, ביצענו הרצה של המודלים השונים על הדאטה שלנו. הנ"ל בוצע עד הגעה לתוצאה סופית ובחינת הרלוונטיות של השינויים שביצענו (כיול היפר-פרמטרים לכל מודל). בחרנו שלא לבצע הורדת מימדים זאת אל מול ההנחה שההורדה אינה תורמת רבות לשיפור התוצאות.

**מודלים ראשוניים - KNN, Logistic Regression:**

**מודל KNN:** מודל זה משמש למשימות סיווג ומשימות רגרסיה. מודל זה משתמש במופעי האימון לטובת ביצוע תחזיות. המודל מתייחס למספר השכנים שנגדיר לו (במקרה שלנו רצינו לבדוק K בין 5 ל20), ובעזרת הנתונים של K הנקודות הקרובות אליו, המודל מסווג את תוצאת הבדיקה. המודל הינו מודל יקר בזמן חישוב מהסיבה הפשוטה שהוא מחשב את המרחק בין הנקודה החדשה לבין הנקודות באימון.
ההיפר-פרמטר שבחרנו לשנות במודל הוא מספר השכנים עליו מופעל תהליך הבדיקה של המודל. אנחנו בדקנו מהי כמות השכנים המיטבית מבחינת המודל בין כמות של 5 שכנים ל20 שכנים ומצאנו שהפרמטר האופטימלי של כמות השכנים לטובת בניית מודל עם מספר השכנים האופטימלי שאומן על הTRAIN ונבדק על הואלידציה הוא 5.
ההשפעה על השונות וההטיה ניתנת להסבר בעזרת הגרף המצורף בנספח א' - חלק שלישי. בגרף ניתן לראות שבכל שכמות השכנים עולה, השונות קטנה עד נקודה מסוימת (עד 5 שכנים) ומשם מתחילה לעלות, לכן זאת הנקודה האופטימלית. לעומתה, ההטיה גדלה עם הגידול בשכנים. כמות השכנים האופטימלית, אשר ניתן ללמוד מהגרף, מראה כי אנחנו במצב של Overfitting . מצב זה ניתן להסבר בעקבות ההבנה כי המודל "משנן" את האימון בK-Fold ומגיעה לתוצאות AUC גבוהות (מעל 0.9) אך כאשר מנבאים את התוצאות ה-AUC לואלידציה, מקבלים צניחה משמעותית בניבוי התוצאות (AUC=0.58).

**מודל Logistic Regression:** מודל זה הינו מודל סיווג. מטרת המודל הינה לדמות את הקשר בין המשתנים הבלתי תלויים לבין ההסתברות של תוצאה מסוימת השייכת למשתנה היעד. המודל חוזה את ההסתברות שמופע שייך למחלקה מסוימת. בהתבסס על הסתברות זו, המודל מחיל בעזרת סף החלטה את הנקודה לאחת המחלקות. מודל זה הינו מודל פשוט ומתמודד בצורה טובה עם קשרים לינאריים ולא לינאריים, אך חשוב להגיד כי רגרסיה לוגיסטית מניחה קשר לינארי - הנחה זו עלולה לגרום למצב בו לא ניתן להבין אינטראקציות מורכבות בין משתנים. במהלך ביצוע המודל השתמשנו בפונקציית הפסד "L1" (Lasso) אשר משמעותה היא שכל הפיצ'רים "הפחות חשובים" יאופסו ע"י פונקציית ההפסד ולא ישפיעו על התוצאות.
ההיפר-פרמטר שנשנה במודל הוא פרמטר למבדה (1 חלקי C). ההשפעה של שינוי זה באה לידי ביטוי ע"י שינוי גודל "העונש" אותו המודל יקבל, כלומר ככל שפרמטר C קטן יותר אזי "העונש" של המודל יהיה גדול

יותר. משמעות "העונש" היא כאשר המודל ישווה בין התחזית הצפויה לבין התוצאה האמיתית ללא מגבלות וללא מניפולציות, הוא יכול לחפש את המודל הטוב ביותר שיבצע Overfitting וה- C בא לשרת אותנו בדיוק המודל כדי למנוע את ה-Overfitting. התוצאה האופטימלית שמצאנו ללמבדה היא 0.001.
ההשפעה של השינוי על השונות וההטיה ניתנת להסבר בעזרת הגרף המצורף בנספח א' - חלק שלישי המאתר את היחס בין האימון לואלידציה על פרמטר C (1 חלקי למבדה). בגרף ניתן לראות כי ככל ש- C גדל עד 1000 השונות וההטיה יורדות אך החל מנקודה זו והלאה שימוש ב-C גדול יותר גורם להטיה והשונות להתייצב ולכן "עונש" גדול יותר לא ישפר את ביצועי המודל ולכן נבחר בפרמטרים C ולמבדה ע"פ הרשום מעלה. זאת ועוד, ניתן לראות בגרף ה-K-Fold כי המודל מצליח להימנע ממצב של Overfitting ומבנא את סט הטסט המדגימי בצורה קורבה מאוד לנתוני AUC של ה-Fold.

**<u>מודלים מתקדמים - Multi-Layer Perceptron ,Random Forest</u>**

**<u>מודל Multi-Layer Perceptron (ANN):</u>** מודל זה הוא למעשה סוג של רשת מלאכותית, המודל משמש לפתרון בעיות מורכבות - משימות סיווג ומשימות רגרסיה. המודל מורכב ממספר רב של צמתים מחוברים הנקראים נוירונים. המידע הזורם ברשת הנוירונים זורם בכיוון אחד, המידע מגיע לכל שכבה, עובר את התהליך של הפונקציה ויוצא פלט. הפלט של שכבה אחת הוא למעשה הקלט של השכבה הבאה, עד שמגיעים לשכבה האחרונה. המודל יודע ללמוד דפוסים מורכבים, עם זאת המודל דורש כמות גדולה של נתוני אימון ומשאבים חישוביים. בעזרת הגמישות של המודל והיכולת לטפל בריבוי נתונים, הוא נחשב למודל חזק מאוד. בהפעלת המודל השתמשנו במתודת RELU. במהלך בניית המודל החלטנו לבצע את המודל בעזרת 3 שכבות, הראשונה עם 10 נוירונים, השניה עם 10 נוירונים והאחרונה עם 10 נוירונים. ההחלטה לבחור שלוש שכבות עם 10 נוירונים התקבלה לאחר מספר בדיקות על כמות שכבות וכמות נוירונים, בנוסף למדנו במודלים הקודמים כי הבעיה הינה בעיה מורכבת בעלת מספר רב של אופציות ולכן חשבנו כי מודל MLP בעל מספר שכבות עם 10 נוירונים, יתן את המענה לבעיה ע"י התאמה מדויקת לדאטה. הנ"ל כמובן תוך הימנעות מהוספת שכבות מיותרות אשר עלולות להוביל את המודל ל- Overfitting.
ההיפר-פרמטר שבחרנו לשנות במודל הוא אלפא. גודל האלפא הוא קצב הלמידה של המודל, כלומר גודל הצעד אותו יעשה במהלך פעולת ה-Gradient Descent. אלפא קטנה משמעותה היא שהצעדים קטנים וזמן ההגעה למינימום הוא ארוך לעומת אלפא גדולה (צעדים גדולים) שעלולה ליצור Overshooting בצעדים.
האלפא האופטימלית שמצאנו היא 1.
בהקשר השונות וההטיה, כמו שניתן לראות בגרף המצורף בנספח א' - חלק שלישי, ככל שאלפא גדלה ישנה ירידה בשונות עד לנקודה בה האלפא אופטימלית (אלפא =1). לעומת זאת ניתן לראות בגרף ההטיה שישנה עליה קלה עד לאלפא האופטימלית ומשם "שבירה" בגרף למעלה באופן קיצוני המעיד על "שינון" המודל. בנוסף, ניתן לראות את המסקנה בגרף הנוסף (הימיני) המתאר את הדיוק של המודל אל מול האלפא. החל מנקודת האופטימום הדיוק צונח מטה. זאת ועוד, ניתן לראות בגרף K-Fold כי המודל נמצא במצב קל של Overfitting מכיוון שישנו הבדל של 0.13 בין ממוצע נתוני AUC של ה-Fold לבין תוצאות AUC של המדגם המדגמי.

**<u>מודל Random Forest:</u>** מודל זה משמש למשימות סיווג ולמשימות רגרסיה. שיטה זו משלבת מספר עצי החלטה ההופכים יחד ליער. ל- Random Forest יש דיוק גבוה ויכולת להתמודד עם מערכי נתונים גדולים ומורכבים. עם זאת, המודל עלול להיות יקר מבחינה חישובית ומאתגר לפרשנות ביחס לעצי החלטה בודדים. במודל זה, החלטנו לשנות מספר היפר-פרמטרים לטובת דיוק התוצאות. בחרנו שהיער יכלול 200 עצי החלטה, כמות הפיצ'רים המקסימלית תהיה 14 וכמות העלים המקסימלית בכל עץ תהיה האופטימלית מבין מספר אופציות שבחרנו לבדוק. בנוסף בדקנו את קריטריון ביצוע המודל, האם נבצע עם "gini" או "entropy". כמות העלים המקסימלית בכל עץ מסייעת למודל להימנע ממצב של התאמת יתר בין הדוגמאות לבין המבחן. בהיבט שיטת ביצוע המודל, בחרנו לבחון האם לבצע בשיטת "gini" אשר משמעותה היא ההסתברות לסיווג עלה בצורה שגויה בתוך עץ החלטה או לבצע בשיטת "entropy" אשר משמעותה אחידות הסיווגים בעלה מסוים, כלומר ככל שהאנתרופיה גדולה יותר אז יותר חוסר ודאות לגבי סיווג העלה

לתוצאה האמיתית. מצאנו כי נכון לבצע עם כמות מקסימלית של 700 עלים ובשיטת "entropy ". מהצד השני פרמטר כמות העצים ביער (N_estimators) אשר הוחלט להיות 200 עוזר להתמודד עם תופעת ה-Overfitting שעלול להיווצר מהתאמת יתר כאשר מגדילים את מספר העלים בעץ. בנוסף, ניתן לראות בגרף הנמצא בנספח א' - חלק שלישי כי המודל מצליח ללמוד בצורה טובה את הנתונים לפי ההבדל בין נתוני ממוצע AUC ב-Fold אל מול נתון הAUC של הסט המדגמי שבחנו (0.97 אל מול 0.95 בקירוב). כלומר, המודל לומד את הנתונים בצורה טובה, לא משנן ואין פה מצב של Overfitting. לאחר הגעה לתוצאות, ביצענו שיפור למודל ע"י שימוש בכלי חדש (חלק שישי) המשפר את התוצאות ב-0.01.

**<u>תרומה הפיצ'רים השונים להצלחת המודל:</u>**

במהלך הפרויקט רצינו להבין את מידת התרומה של הפיצ'רים למודל שבחרנו. כמו שניתן לראות בנספח א' - חלק שלישי, הפיצ'רים מסודרים בסדר יורד בהיבט חשיבותם. החלטנו כי לא נכון לנתח את המשמעות של כל פיצ'ר אלא להתמקד ב- TOP5 , זאת לטובת הסקת מסקנות איכותית על המודל ולא רק כמותית כמו שמבוצע בשאר הפרויקט. בחרנו רק 5 פיצ'רים מרכזיים כדי להיות מסוגלים להתמקד ולהגיע למסקנות ממוקדות. חמשת הפיצ'רים הינם:

1. Avlength - אורך הממוצע של מחרוזת. פיצ'ר זה מעיד על סיבוכיות הקובץ ואף לכלול קוד המנסה להטות את הגורם המפענח של אמינות הקוד.
2. B - עמודה ללא משמעות ידועה.
3. Imports - כמות הפונקציות שייבאנו. ככל שהקובץ מייבא יותר פונקציות מ"העולם החיצון", כך הוא חשוף הסתברותית יותר לפגעים ולייבא פונקציות המכילות מידע זדוני. בנוסף, האקרים יזהו קבצים בעלי כמויות גדולות של ייבוא לטובת השתלת פונקציה שנראת תמימה אך לבסוף תהיה זדונית, זאת בין מספר רב של פונקציות. בגלל כמות גדולה של פונקציות, בעל הקובץ עלול לפספס את הפונקציה הזדונית במהלך בדיקות הקבצים.
4. Urls - כתובות של אתרי אינטרנט. ייתכן כי כמות גדולה של קישורים/כתובות אינטרנט בתוך הקובץ מצביעות על ניסיון לביצוע "פישינג" או משיכת הלקוח ללחוץ על קישור כאשר הוא לא יודע מה המשמעות הנוספת - אישור מידע אישי/הזרקת קבצים למחשב/קבלת הרשאות גישה למחשב ועוד...
5. File_Type_Prob_Trid - ההסתברות שהקובץ הינו באמת מהסוג שהצהירו עליו. ההיגיון האנושי אומר כי אין סיבה שקובץ מסוים לא יהיה מהסוג שהצהירו עליו ואם הדבר קורה, ישנו חשד אוטומטי ועלינו לשאול את שאלת הספק האם הקובץ זדוני ומה יש להסתיר בסוג הקובץ. לכן ככל שההסתברות לסוג קובץ שונה גבוהה, יש סיכוי גבוה יותר להשתלת קובץ זדוני.

מסקנות - מניתוח של חמשת הפיצ'רים המרכזיים, ניתן ללמוד כי ישנם שני גורמים עיקריים להעלאת החשד לקובץ זדוני - ממשק עם "העולם החיצון" וסיבוכיות הקובץ. הממשק של "העולם החיצון" אצלנו נראה בתצורה של כתובות אינטרנט ופונקציות מיובאות והממשק של הסיבוכיות בא לידי ביטוי באורך הקובץ וחוסר התאמה בין ההצהרה לסוג הקובץ עצמו. בנוסף לאמור לעיל, בגרף אנו רואים כי קידודים בינאריים קיבלו חשיבות נמוכה, ככל הנראה כי ברמה האיכותית קשה להסיק מסקנות על תוצאות בינאריות.

## <u>חלק רביעי - הערכת המודלים</u>

בחלק זה הגענו למסקנה שהמודל הכי מתאים לביצוע המשך ההערכות וביצוע פרדיקציה על סט ה-Test הוא מודל RandomForest. מודל זו הינו המודל בעל הביצועים הטובים ביותר שראינו מבין ארבעת המודלים.

שלב זה של הערכת המודלים חולק למספר לשלבים - בניית Confusion Matrix, ביצוע K-Fold Cross Validation, והצגת פערי הביצועים בין הTrain לValidation.

**בניית Confusion Matrix -** כמו שניתן לראות בנספח א'-חלק רביעי, הטבלה מורכבת מארבעה תאים. ציר הX מסמן את ניבוי המודל ("התוצאות ללקוח") וציר הY מסמן את הLabels (המציאות). תוצאות הטבלה

מחולקות לארבע אפשרויות (התוצאות הינן לפני שיפור המודל ע"י שימוש בכלי החדש – חלק שישי):
1. TP - הדוגמאות שסווגו כקובץ זדוני ואכן הקובץ היה זדוני, 5999 מקרים.
2. FP - הדוגמאות שסווגו כקובץ זדוני אך הקובץ היה אמין בפועל, 619 מקרים.
3. TN - הדוגמאות שסווגו כקובץ אמין ואכן הקובץ היה אמין, 6868 מקרים.
4. FN - הדוגמאות שסווגו כקובץ אמין אך הקובץ היה זדוני, 1514 מקרים.
מהתוצאות המספריות בטבלה ניתן ללמוד כי הרוב המוחלט של הדוגמאות סווגו בצורה נכונה, או סווגו כזדוניות ואכן היו כך או סווגו כאמינות ואכן היו כך. המקרה החמור ביותר הוא FN כי שם המודל חוזה שהקובץ תקין למרות שהוא זדוני.

למעשה, האלכסון הראשי מעיד על איכות המודל - האם ובאיזה אחוז המודל מנבא בצורה נכונה את סוג הקובץ (זדוני/אמין). הנתונים מראים כי אחוז הרגישות הוא 79% (TP/TP+FN) ואחוז הדיוק הוא 90.6% (TP/TP+FP).

**הערכת המודל באמצעות K-Fold Cross Validation -** במהלך ההרצה של K-Fold Cross Validation (מצורף הגרף בנספח א' - חלק רביעי), מצאנו כי כל Fold הראה יציבות עם ציון AUC של 0.97 (בקירוב). מיציבות זו, ניתן להסיק כי המודל אמין ועקבי, זאת ע"פ השוואה לואלידציה.

**פערי ביצועים בין הרצת המודל על ה-Train ועל ה-Validation:**

להערכתנו המודל אותו בחנו מעלה אינו מבצע Overfitting מהסיבה שכיול ההיפר-פרמטרים וחישוב ROC, AUC של סט האימון אל מול סט הואלידציה עם ההיפר-פרמטרים האופטימליים מעיד עד כך המודל אינו "משנן" את הדאטה אלא מצליח ללמוד דפוסי התנהגות של הדאטה לטובת ניבוי התוצאות בהמשך. בנוסף, ניתן לראות שתוצאות הניתוח של סט הואלידציה גבוהות יותר מאשר תוצאות סט האימון. הדבר מעיד באופן מוחלט שהמודל אינו במצב של Overfitting. כדי להגיע למצב זה (ללא Overfitting) השתמשנו בפונקציית GridSearchCV המוצאת את השילוב האופטימלי בין כלל ההיפר-פרמטרים שבעזרתם נבצע את הפרדיקציה.

## חלק חמישי - ביצוע פרדיקציה

במהלך שלב זה הרצנו את המודל על נתוני סט ה-Test. את הנתונים המופיעים בסט ה-Test "העברנו" את אותה הדרך שהעברנו את נתוני ה-Train - בוצע אותו תהליך עיבוד מקדים הכולל סידור הדאטה והתאמתו למודל הרלוונטי (Random Forest). כלל השלבים מפורטים במחברת הקוד. בנוסף, ניתן לראות בנספחים את התפלגות הקבצים הזדוניים והלא זדוניים כאשר סף ההחלטה היה הסתברות 0.6. הפרדיקציה בוצעה לאחר שיפור התוצאות בעזרת כלי שלא נלמד (CalibratedClassifierCV).

## חלק שישי - שימוש בכלי שלא נלמד בקורס (CalibratedClassifierCV)

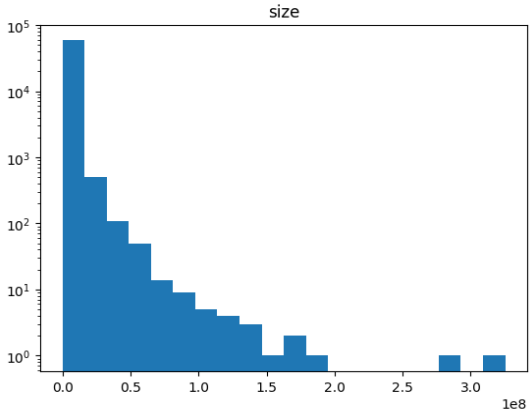
במהלך הפרויקט, החלטנו להשתמש בכלי אשר לא נלמד במהלך הסמסטר לטובת שיפור תוצאות המודל. את הכלי בחרנו לממש בשלב הערכת המודל זאת לאחר הבנה של התוצאות הקיימות ורצון לשפר את התוצאות של המודל הטוב ביותר. הבנו כי אנחנו טועים יותר במצב של FN (השגיאה החמורה ביותר) ורצינו לנסות להקטין למינימום את כמות הטעויות במצב זה.

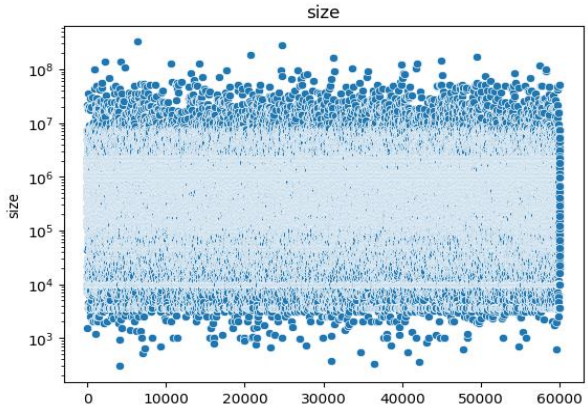
הכלי שבחרנו הוא פונקציית CalibratedClassifierCV. הפונקציה הנ"ל מקבלת את המודל הנבחר (במקרה שלנו - RandomForest) לאחר שאומן ובעזרת Sigmoid אשר מטייבת את ההסתברויות של אי ההתאמות, כלומר לייבלים שסווגו בצורה שגויה בעזרת ההסתברויות של הדוגמאות שסווגו נכון (בחנו את התוצאות בעזרת שימוש ב- Isotonic Regression אל מול Sigmoid, והמתודה שהורידה את ה-FN למינימום הייתה Sigmoid). לאחר מכן, מבוצע אימון למודל מחדש אשר משפר את ההסתברויות שכל לייבל יסווג נכון. כך הצלחנו להקטין את כמות המקרים של FN. ניתן לראות כי הפונקציה משפרת את תוצאות המודל ב-0.01 ומקטינה את כמות ה- FN מ-1514 ל-1385.

# נספח א' - ויזואליזציה

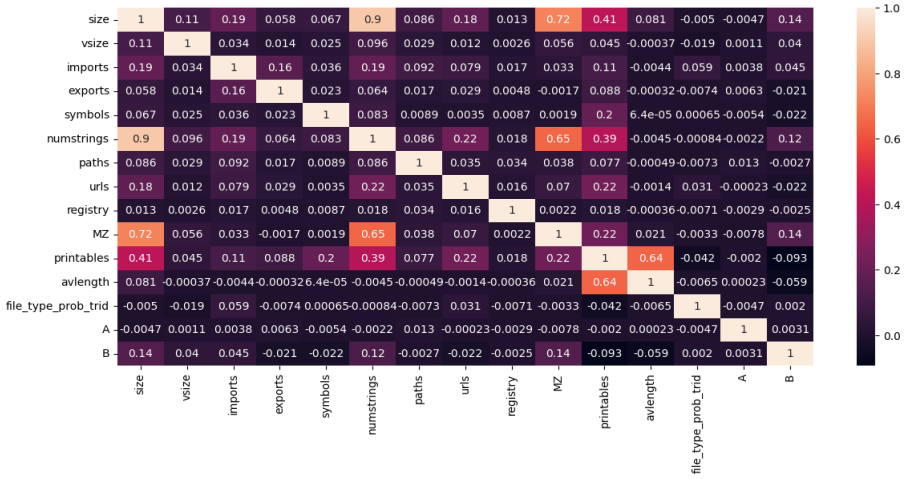
## גרפים חלק ראשון - אקספלורציה:

**גרפים לדוגמא עבור משתנה Size -** גרף עמודות (היסטוגרמה) המתאר את כמות התצפיות (ציר Y) אל מול גודל התיקיה (ציר X). את שני הגרפים הנ"ל ביצענו לכל אחד מהפיצ'רים.

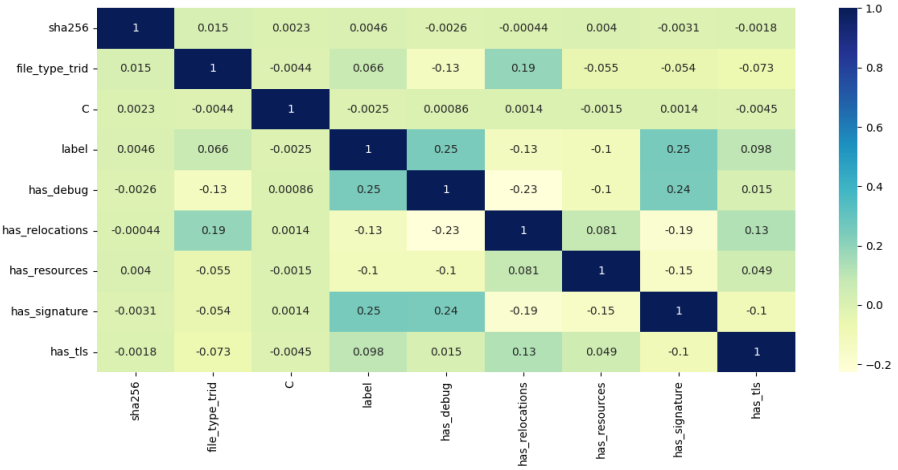



**גרף קורלציה בין המשתנים הנומריים -**

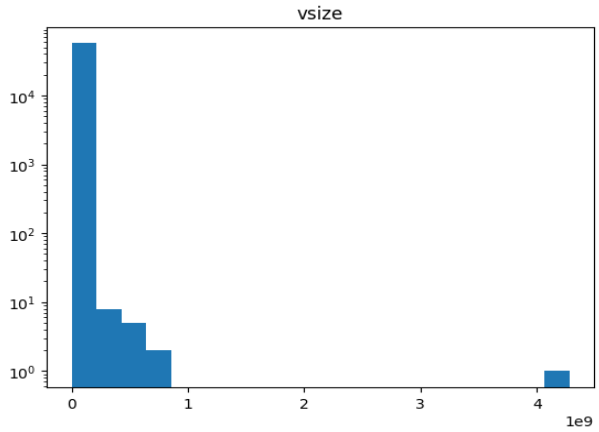


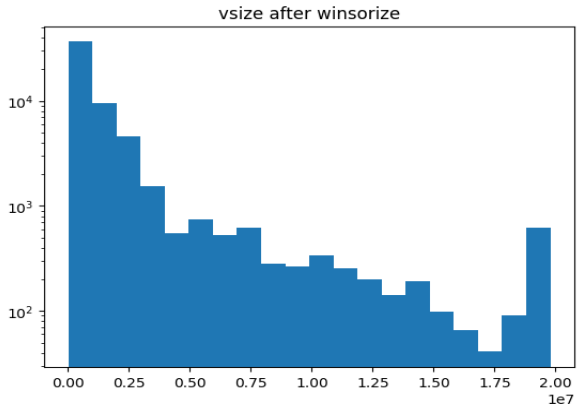

**גרף קורלציה בין המשתנים הקטגוריאליים -**



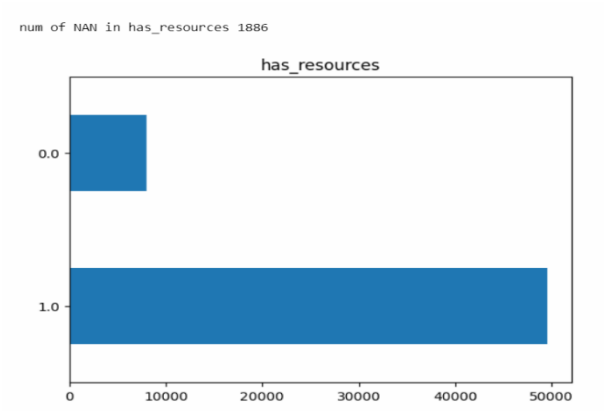
**גרפים חלק שני - עיבוד מקדים:**

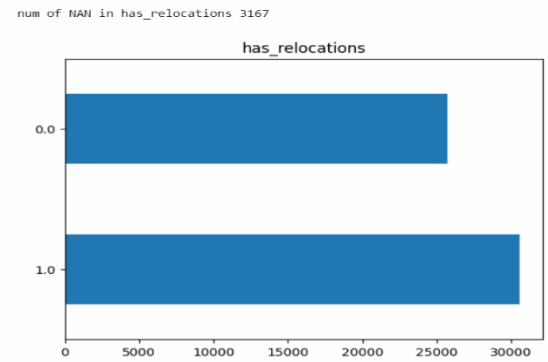
**גרף לדוגמא עבור Vsize -** קירוב הנתונים בוצע בהחלטה רק מחלקו העליון של הדאטה בעקבות התפלגות הנתונים אשר ניתן לראות כי מתרכזים בערכים נמוכים. משמאל נמצא גרף Vsize לפני ביצוע הקירוב ומימין הגרף לאחר ביצוע הקירוב. את ביצוע קירוב הנתונים ביצענו לשאר המשתנים ע"פ המדיניות המפורטת במחברת הקוד.

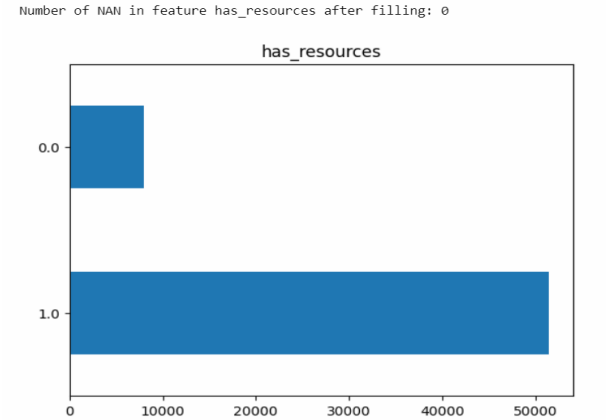


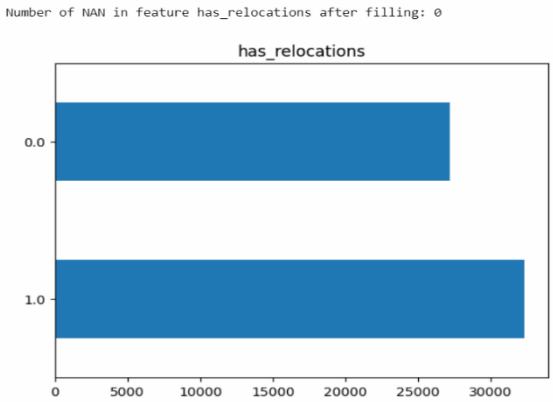




**החלטות מילוי הנתונים החסרים במשתנים קטגוריאליים -** מצורפים שני גרפים - משמאל (גרף Has_Resources) המראה את הסיבה להחלטה של מילוי הנתונים במספר מסוים (במקרה זה 1) שביצענו, ניתן לראות בגרף העליון שרוב התוצאות לפני ההוספה שלנו הן 1 ולכן ההחלטה למלא באחדים. לעומת זאת, בגרפים מימין (גרף Has_relocation) ניתן לראות כי הנתונים מתפלגים באופן יחסית זהה בין 0 ל-1 ולכן מילוי הנתונים בוצע לפי היחסים כלומר שמרנו על היחס לפני ואחרי המילוי של הנתונים.



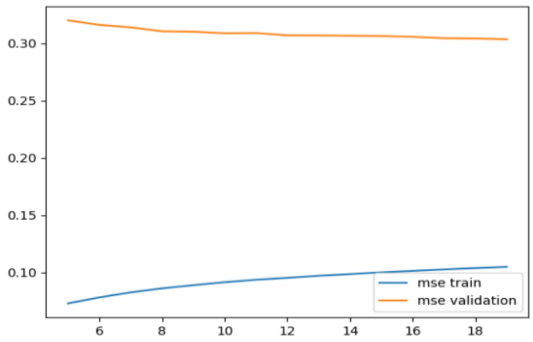



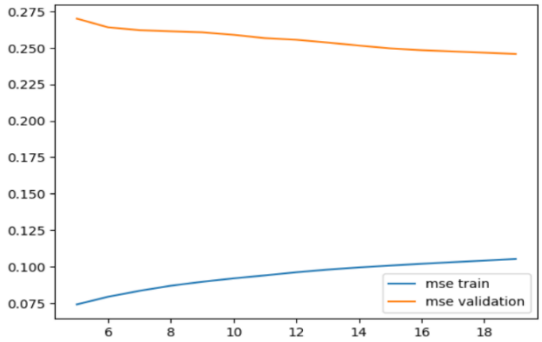



## גרפים חלק שלישי - הרצת המודלים:

**מודל KNN -** מצורפים שני גרפים, מצד שמאל ניתן לראות את הגרף עם שימוש במתודת PCA ובצד ימין ללא שימוש במתודה. ניתן לראות שהתוצאות טובות יותר בגרף ללא שימוש בPCA. הגרפים מתארים את ההשפעה של השונות וההטיה ולמעשה מסייעים לנו בבחירת ההיפר-פרמטר האופטימלי (מספר השכנים).

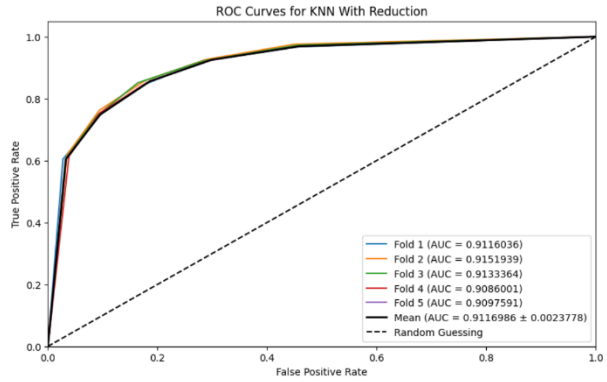



```
The best amount of neighbors is: 5
Score of the best amount of neighbors: 0.569
```

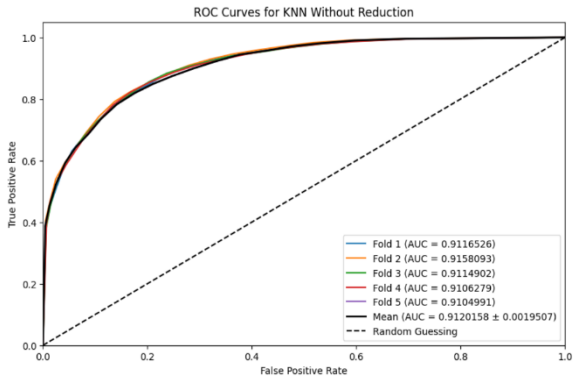



```
The best amount of neighbors is: 19
Score of the best amount of neighbors: 0.6342666666666666
```

בנוסף לגרפים מעלה, מצורף גרף ROC Curves המתאר את האופן שבו משתנים נתוני ה-AUC אל מול ה-Fold הרלוונטי

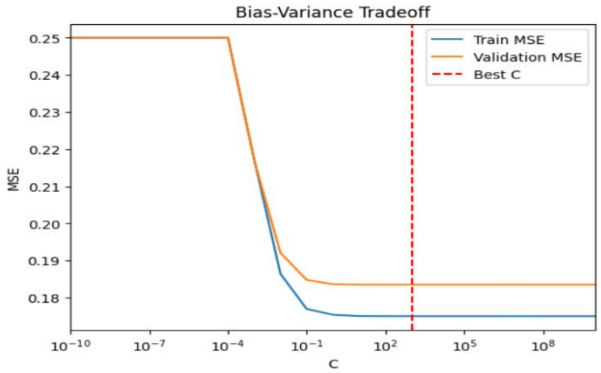


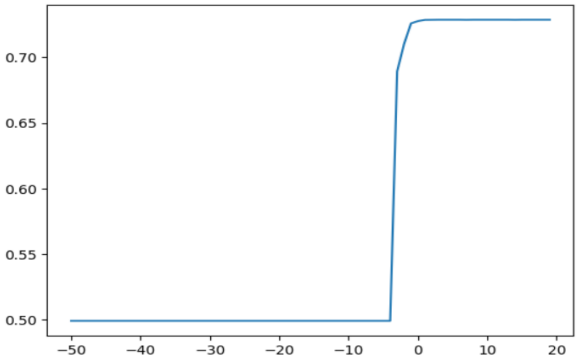

```
KNN validation AUC score with dimension reduction: 0.583958812249587
```




```
KNN Validation AUC without dimension reduction: 0.687586119147629
```

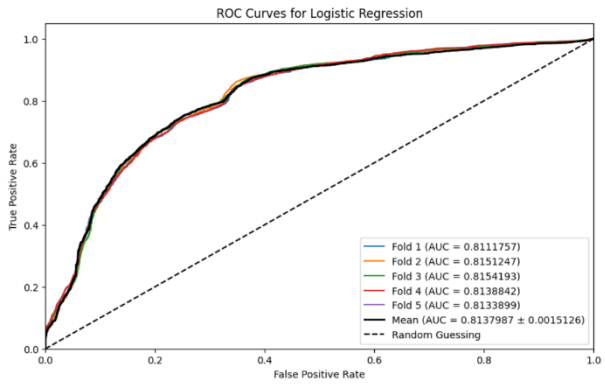
**מודל רגרסיה לוגיסטית -** מצורפים שני גרפים המתארים את ההשפעות על השונות וההטיה ביחס להיפר-פרמטר שבחרנו לשנות (למבדה = 1/C). בגרף מצד שמאל ניתן לראות את השינוי בשונות ובהטיה אל מול שינוי בפרמטר C ובגרף מימין ניתן לראות את שיפור הדיוק של המודל אל מול הגדלת ה-C (1 חלקי למבדה).



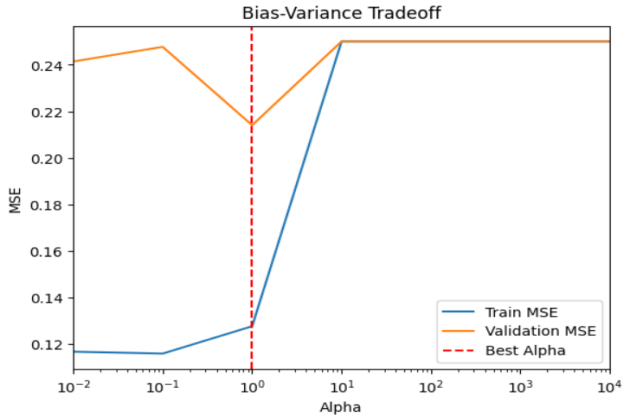
```
The optimum C is: 1000
The optimum Lambda is: 0.001
The score of the validation set with optimum lambda: 0.7286
```

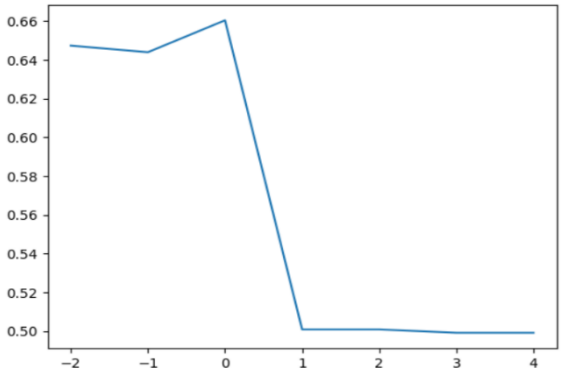
בנוסף לגרפים מעלה, מצורף גרף ROC Curves המתאר את האופן שבו משתנים נתוני ה-AUC אל מול ה-Fold הרלוונטי.




```
Logistic regression validation AUC score: 0.7996598958670649
```

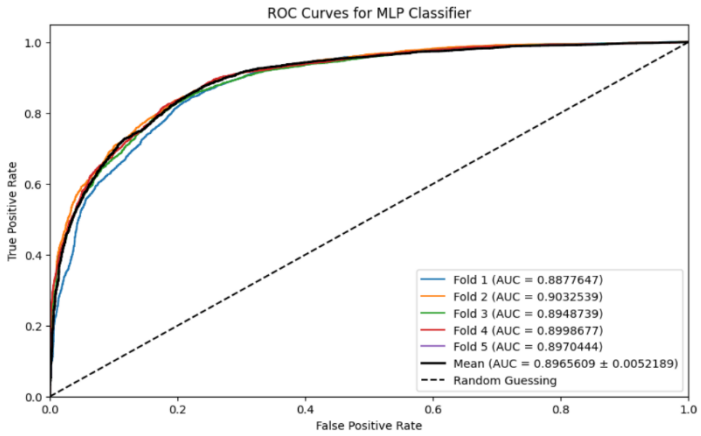
**מודל MLP -** מצורפים שני גרפים, משמאל ניתן לראות את הגרף המתאר את ההשפעה של השונות ההטיה ביחד להיפר-פרמטר שבחרנו לשנות (אלפא) ומצד ימין ניתן לראות את הגרף המתאר את בחירת פרמטר אלפא האופטימלי.



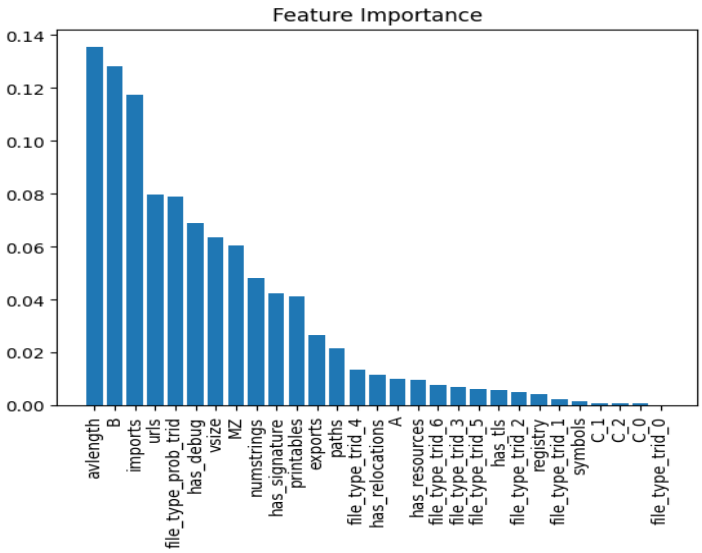
```
The optimal alpha is:  1
The score of the validation set with optimum alpha: 0.6604
```

בנוסף לגרפים מעלה, מצורף גרף ROC Curves המתאר את האופן שבו משתנים נתוני ה-AUC אל מול ה-Fold הרלוונטי



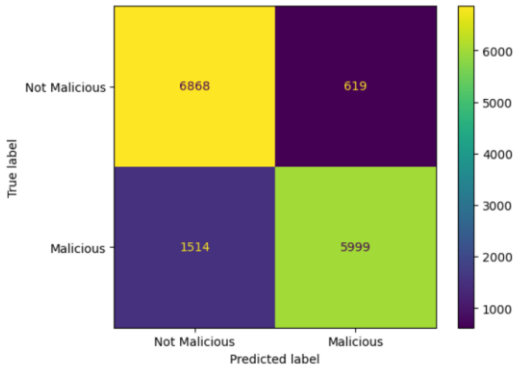

```
MLP AUC validation score: 0.7684573310095812
```

**גרף חשיבות הפיצ'רים -** בגרף המצורף ניתן לראות את סדר החשיבות היורד של הפיצ'רים.



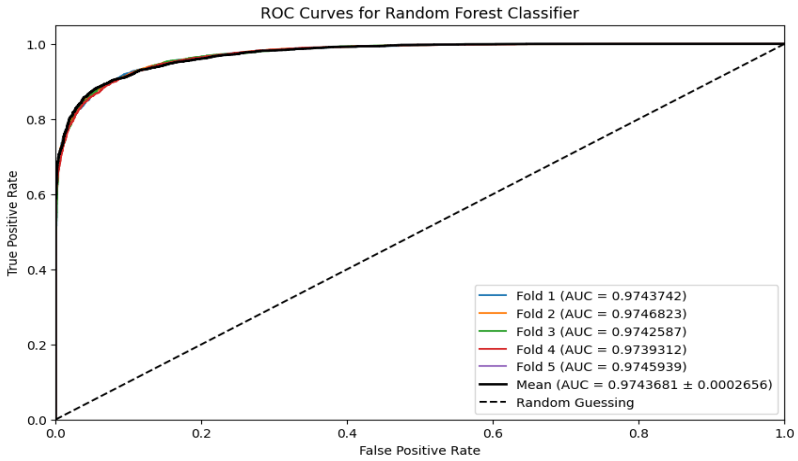

**<u>גרפים חלק רביעי - הערכת המודלים:</u>**

**Confusion Matrix -** בגרף זה ניתן לראות את חלוקת התוצאות ע"פ הפירוט שרשמנו מעלה, לפני שיפור המודל ע"י CalibratedClassifierCV.



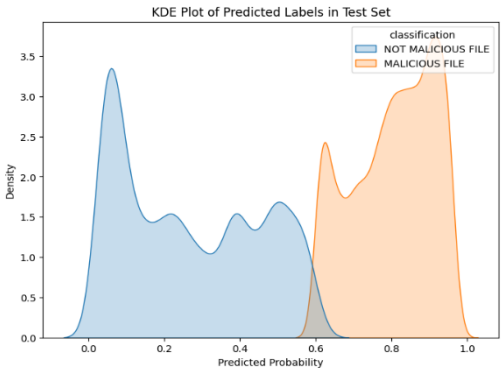

```
Optimal hyper-parameters of the Random Forest model: {'criterion': 'entropy', 'max_features': 14, 'max_leaf_nodes': 700, 'n_estimators': 200}
Random forest validation AUC score: 0.9512902536542732
```

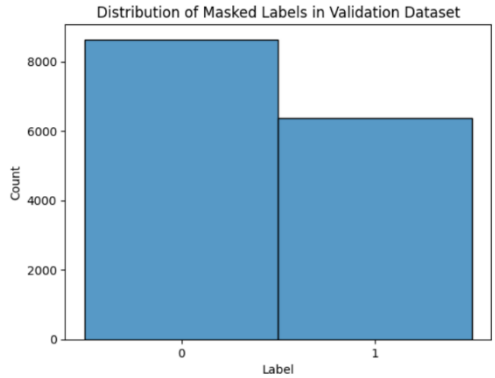
**- K-Fold Cross Validation**



## גרפים חלק חמישי - ביצוע הפרדיקציה:

**גרף KDE –** מצורפים גרפים המתארים את סיווג התוצאות, מימין את סט ה-Validation ומשמאל סט ה-Test, זאת לאחר מציאת סף ההחלטה האופטימלי.

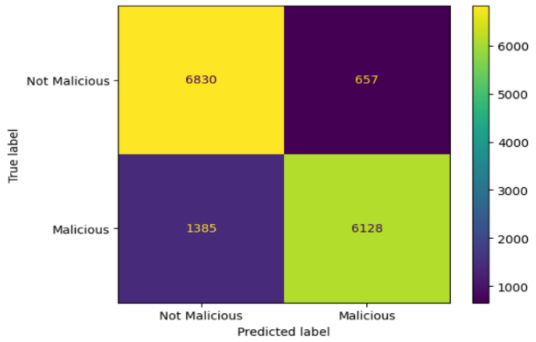


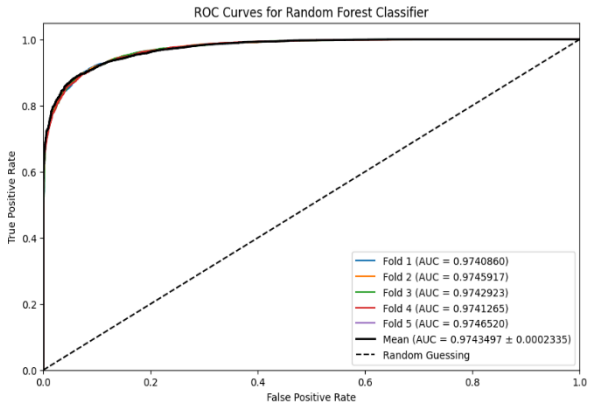

```
Best threshold: 0.6
NOT MALICIOUS FILE    10175
MALICIOUS FILE         7825
```

## גרפים חלק שישי - כלים שלא נלמדו בקורס:

**גרפים של שימוש ב-Sigmoid -** מצורפים הגרפים עם שימוש בפונקציית CalibratedClassifierCV עם שיטת Sigmoid. הגרפים מראים את השיפור ב-FN אל מול המצב ללא שימוש בכלי המתואר בנספח א'-חלק רביעי.




```
Random forest validation AUC score: 0.9522162475474815
```



## נספח ב' - חלוקת אחריות בצוות

הפרויקט בוצע בצורה של עבודה יחד ולא עבודה מקבילית למרות חוסר היעילות בשיטה זו. החלטנו לבצע את העבודה כאשר אנחנו באותו מקום פיזי ועובדים יחד על אותם הדברים בגלל שרוב הניתוחים וההחלטות משפיעים על מהלך הפרויקט. הבנו כי לא ניתן למקבל את השלבים השונים כי כל שלב משפיע על המשך הדרך.

בכל שלב ניסינו לחלק את העבודה שאחד מחברי הצוות כותב את הקוד ובמקביל השותף השני מבצע את ניתוח וכתיבת הדו"ח, כלומר בכל מקטע החלפנו את התפקידים כדי ששנינו נחווה ניסיון בשתי האסכולות.