**תקציר**

בעבודה זו עסקנו בבעיית סיווג בינארי במסגרתה בנינו מודל שמטרתו לחזות האם קובץ מסוים הינו זדוני (1) או לא (0).

המודל אומן באמצעות קובץ דוגמאות, כל דוגמה כוללת מספר פרמטרים (פיצ'רים) אודות הקובץ. לכל קובץ ישנו מזהה חד ערכי הנקרא sha256.

תהליך בניית המודל כלל מספר חלקים, בין היתר – הבנת אופי ומבנה הנתונים, אפיון מימדיות הבעיה, חלוקת הדוגמאות לאימון וולידציה, והתמודדות עם פערים בנתונים (ערכים חריגים, השלמת ערכים חסרים ונרמול)

בחנו מספר מודלים אפשריים לפתרון בעיית הסיווג, בכל מודל בחנו היפר-פרמטרים שונים בהתאם לסוג המודל על מנת להגיע למודל הטוב ביותר.

על מנת לבחור את המודל המדויק ביותר ביצענו הערכה לכל מודל, השוונו בין תוצאות האימון לולידציה ופירשנו את התוצאות בהתאם.

לסיום, בהתבסס על תוצאות הרצת המודלים השונים על הולידציות, בחרנו את המודל שמאפשר לנו למקסם את ה-ACU, כלומר מקסום הפעמים שמודל חזה נכון שהקובץ זדוני ומזעור הפעמים שמודל חזה זדוני והקובץ היה רגיל.

AUC – שיטה למדידת אפקטיביות המודל המציגה על ציר ה-X את ה- False Positive Rate (המודל חזה שהקובץ זדוני אך הקובץ היה לא זדוני) ועל ציר ה-Y את ה- True Positive Rate (המודל חזה שהקובץ זדוני והוא באמת היה זדוני). כל נקודה על הגרף מציגה בהינתן threshold מסוים מהם ערכי ה-X וה-Y של המודל. ה-AUC זה השטח שמתחת לעקומה שהוסברה לעיל.

המודל שחזה את ה-AUC הגבוה ביותר הינו מודל ה-Random Forest, לכן בחרנו בו בתור במודל שאותו נאמן על כלל ה-train data ונבצע את החיזוי לקובץ ה-test.

**ניתוח הנתונים ובניית המודל**

בחלק זה נתאר את כלל הפעולות שביצענו על הנתונים ומידול השאילתה המרכזית, האם קובץ מסוים הינו זדוני או לא. את המודל בנינו באמצעות מניפולציות שונות על קובץ הנתונים train.csv.

מעתה נשתמש במונחים label =1 לקובץ זדוני ו-label = 0 לקובץ רגיל.

חלק ראשון - אקספלורציה של הנתונים

בחלק זה עסקנו בחקר הנתונים, בדקנו מה הם הפיצ'רים, מה כל פיצ'ר מסמן, ואפיינו אותם לפי פרמטרים שונים – סוג ערכי הפיצ'ר (בינארי / קטגוריאלי / מספרי) ובנינו תתי רשימות לכל סוג, מה היא התפלגות הפיצ'רים, הקורלציה בין נתוני הפיצ'ר ל-label, והקורלציה בין כל זוג פיצ'רים. בנוסף אפיינו פערים או נקודות ששווה לשים אליהם לב בנתונים כמו ערכים חסרים, שכיחות ערכים בנתונים וערכים חריגים. עיקר האקספלורציה בוצעה באמצעות ויזואליזציות והדפסות, חילקנו את סוג הויזואליזציות וההדפסות לפי סוג ערכי הפיצ'ר.

במשתנים הנומריים (נספח 3) התמקדנו באפיון ההתפלגות באמצעות הדפסת היסטוגרמה של הנתונים, מציאת ערכים חריגים באמצעות scatter-plot של הפיצ'ר בחלוקה ל-label (באמצעות חלוקה לצבעים), ובאפיון הצפיפות של הפיצ'ר לפי label (גם כן באמצעות חלוקה לצבעים).

במשתנים הקטגוריאליים והבינאריים (נספחים 4,5 ו-6) התמקדנו באפיון ובהתפלגות הדאטה בין הקטגוריות השונות באמצעות גרף עמודות ובחלוקה של כל תת קטגוריה לפי label באמצעות גרף עמודות עם צבעים לפי קטגוריה. על פי קטגוריית file type trid ניתן לראות כי רוב הקבצים שייכים לסוג Win. בעבור C ניתן לראות כי ההתפלגות בחלוקה ל-label שווה, כלומר לא נראה שערך מסוים של A משפיע על ה-label.

הטיפול בסוגיות אלו ואחרות שעלו כחלק מהליך האקספלורציה של הנתונים יוצג תחת החלק השני בעבודה. כעת נציג את המסקנות העיקריות שעלו בחלק זה, כאשר עיקר המסקנות מפורטות במחברת הפרויקט;

ניכר כי כלל הפיצ'רים המספריים פרט לפיצ'רים האנונימיים A ו-B מתפלגים בהתפלגות המזכירה Power law (נספח 2) שהיא התפלגות אקספוננציאלית שלילית, כלומר עיקר צפיפות הנתונים נמצאת בערכים הקטנים, התפלגות זאת מקשה על מציאת ערכים חריגים שכן ערכים חריגים יופיעו רק מצד אחד של הגרף וגם שם יהיה קושי לזהות אותם כי טיב ההסתברות הוא כזה שמאפשר ערכים גדולים מאוד.

ניתן לראות כי משתנה ה-label מתפלג שוויוני לאורך הנתונים, כלומר כמות הקבצים הזדוניים והלא זדוניים שווה. בנוסף, ההתפלגות של המשתנה A בחלוקה ל-label שווה, כלומר לא נראה שערך מסוים של A משפיע על ה-label. לגבי משתנה B נראה כי מתפלג בצורה אקספוננציאלית (הפוך מ-powerlaw), ישנם ערכים מסוימים שנראה כי אינם מתפלגים בהתפלגות מסוימת.

ניתן לראות שלכל הפיצ'רים פרט לארבעה ישנם בין 1739 – 3751 ערכים חסרים, שזה בין 0.03%-0.06%, ובנוסף אין sample שיש לו יותר מ-7 ערכים חסרים. (נספח 7)

ממטריצת הקורלציות (נספח 8) ניתן לראות כי ישנה קורלציה חזקה בין הערכים MZ, size, numstrings ובין הערכים printables ו-avleangth, אך מרבית הנתונים אינם תלויים במידה גבוהה. בנוסף ניתן לראות כי המשתנה שמופיע בצורה חיובית הרבה ביותר על ה-label הינו B. בנוסף ניתן לראות כי ל-has debug ו-has signature יש קורלציה שלילית חזקה עם ה-lbelץ קורלציה זו תואמת את המידע שנמצא בעת מחקר הפיצ'רים, ראה טבלה.

חלק שני – עיבוד מקדים

בשלב זה התמודדנו גם עם בעיית הערכים החסרים, במשתנים הבינאריים בחרנו את הערך השכיח ביותר בכל פיצ'ר והשלמנו בעזרתו את כל הערכים החסרים. במשתנים המספריים בחרנו את הממוצע של כל פיצ'ר והשלמנו בעזרתו את כל הערכים החסרים.

בחלק זה ביצענו עיבוד של הנתונים הגולמיים לטובת טיוב המודל. בחלק הראשון ראינו כי פרמטרים מספריים רבים מתנהגים תחת צפיפות לוגריתמית שלילית, לכן בחרנו להפעיל טרנספורמציה מסוג log(x+1) על כלל הערכים המספריים פרט ל-A ו-B (ניתן לראות בנספח 9 כי התפלגותם שונה מההתפלגות המתוארת) על מנת להגיע להתפלגות שוויונית יותר המזכירה התפלגות גאוסיאנית, הוספנו את הערכים החדשים לdata set בנוסף לערכים המקוריים על מנת לאפשר למודל התמודדות לא לינארית (כלומר, לשלב גם את הערכים המקוריים עם הערכים החדשים).

כעת נבדוק האם קיימים ערכים חריגים בדאטה, ניסינו לבדוק האם הורדת תצפיות בעלות הסתברות נמוכה (קטנה מאחוז אחד) תעזור לנו לקבלת תוצאות ולידציה טובות יותרת זאת תחת ההנחה שתצפיות שההסתברות לקבל אותן נמוכות מאחוז הנן תצפיות חריגות. לאחר ההשוואה גילינו כי תוצאות הולידציה דומות עד כדי זהות. סיבה מרכזית יכולה להיות בגלל שהדאטה שלנו לא מתפלג נורמלית. החלטנו לא להוריד תצפיות חריגות בדאטה מכיוון שאיננו יכולים לאפיין מה היא תצפית חריגה, והדרכים שניסינו רק פגעו או לא שינו את תוצאות המודל.

בשלב זה בחרנו גם את דרך ההתמודדות שלנו עם המשתנים הקטגוריאליים, נזכיר כי תת הקטגוריות של הפיצ'ר C אינן משפיעות על ה-label כיוון שההתפלגות של ה-label זהה בין ערכיה השונים, לכן הוחלט להסירו בהמשך. כעת, נשארנו עם משתנה קטגוריאלי יחיד file\_type\_trid שמכיל 89 ערכים (נספח 10), הוספת כלל הערכים בתור משתני דאמי באמצעות One-Hot-Encoder תוביל ל-complexity גבוה דבר שיגרע מאיכות המודל, לכן בחרנו לבדוק את ההתפלגות הפנימית של הערכים לפי תת קטגוריה וראינו שהצפיפות העיקרית בפיצ'ר הינה במספר ערכים מצומצם, לכן בחרנו לקחת 2% מהנתונים ולאחד אותם תחת קטגוריה חדשה other.

פעולה זו גרמה למספר תתי הקטגוריות להצטמצם ל-11 קטגוריות בלבד אותן הכנסנו כעמודות נפרדות ל-dataset עם ערכים בינאריים האם תצפית מסוימת שייכת לתת הקטגוריה או לא. (נספח 11)

בדקנו את התפלגות ה-label בין ערכי file\_type\_triod לאחר האגרגציה (נספח 12), חשיבה מאחורי פעולה זו הייתה שאם התפלגות ה-label הייתה דומה בין הערכים השונים ייתכן והיינו מסירים פיצ'ר זה. התפלגויות יצאו שונות ולכן השארנו את הערכים.

את הערכים הסופיים הוספנו ל-data באמצעות One Hot Encoder על מנת להציגם כמשתני דאמיז בעלי משקל שווה.

כעת נבחן אילו פיצ'רים ברצוננו להוריד כבר בשלב זה מתוך הבנה שהם לא יתרמו לנו למודל, בחרנו להוריד את A ו-C מכיוון שההתפלגות שלהן בהינתן label שווה ל-1 ובהינתן label שווה ל-0 דומה עד כדי זהה, כלומר הפיצ'ר לא תורם להבנה האם תצפית מסוימת הינה זדונית או לא. המשתנים log\_symbols ו-symbols – ראינו כי תרומתם למודלים מוגבלת מאוד ולכן הוחלט להורידם. הוחלט להוריד אף את משתנה הדאמי category other שמורכב מהרבה משתני דאמיז מקטגורית האב file type trid ואינו אינדיקטיבי לניבוי ה-label.

תחילה נפצל את סט הנתונים ל-X ו-Y (הנתונים הגולמיים וה-label בהתאמה), נשים לב כי על מנת להוריד מימדים בצורה נכונה ויעילה עלינו לנרמל את הנתונים שברשותנו מאחר ולנתונים יש ערכים בסדרי גודל שונים, לשם כך נשתמש בפונקציה StandardScaler שמטרתה לנרמל את הנתונים לקבלת ממוצע 0 ושונות 1. כעת נשים לב כי כמות הפיצ'רים שברשותנו הינה 39 (לא כולל ה-label), נפעל לנסות ולצמצם כמות זו ע"מ להימנע מבעיית שונות שעלולה להיווצר בעקבות ה-complexity הגבוה.

השיטה הראשונה שבחרנו להוריד עימה מימדים הינה שיטת PCA, בשיטה זו מימדיות הבעיה ירדה מ-39 ל-36, תוך כדי שהבטחנו שמירה של 99% מהשונות, לאחר מכן בחנו את הורדת מימדיות הבעיה תוך שימוש במתודת forward selection (נספח 13). השימוש במתודה זו הובילה לכך שנותרנו עם 35 מיימדים, כאשר בחנו את הקורלציות (נספח 14) ביניהם ראינו כי קיימת קורלציה חיובית חזקה בין הפיצ'רים הבאים 'log\_numstrings' and 'log\_printables' (0.89), log\_urls and has\_signature (0.74), size and MZ (0.7), B and log\_avlength (-0.6). לאחר בדיקה של השפעת הפיצ'רים השונים על המודל הוחלט כי מבין כל הקורלציות הגבוהות שצוינו להוריד את המשתנה log\_numstrings מכיוון ששאר הפיצ'רים תורמים במידה רבה לתוצאות המודל.

לבסוף, מודל הכולל את הורדת המימדים נותן תוצאות טובות יותר בעקבות הפחתת ה-complexity שתורמת להורדת שונות המודל ולאיזון בין ה-bias ל-variance.

חלק שלישי – הרצת המודלים

לפני הרצת המודלים חילקנו את הdata הסופי שלנו ל-train ו-validation. על ה-train נבצע cross validation ולבסוף נריץ את המודל גם על ה-validation. (נספחים 15-21)

המודלים שבחרנו להריץ הם: רגרסיה לוגיסטית, KNN, Decision Tree, Random Forest, ו-XGBOOST, את המידול ביצענו במספר שלבים: תחילה ביצענו בדיקה למציאת ההיפר-פרטמטרים האופטימליים באמצעות שימוש במודל GridSearchCV (פרט לרגרסיה לוגיסטית שבדקנו ידנית C אופטימלי) עם ערכים אופציונליים. לאחר מכן נגדיר את המודל עם ההיפר-פרמטרים האופטימליים שנמצאו.את המודל הערכנו באמצעות K cross fold validation, תוך בניית גרף roc curve וחישוב ה-AUC של ה-cross validation ושל ה-validation בנפרד.

|  |  |  |
| --- | --- | --- |
| מודל | היפר פרמטרים אופציונליים | היפר פרמטרים שנבחרו |
| Logistic Regression | C: [1000,100,10,1,0.1,0.01,0.001,  0.0001,0.00001,0.000001, 0.0000001, 0.00000001, 0.000000001, 0.0000000001] | C: 1   * Lambda: 1 |
| KNN | n neighbors: [200,120,100,80,60,50,20,10]  weights: ['uniform', 'distance'] | n neighbors: 20  weights: distance |
| Decision Tree | criterion: ['gini', 'entropy']  max depth: [3, 5, 10, None]  min samples split: [2, 5, 10]  min samples leaf: [2, 5,10] | criterion: 'entropy'  max depth: None  min samples leaf: 10  min samples split: 5 |
| Random Forest | criterion: ['gini', 'entropy']  n estimators: [20,50]  max depth: [5, 10,20]  min samples split: [2, 5, 10]  min samples leaf: [1, 2, 5,10] | criterion: 'gini'  max depth: 20  min samples leaf: 1  min samples split: 2  n estimators: 50 |
| XGBOOST | Max depth: [10,20]  learning rate: [0.1, 0.01]  n estimators: [20,50]  gamma: [0, 0.1, 0.2]  subsample: [0.5,1] | gamma: 0.2  learning rate: 0.1  max depth: 20  n estimators: 50  subsample: 0.5 |

חלק רביעי – הערכת המודלים

|  |  |  |  |
| --- | --- | --- | --- |
| מודל | Train mean score | Validation mean score | השערות לגבי הפערים |
| Logistic regression | 0.856 | 0.79 | 1. המודל מניח שיש מעט מולטיקולינאריות בין משתנים, אך לפי המפת הקורלציה שביצענו לאחר הורדת המימדים, נראה שיש משתנים שעדיין יש בינהם תלות ולכן לא מתקיימת ההנחה של המודל והדבר עשוי לפגוע בביצועי המודל. |
| KNN | 0.964 | 0.9 | 1. ייתכן כי מספר השכנים שנבחר קטן ומוביל לoverfitting.  2. העובדה שלא הסרנו ערכים חריגים יכולה לגרום לכך שתצפיות מסויימות יסווגו לא נכון בעקבות קרבתם לתצפית חריגה |
| Decision Tree | 0.937 | 0.88 | עבור 3 מתוך 4 מההיפר פרמטרים שנבחרו קיימים ערכים שעלולים להוביל לoverfitting (להרחבה ראה מחברת) |
| Random Forest | 0.98 | 0.92 | 1. עבור 4 מתוך 5 מההיפר פרמטרים שנבחרו קיימים ערכים שעלולים להוביל לoverfitting (להרחבה ראה מחברת)  2. עבור מודל זה, ככל שיש יותר תצפיות כך היכולת סיווג שלו עולה. דבר זה יכול להסביר את הפער בין הביצועים על נתוני האימון לעומת הביצועים בסט הולידציה (בסט האימון הרבה יותר תצפיות מסט הולדיציה). |
| XGBOOST | 0.96 | 0.88 | 1. עבור 4 מתוך 5 מההיפר פרמטרים שנבחרו קיימים ערכים שעלולים להוביל לoverfitting (להרחבה ראה מחברת)  2. עבור מודל זה, ככל שיש יותר תצפיות כך היכולת סיווג שלו עולה. דבר זה יכול להסביר את הפער בין הביצועים על נתוני האימון לעומת הביצועים בסט הולידציה (בסט האימון הרבה יותר תצפיות מסט הולדיציה). |

כפי שניתן לראות בנספחים ובמחברת הפרויקט Random Forest קיבל הן את ציון ה-train הגבוה ביותר והן את ציון ה-validation הגבוה ביותר, בנוסף, בעל הפער הקטן ביותר בין ציון הביצועים על מדגם האימון לבין מדגם הולידציה בהשוואה למודלים האחרים שנבדקו, לכן בחרנו לחזות את בעיית הסיווג תוך שימוש במודל זה.

מנספח 18 ג, ניתן לראות כי הפיצ'רים שמשפיעים במידה הגבוהה ביותר על החיזוי (תרומתם גדולה מ-5%) הינם log\_avlength, B, Imports, log imports, file type prob trid, log urls ו-size.

נספחים

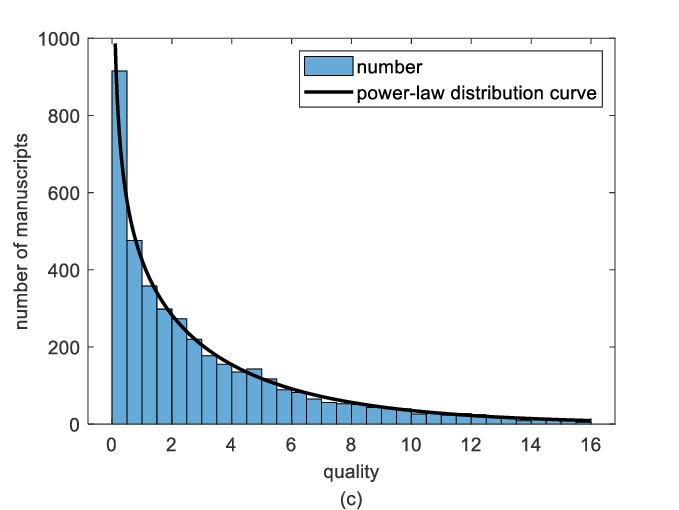
1. חלוקת עבודה –

רונית – אקספלורציה, אגרגציה במשתנים הקטגוריאלים, XGBOOST, random forest, pipline.

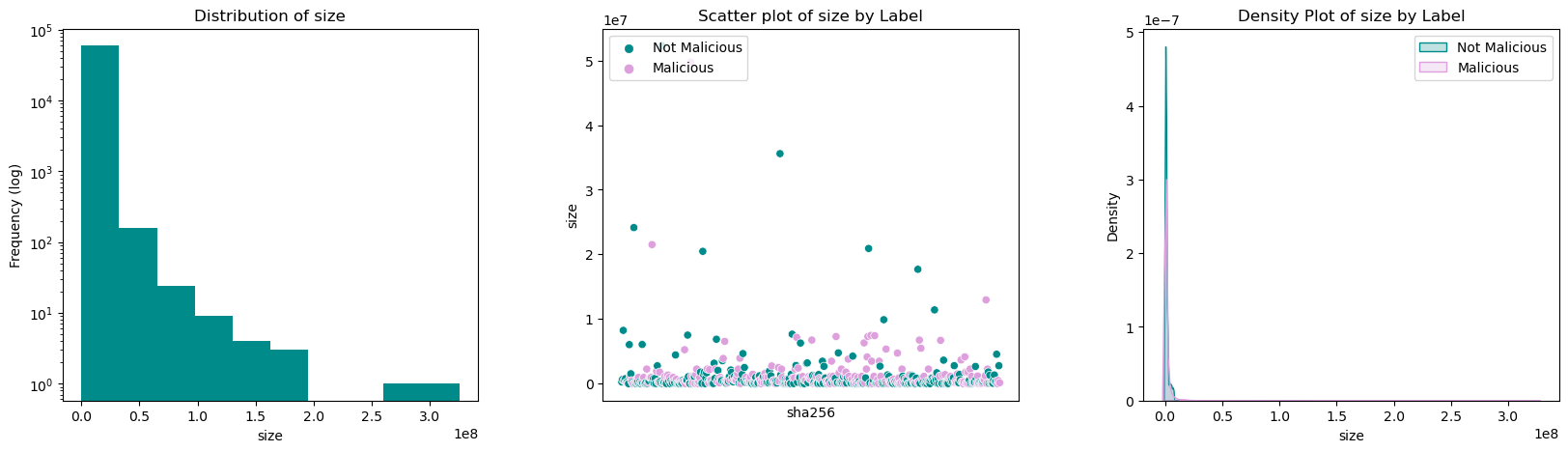
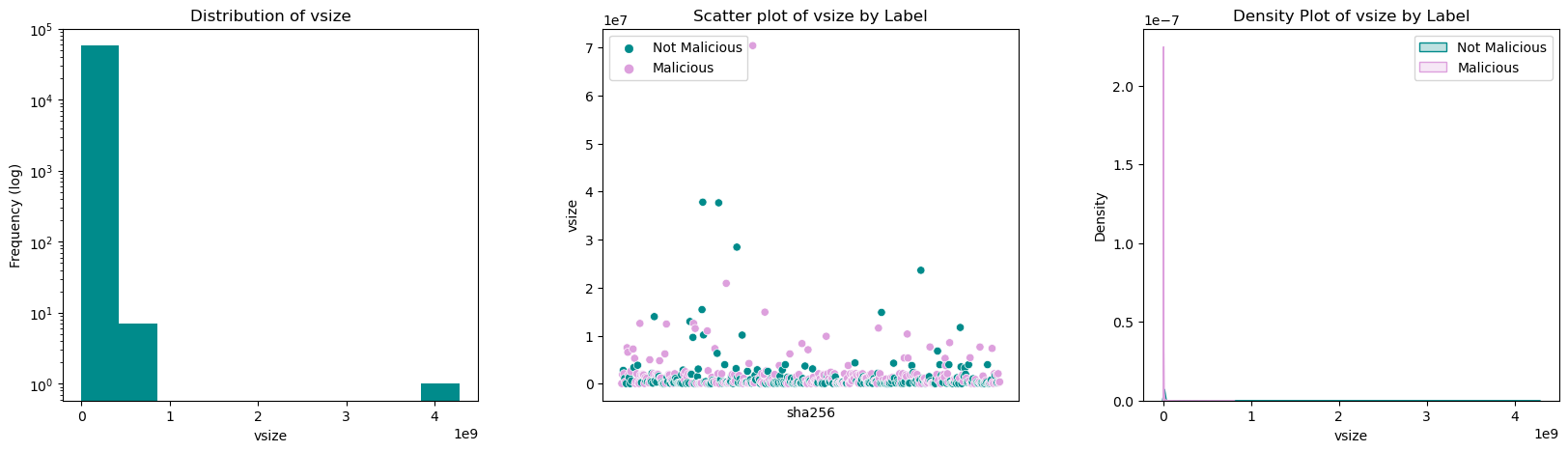
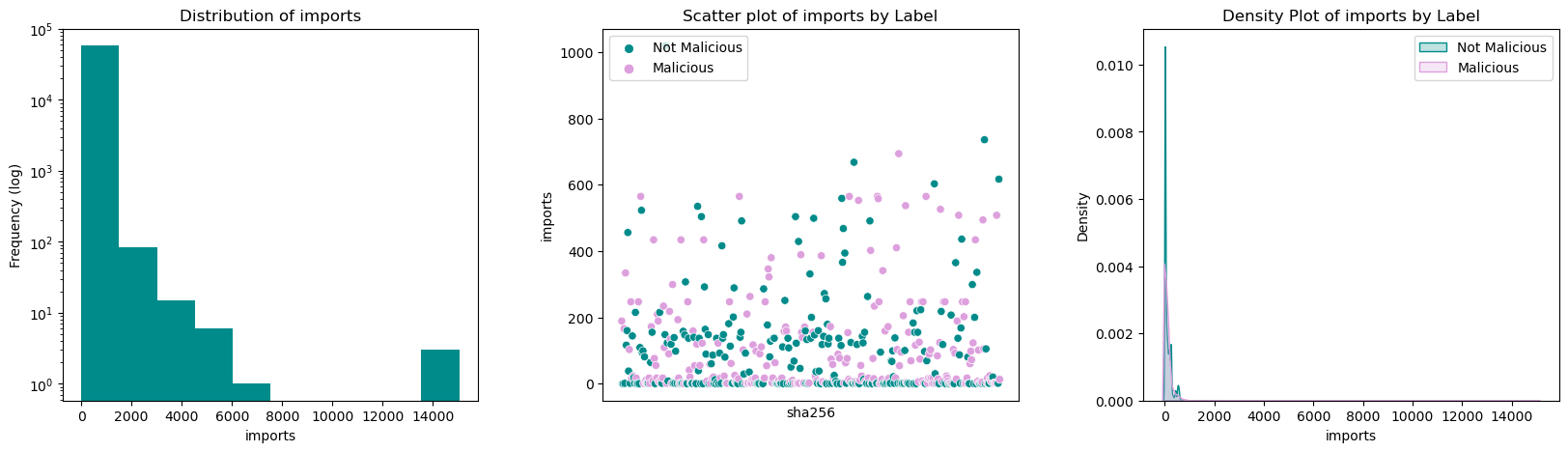
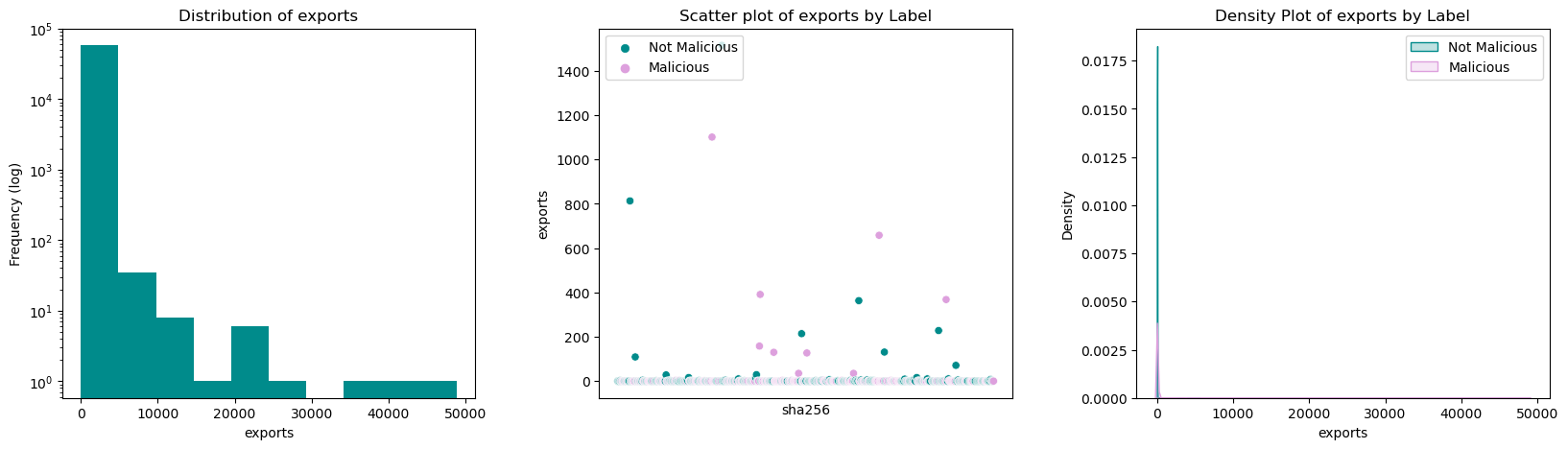
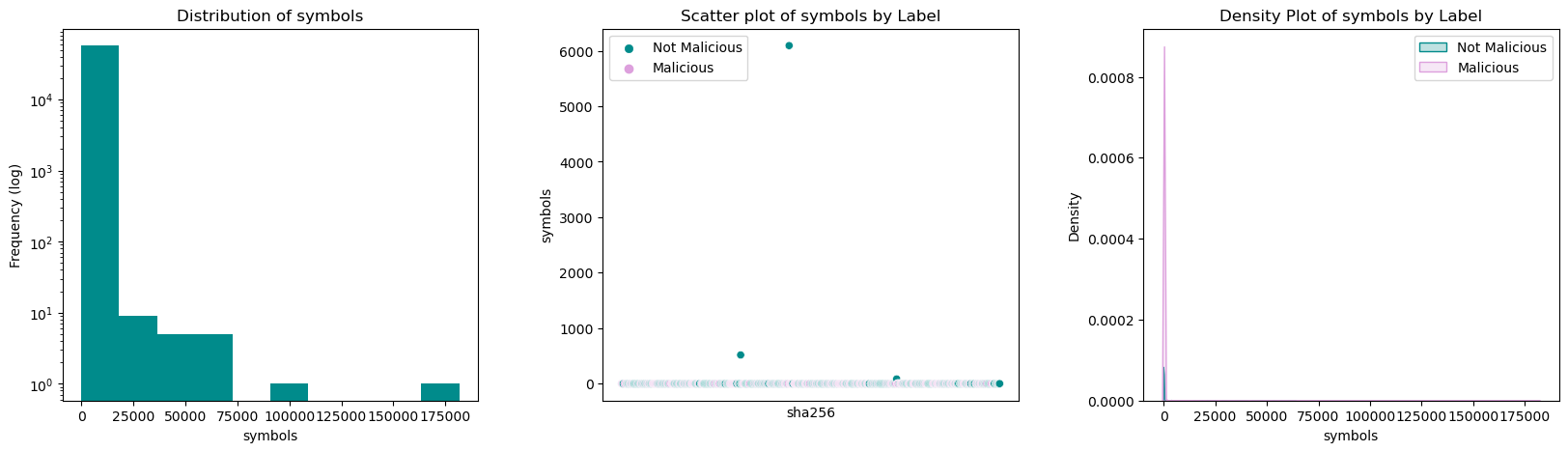
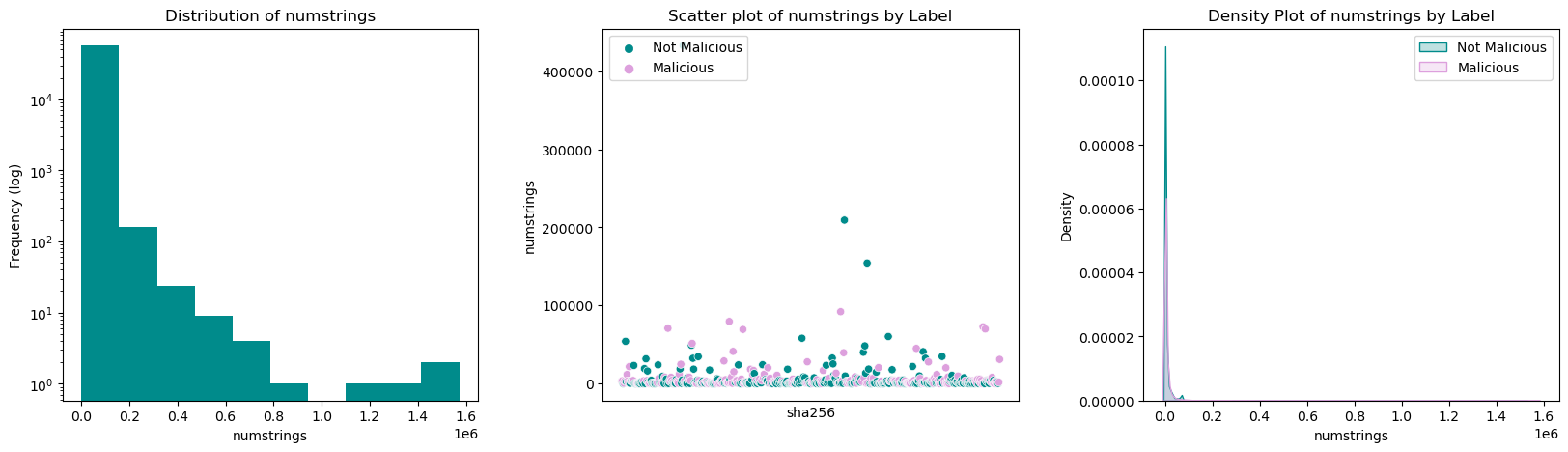
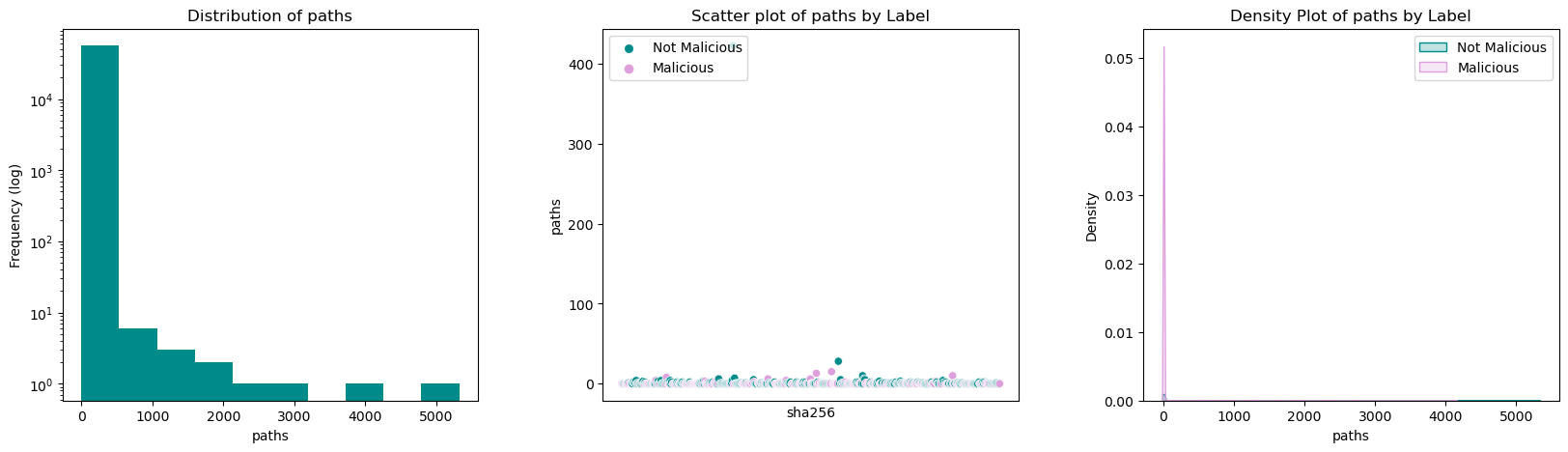
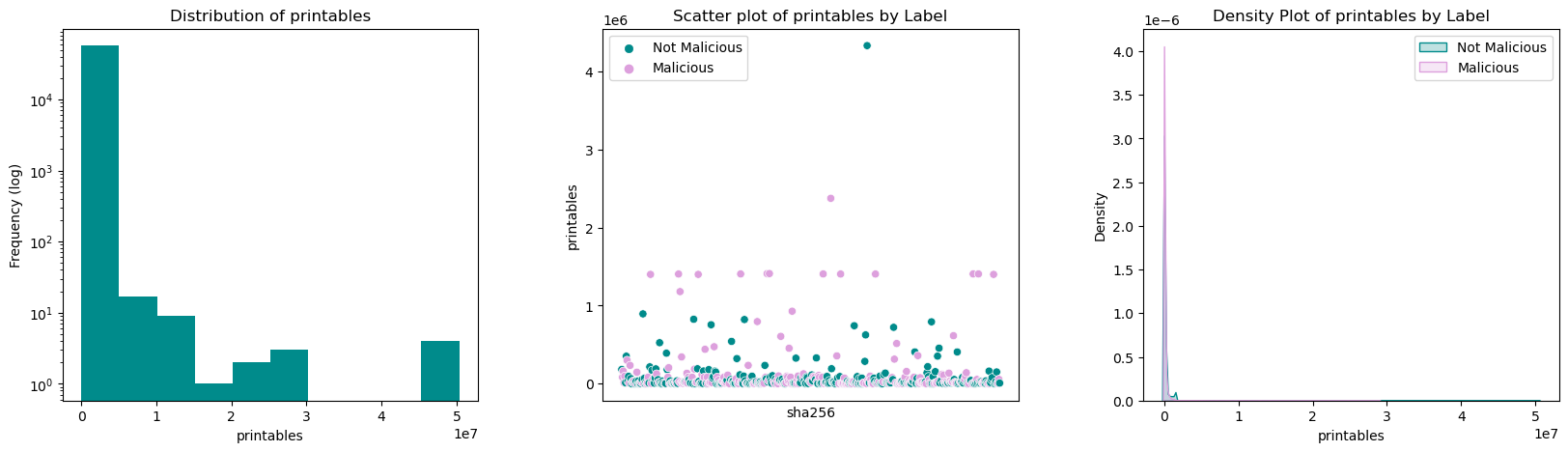
עמית – מחיקת פיצ'רים והורדת מימדים, מילוי ערכים חסרים, התמודדות עם ערכים חריגים, הוספת פיצ'רים חדשים, נרמול, KNN, logistic regression, decision tree

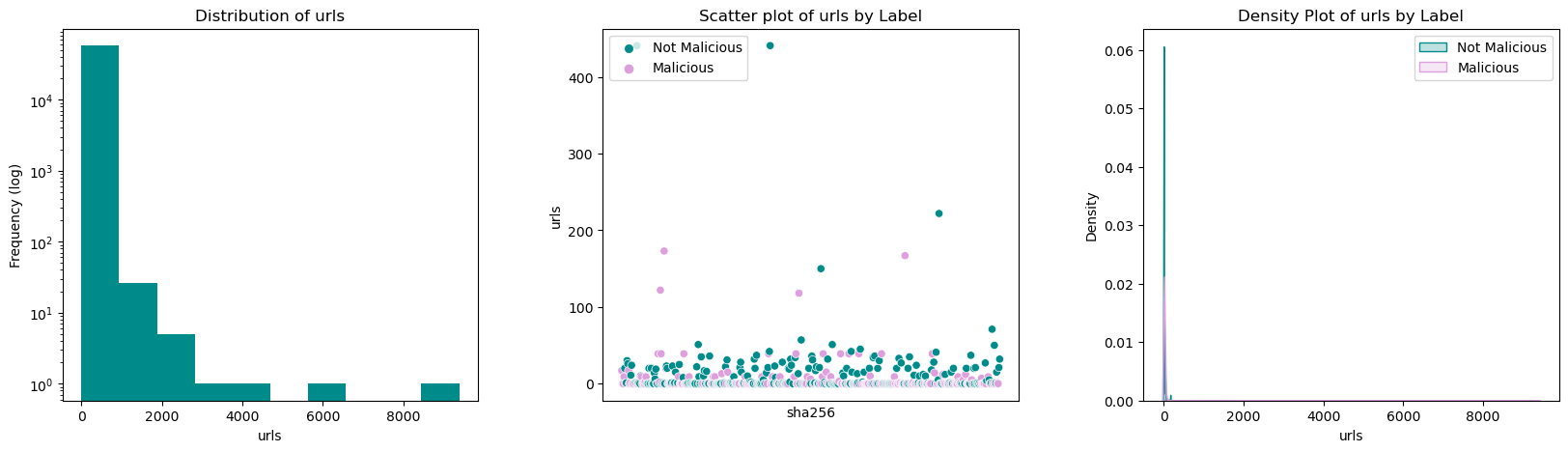
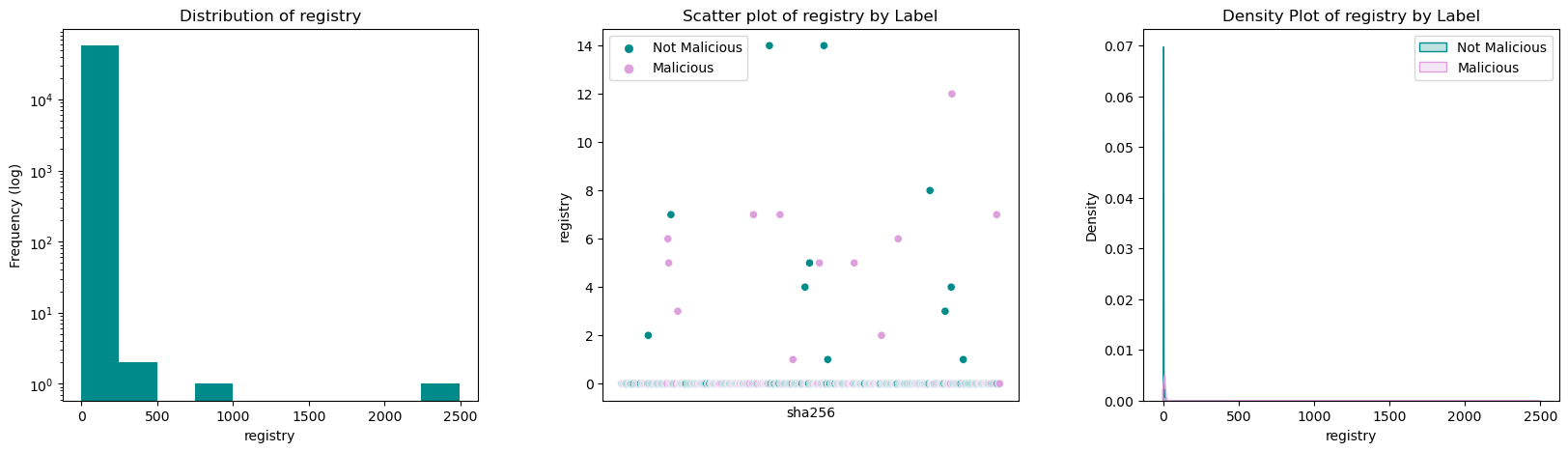
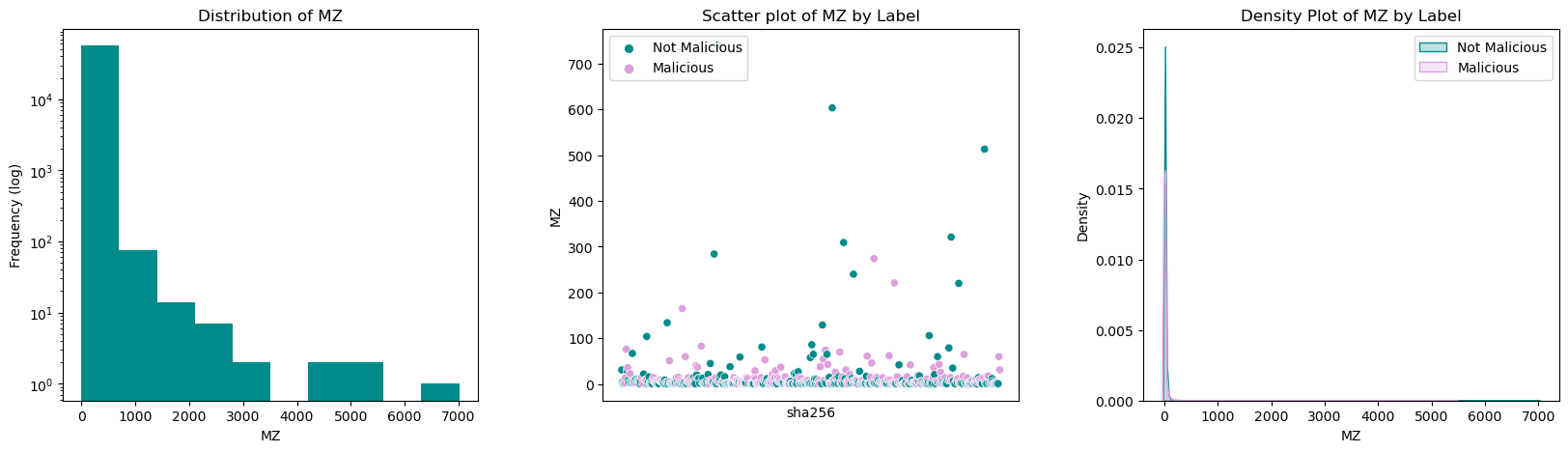
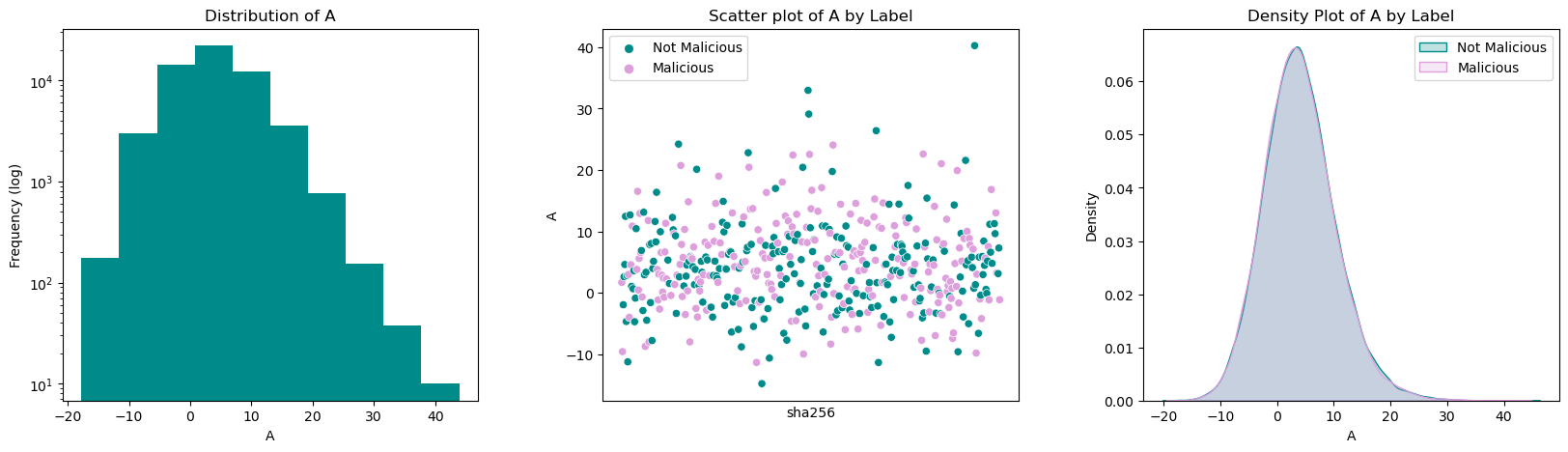
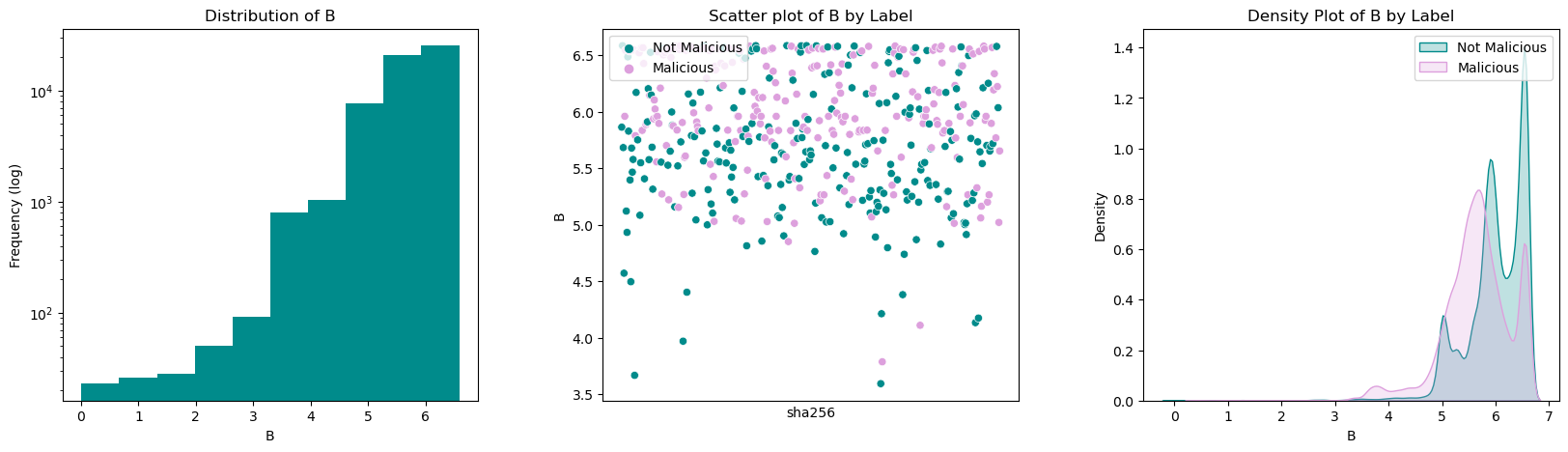
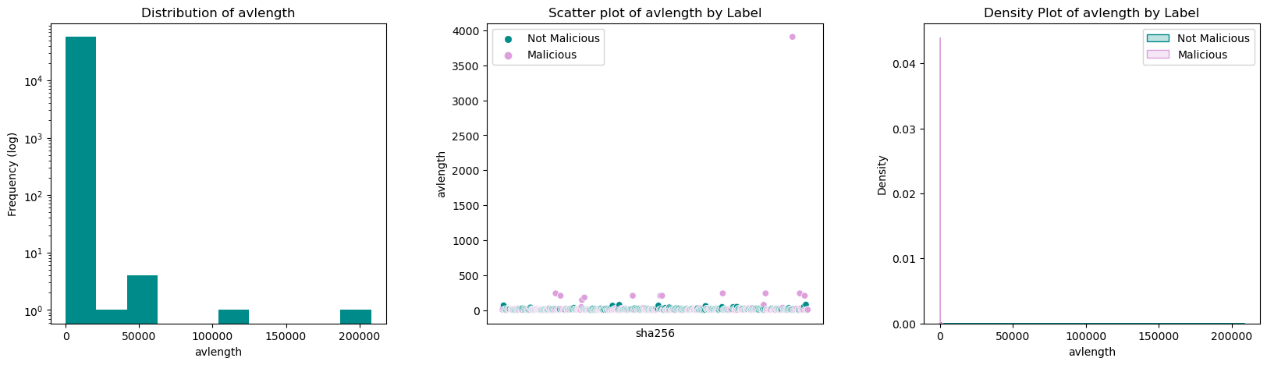
מלבד החלקים המצוינים לעיל נעשתה עבודה משותפת על כתיבת הדוח, כתיבת הערות לקוד, יצירת הפלטים וניתוחם וניתוח תוצאות המודלים.

1. התפלגות power law –

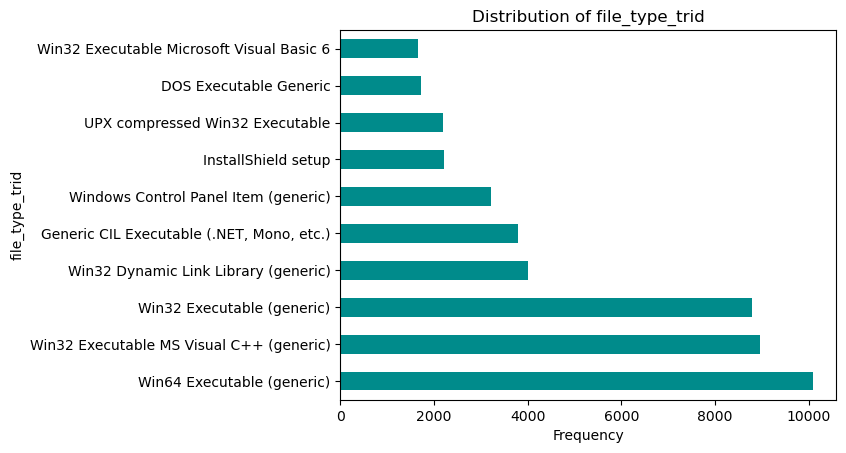


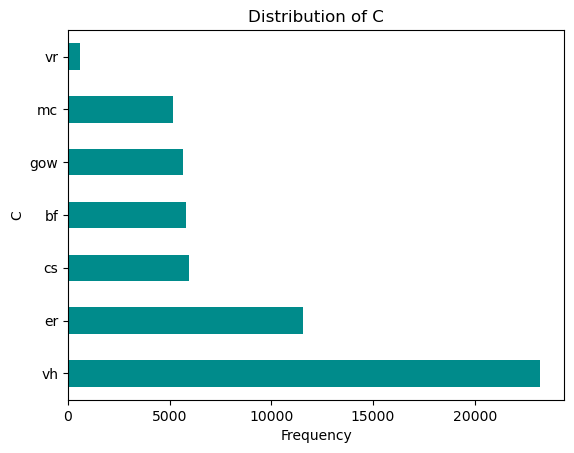
1. תרשימים של הפיצ'רים המספריים –

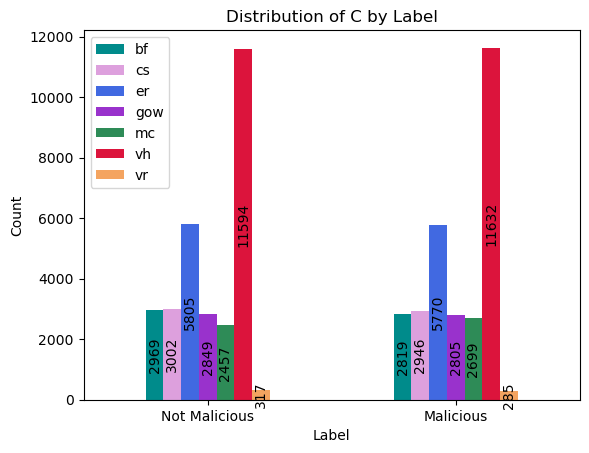
       

1. תרשימים של הפיצ'רים הקטגוריאליים –

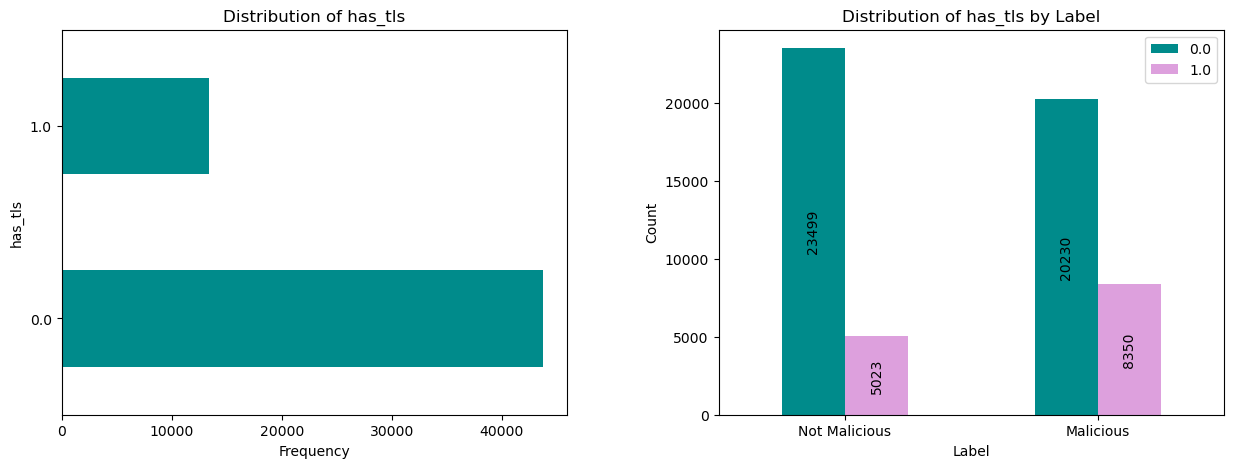


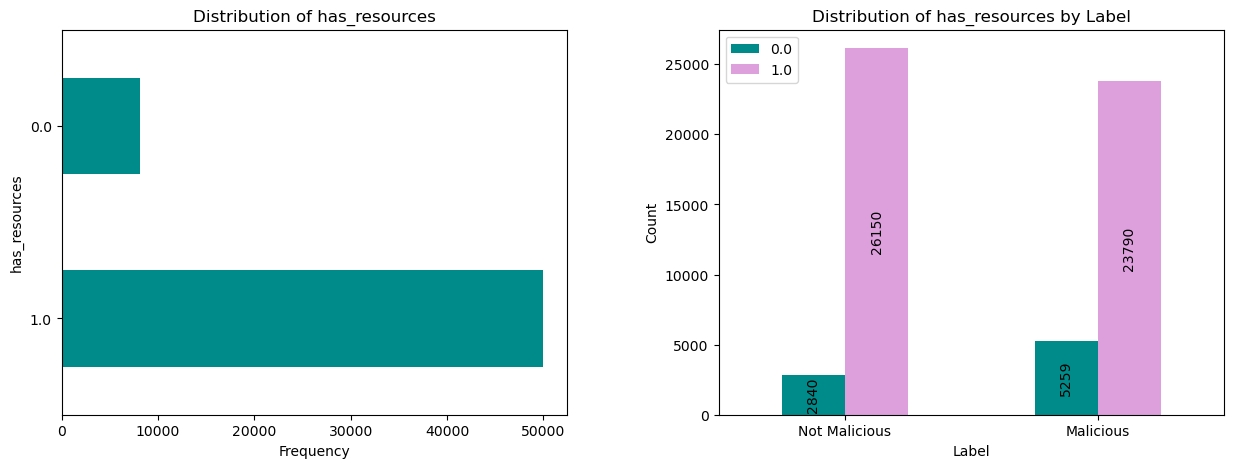
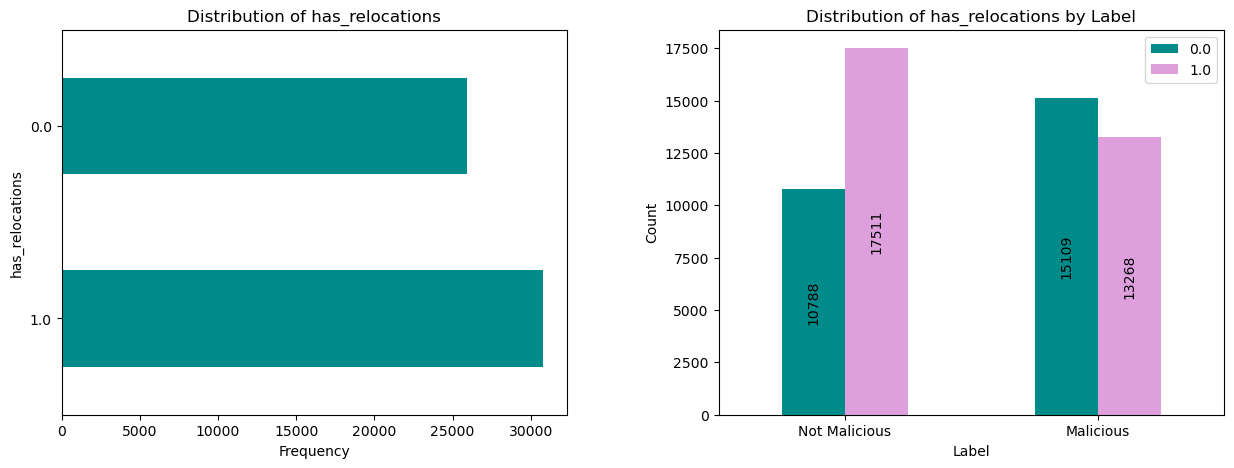
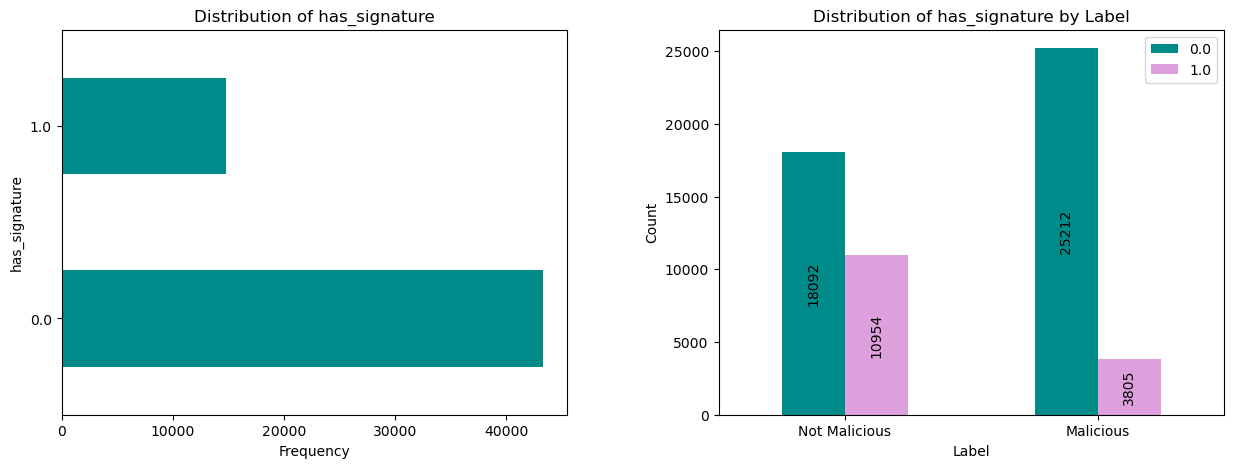
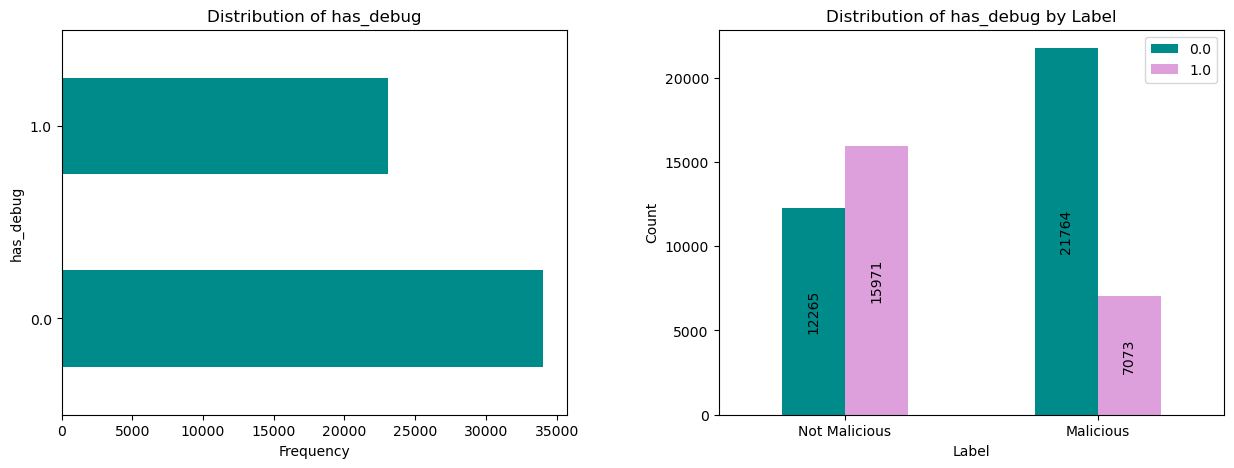




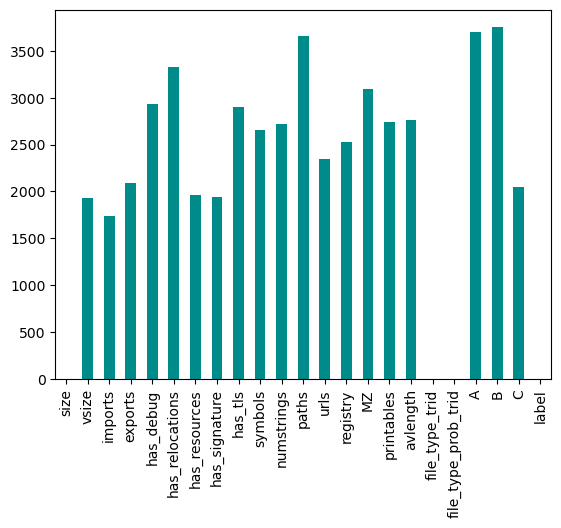
1. הרחבה על המשתנים הקטגוריאליים –

|  |  |  |
| --- | --- | --- |
| פיצ'ר | הערך הנפוץ ביותר | ערך מתאים לקבלת קובץ זדוני |
| Has\_debug | 0 | 0 – נראה כי עבור ערך זה יש את הסבירות הגדולה ביותר לקבלת ערך זדוני, בהתאמה מצאנו שקובצים זדוניים מנסים להעלים את נוכחותם בכך שהם לא מכילים מידע debug |
| Has\_tls | 0 | 1 – נראה כי עבור ערך זה יש את הסבירות הגבוה ביותר לקבלת ערך זדוני, בהתאמה קבצים זדוניים ישתמשו בtls |
| Has\_resources | 1 | 0 - נראה כי עבור ערך זה יש את הסבירות הגבוה ביותר לקבלת ערך זדוני |
| Has\_relocation | 1 | 0 - נראה כי עבור ערך זה יש את הסבירות הגבוה ביותר לקבלת ערך זדוני |
| Has\_signature | 0 | 0 - נראה כי עבור ערך זה יש את הסבירות הגבוה ביותר לקבלת ערך זדוני, בהתאמה לקבצים זדוניים יש יותר סיכוי כי לא ימצאו בהם signature |

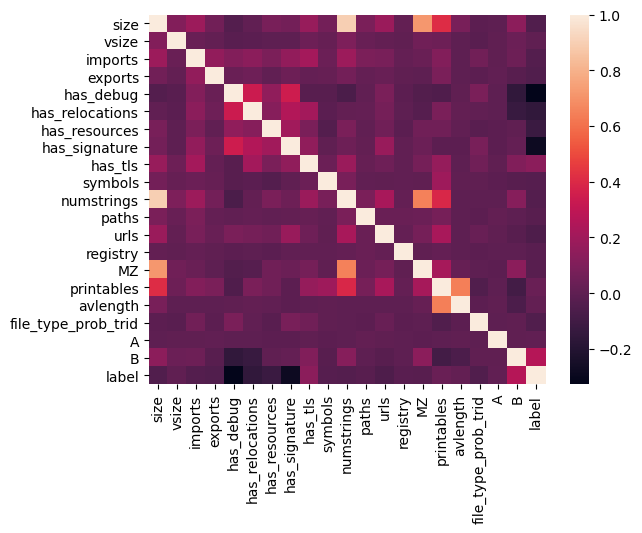
1. תרשימים של פיצ'רים בינאריים –

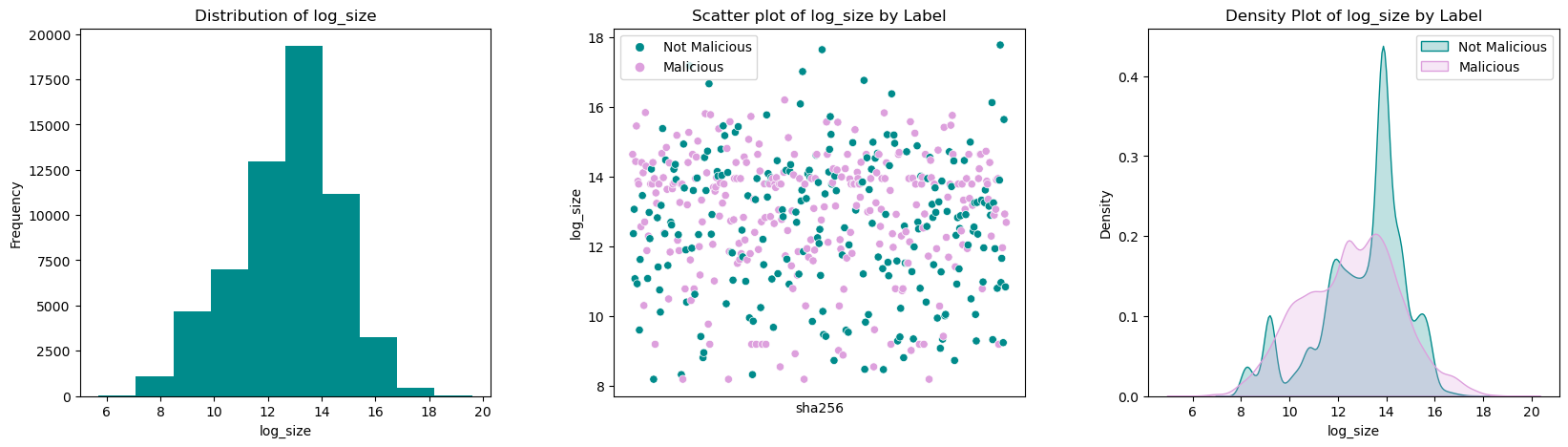
1. תרשים ערכים חסרים –

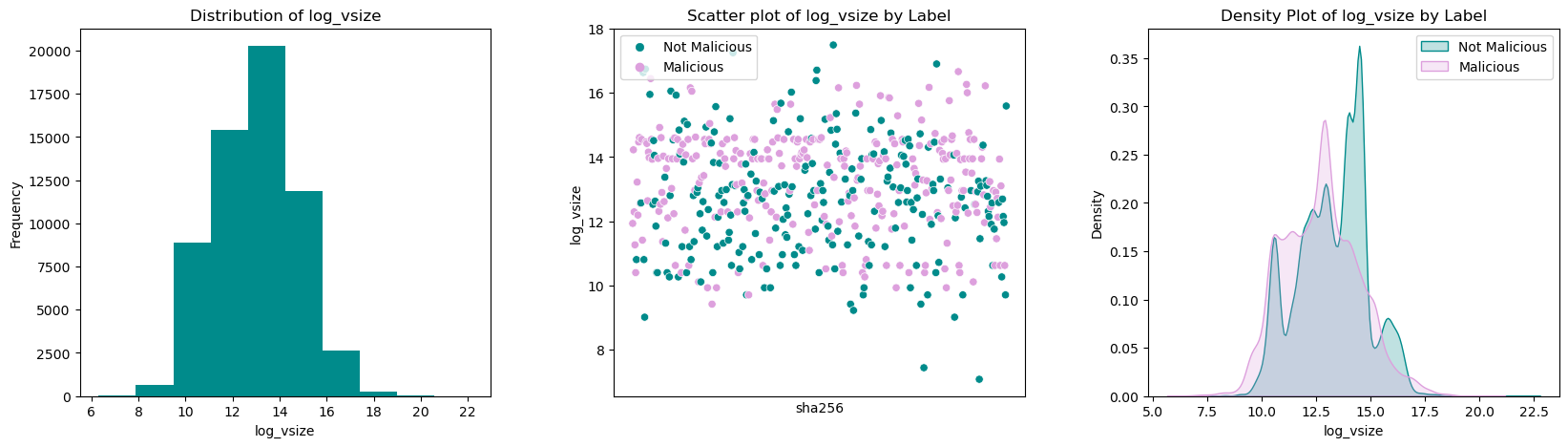


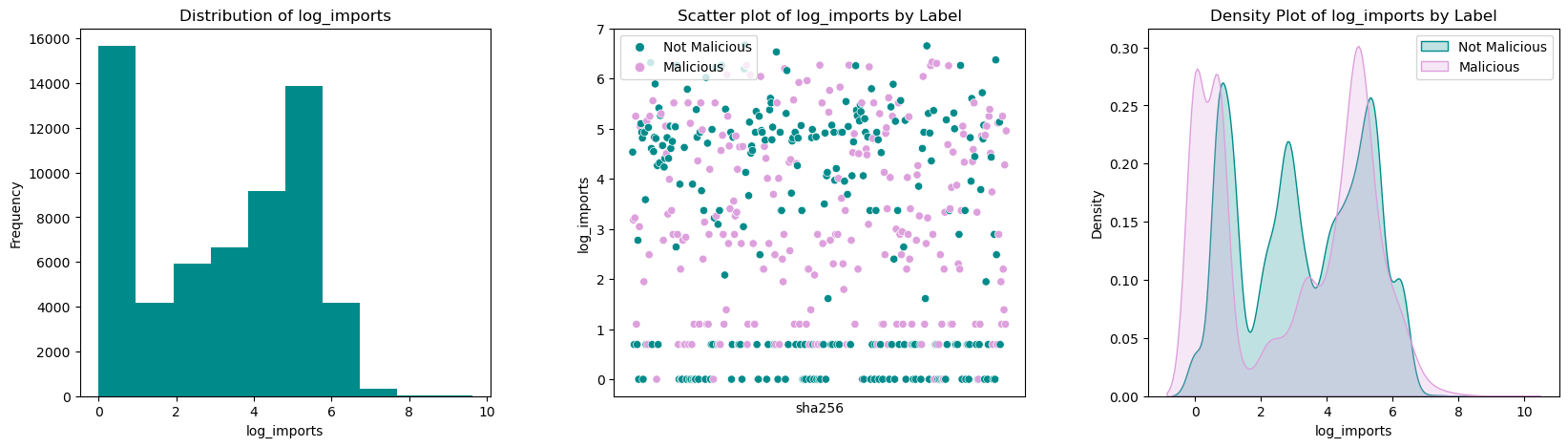
1. תרשים של הקורלציה הראשונית –

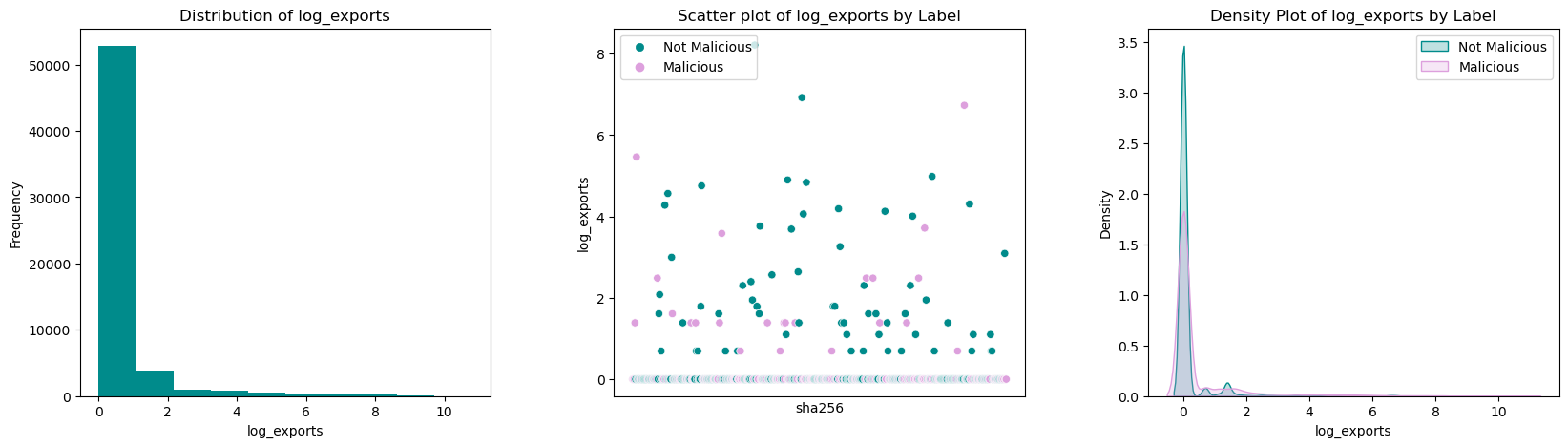


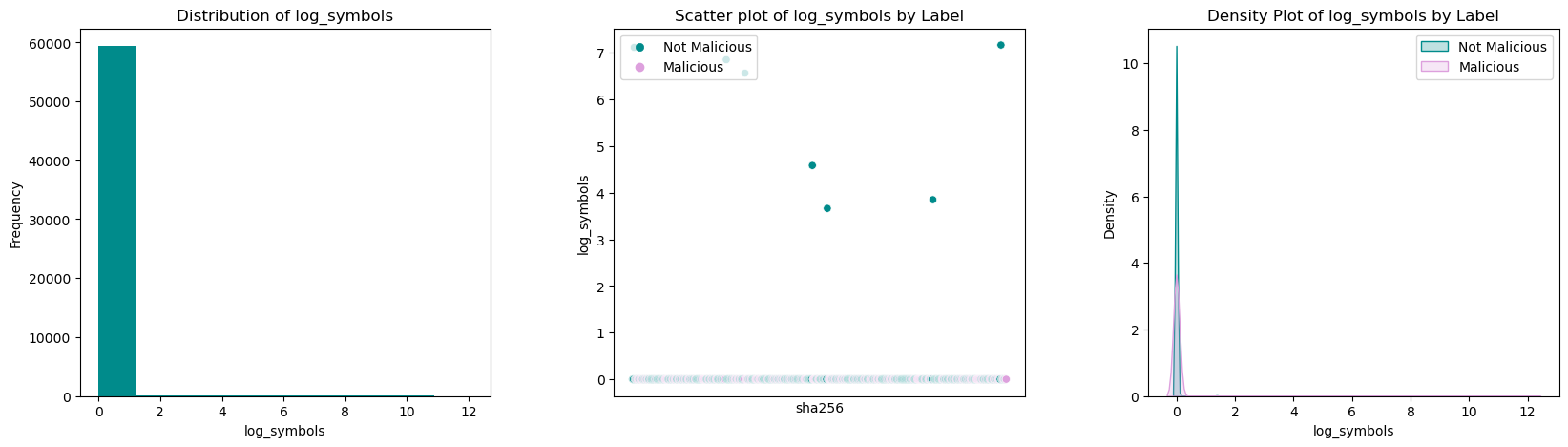
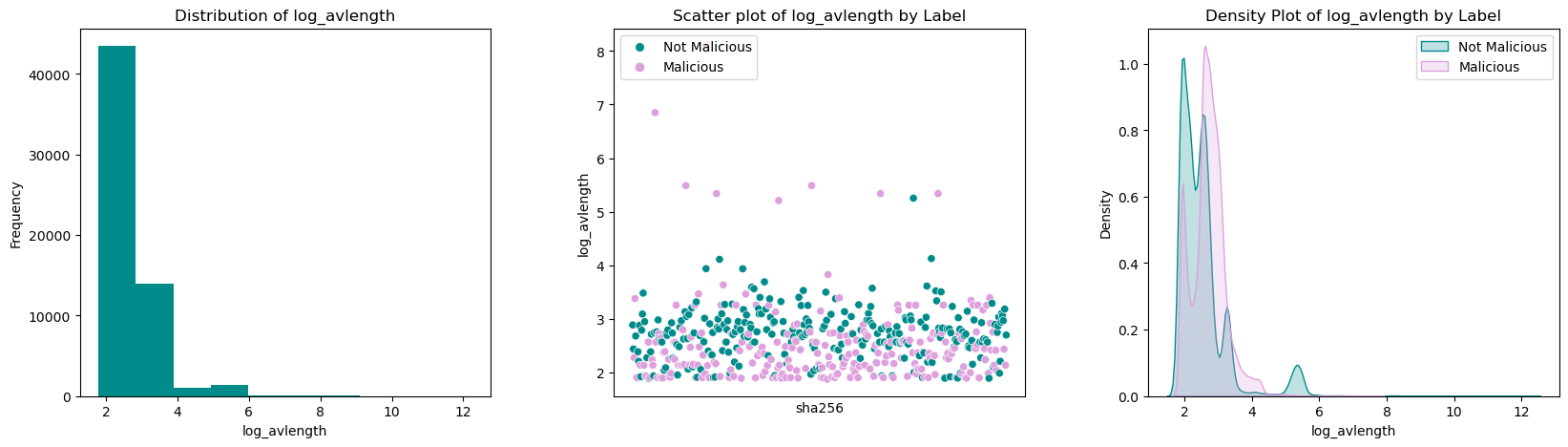
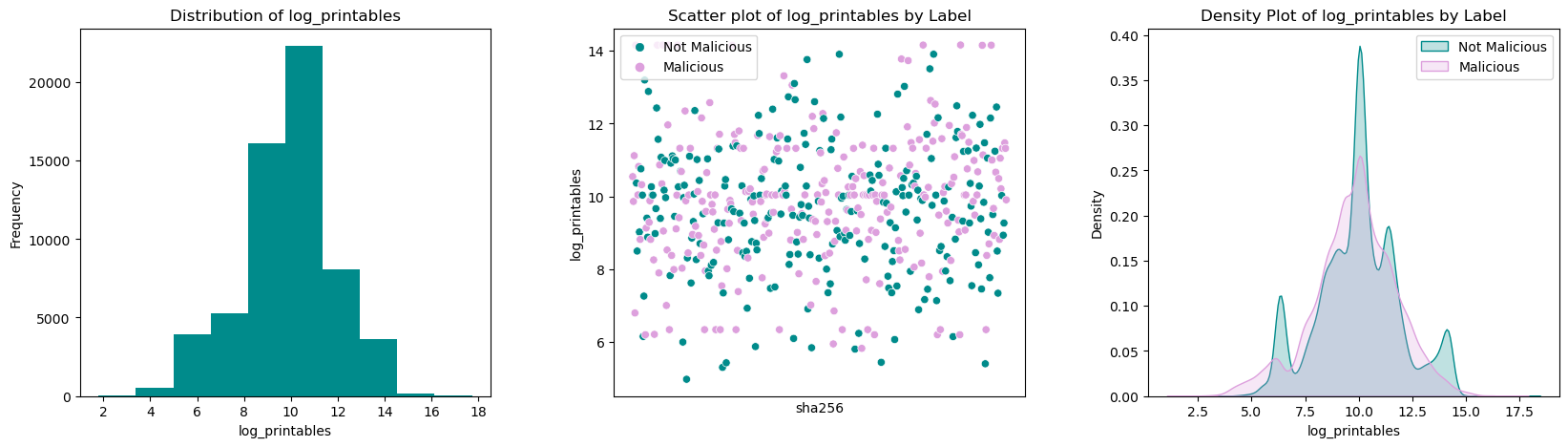
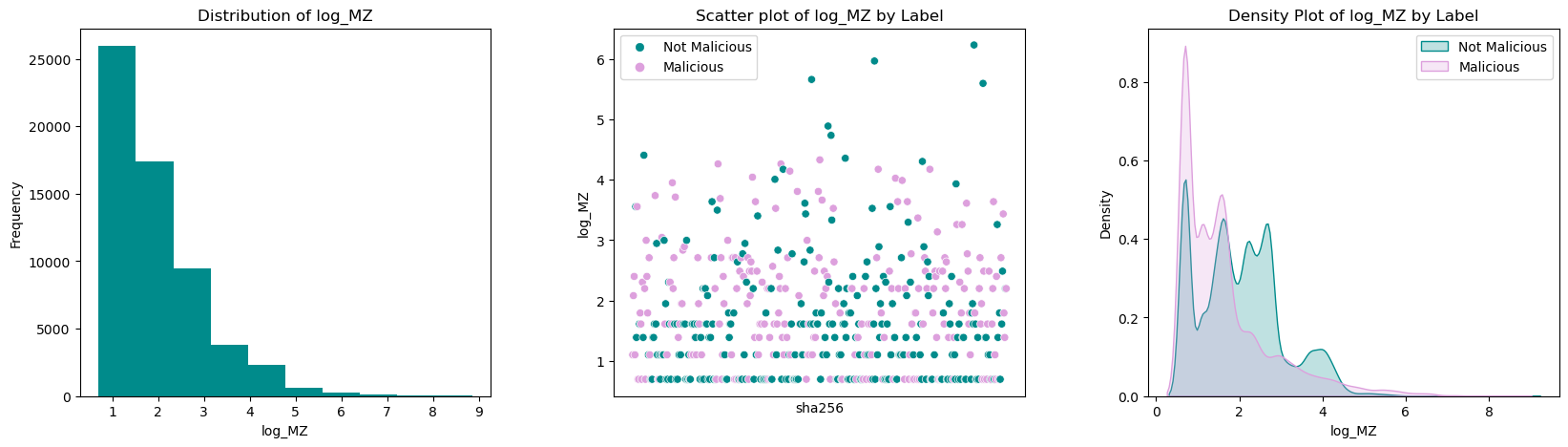
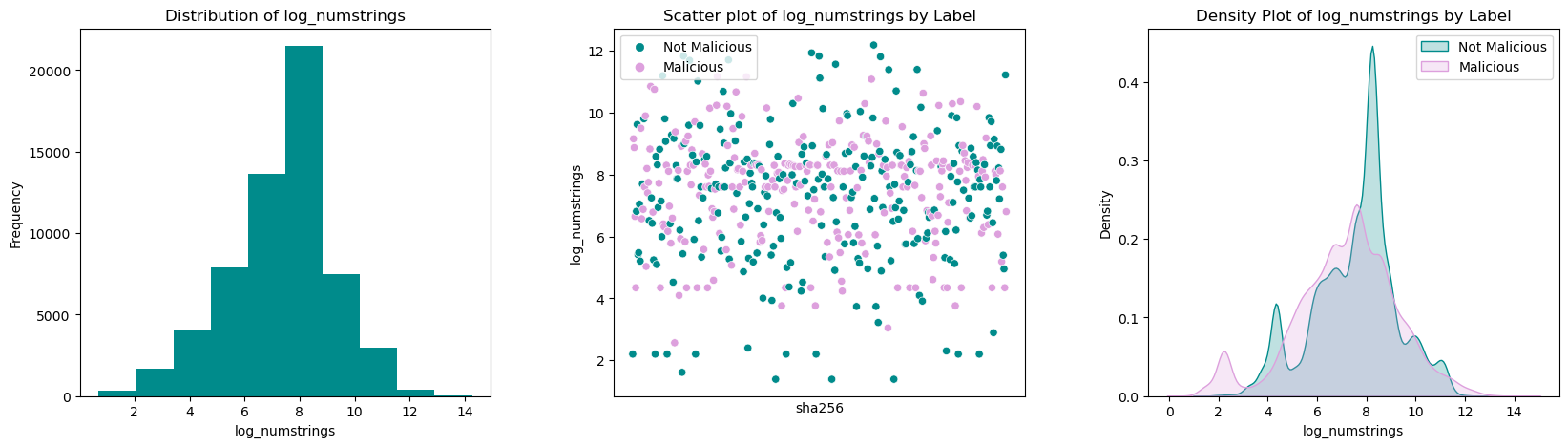
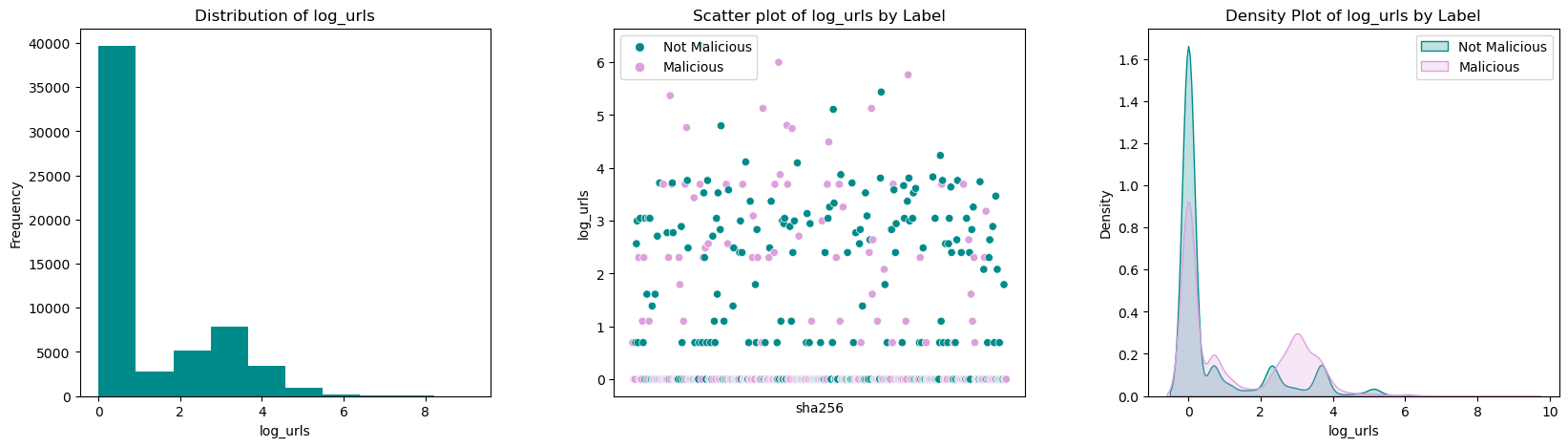
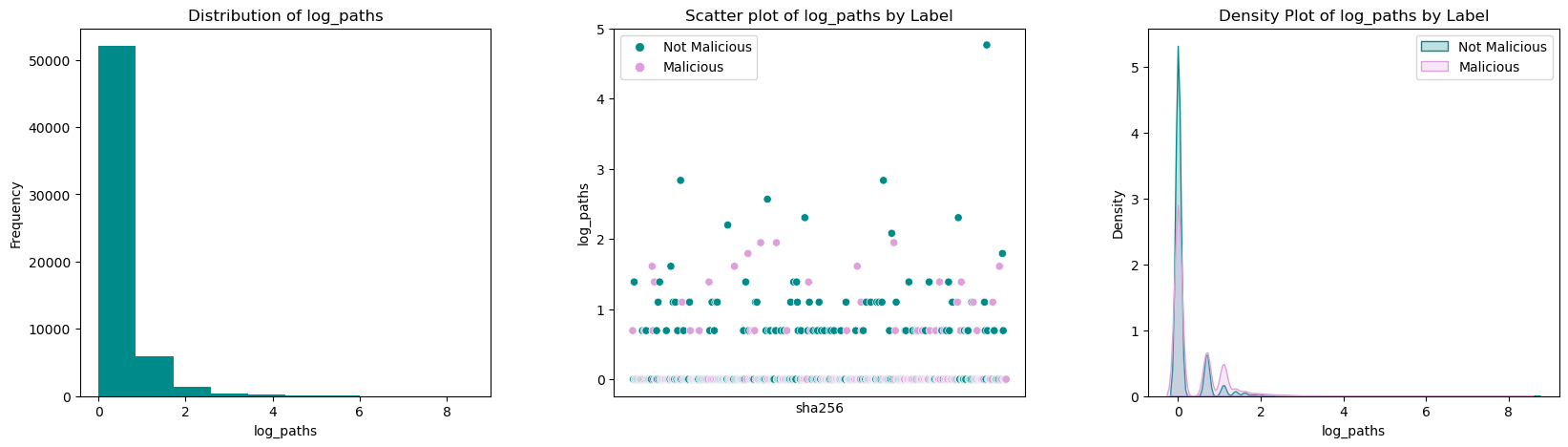
1. תרשים פיצ'רים מספריים חדשים (לאחר מניפולציית log) –

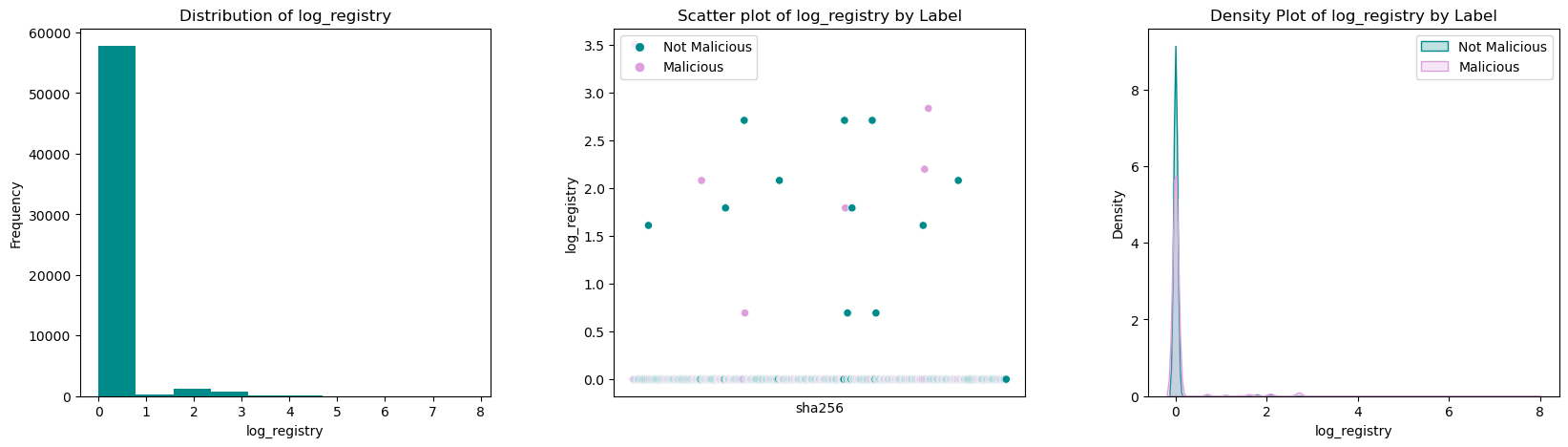




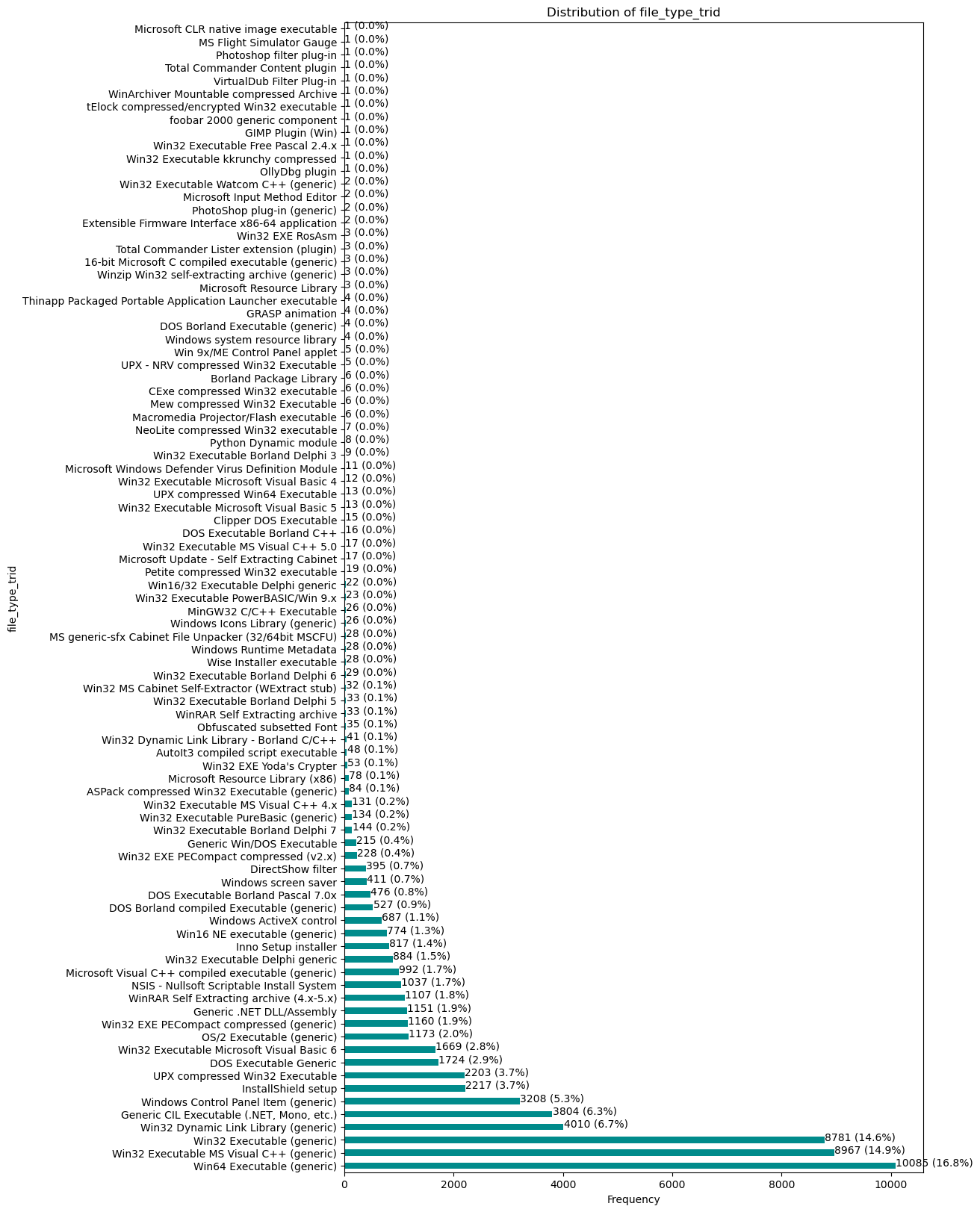




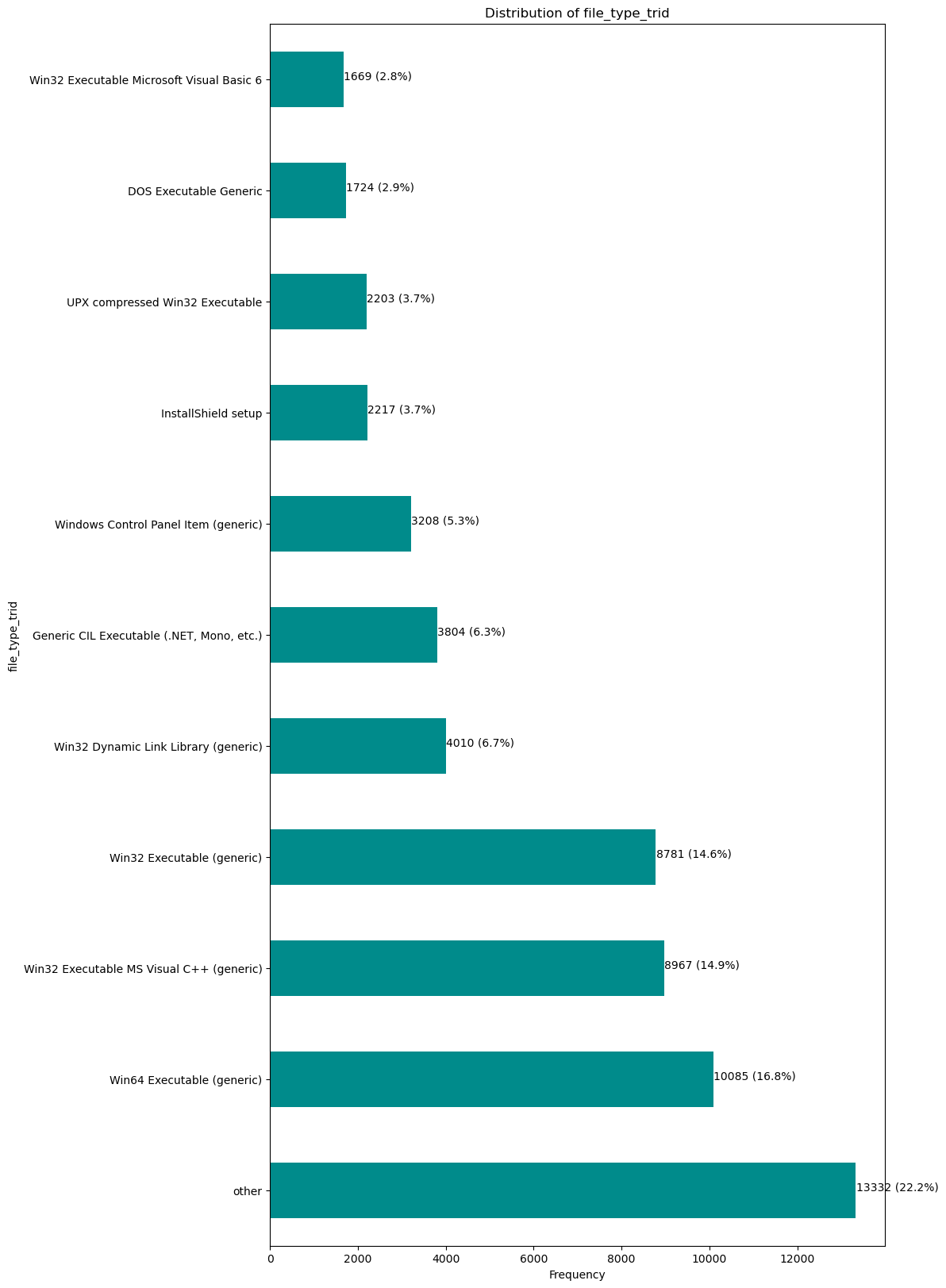
   



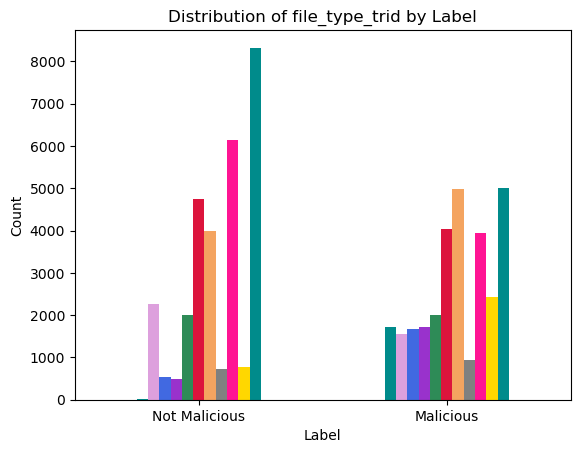
1. תרשים של התפלגות הערכים של פיצ'ר file type trid –



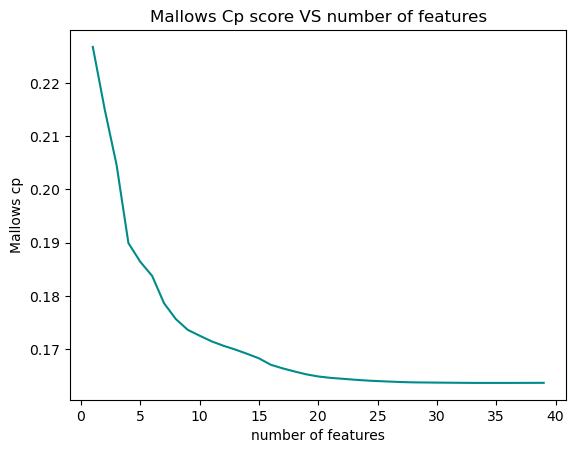
1. תרשים של התפלגות הערכים ב-file type trid לאחר האגרגציה –



1. תרשים של התפלגות ערכי file type trid לפי label –



1. תרשים שמתאר את ה-mallow cp עבור כל מספר פיצ'רים בעת הרצת forward selection –

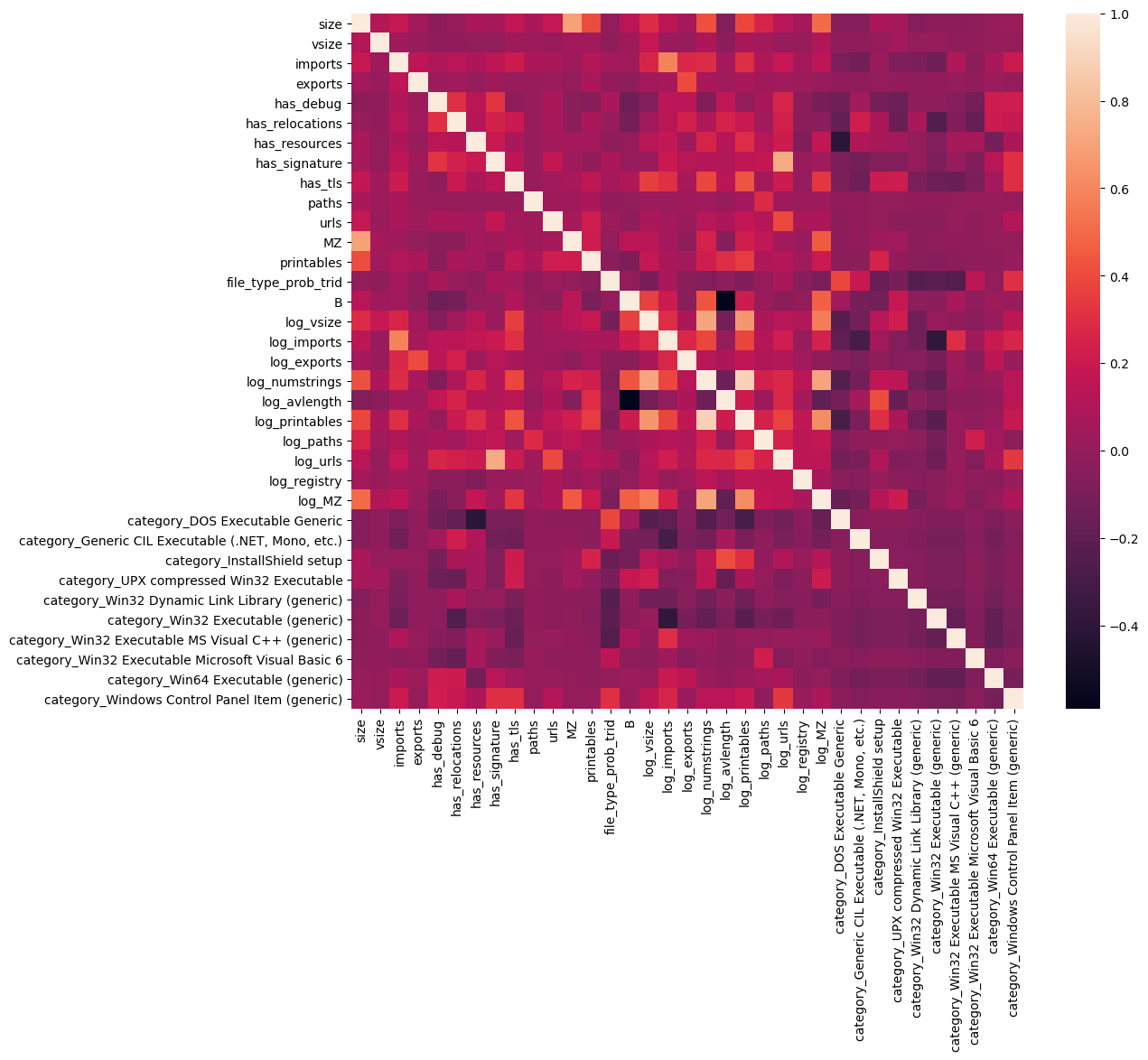


1. פלט –

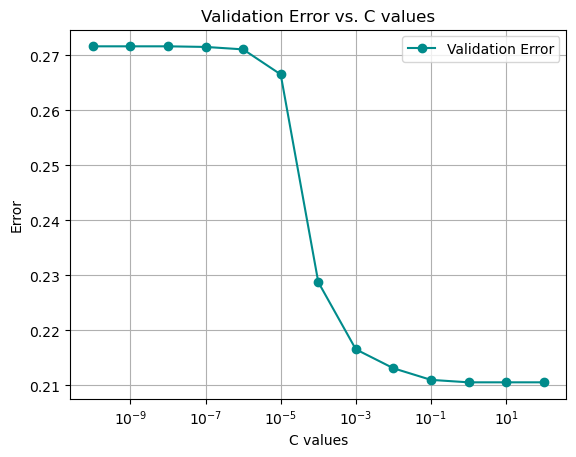
Number of Chosen Features: 35

Chosen features: ['size', 'vsize', 'imports', 'exports', 'has\_debug', 'has\_relocations', 'has\_resources', 'has\_signature', 'has\_tls', 'paths', 'urls', 'MZ', 'printables', 'file\_type\_prob\_trid', 'B', 'log\_vsize', 'log\_imports', 'log\_exports', 'log\_numstrings', 'log\_avlength', 'log\_printables', 'log\_paths', 'log\_urls', 'log\_registry', 'log\_MZ', 'category\_DOS Executable Generic', 'category\_Generic CIL Executable (.NET, Mono, etc.)', 'category\_InstallShield setup', 'category\_UPX compressed Win32 Executable', 'category\_Win32 Dynamic Link Library (generic)', 'category\_Win32 Executable (generic)', 'category\_Win32 Executable MS Visual C++ (generic)', 'category\_Win32 Executable Microsoft Visual Basic 6', 'category\_Win64 Executable (generic)', 'category\_Windows Control Panel Item (generic)']

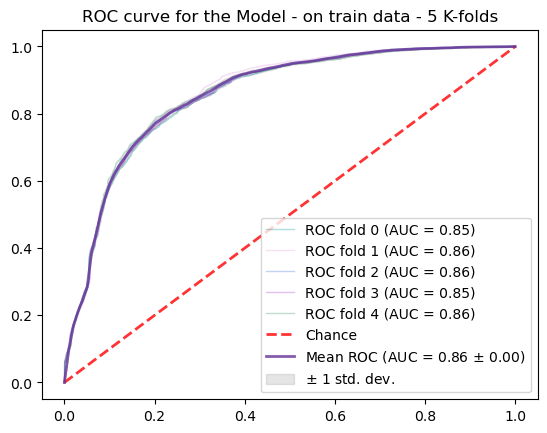
1. תרשים מפת קורלציה לאחר שלב העיבוד המקדים –



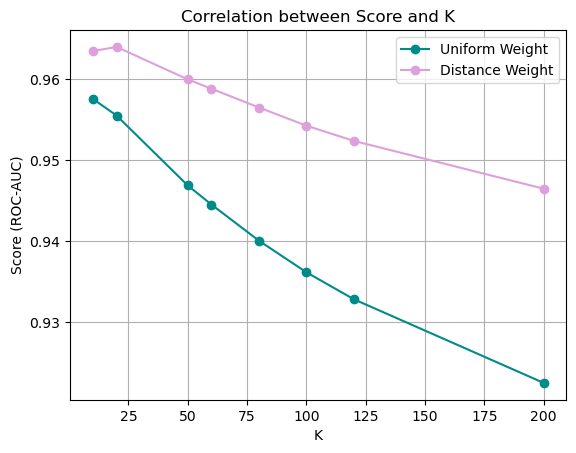
1. רגרסיה לוגיסטית –
2. תרשים המתאר את מציאת ההיפר-פרמטר C האופטימלי –



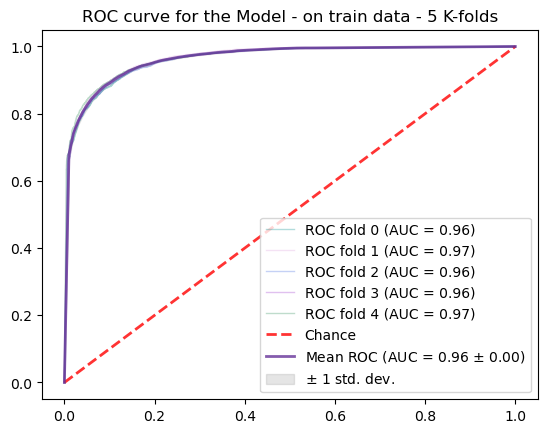
1. תרשים ROC ו-AUC על נתוני האימון –



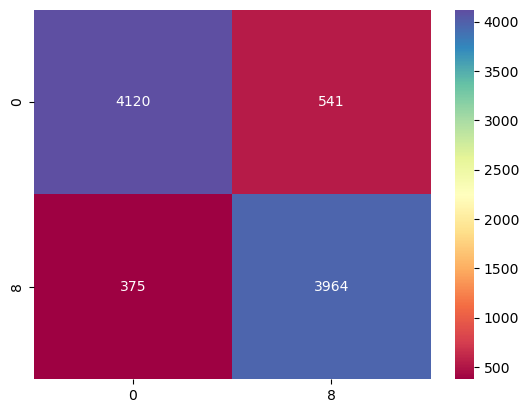
1. KNN
2. תרשים המתאר את מציאת ההיפר-פרמטרים האופטימליים עבור המודל –



1. תרשים ROC ו-AUC על נתוני האימון –



1. Validation confusion matrix plot –

פלט –

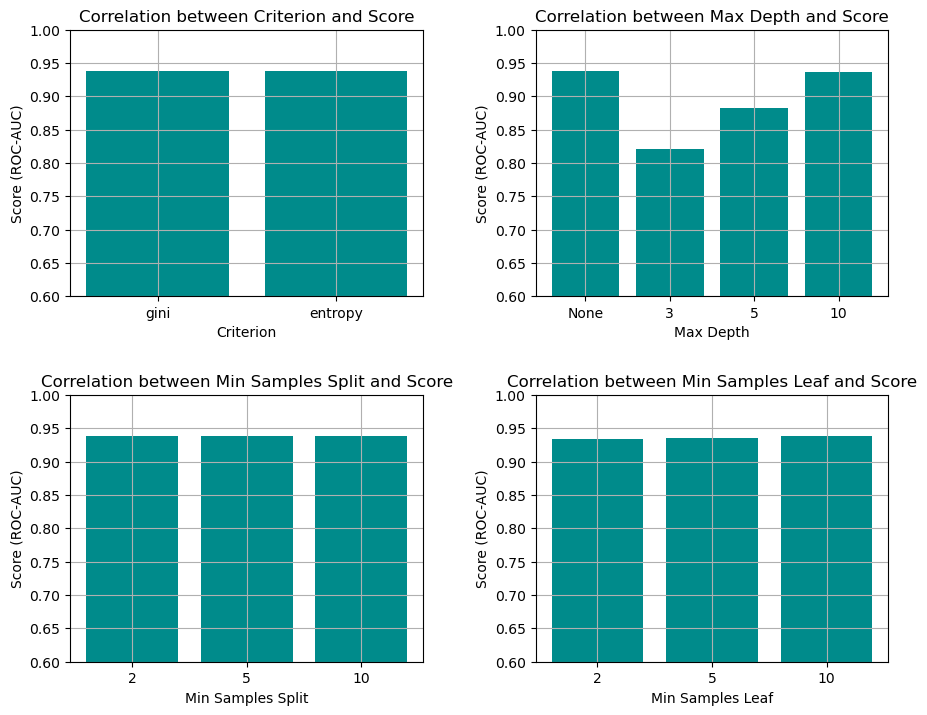
Validation Accuracy:

0.8982222222222223

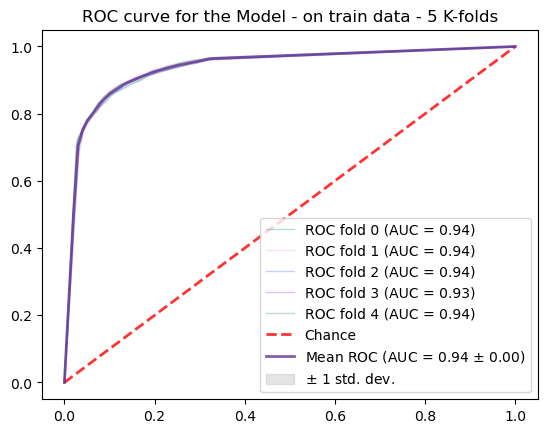
Validation Recall:

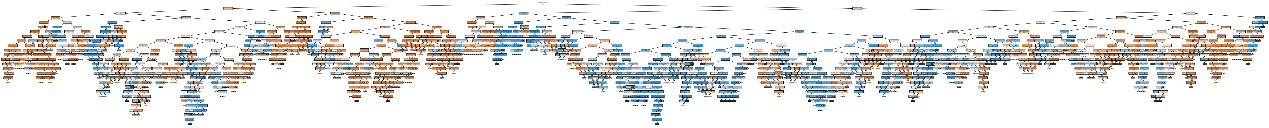
0.9165739710789766

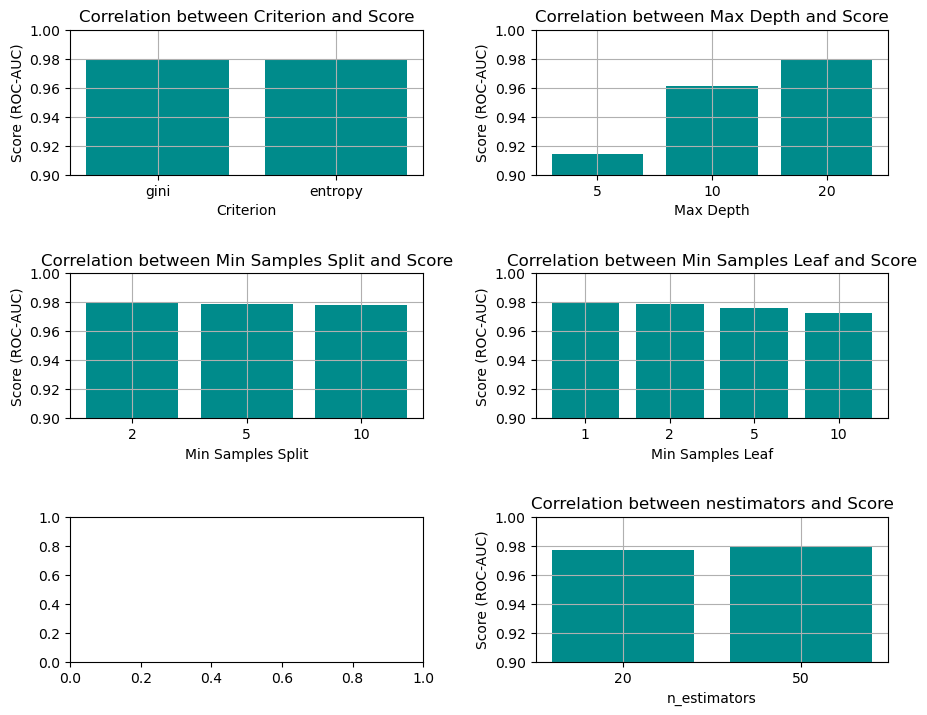
1. Decision Tree –
2. תרשים המתאר את מציאת ההיפר-פרמטרים האופטימליים עבור המודל –



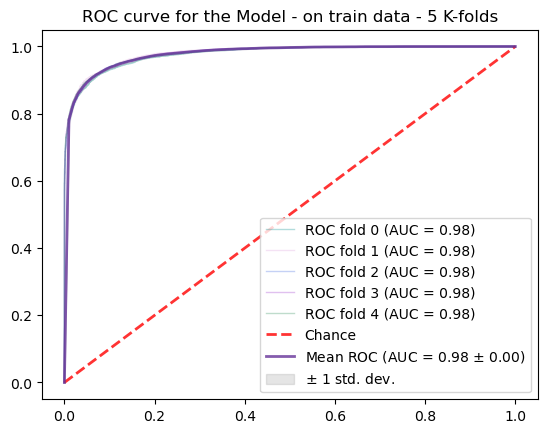
1. תרשים ROC ו-AUC על נתוני האימון –



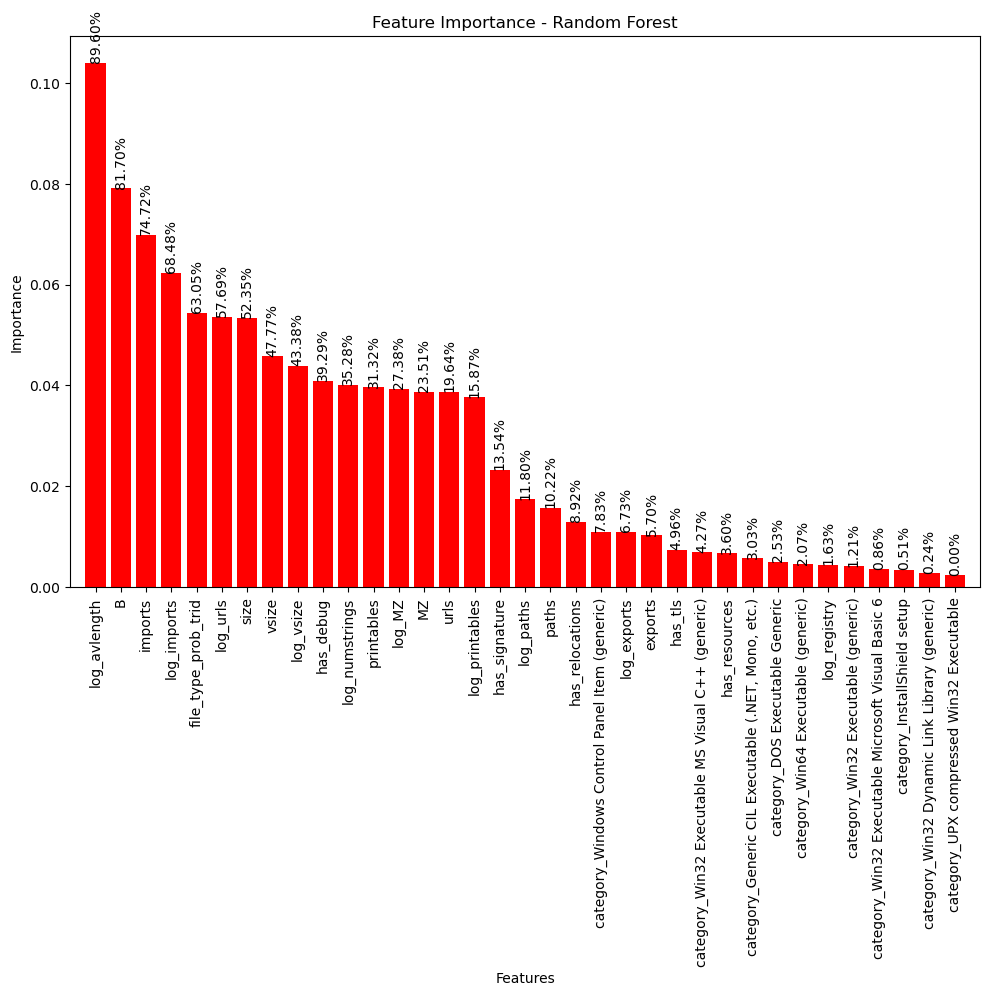
1. תרשים העץ –
2. Random Forest –
3. תרשים המתאר את מציאת ההיפר-פרמטרים האופטימליים עבור המודל –

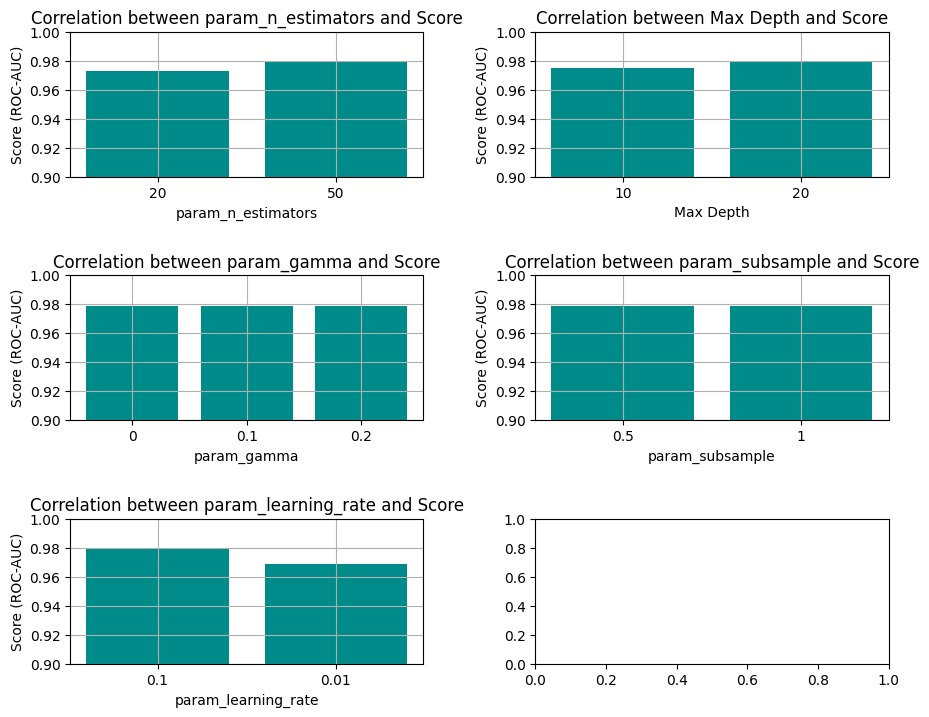
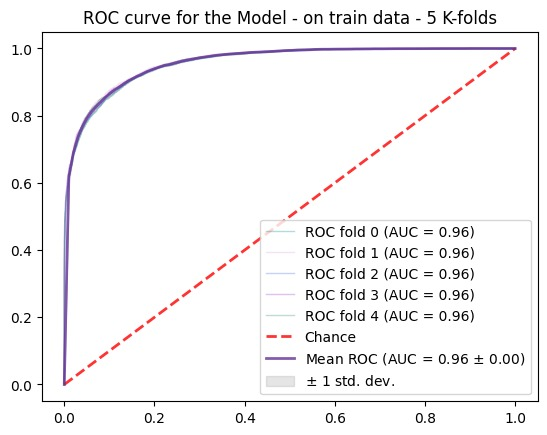


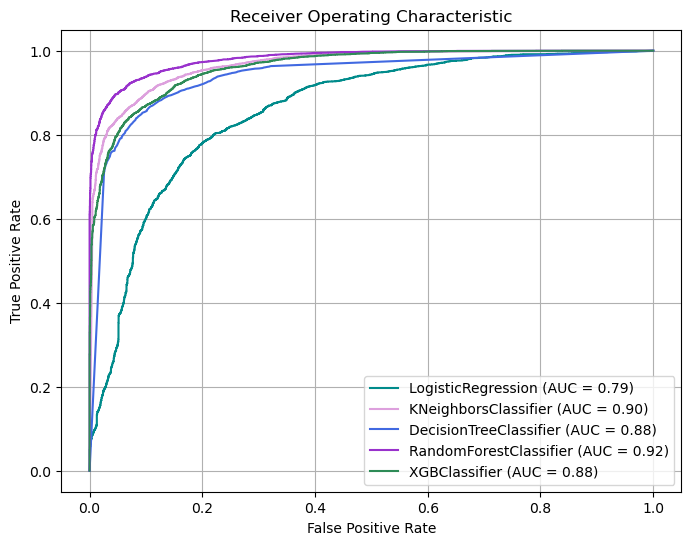
1. תרשים ROC ו-AUC על נתוני האימון –



1. Feature importance –



1. XGBoost –
2. תרשים המתאר את מציאת ההיפר-פרמטרים האופטימליים עבור המודל –
3. תרשים ROC ו-AUC על נתוני האימון –
4. תרשים המתאר את ה-ROC ו-AUC על כל המודלים –



1. משמעות היפר הפרמטרים:

|  |  |
| --- | --- |
| היפר הפרמטר | משמעות |
| C | הפרמטר C קובע את עוצמת ה-regulation. C קטן יותר מגביר את ה-regulation, מה שמוביל למודלים פשוטים יותר (מוריד שונות ומעלה bias).  C גדול יותר מפחית את ה-regulation, ומאפשר למודל להתאים עצמו לנתונים (מעלה שונות ומוריד bias). |
| weights | Weights קובע איך ממושקלים השכנים, עם משקל זהה (**uniform**) או משקל בהתאם למרחק (distance). |
| criterion | הפרמטרcriterion קובע את שיטת חישוב ה-impurity. Entropy מקושר עם עץ החלטה בעל bias נמוך יותר ובהתאמה שונות גבוהה יותר. |
| min samples split | הפרמטר min sample split קובע את המספר המינימלי של דגימות הנדרשות על מנת לפצל צומת פנימי לצמתים צאצאים. ככל שהמספר גבוה השונות קטנה וה-bias עולה בהתאמה. |
| min samples leaf | הפרמטר min sample leaf קובע את המספר המינימלי של דגימות הנדרשות על מנת לפצל עלה. ככל שהמספר גבוה כך השונות קטנה וה-bias עולה בהתאמה. |
| Max depth | הפרמטר max depth קובע את המספר המקסימלי של הרמות שהעץ יכול לקבל. ככל שיש יותר רמות, השונות עולה (overfitting) ובהתאמה ה-bias יורד. |
| learning rate | קובע את הגודל הצעד בכל איטרציה. ככל שהוא נמוך יותר המודל נותן פחות שפעה לכל עץ, מפחית את השונות ומעלה את ה-bias. |
| n estimators | בעבור KNN מגדיר את כמות השכנים שבאמצעותם נקבע ה-label של הדגימה החדשה. ערך גבוה של N מעלה את ה-bias בעוד שערך נמוך זה מודל שיכול להוביל ל-overfitting (שונות גדלה על חשבון bias)  בעבור מודלים המתבססים על עצי החלטה המשתנה קובע את מספר העצים שנשתמש, ככל שהמספר גבוה ה-bias קטן והשונות גדלה. |
| gamma | משמש כרגולריזציה השולטת במורכבות המודל, ככל שיותר גבוה השונות קטנה וה-bias עולה |
| subsample | מתייחס לבחירה רנדומלית של כמות הדאטה לכל עץ, ככל שגבוה יותר השונות קטנה, זאת מכיוון שהמודל מתאמן על סט נתונים קטן, במקביל ה-bias גדל |