מדעים דיגיטליים להייטק

הפקולטה להנדסה

דוח פרויקט גמר
בקורס מבוא ללמידת מכונה

מוגש ע"י

318252160רן אסנטה

עידן בעבור 314918699

מרצה: דור בנק

מתרגל: אילן וסילבסקי

יוני 2023

הרצת מודל ללא הבנת הפיצ'רים או עיבוד הנתונים

נפתח את הפרויקט עם ביצוע מודל ראשוני באמצעות מודל פשוט - רגרסיה לוגיסטית, כדי לקבל תוצאה ראשונית של מודל שממנה נרצה להשתפר ולבצע עבודה מעמיקה יותר על הנתונים, בהתאם להמלצות מההרצאה. כדי לבצע זאת, הורדנו מהנתונים את כל הפיצ'רים שאינם מספריים. ביצענו רגרסיה לוגיסטית עם הנתונים שנשארו לנו - בערך 32.5K דוגמאות עם 15 פיצ'רים לעומת 60K דוגמאות עם 22 פיצ'רים. עשינו Cross-Validation עבור K=5 והצגנו גרף של עקומת ה-ROC לכל fold. קיבלנו שציון ה-AUC הממוצע על סט הוולידציה הוא 0.5704=1-0.4296, כלומר חיזוי דגימה של 57.07%. על סט האימון קיבלנו ציון ממוצע של 0.4281, כך שלפי אותו החישוב נקבל 0.5719=1-0.4281. גרף המודל יופיע בנספחים תחת הכותרת "גרפים של ROC-AUC עבור כל מודל".
חיזוי סביב 57% קרוב לניחוש מוחלט לגבי זדוניות הקובץ (ניחוש מוחלט יהיה חיזוי של 50%), ולכן נרצה לשפר את המודל באופן משמעותי בחלקי הפרויקט הבאים. קיבלנו ציונים קרובים בין שני הסטים, דבר אשר עשוי להצביע על איזון טוב בין ההטיה לשונות, אך לא ניתן לייחס לכך חשיבות בשלב זה של הפרויקט עקב העובדה שלא בוצעו שינויים על הנתונים, והתחזיות של המודל קרובות לניחושים על סמך הנתונים הקיימים במתכונת המקורית שלהם.

חלק ראשון – אקספלורציה

ייבאנו את הספריות וטענו את קובץ ה-train.csv, אותו חילקנו למטריצת הפיצ'רים ולוקטור הלייבלים.
כל הגרפים והתרשימים המפורטים מטה נמצאים בנספחים תחת הכותרת "אקספלורציה – גרפים".

בדיקת התפלגויות הפיצ'רים:

נבדוק כיצד כל פיצ'ר מתפלג על ידי גרף היסטוגרמה עבור כל פיצ'ר בנפרד. הפיצ'רים מתחלקים לשלושה סוגים: בינארי, קטגוריאלי או מספרי, וכל אחד דורש טיפול אחר והצגה שונה של התפלגויות הנתונים. נבצע זאת בעזרת לולאה על כל פיצ'ר ונסיק מסקנות למען שלב עיבוד הנתונים.

1. **פיצ'רים מספריים:**
ניתן לראות שישנם פיצ'רים שבכל הנראה מתפלגים נורמלית (כגון: size, vsize, numstrings, printables) ועבור שאר הפיצ'רים לא נראה שיש התפלגות מוגדרת.
למדנו בכיתה שבדרך כלל נרמול הנתונים זו פעולה הכרחית וחיובית, במיוחד אם לפיצ'רים קנה מידה שונה או שהם בעלי התפלגות שונה. נרמול הפיצ'רים עשוי לעזור לשפר את הביצועים והיציבות של המודלים השונים שנבצע בהמשך, ונתייחס לנרמול כשיבוצע בשלב העיבוד המקדים.

2. **פ'יצרים קטגוראליים:**
ישנם 2 פיצ'רים קטוגוריאליים: "C" ו-"file_type_trid".
"C" - הוא פיצ'ר אנונימי וניתן לראות שהוא מתפלג באופן די אחיד לכל קטגוריה בו, למעט "vh" - הוא גבוה מאוד ביחס לשאר הקטגוריות, ולמעט "vr" - הוא נמוך מאוד ביחס לשאר הקטגוריות. מלבד זאת, לא ניתן לדעתנו להסיק על משתנה זה מסקנה קונקרטית נוספת.
"file_type_trid" - ניתן לראות שיש הרבה קטגוריות בפיצ'ר זה. ל-3 סוגים קטגוריות של סוגי קבצים יש כמות שגדולה יותר משמעותית משאר הקטגוריות, בעוד שיש מספר רב של קטגוריות של סוגי קבצים עם כמות זניחה ביחס לכמויות הגדולות בפיצ'ר או כמות כמעט אפסית.
לא מצאנו לנכון להסיק מסקנות משמעותיות מהפיצ'רים הנ"ל ולכן נעבור אל סוג הפיצ'רים הבא.

3. **פיצ'רים בינאריים:**
ישנם 5 פיצ'רים בינאריים ובדקנו את הפרופורציה של כל אחד מהם (0 או 1). ניתן לראות ש: has_debug ו-has_relocation מתחלקים באופן די שוויוני, בעוד שהשאר לא. בדקנו בנתונים, על פי הלייבלים, ומצאנו כי חצי מהקבצים הם זדוניים וחצי אינם. ניתן מכאן להסיק כי הפיצ'רים שמתחלקים באופן שווה או קרוב לכך עשויים להיות משמעותיים יותר למודלים שנבצע, ואנו מניחים כי הם יקבלו משקולות משמעותיות יותר לעומת הפיצ'רים שאינם מתחלקים באופן שווה. בכל אופן נשאיר את הפיצ'רים האלה בנתונים.

בנוסף נשים לב ש-"file_type_trid" ו-"file_type_prob_trid" קשורים אחד לשני (סיווג הקובץ וההסתברות לסיווגו כקובץ כזה) ומכיוון שאחד מהם קטגוריאלי לא נראה את הקשר הנ"ל במפת חום של קורלציה. בשלב העיבוד המקדים נבדוק האם ניתן לבצע פעולה כלשהי לגבי הקשר ביניהם והאם כדאי להוריד את אחד מהם או את שניהם מהנתונים.

<u>קורלציה בין הפיצ'רים:</u>

ביצענו מפת קורלציה בין הפיצ'רים כדי למצוא קשרים אפשריים ביניהם. מצאנו כי כמעט כל הפיצ'רים אינם קשורים אחד בשני לפי מתאם פירסון מלבד הפיצ'רים הבאים: corr(size, numstrings) = 0.89, corr(size, MZ) = 0.71, corr(numstrings, MZ) = 0.64, corr(printables, avlength)=0.64. ניתן לראות כי יש קשר חזק מאוד (מעל 0.75) ושלושה קשרים בינוניים (0.6 - 0.75) ונרצה בשלב העיבוד המקדים לנסות להשתמש במידע הנ"ל כדי לבצע טרנספורמציות וחישובים שישדרגו את השימוש בפיצ'רים.

מפת הקורלציה (וגם המפה הבאה שנבצע) מופיעה בנספחים תחת הכותרת "מפת קורלציות בין הפיצ'רים".

הצגנו סוגים נוספים של גרפים עבור כל פיצ'ר מספרי: בוקס פלוט, ויולין פלוט, סקטר פלוט. גרפים אלו נותנים לנו את הממוצע, החציון ואת ערכי הקצה. זאת ועוד, הם מציגים בפנינו את התפלגות הנתונים ואת שכיחות הערכים בכל פיצ'ר ביחס לשאר הערכים. כמו כן הצגנו בטבלה את הנתונים המוזכרים ואת האחוזונים הקיצוניים של כל פיצ'ר (עשרת האחוזים בכל קצה מוצגים בקפיצות של שני אחוזים). הגרפים מופיעים בנספחים תחת הכותרת: "אקספלורציה – גרפים". המסקנות המתקבלות מן הגרפים הן:

1. הממוצע של size קטן יותר מהממוצע של vsize כמצופה, כיוון שאנו מצפים שהגודל הוירטואלי המוקצה יהיה לכל הפחות שווה לגודל האמיתי של הקובץ. על אף זאת ראינו כי החציון של vsize קטן יותר מהחציון של size. מכאן ניתן להבין שכנראה יש קבצים שמקבלים הקצאות זיכרון קטנות יותר, דבר אשר עלול להשליך על סיווג הקובץ כזדוני (למשל האם קבצים זדוניים מקבלים נפח זיכרון גדול יותר במטרה לנפח את הדיסק). כמו כן שמנו לב שישנם פעמים שהזיכרון הוירטואלי המוקצה נמוך מגודל הקובץ וזה עלול גם כן לתת לנו אינדיקציה לגבי חשיבות ההבדלים בין ערכי הפיצ'רים הללו. נתייחס לכך עם פיצ'ר נוסף שיציג את היחס בין הגדלים בשלב העיבוד המקדים.
2. הגרפים של הפיצ'רים הבינאריים מציגים את היחס בין הקטגוריות באופן התואם ליחס שיצא בתרשימי העוגה שהוצגו.
3. עבור הפיצ'ר "file_type_prob_trid" המראה לנו את ההסתברות לסוג הקובץ - לא מצאנו מסקנות חדשות מנתוניו הסטטיסטים.
4. עבור הפיצ'רים numstrings, printables, avlength מצאנו כי הממוצע גדול משמעותית מהחציון ולכן מסתמן שיש הרבה ערכי קיצון. נחליט בשלב העיבוד המקדים כמה דגימות להוריד לפי הפערים באחוזונים.
5. עבור שאר הפיצ'רים המספריים אין פער גדול באותה מידה כמו הפער שהיה עבור הפיצ'רים הקודמים, לכן נשער כי אין בפיצ'רים אלו מספר גדול של ערכי קצה.

<u>חלק שני – עיבוד מקדים</u>

<u>פיצול לסט אימון וסט וולידציה</u>

פיצלנו את הנתונים לסט אימון ולסט וולידציה של 20% מכלל הנתונים. פעולה זו נועדה להפריד את סט הוולידציה ולתרום להיותו מידע נפרד ועצמאי מסט האימון. זה מבטיח לנו שנוכל להעריך את המודלים שנבצע בעזרת סט הוולידציה שישמש מידע שהמודל לא ראה, וידמה את סט המבחן. ראוי לציין שעקב ביצוע Cross-Validation בעת הרצת המודלים, נשתמש בפיצול זה בעיקר למען הורדת המימדיות. לכן, בנוסף לביצוע העיבוד המקדים על סט האימון ועל סט הוולידציה, נבצע אותו גם על סט הנתונים כולו כדי לא לאבד 20% מהדגימות על סט וולידציה שלא ישמש אותנו בעת הרצת המודלים. נשמור אותו בנפרד במשתנה אחר עד לרגע הרצתם, לכן יש לשים לב שכל שלבי העיבוד המקדים פועלים על סט האימון וסט הוולידציה וכל סט הנתונים כולו בנפרד, אך בכולם משתמשים בערכי סט הנתונים המלא.

מילוי ערכים חסרים

כדי לטפל בערכים חסרים תחילה בדקנו באילו פיצ'רים בכלל חסרים נתונים. מילאנו אותם בשיטות שונות והן מפורטות, יחד עם הסיבות לשימוש בכל שיטה, בנספחים תחת הכותרת "מילוי ערכים חסרים".

התמודדות עם משתנים קטגוריאליים:

לשם הרצת המודלים יש להפוך את הפיצ'רים הקטגוריאלים לרציפים\מספריים. השתמשנו בשיטת One Hot Encoding. לפני הקידוד היו 24 פיצ'רים ולאחר הקידוד 118 פיצ'רים. מדובר בהרבה פיצ'רים ולכן השיטה אינה מיטבית, עקב כך נרצה להוריד פיצ'רים לפני השימוש באלגוריתם הורדת מימדים. בדקנו באילו משתנים יש פחות מ-0.005% ערכים השונים מ-0. מתוך 60000 דגימות קיימות, פיצ'רים בהם יש ערכים בודדים כאלו יהיו חסרי חשיבות ונוריד אותם באופן ידני. כך נתגבר על ריבוי הפיצ'רים עוד לפני הרצת אלגוריתם הורדת המימדים שעליו נדבר בהמשך, ונוכל לתת לו בסיס התחלתי מיטבי יותר.

בניית פיצ'רים חדשים ו/או מניפולציה מתמטית על פיצ'רים קיימים:

לאחר בדיקה נוספת שלא חסרים ערכים בנתונים, נעשה בפעם השנייה מפת חום (מוצגת בנספחים) כדי לראות אם הקורלציה בין הפיצ'רים השתנתה בעקבות מילוי הערכים החסרים. מצאנו כי לא היה שינוי משמעותי לאחר מילוי הערכים החסרים. נרצה לאחד את הפיצ'רים שנמצאו בקשר של מעל 0.7 או להוריד אחד מהם, כי אין סיבה לשמור שני פיצ'רים שנותנים מידע דומה. לאחר מילוי החסרים קיבלנו: corr(size, numstrings)=0.88 , corr(size,MZ)=0.70, corr(MZ,numstrings)=0.62.

בחרנו להוריד את עמודת הפיצ'ר numstrings מכמה סיבות: ראשית יש לו קורלציה גבוהה מאוד עם הפיצ'ר size וקורלציה גבוהה עם הפיצ'ר MZ. שנית, זה יפשט את המודל ויוריד את שונות הנתונים. שלישית, הקורלציה בין size ו-MZ נמוכה יותר ונרצה להשאיר אותם, אך נשנה את הקשר ביניהם על ידי איחודם לפיצ'ר אחד בעזרת מכפלתם. קראנו למשתנה החדש size_MZ_interaction. טרנספורמציה נוספת שנבצע היא בעקבות האינטואיציה שקיים קשר בין size ו-vsize. יצרנו משתנה חדש שבודק לכל דגימה את היחס בין size ל-vsize לפי: $\frac{size}{vsize}$. קראנו לפיצ'ר החדש size_ratio. בעקבות שמירת היחס בין הפיצ'רים, מצאנו לנכון שלא לשמור על הפיצ'רים האלו ומחקנו אותם מהנתונים. לאחר הוספה והורדה של פיצ'רים - נשארו 98 פיצ'רים בסט הנתונים המלא, 96 באימון ו-92 בוולידציה. בהמשך נתאם בין מספרי הפיצ'רים על ידי הוספתם החסרים עם ערכי 0 בלבד.

נרמול הנתונים:

הנתונים אינם מנורמלים, כלומר הערכים לא בסקלה אחידה. חשוב לנרמל את הפיצ'רים המספריים כדי להימנע מהבדלים ביחידות מידה בין הפיצ'רים והנרמול אף ישפר את יכולת המודל לזהות תבניות בין הפיצ'רים השונים של הנתונים. בנוסף כאשר יש נתונים מספריים חריגים, יתכנו השפעות לא רצויות על המודל ונרמול הנתונים מסייע למנוע השפעות אלו. ביצענו נרמול באמצעות פונקציה מובנית עם סקלר Min-Max. מצאנו לנכון להשתמש בשיטה זו כי השמת הערכים בין 0 ל-1 מאפשרת אחידות ושמירה על עיקר התפלגות הנתונים. אמנם מצטמצמים טווחי הערכים, אך בשביל להשתמש בשיטת הסטנדרטיזציה - תוחלת 0 ושונות 1 - יש להניח כי הפיצ'רים מתפלגים נורמלית, דבר שלא קורה בכולם, ולא ראינו צורך בלשנות אותם כך שנוכל להשתמש בסטנדרטיזציה.

הורדת ערכי קצה:

כדי להוריד את ערכי הקצה השתמשנו באמצעות מרחק מהלונוביס. חישבנו עבור כל דגימה את המרחק מהממוצע והורדנו את הדגימות ב-5% הרחוקים ביותר. כדי להשתמש בדרך זו, ביצענו את הנרמול לפני הורדת הערכים כדי שהסקלות השונות בין הפיצ'רים לא יצרו הטיה בעת חישובי המרחק. בתכנון המקורי הורדת הערכים הקיצוניים בוצעה לפני נרמול הנתונים כי התעסקנו בגודל הערכים לפי האחוזונים שלהם בכל פיצ'ר. פירוט נרחב על כך מופיע בנספחים תחת הכותרת "התמודדות עם נתונים חריגים – דרך שוויתרנו עליה".

הקטנת המימדיות בעזרת PCA או Backward-Selection

מימדיות גבוהה מדי עלולה לגרום לעלייה במאמצים החישוביים בניתוח הנתונים בסט הנתון. בנוסף כמות מימדים גבוהה תגרור צורך ביותר מידע למען שיפור יכולת ההכללה של מודלים על מידע שלא התאמנו עליו. מימדיות הנתונים בתצורתם המקורית אכן גדולה מדי, לכן אנו נפתור אותה באמצעות שימוש באלגוריתם הקטנת מימדים וב- cross validation בעת הרצת המודלים. הקטנת המימדיות תורמת למניעת ה-overfitting על ידי הפיכת המודל לפחות מסובך, בהתאם לטריידאוף בין השונות להטיה.

כדי לפתור זאת, ביצענו PCA ו-Backward-Selection כדי להשוות את ה-MSE עבור סט הוולידציה בכל טכניקה. השתמשנו בטכניקת PCA עם שמירה על 99% מהשונות ונותרנו עם 31 פיצ'רים, בעוד שבטכניקת Backward-Selection נותרנו עם 39 פיצ'רים. מבחינת ה-MSE ההבדלים היו מהותיים: 0.605 ל-Backward ו-0.181 ל-PCA.

לפי התוצאות כדאי היה לבחור ב-PCA, אך הפיצ'רים שנבחרו ב-PCA היו קטגוריות מקודדות בעוד שהפיצ'רים שנבחרו ב-Feature-Selection כללו פיצ'רים מספריים בנוסף למקודדים. הנחנו שיש חשיבות לפיצ'רים המספריים, לכן החלטנו לספוג את השגיאה הגדולה יותר ולבחור ב-Backward-Selection. סיבה נוספת היא שנרצה לשמור על המשמעות המקורית של הפיצ'רים: כשנציג את הפיצ'רים שתרמו הכי הרבה לתוצאות החיזוי, נוכל להתייחס לפיצ'רים ולמשמעותם המקורית בניגוד לתוצאת ה-PCA שבה לא ניתן לדעת את משמעות הפיצ'רים. לאחר שקיבלנו את הפיצ'רים מסט האימון, נחיל אותם על סט הוולידציה וסט הנתונים כולו.

גרף ה-Mallows Cp Score עבור Backward-Selection נמצא בנספחים תחת הכותרת "גרף Mallows Cp Score עבור Backward-Selection".

החלת העיבוד המקדים על סט ה- Test:

ביצענו את השינויים שביצענו על סט הנתונים גם על סט המבחן: ביצענו הוספה והחסרה של הפיצ'רים עליהם דנו בסעיפים הקודמים באופן ידני, ביצענו קידוד למשתנים הקטגוריאליים, מילאנו חוסרים לפי סט הנתונים וביצענו נירמול min-max. נדגיש כי לא ביצענו הורדת מימדים בעזרת אחת השיטות המוזכרות לעיל כנדרש והחלנו את הפיצ'רים שנותרו בצורה ידנית.

הרצת והערכת מודלים:

עבור כל המודלים ביצענו Grid-Search כדי למצוא מבין מספר אפשרויות את ההיפר-פרמטרים הטובים ביותר. כמו כן ביצענו Cross-Validate כאשר K=5 וחישבנו את תוצאת ה-ROC-AUC הממוצעת בין כל חמשת החישובים. מימשנו גרפים לכל מודל המכילים את ה-AUC של כל חמשת החישובים והם מוצגים, יחד עם קטעי הקוד ופירוט לגבי ההיפר-פרמטרים והשפעתם על ההטיה והשונות בנספחים תחת הכותרת "הרצת והערכת מודלים". בנוסף, פירוט רחב יותר לגבי ההטיה, השונות, ויכולת ההכללה של כל מודל נמצא במחברת הפרויקט לאחר הרצת כל מודל.

Logistic Regression:

בניגוד לפעם הראשונה, הפעם התעסקנו עם ההיפר-פרמטרים של המודל. ביצוע העיבוד המקדים תרם לכך שקיבלנו ציון AUC הממוצע על סט הוולידציה של 0.8371, כלומר חיזוי דגימה ממוצע של 83.71%. הציון הממוצע על סט האימון הוא 0.8393. ניתן לראות שהשינויים המתוארים תרמו לעלייה משמעותית בתוצאת החיזוי. נצפה לראות במודלים המתקדמים תוצאות טובות אפילו יותר עקב המורכבות הרבה שלהם ביחס למודל הרגרסיה הלוגיסטית.

Naive Bayes Classifier:

ביצענו מודל בייס נאיבי, עבורו ציפינו לתוצאת החיזוי הכי נמוכה מבין כל המודלים. במודל זה קיבלנו שציון ה-AUC הממוצע על סט הוולידציה הוא 0.7951, כלומר חיזוי דגימה ממוצע של 79.51%. הציון הממוצע על סט האימון הוא 79.81%. כפי ששיערנו - ציון זה הוא נמוך יותר מציון המודל הלוגיסטי ואנו משערים כי הוא יהיה נמוך גם מהמודלים הבאים.

Multi-Layer Perceptron:

ביצענו מודל רשת נוירונים (MLP) שאנו מניחים שנקבל בו את החיזוי הטוב ביותר, משום שאנחנו מתעסקים בבעיית

סיווג כאשר אין קשר מובהק בין רוב הפיצ'רים. במודל קיבלנו שציון ה-AUC הממוצע על סט הוולידציה הוא 0.9236, כלומר חיזוי דגימה ממוצע של 92.36% ועל סט האימון קיבלנו ציון ממוצע של 93.79%. התוצאות אכן עלו מהמודלים הפשוטים יותר כצפוי, אך אין לזלזל במודל ה-ADABoost וביכולתו לעקוף את מודל זה.

Adaptive Boosting:

ביצענו מודל עצים מתקדם שהנחנו שנקבל גם בו חיזוי טוב, אולי אפילו יותר טוב מ-MLP, מפני שמודל זה יעיל יותר כאשר, כפי שלמדנו בהרצאה. נצפה גם שמודל זה יהיה מהיר יותר ממודל ה-MLP מבחינת זמן הריצה שלו. במודל קיבלנו שציון ה-AUC הממוצע הוא 0.9595, כלומר חיזוי דגימה ממוצע של 95.95%. על סט האימון קיבלנו ציון ממוצע של 98.17%. ציון החיזוי במודל זה גבוה יותר מציון ה-MLP ומודל זה כרגע מוביל לבצע את התחזית על סט המבחן.

Voting Classifier (חלק שלא נלמד בקורס):

ביצענו מודל אנסמבל - מודל המשתמש בכמה מודלים (כפי שלמדנו בדומה למודל עצים אקראיים) ומצביע על סיווג או הסתברות לסיווג על פי רוב הצבעות בין המודלים. הנחת היסוד במחשבה על מודל זה הייתה שנוכל לשלב את השיקולים השונים של כל מודל ולתת להם את המשקל היחסי כדי שיגיעו לסיווג המדויק ביותר. השתמשנו במודלים הקודמים, למעט מודל בייס נאיבי שנתן חיזוי נמוך מהשאר, וכדי להישאר עם מספר מודלים אי-זוגי במקרה של שוויון הצבעות. השתמשנו בהיפר פרמטרים המיטביים שנבחרו בחיפוש גרידי עבור כל מודל. קיבלנו שציון ה-AUC הממוצע על סט הוולידציה הוא 0.9371, כלומר חיזוי דגימה ממוצע של 93.71% ועל סט האימון קיבלנו ציון ממוצע של 94.7%. ציפינו שמודל זה יניב את התוצאות הטובות ביותר עקב שילוב המודלים עם ההיפר-פרמטרים המיטביים שמצאנו, ועקב חלוקת משקולות נבונה בין המודלים השונים. בניגוד לציפיות, מודל זה לא היה יותר טוב מה-ADABoost.

בניית Confusion Matrix על מודל ה-ADABoost:

עשינו זאת עבור המודל שסיפק את ציון החיזוי הגבוה ביותר. ההסברים המפורטים לגבי המטריצה נמצאים בנספחים תחת הכותרת "בניית Confusion Matrix על ה-Voting Classifier".

נסכם את התוצאות: אחוז הקבצים שסווגו נכון הוא 89.57%. אחוז ה- False Negative - 0.049%, אחוז ה- False Positive - 0.055%. המודל שלנו מסווג יותר קבצים כדזוניים מאשר מפספס קבצים דזוניים באמת. סיווג קובץ לא דזוני כדזוני עשוי לגרום למידע לגיטימי לא לעבור, אך מדיניות זו עדיפה במקרה שלנו בדומה למה שהוסבר בהרצאה: עדיף למנוע מעבר של קובץ אם יש חשד לגביו מאשר להבליג ולהכניס קובץ דזוני למערכת.

תיאור פערי הביצועים בין סט הוולידציה לסט האימון ו-overfitting:

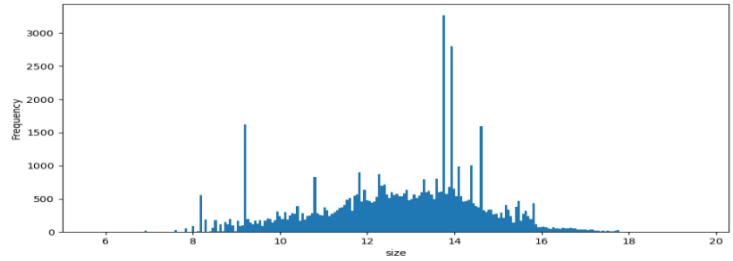
ניתן לראות שהפער בין הציון הממוצע על סט הוולידציה ועל סט האימון אינו גדול אך גם אינו זניח – 2.22%. ניתן בכל זאת להסיק כי המודלים אינם מותאמים יתר על המידה: תוצאת הוולידציה אינה נמוכה משמעותית מציון האימון ושניהם גבוהים מאוד. מסתמן כי הטיית המודל קיימת אך היא קטנה, ושונות המודל אינה גבוהה כלל. הפיצול לסט אימון וולידציה וביצוע cross-validate יחד עם grid search תרמו לשיפור יכולת ההכללה ומיגור ה-overfitting במודל. ניתן לבדוק אילו היפר-פרמטרים או ערכים שלא בדקנו עבורם עשויים לתרום להפחתת הפער בין הציון הסט לציון על סט הוולידציה במודלים. מנגד, לא ניתן לקבוע באופן ודאי שיכולת ההכלה היא טובה משום שפונקציית המטרה האמיתית אינה ידועה לנו, אבל אנו מאמינים כי הציונים שהתקבלו הם הוכחה טובה לכך שהיכולת איכותית.

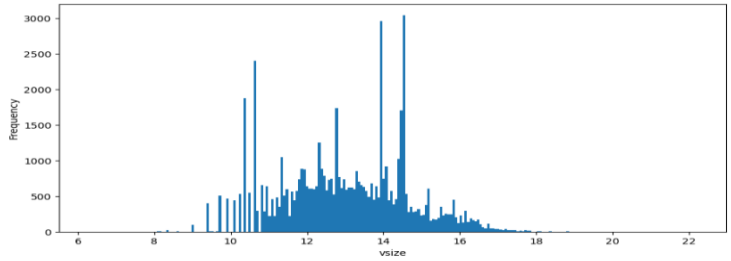
בניית גרף מידת חשיבות הפיצ'רים:

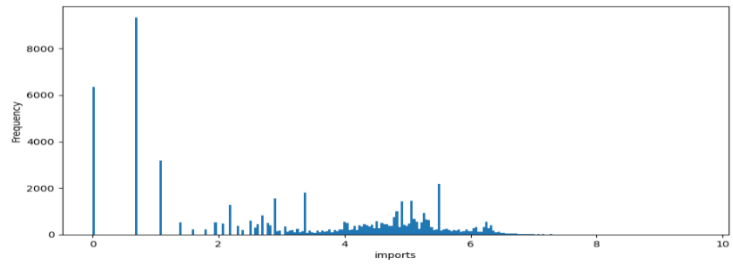
עבור מודל ה-ADABoost, בדקנו אילו פיצ'רים תורמים לחיזוי של המודל עבור דזוניות הדגימות. מצאנו ש-3 פיצ'רים עיקריים תורמים באופן המירבי לחיזוי: 'B', 'printables', 'imports'. ראוי לציין כי במקום הרביעי נמצא הפיצ'ר שיצרנו 'size_ratio'. קבצים דזוניים פועלים בדרכים שונות ורבות, ומהתוצאות ניתן לשער כי קבצים דזוניים ייבאו איתם כמות פונקציות גדולה שלא קיימות במערכות הפעלה סטנדרטיות. כמו כן, קבצים דזוניים עשויים לנסות ולנפח את הזיכרון בנגיש במערכת מסוימת, לכן אין זו הפתעה שיחס הגודל האמיתי והגודל הווירטואלי הגיע גבוה ברשימה.

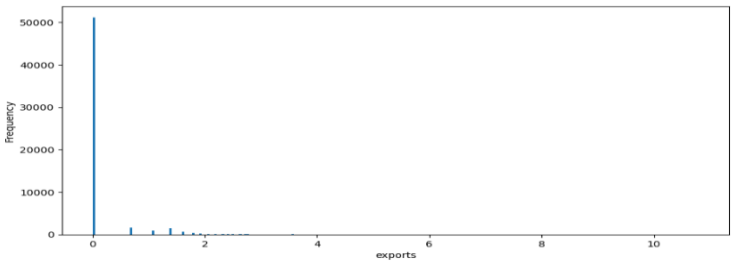
נספחים:

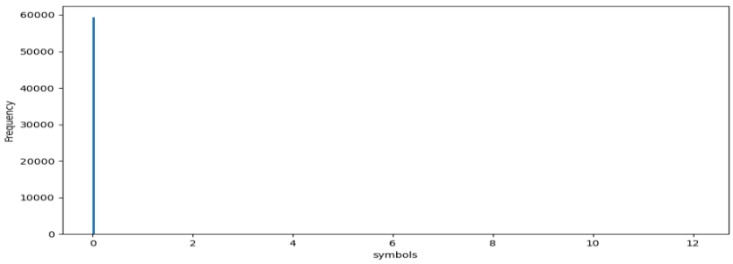
אקספלורציה – גרפים:

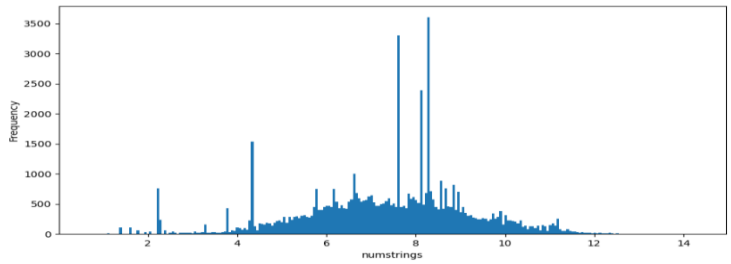


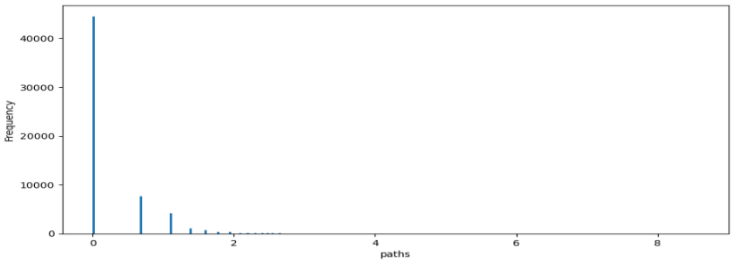


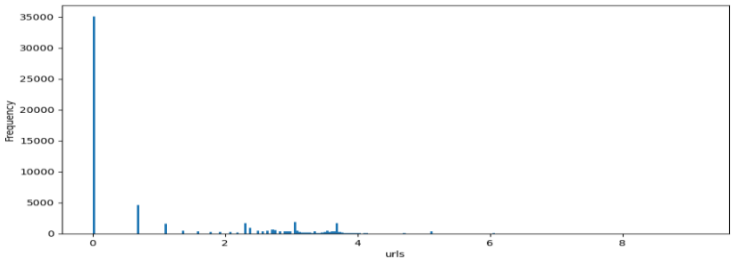


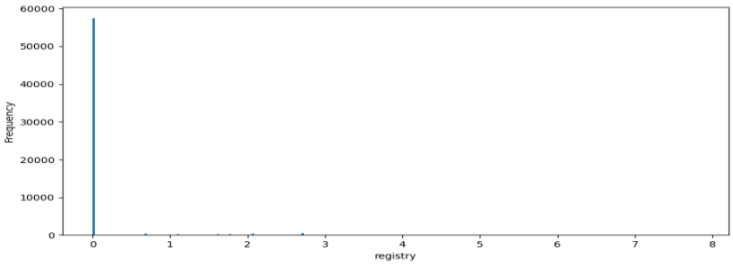


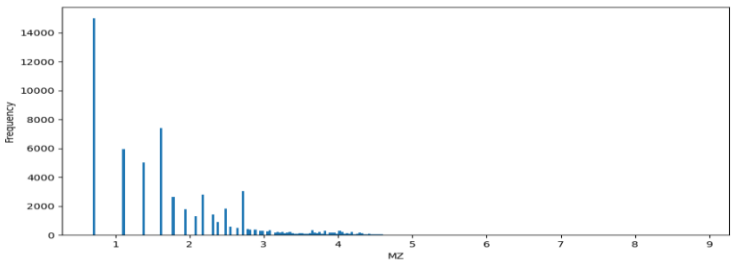


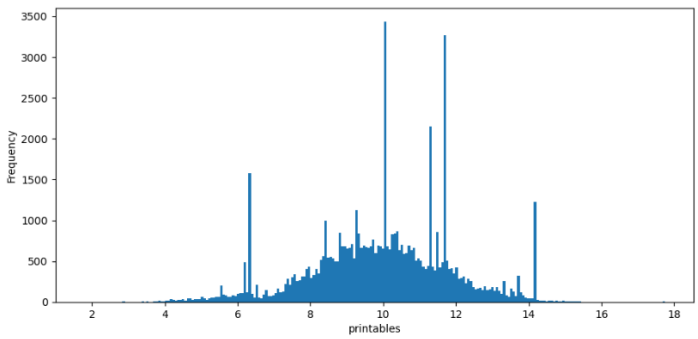


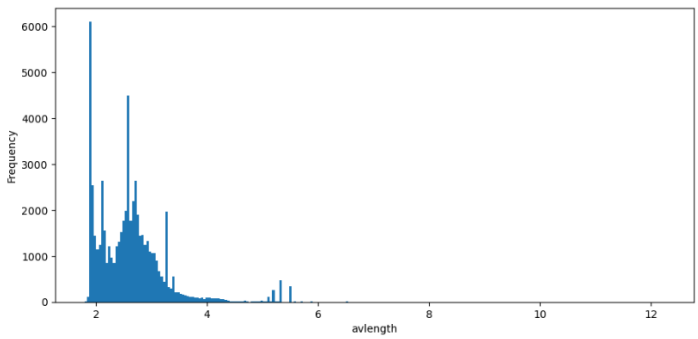


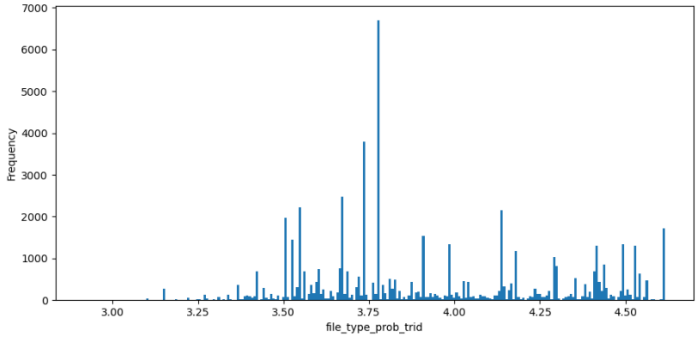


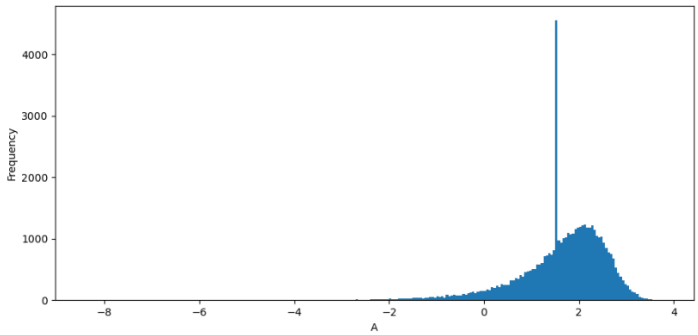


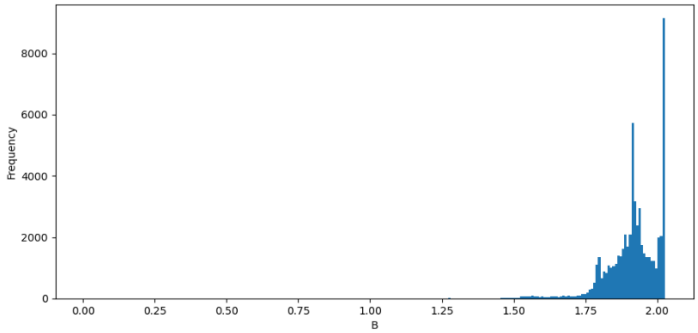


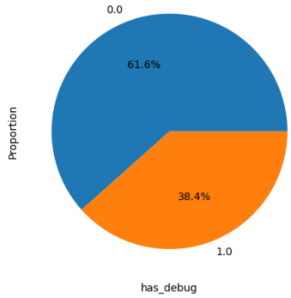

Frequency
printables

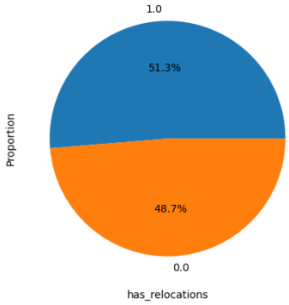

Frequency
avlength

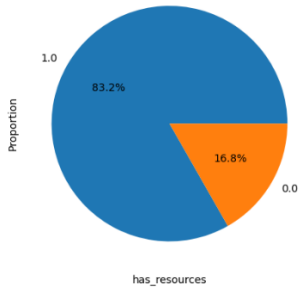

Frequency
file_type_prob_trid

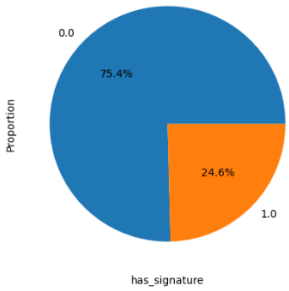

Frequency
A

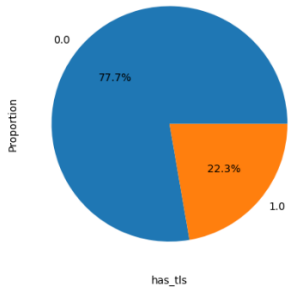

Frequency
B

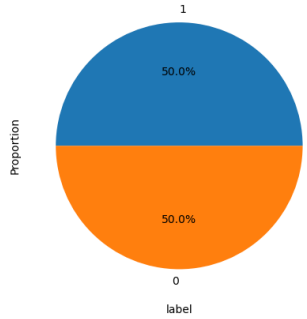

0.0
61.6%
38.4%
1.0
Proportion
has_debug

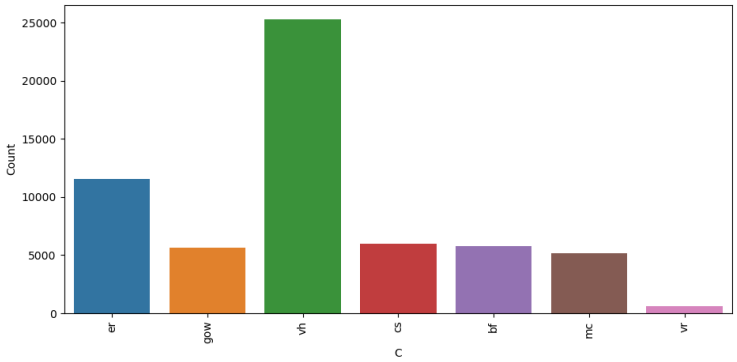

1.0
51.3%
48.7%
0.0
Proportion
has_relocations

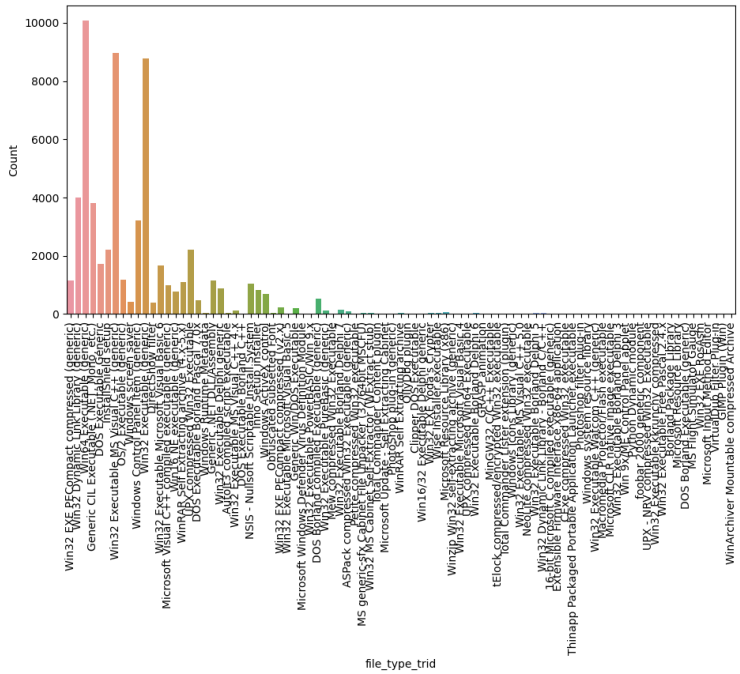

1.0
83.2%
16.8%
0.0
Proportion
has_resources

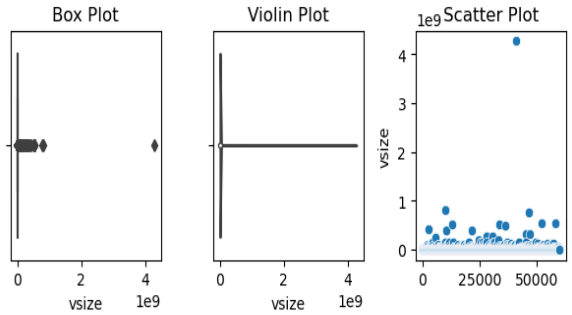

0.0
75.4%
24.6%
1.0
Proportion
has_signature

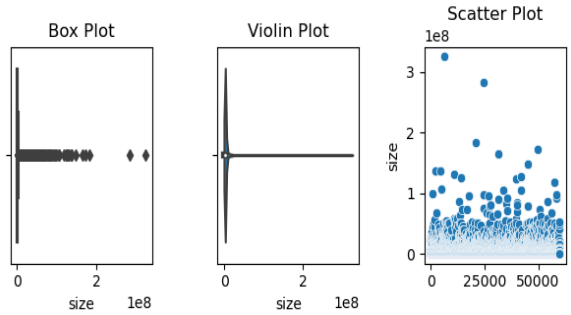

0.0
77.7%
22.3%
1.0
Proportion
has_tls

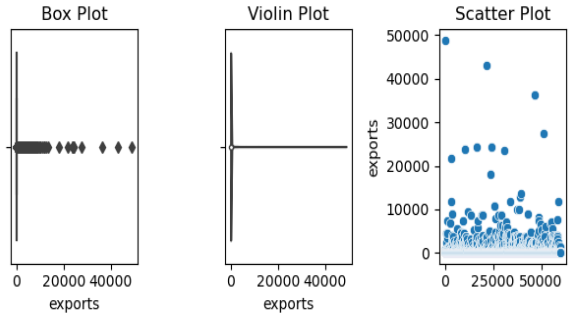

1
50.0%
Proportion
50.0%
0
label

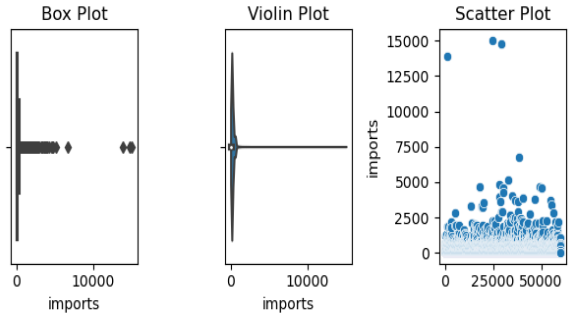

Count
er
gow
vh
cs
bf
mc
vr
c

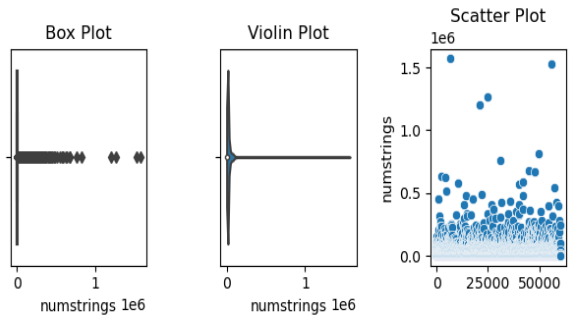

Count
file_type_trid


Box Plot
Violin Plot
Scatter Plot
vsize
1e9

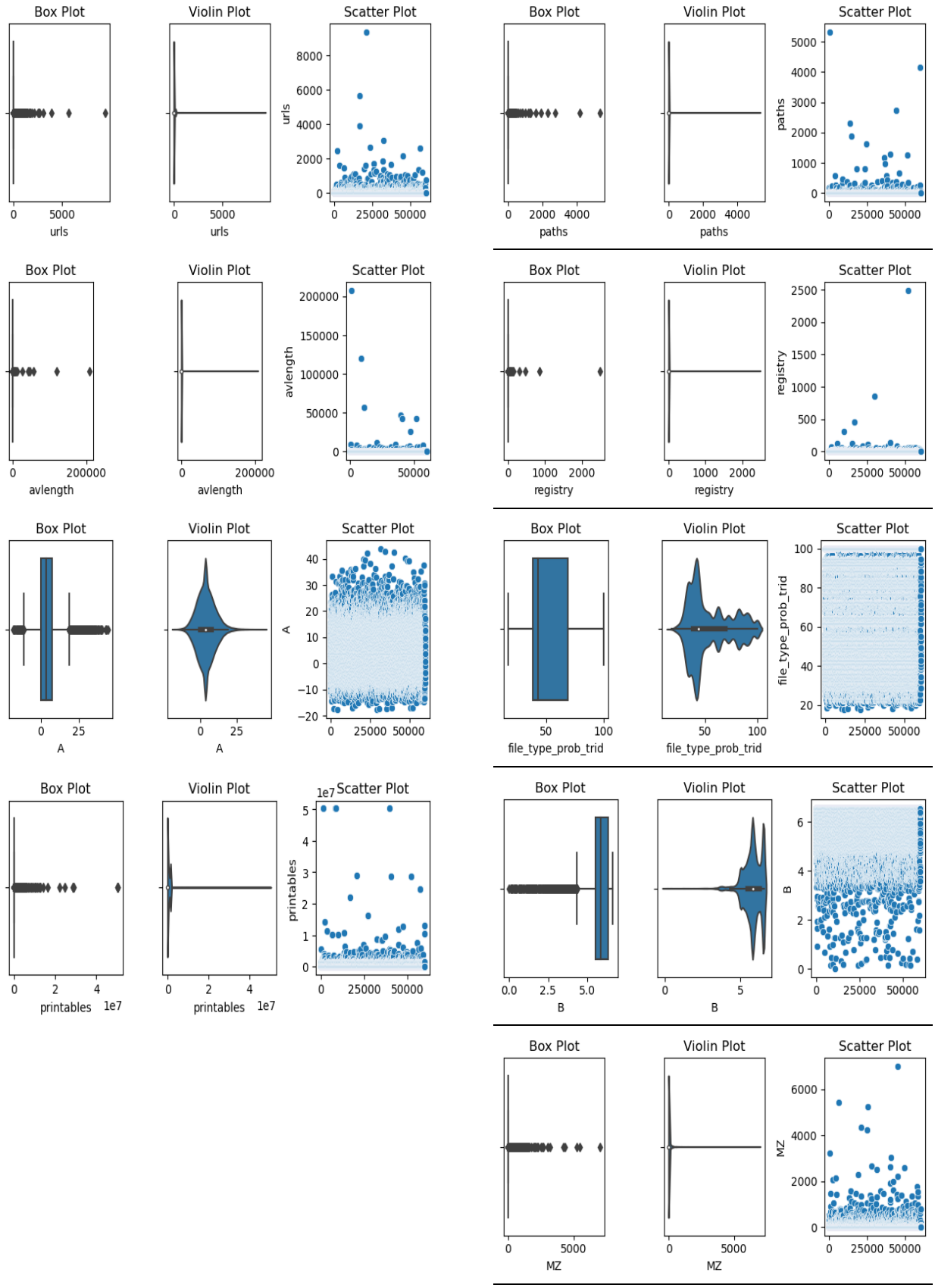

Box Plot
Violin Plot
Scatter Plot
size
1e8


Box Plot
Violin Plot
Scatter Plot
exports


Box Plot
Violin Plot
Scatter Plot
imports

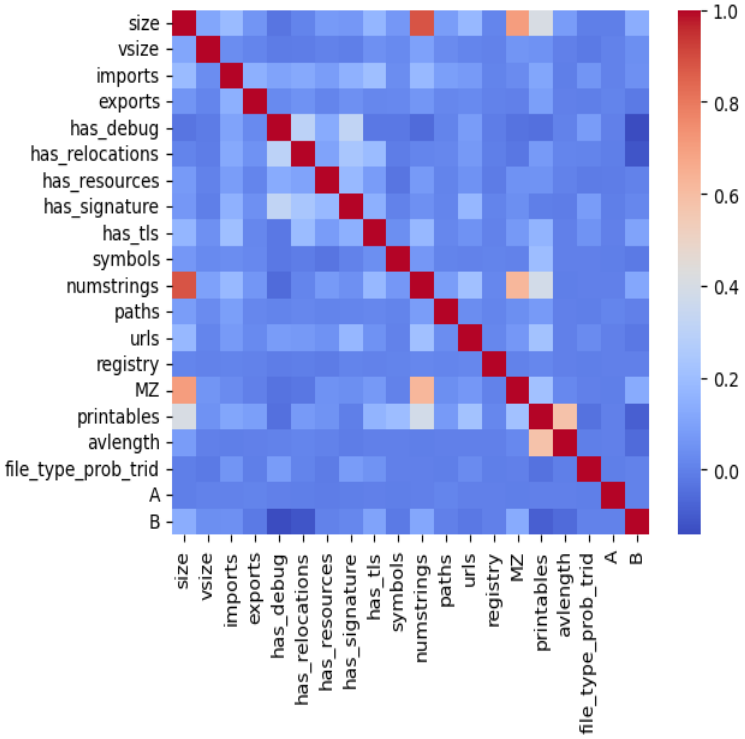

Box Plot
Violin Plot
Scatter Plot
numstrings
1e6


Box Plot
Violin Plot
Scatter Plot
symbols

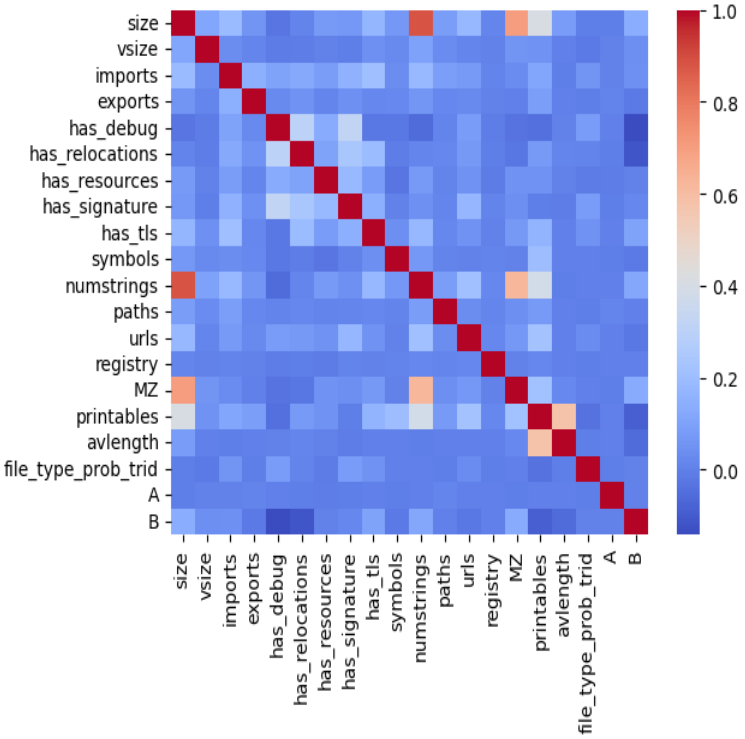

Box Plot
Violin Plot
Scatter Plot
urls
urls
urls
Box Plot
Violin Plot
Scatter Plot
paths
paths
paths
Box Plot
Violin Plot
Scatter Plot
avlength
avlength
avlength
Box Plot
Violin Plot
Scatter Plot
registry
registry
registry
Box Plot
Violin Plot
Scatter Plot
A
A
A
Box Plot
Violin Plot
Scatter Plot
file_type_prob_trid
file_type_prob_trid
file_type_prob_trid
Box Plot
Violin Plot
Scatter Plot
printables
1e7
printables
1e7
1e7
printables
Box Plot
Violin Plot
Scatter Plot
B
B
B
Box Plot
Violin Plot
Scatter Plot
MZ
MZ
MZ

מפת קורלציות בין הפיצ'רים:

לפני מילוי חסרים:



אחרי מילוי חסרים:



מילוי ערכים חסרים:

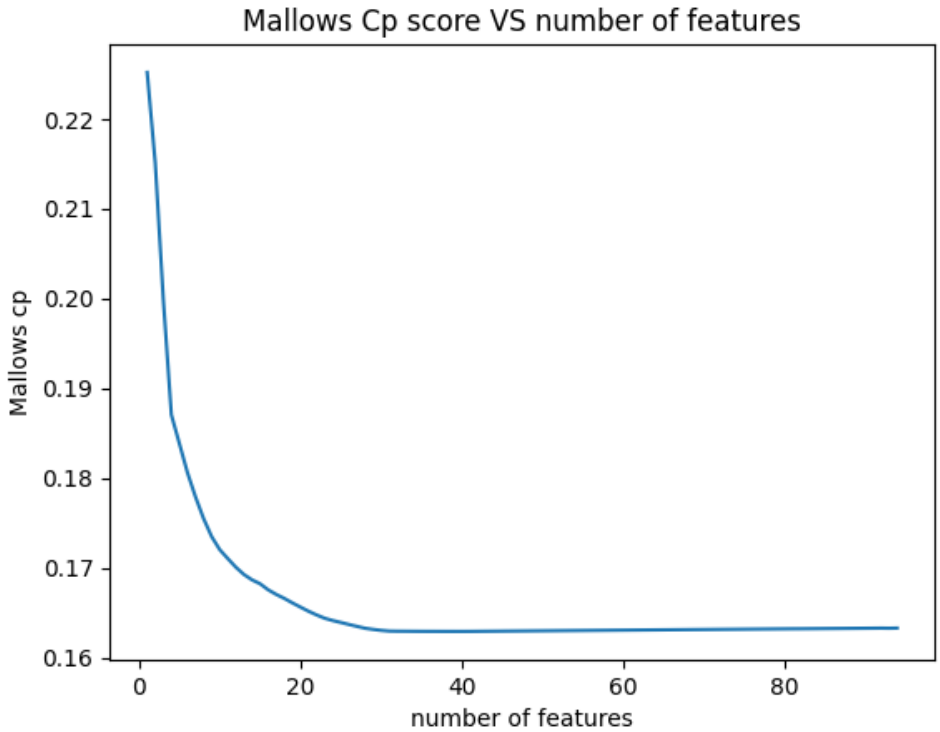
- בפיצ'רים הרציפים הבאים בחרנו למלא את ערכים עם החציון מפני שטווח הערכים בהם רחב ולא נרצה לשנות את התפלגות הנתונים: A, imports, exports, numstrings, avlength, paths, urls, MZ.
- בפיצ'רים הבאים (רציפים) בחרנו למלא 0 בערכים החסרים מפני שהרוב המוחלט של הערכים בהם הוא 0: symbols, registry.
- בפיצ'רים הבינאריים הבאים נמלא 0 כי לא ניתן לדעת איזה ערך אמור להיות עבור כל דגימה. לכן אנחנו יוצאים מנקודת הנחה שאם הוא לא מופיע, הערך אמור להיות 0. הפיצ'רים הם: has_debug, has_resources, has_relocations, has_tls, has_signature.
- בפיצ'רים הרציפים הבאים בחרנו למלא את הממוצע כי יש מעט מאוד ערכי קצה והממוצע לא יחזק את השפעת אותם ערכי קצה: B, printables.
- בפיצ'ר האנונימי C (קטגוריאלי) נבדוק איזו קטגוריה הכי נפוצה ונבחר למלא איתה את החסרים.
- בפיצ'ר vsize (רציף) שמנו לב שהערכים לרוב גדולים מהערך בפיצ'ר Size של הקובץ אך לא ברור מה משפיע על כך. אינטואיטיבית הפיצ'ר קשור לגודל האמיתי של הקובץ ולכן נשתמש ב-KNN (כמות השכנים הרצויה תהיה שורש כמות הדגימות בנתונים) כדי למלא את הערכים החסרים בפיצ'ר vsize על פי הפיצ'ר size.

התמודדות עם נתונים חריגים – דרך שוויתרנו עליה

הצגנו במטריצה את כמות החריגים בנתונים עבור הפיצ'רים המספריים, עבור ערכי הקצה בקפיצות של 2% בעשירון העליון והעשירון התחתון. עשינו זאת בעזרת טבלת הסטטיסטיקה המצביעה על ערכי הסף של כל אחוזון.
לא רצינו להתייחס לפיצ'רים הבינאריים, הקטגוראלים והפיצ'ר ההסתברותי: file_type_prob_trid מפני שערכי הקצה בפיצ'רים אלו לא קיימים (הבינאריים) או שאינם רלוונטים לניתוח החריגים.
לא רצינו להוריד חריגים מה-size וה-vsize מכיוון שעשוי להיות קשר בין ההפרשים ביניהם לסיווג הקובץ, כפי שציינו מקודם.

השתמשנו בטבלאות המתוארות כדי להחליט מה ערך הסף שממנו נרצה להסיר וכמה דוגמאות נמחק עקב כך.
הסתכלנו על ההפרשים בכמויות בין האחוזונים: כאשר ההפרש יהיה גדול יותר, סימן שההבדל בין ערכי הסף גדול יותר ולכן ניתן יהיה להסתכל על ערך הסף של האחוזון המתאים מביניהם כגבול הקובע האם ערך מסוים בפיצ'ר הוא חריג ויוסר או לא.
הסטטיסטיקה הזו היא אבסולוטית ואין לחזור עליה, מכיוון שמחיקת של נתונים משפיעה על ערכי הקצה (ישתנו) ולכן ערכי הקצה שנקבע שלכל פיצ'ר יהיו קבועים גם אחרי המחיקה - ולכן אין צורך לבדוק את הסטטיסטיקה שוב אלא רק להשלים את הכמויות שנותרו להחסיר משאר הפיצ'רים.

גרף Mallows Cp Score עבור Backward Selection:



הרצת והערכת מודלים:

היפר פרמטרים:

1. **Logistic Regression:**

```
clf = LogisticRegression(max_iter=100, solver='newton-cholesky', penalty='l2', C=1.0,
fit_intercept=True, class_weight=None, random_state=None, solver='lbfgs',
max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None,
l1_ratio=None)
```

הסולבר נבחר ככזה בעקבות המלצת תיעוד המודל לבחור בסולבר הזה לדאטא שיש לו מספר גבוה משמעותית בדגימות מאשר מכמות הקטגוריאלים - כפי שמתאים למקרה שלנו.
מקסימום איטרציות נבחר על 100 כי אנחנו מצפים מהמודל עם הסולבר הנ"ל להתכנס מהר (זו גם ברירת המחדל).
את כל השאר השארנו כברירת מחדל מפני שלא ראינו צורך לשנות אותם (למשל כמו C-ענישה שהקטנתו תביא להתאמת יתר, או L2 כרגולריזציית ברירת מחדל מפני שהיא נפוצה יותר (מרחק ריבועי ולא אבסולוטי). חלק מההיפר פרמטרים לא ניתנים לבחירה כי לא השתמשנו בסולבר ליניארי כמו intercept_scaling, dual וחלק כי לא צריך לשנות אותם כמו class_weight – הדגימות מתחלקות באופן שווה בין שתי הקטגוריות.

2. **Naive Bayes Classifier:**

```
clf = GaussianNB(priors=None,var_smoothing=1e-9)
```

מכיוון שאין לנו שום מידע מקדים על התפלגות הקבצים במציאות בכלל, לא נותן לקבוע את משתני הפריור.
לכן, השארנו את ערכי ההיפר-פרמטרים המודל כערכי ברירת המחדל.

3. **Multi-Layer Perceptron:**

```
clf = MLPClassifier(alpha=0.0001, batch_size='auto', beta_1=0.9, beta_2=0.999, epsilon=1e-08,
max_iter=200, learning_rate_init=0.001, n_iter_no_change=10, random_state=123, solver='adam',
learning_rate ='constant', power_t=0.5, shuffle=True, tol=1e-4, verbose=False,
warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False,
validation_fraction=0.1)

params = {'activation': ['relu','tanh'],
        'hidden_layer_sizes': [(100,), (50,100,),]
       }

mlp_classif_grid = GridSearchCV(clf, param_grid=params, cv=5, scoring='roc_auc',n_jobs=2)
```

הסולבר נבחר כמו בברירת המחדל, תיעוד הפונקציה המליץ עליו כבעל ביצועים טובים עבור קבצי דאטא גדולים.
המשתנים המופיעים יחד איתו, למעט מקסימום איטרציות, הם ברירת המחדל (עבור הסולבר).
מקסימום האיטרציות נקבע ב-200 (ולא מעליו כדי לא ליצור התאמת יתר), ואילו מתחת לכמות זו מאבדים מכוחו של המודל לבצע הרבה איטרציות עבור התכנסות. ראוי לציון כי כחלק משלבי הניסוי והטעיה עם ה-GridSearch, לאחר שהזנו לאלגוריתם להשתמש ב150, 200, ו-300 איטרציות, קיבלנו כי גם ב-200 וגם מעל לכך אין התכנסות לפתרון. בכל זאת בחרנו ללכת על 200 איטרציות בטענה שהשגיאה מספיק נמוכה והמודל קרוב למינימום לוקאלי (בסופו של דבר אנחנו גם מוגבלים בכמות האיטרציות האפשריות). לכן בחרנו לדבוק ב-200 כקבוע.

עבור המשתנים הבאים הרצנו חיפוש גרידי כדי למצוא את הערכים הנותנים עבורנו את החיזוי הטוב ביותר:
הפעלת הפונקציה הלא-לינארית: טנגנס או רליו.
שכבות חבויות: שכבה אחת של 100 או 2 שכבות כשהראשונה 50 והשנייה 100.
נמצא שרליו עדיף יחד עם 2 השכבות. בחרנו באקטיבציה כיוון שהיא קובעת את התנהגות הרשת כולה ואנחנו לא רוצים להגביל את היכולת של רשת הנוירונים ללמוד תבניות מורכבות בנתונים. כמות השכבות החבויות והגודל שלהן גם היא משליכה באופן מהותי על ביצועי המודל לכן רצינו לתת לזה לפחות שתי אופציות שונות.

4. **AdaBoost:**

```
clf = AdaBoostClassifier(algorithm='SAMME.R',random_state=None)

params = {
   'n_estimators': [50, 200],
   'learning_rate': [0.1, 0.5],
   'base_estimator': [DecisionTreeClassifier(max_depth=2),
                      DecisionTreeClassifier(max_depth=3),
                      DecisionTreeClassifier(max_depth=5)]
          }

ada_classif_grid = GridSearchCV(clf, param_grid=params, cv=5,
scoring='roc_auc',n_jobs=2)
```

עבור מודל עצים מסוג אדא-בוסט רצינו לבדוק מספר היפר פרמטרים ספציפיים אשר לדעתנו משפיעים הכי הרבה על ציון החיזוי. בשביל למצוא את הערכים הכי טובים עבורם ביצענו חיפוש גרידי:
כמות אמדנים: 50 (ערך ברירת המחדל) או 200 (ערך גדול יותר) שמאפשר למודל להיות מורכב יותר.

גודל הצעד לשיפור: 0.5 (ערך ברירת המחדל) או 0.1 שהינו קטן יותר ומאפשר למודל לעשות צעדים קטנים יותר ולהגיע לשגיאה מינימלית יותר כפי שהוצג בהרצאה.
עומק העץ: בחירה בין עומקים שונים: 2, 3 ו-5.
נמצא שהערכים הטובים ביותר:
כמות אמדנים של 200, גודל צעד של 0.1 ועומק של 3.
צעד קטן יותר מביא אותנו למינימום מקומי בסבירות גבוהה יותר מאשר צעד גדול יותר, ועומק של 3 יחד עם 200 אמדנים מאפשר מודל מורכב יותר ולכן להגיע לחיזוי טוב יותר (התייחסות להתאמת יתר נמצאת בחלק הרביעי בעבודה).

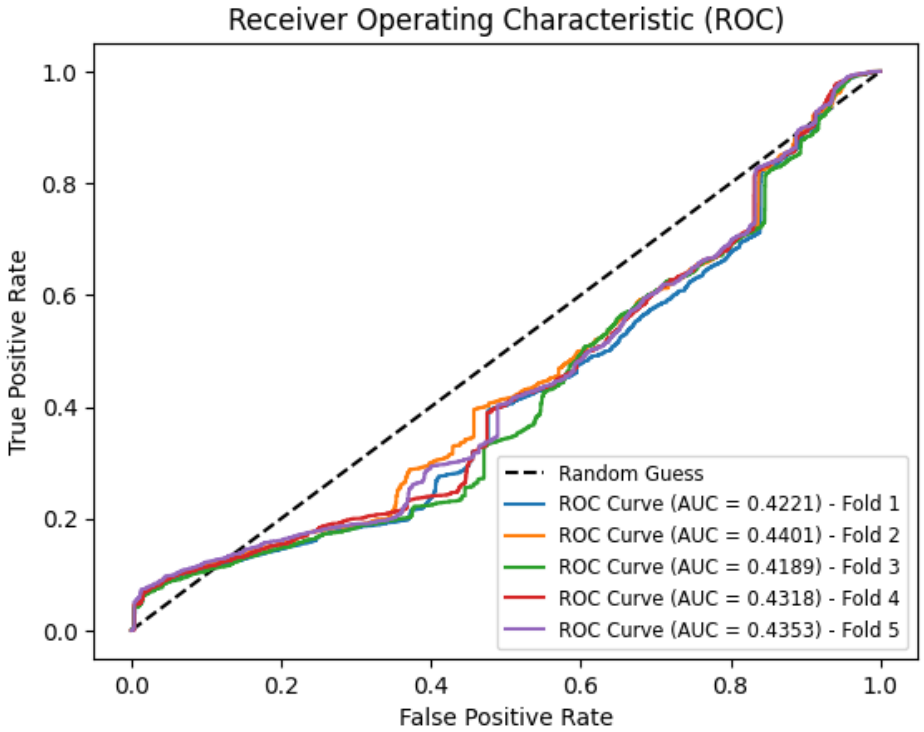
5. **Voting Classifier:**

```
# 1) Logistic Regression = lr
lr = LogisticRegression(max_iter=100, solver='newton-cholesky')
# 2) AdaBoost = ada
ada = AdaBoostClassifier(DecisionTreeClassifier(max_depth=3), learning_rate=0.1,
n_estimators=200)
# 3) MLPClassifier = mlp
mlp = MLPClassifier(activation='relu', hidden_layer_sizes= (50, 100))

clf = VotingClassifier(estimators=[('lr',lr),('ada',ada),('mlp',mlp)], voting='soft')

params = {
   'weights': [(.2, .4, .4), (.1, .4, .5), (.1, .5, .4)]
}
vot_classif_grid = GridSearchCV(clf, param_grid=params, cv=5, scoring='roc_auc',n_jobs=2)
```

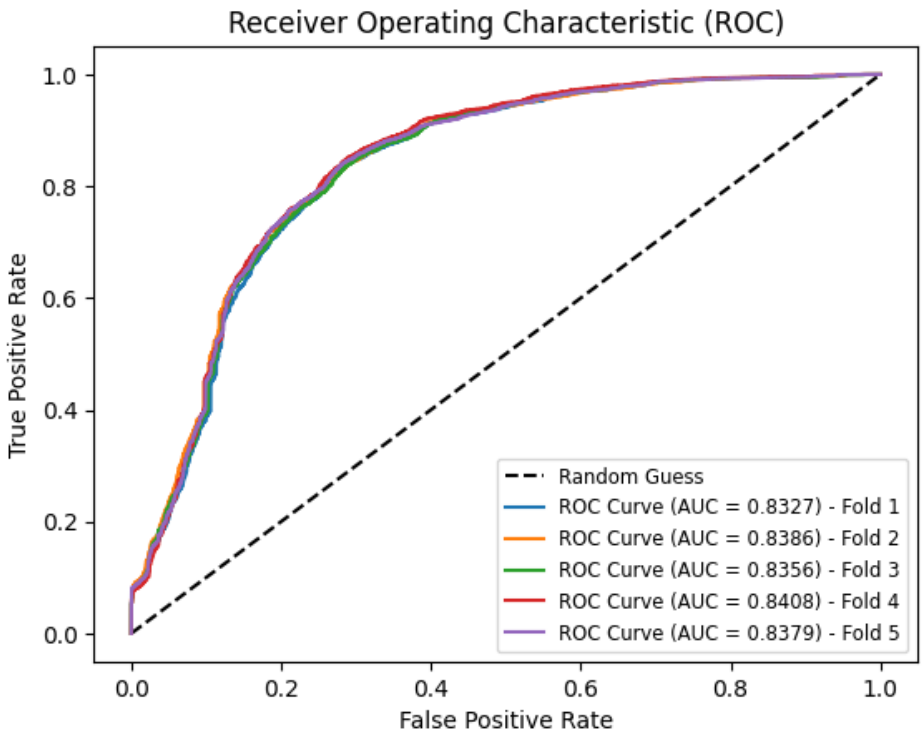
שימוש במודלים הקודמים לצורך קביעת היפר פרמטר "אמדנים".
קבענו לכל מודל כזה את הערכים האופטימליים עפ"י מה שמצאנו לעיל.
לאחר ריצה, לא ניתן היה לתת פרדיקציה עבור היפר פרמטר: הצבעה - 'קשוחה' (ברירת המחדל) ולכן שינינו להצבעה 'רכה' (הצבעה 'קשוחה' - לוקח את הרוב עפ"י האמדנים בעוד שהצבעה 'רכה' מבצעת ממוצע על הסתברויות מכל מודל עבור כל סיווג ובוחרת בגבוה מביניהם).
בנוסף נריץ חיפוש גרידי עבור משקלים שונים עבור המודלים כדי לתת חשיבות עפ"י התוצאות שכבר קיבלנו: מודל לוגיסטי יקבל את החשיבות הנמוכה ביותר, בעוד שנבדוק האם לתת חשיבות שווה לאדא-בוסט ולרשת הנוירונים או לתת לאחד מהם עדיפות.
נמצא שעדיף לתת לאדא-בוסט עדיפות (שבו גם כן קיבלנו את הציון הגבוה ביותר מבין כל השלושה) ולכן נבחר: (0.4, 0.5, 0.1) עבור היפר פרמטר זה.

גרפים של ROC-AUC עבור כל מודל:

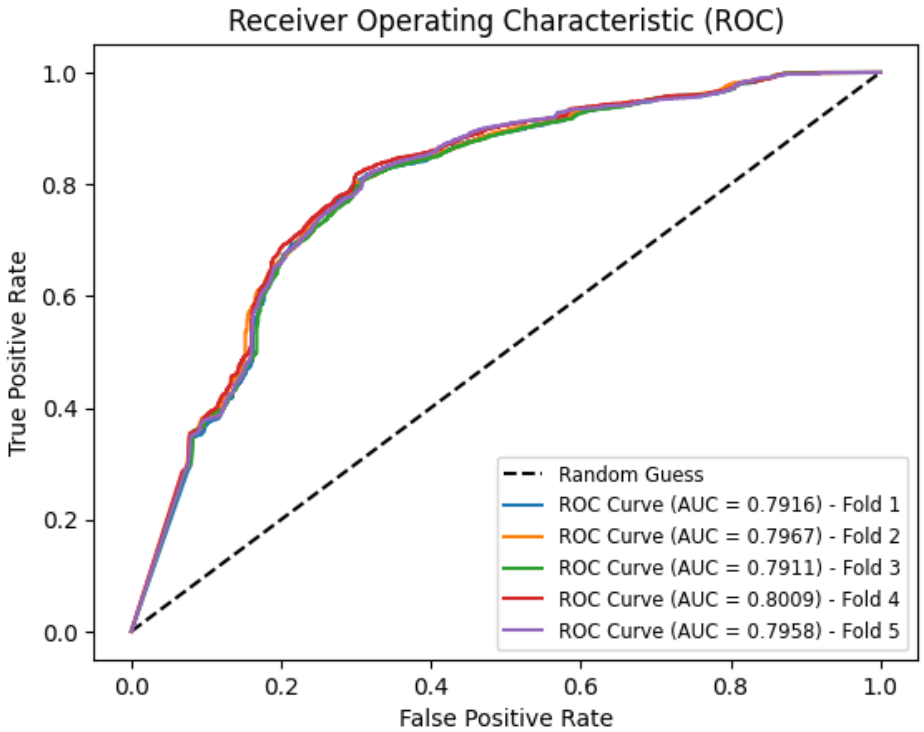
a.1 **Logistic Regression (Without Changes on the Data):**



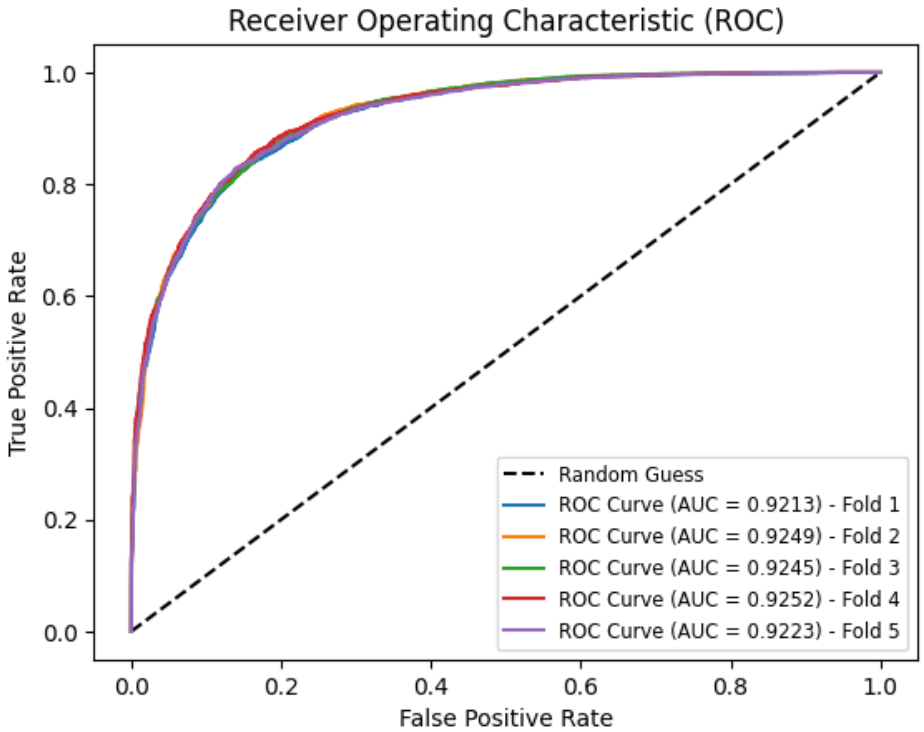
b.1 **Logistic Regression:**



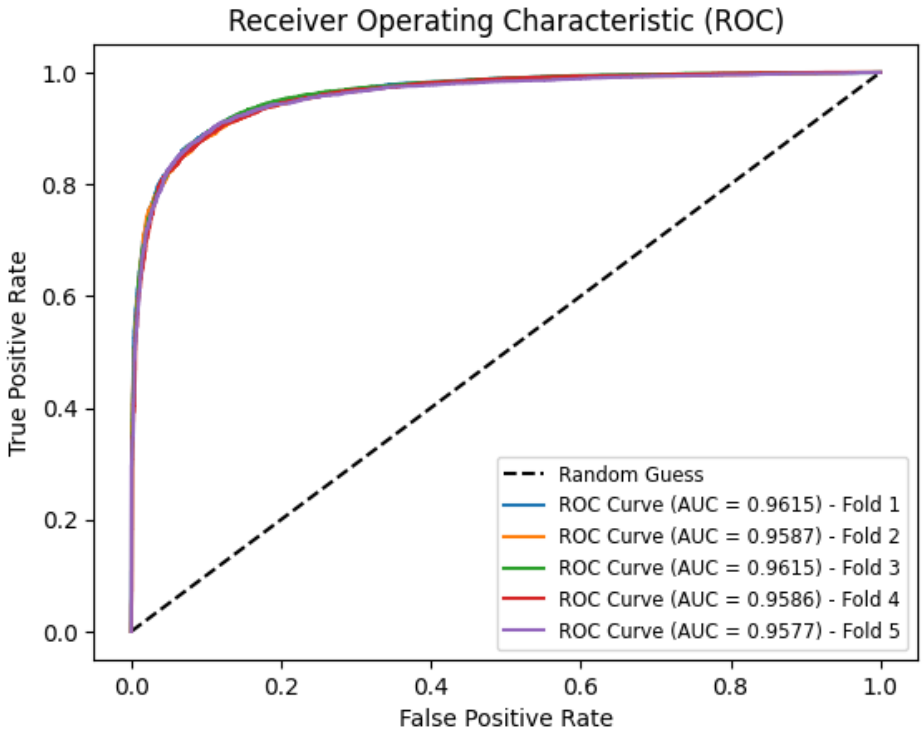
:Naive Bayes Classifier .2



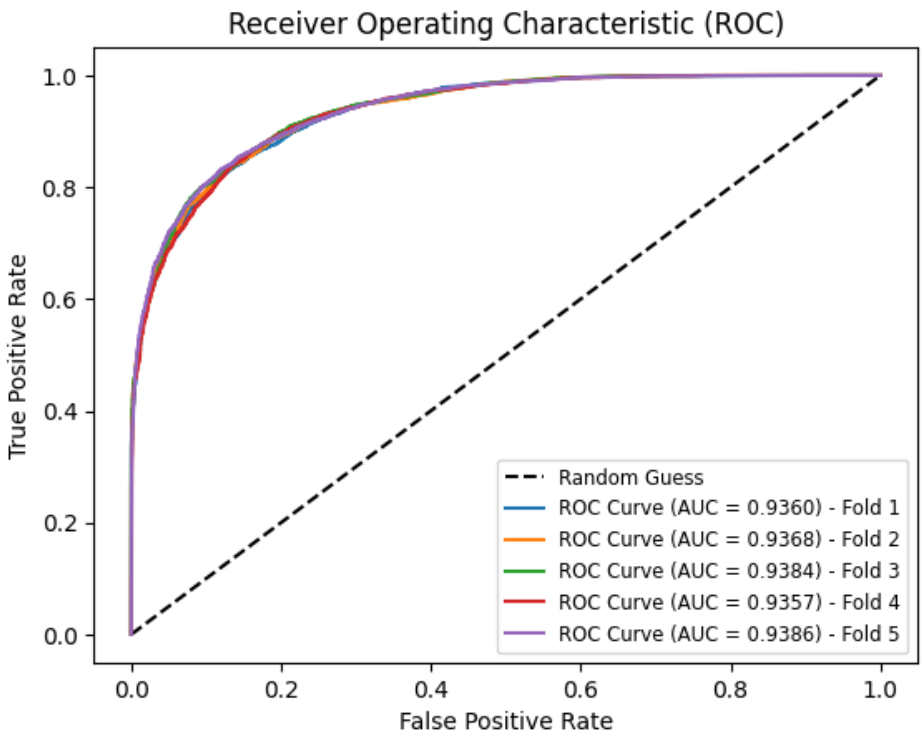
:Multi-Layer Perceptron .3



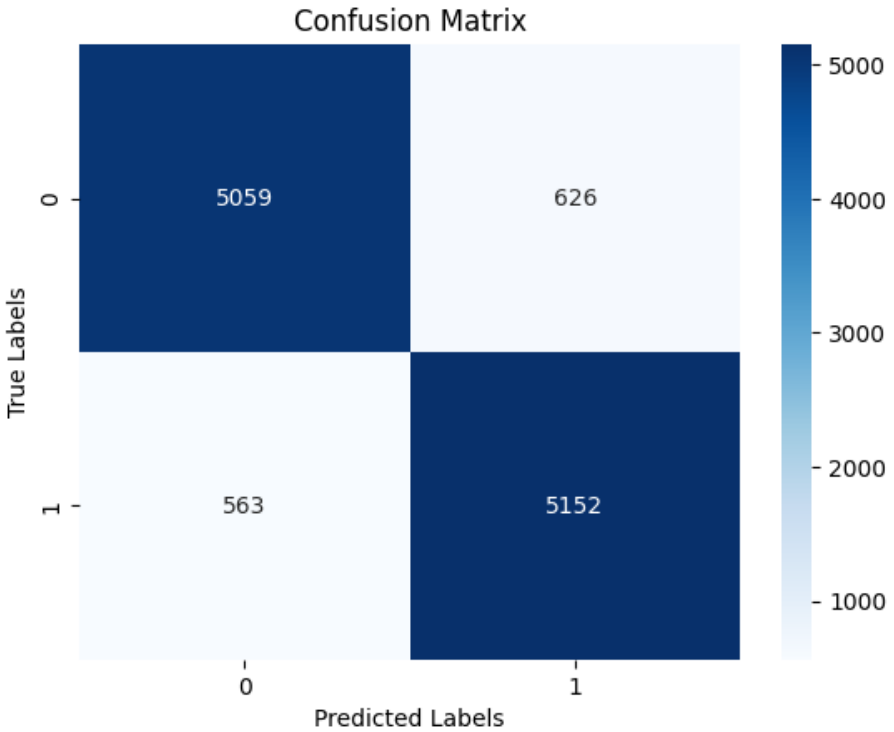
:AdaBoost .4



:Voting Classifier .5



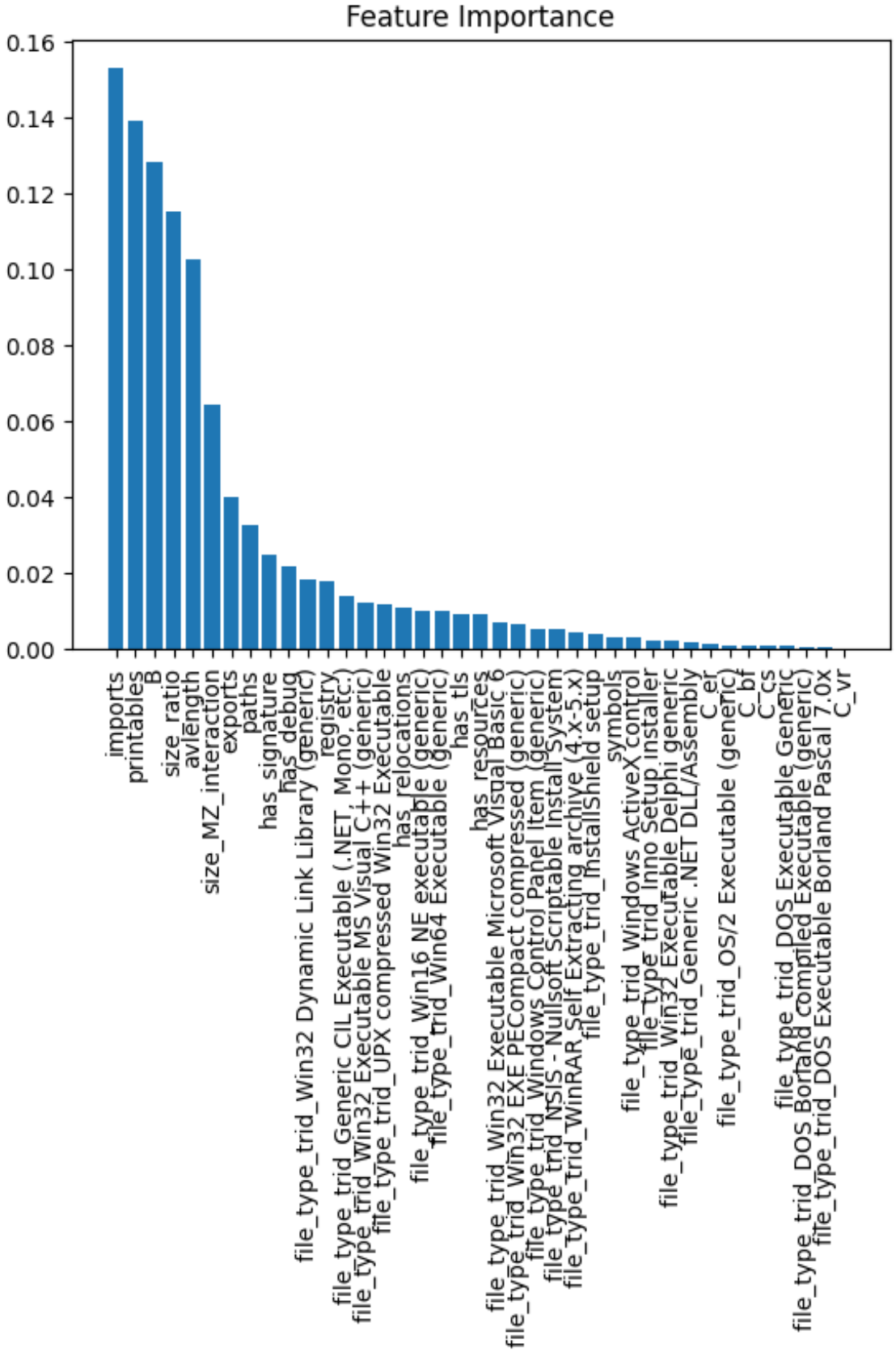
:Confusion Matrix – Voting Classifier



ניתוח Confusion Matrix על ה-ADABoost

המטריצה מראה לנו את הסיווגים הנכונים והלא נכונים על סט הוולידציה, ומחלקת את המטריצה ל-2 עמודות: עמודה שמאלית - כמה דגימות קיבלו חיזוי של קובץ לא זדוני, כאשר המספר העליון מראה בכמה צדקנו (True Negative). המספר התחתון מראה בכמה טעינו, כלומר הקבצים סווגו כלא זדוניים למרות שהם במציאות אכן זדוניים (False Positive). טעות זו היא הטעות החמורה ביותר, כי לא נרצה לסווג קובץ כבטוח כשהוא לא ולסכן את המערכת. בעמודה ימנית נמצא החיזוי שלנו לקבצים זדוניים: המספר התחתון מראה כמה קבצים זדוניים אכן סווגו כזדוניים (True Positive). המספר העליון מראה בכמה דגימות טעינו, כלומר קבצים שסווגו כזדוניים למרות שהם לא (False Positive). במקרה הנ"ל נאמר, כפי שנטען מוקדם יותר, כי עדיף לסווג קובץ לא זדוני כזדוני מאשר לסווג קובץ זדוני כלא זדוני ולהעמיד את כל המערכת בפני סיכונים.

גרף מידת חשיבות פיצ'רים (Voting Classifier):



תרומת כל שותף לפרויקט

רוב הפרויקט נעשה בשיתוף פעולה מלא וללא פיצולי משימות. עבודה על הפרויקט התבצעה בין פעם לפעמיים בשבוע למשך מספר שעות ובהן נכתבו קטעי הקוד של המודל ותועדו ההתקדמות והפעולות שבוצעו בקובץ וורד משותף. ככל שסוף הסמסטר התקרב, כשבועיים לסופו, פגישות העבודה על הפרויקט נהיו קשות להוצאה לפועל והעבודה שנותרה חולקה בין השותפים לפרויקט. לפני שלב זה היה קוד שעבד ורץ מתחילת הפרויקט ועד סופו, אך לפי שאלות בפורום והבנות תאורטיות הוכרע כי היו תהליכים שלא בוצעו כראוי ולכן היה צורך לחזור ולשפץ את הפרויקט בחלקים כמו אופן הסרת ערכי הקיצון או כתיבה מחדש של פונקציות.

כאשר העבודה, שהייתה בשלבים כמעט סופיים שלה (שבוע אחרון לסמסטר):
חולקה בין השותפים, רן לקח על עצמו להתעסק יותר בקטעי הקוד: הוא הפך קטעי קוד שהיו בתאים במחברת לפונקציות, התעסק בפתירת בעיות שנוצרו בעיבוד המקדים בכל הנוגע לאינדקסים של מערכי הנתונים השונים וקידוד המשתנים הקטגוריאליים, ועוד. החלק החשוב הוא שהוא תמיד שמר על מורל גבוה וניגש לעבודה עם חיוך על הפנים, ואף לקח את עניין החיזוי הסופי ברצינות רבה והכניס פן תחרותי שהגיע גם לצוותים אחרים.
עידן מנגד לקח על עצמו בעיקר את החלק המילולי: הוא עבר על כל קטעי ה-markdown במחברת וכתב הסברים תואמים ותיעודים לכל הפעולות שבוצעו בקוד. כמו כן הוא לקח את שלד התיעודים שנכתב בוורד המשותף והפך אותו לדוח המוגמר המוצג כאן לפניכם. מלבד זאת, היה זמין ועזר למציאת פתרונות לבעיות שצצו בקוד, שכן סיעור המוחות עזר מאוד לשנינו. ראוי לציין כי החלוקה בוצעה באופן כזה משתי סיבות: הראשונה היא העדפה אישית – כל אחד פנה אוטומטית למה שהוא מעדיף לבצע. הסיבה השנייה היא שלעידן היה מבחן לפני סוף הסמסטר ולכן מתוך התחשבות יתרה, רן ביצע שינויים הכרחיים בקוד למען התקדמות הפרויקט אל מצבו הסופי.

העבודה על הפרויקט הייתה מלאה בעליות ומורדות, הן מבחינת הביצועים והן מבחינת המוטיבציה, אך בנימה אישית המטלה כולה הייתה מהנה ומלמדת ושותפות העבודה הייתה נפלאה. ביצוע עצמי של פרויקט שלם מההתחלה ועד הסוף, כמובן בהדרכת סגל הקורס והוראות הפרויקט, היה חוויה חשובה ותורמת לפעולות ושיקולים שעשויים להתבצע וכנראה מתבצעים בתהליכים אמיתיים בתעשייה והשכלנו רבות.