הפקולטה להנדסה
ע"ש איבי ואלדר פליישמן
אוניברסיטת תל אביב

פרויקט למידת מכונה

דו"ח הפרויקט

מרצה: דור בנק
מתרגל: אילן וסילבסקי

קבוצה מספר 44

שגיא חולי 314696550
סיון יצחקי 207232570

תקציר מנהלים:

פרויקט זה עוסק בבעיית Binary Classification אשר מטרתה לחזות רכישה או אי-רכישה של לקוח בביקור באתר אי-קומרס (e-commerce). זאת באמצעות ארבעה מודלים: Logistic Regression, KNN, ו-Random Forest Decision Tree.

בפרויקט חקרנו את הנתונים ע"י בדיקת נתונים סטטיסטיים, קורלציות וגרפים, מהם הסקנו תובנות רבות בעלות ערך רב על סט הנתונים. לאחר מכן, ביצענו עיבוד מקדים לסט האימון באמצעות נרמול הנתונים, הסרת ערכים חריגים, מילוי ערכים חסרים ויצרנו פיצ'רים חדשים אשר נראו לנו רלוונטיים לסט הנתונים ולמשימת החיזוי. בהמשך, בחנו דרכים שונות להורדת המימדיות של הבעיה כגון PCA ו-Feature Selection. בסיום שלב זה, השתמשנו ב-Grid search עם Cross Validation בכדי לבחור את ההיפר פרמטרים אשר נותנים את ה-AUC האופטימאלי עבור כל מודל. להערכת המודל הטוב ביותר, התבססנו על K-Fold Cross Validation , כך שלבסוף בחרנו במודל בעל ה-mean test AUC הגבוה ביותר, שלאחריו בוצעה בדיקת Overfitting, וזאת על מנת לחזות את ערכי הנתונים שצורפו לנו בקובץ test.csv בצורה הטובה ביותר. המודל הנבחר הוא Random Forest. את תוצאות החיזוי ייצאנו לקובץ Submission_group_44.csv.

חלק ראשון- אקספלורציה:

בשלב זה בוצעו ניתוחים ויזואליים על הדאטה, בכדי לאפיין את הנתונים ואת הקשר ביניהם, ולהבין כיצד נפעל לגביהם. סט הנתונים מכיל 23 פיצ'רים ו-10,479 תצפיות. כל תצפית בעלת סיווג (lable) של 1 או 0, כאשר 1 מסמל רכישה ו-0 מסמל אי-רכישה. בהמשך, חילקנו את הפיצ'רים לכאלו המכילים ערכים מספריים-רציפים, ולכאלה המכילים ערכים קטגוריאליים, אך גילינו כי פיצול זה עדיין משאיר אותנו עם פיצ'רים האמורים לייצג ערכים קטגוריאליים ('Region', 'device' 'purchase',) ועל כן, עבורם החלטנו לבצע את החלוקה באופן ידני.

כמו כן, בעת החלוקה החלטנו להוציא את עמודת 'id' מכיוון שאינה מייצגת לנו דבר מלבד מספר סידורי, דבר ש-data frame עושה באופן אוטומטי. בנוסף, בדקנו את כמות הערכים הייחודיים, וכן את כמות הערכים החסרים. בדקנו כיצד כל פי'צר מתפלג, כיצד הוא מתנהג אל מול פיצ'רים אחרים וכיצד נראית חלוקת הלייבלים. מבדיקה זאת קיבלנו כי כ-85% מהתצפיות הן בעלות סיווג 0, וכ-15% מהתצפיות הן בעלות סיווג 1. ניתן להסיק שעובדה זו עשויה להשפיע במידה מסוימת על התחזיות שיבוצעו בהמשך על ידי מודלי החיזוי ועל האופן בו נעריך את טיבם של המודלים, שכן, אם מודל מסוים חוזה נכונה 0.85 מהתצפיות, נוכל לדעת שבאותה מידה יכולנו לסווג את כלל התצפיות כ-0 ועדיין לקבל את אותה תוצאה.

<u>ניתוח הדאטה של העמודות המספריות</u> – שמנו לב כי הפיצ'רים: 'info_page_duration' ו-'product_page_duration' מכילים ערך מספרי (רציף) עם סיומת "minutes", כך שמדובר בעמודות מספריות ועל כן, החלטנו בשלב זה להפוך אותן לנומריות, על ידי מחיקת הסיומת. את המספרים שנותרו, המרנו ל-float, כדי שיהיה אפשר לבצע בהם שימוש כחלק מדאטה נומרי.

לאחר מכן, הצגנו טבלה המציגה עבור כל פיצ'ר את הערכים הבאים: ממוצע, סטיית תקן, מינימום, מקסימום ואחוזונים- 25%,50%, 75%. מצאנו כי ישנו הבדל בטווח הערכים ובשונות הפיצ'רים, ולכן, ביצוע הנרמול בשלב הבא הוא בעל חשיבות עבור חלק מהמודלים בהם נשתמש. בנוסף, בדקנו את הקורלציה בין כל הפיצ'רים באמצעות heatmap המסייעת בוויזואליזציה של מטריצת הקורלציה. מכאן ניתן להסיק בין אילו פיצ'רים יש קורלציה גבוהה ולאילו אין, כך שבהמשך נוריד חלק מהפיצ'רים בעלי קורלציה גבוהה. גילינו שהפיצ'רים 'BounceRates' & 'ExitRates', 'product_page_duration' & 'total_duration' & 'num_of_product_pages' ו- 'num_of_product_pages' הם בעלי קורלציה גבוהה בין זה לזה, ו-'purchase' & 'D' הם בעלי קורלציה שלילית גבוהה ביניהם. מסקנות אלו ינחו אותנו בהמשך, בשלב העיבוד המקדים.

כמו כן, עבור כל עמודה נומרית הוצגה היסטוגרמה המציגה בצורה בהירה את התפלגות הנתונים בכל עמודה, ועוזרת לזהות ערכים חריגים הקיימים בעמודה. ניתן לראות שעמודה 'B' מתפלגת נורמלית באופן כמעט מושלם, וכמה מהפיצ'רים האחרים נראים קרובים יחסית להתפלגות log-normal.

ניתוח הדאטה של העמודות הקטגוריאליות – עבור כל פיצ'ר קטגוריאלי הצגנו היסטוגרמה שמציגה את הערכים השונים שהוא מקבל. רצינו לנתח את הפיצ'ר 'internet_browser' על מנת להבין כיצד הנתונים מחולקים מבחינת דפדפנים, אך ראינו כי פיצ'ר זה מכיל בתוכו 126 ערכים שונים, המבטאים גרסאות שונות של אותם 4 סוגי דפדפנים. לאחר חקירה, עלה כי ניתן לחלק ערכים אלו לארבעה ערכים – 'chrome', 'safari', 'edge', 'browser_v' כך שכל ערך המכיל את אחד משמות אלו, ישתנה אליו. לאחר צמצום מספר הערכים ל-4 דפדפנים שונים, הצגנו את התפלגות הכניסות מארבעת הדפדפנים השונים. שמנו לב כי אנשים נוטים להשתמש יותר בדפדפן "chrome" (ראה נספח 6). כמו כן, בדקנו מהם המכשירים מהם מתבצעות הרכישות הרבות ביותר. מהצגת הגרף מתקבל כי בעלי מכשיר מספר 2 רוכשים הכי הרבה, דבר שיכול לסייע בהבנת אופי הרוכשים, כך שבשילוב עם תחזיות המודל, תוכל להיות מתורגמת למסקנות עסקיות.

חלק שני- עיבוד מקדים:

השינויים והטרנספורמציות שהופעלו על ה-train הופעלו גם על ה-test כדי שהסט הסופי שיועבר למודל הפרדיקציה יהיה בעל מאפיינים דומים לאלו שמהם למד המודל, מלבד הוצאת חריגים, אשר צריכה להתבצע רק עבור ה-train, שכן, יש לבצע פרדיקציה לכל הרשומות ב-test.

"תיקון" פיצ'רים- כפי שציינו בשלב האקספלורציה, המרנו את פיצ'רים 'info_page_duration' ו-'product_page_duration' לנומריים ושינינו את 'internet_browser' כך שיכיל רק 4 ערכים ייחודיים. בנוסף, החלטנו להפוך את פיצ'ר 'Weekend' למשתנה בינארי (0/1), במקום TRUE/FALSE, זאת על מנת שבהמשך נוכל ליצור פיצ'רים חדשים, עליהם נדבר בהרחבה בתת נושא- "בניית פיצ'רים חדשים". לפני ההמרה, מילאנו ערכים חסרים עבור 'Weekend' ב-0 תחת ההנחה שרוב ימות השבוע אינם סוף שבוע. כמו כן, ראינו לנכון להוריד את כמות הערכים גם לעמודה 'A', וכך גם פעלנו עבור עמודת 'device', שכן, בשלב האקספלורציה ראינו כי ישנם בעלי מכשירים אשר לא מופיעים כלל בסט ה-train. עבור כל אחת מעמודות אלו בחרנו ליצור ערך חדש "other" שיקבל את כל הערכים שמופיעים מתחת ל-200 פעמים בסט ה-train. באופן זה, איחדנו ערכים שונים עבור שני פיצ'רים אלו. כמות הערכים השונים בעמודה 'A' ירדו מ-96 ערכים שונים, ל-9. כמות הערכים השונים בעמודה 'device' ירדו מ-8 ערכים שונים, ל-5.

מחיקת פיצ'רים- לפי טבלת heatmap שהצגנו, ראינו כי בין 'BounceRates' ו-'ExitRates' יש קורלציה גבוהה (0.91) וכן, בין 'product_page_duration' ו-'num_of_product_pages' יש קורלציה גבוהה (0.86), על כן, כחלק מתהליך הורדת מימדים, החלטנו להוריד מהנתונים גם את עמודת 'BounceRates' ואת עמודת 'product_page_duration'. בדומה לכך, עמודת 'total_duration' נמצאת בקורלציה גבוהה עם 'num_of_product_pages' ובעלת 40% ערכים חסרים, על כן, החלטנו להוריד עמודה זו מהנתונים. נציין כי שקלנו למחוק את עמודה 'D' בשל היותה בעלת 60% ערכים חסרים, אך בעקבות הקורלציה עם 'purchase' החלטנו להשאירה ולמלא ערכים חסרים כפי שנציין בהמשך.

הוצאת חריגים (רק עבור train)- לפני שלב הוצאת החריגים, שמנו לב שבהיסטוגרמה, עמודת 'ExitRates' נראית די ייחודית ולכן החלטנו לבצע עליה פלט של Box-Cox, שם נראה באופן יותר ברור כי יש שתי התפלגויות גאוסיאניות (ראה נספח 7). לכן, עבור עמודה זו לא ניעזר ב-Box plot על מנת להסיר חריגים.

לעומת זאת, עמודה 'B' אכן נראית כמתפלגת נורמלית ועבורה החלטנו להציג Box plot כדי שנוכל לזהות חריגים, אותם הוצאנו בעזרת הפעלת פונקציית zscore עם 3 סטיות תקן, אשר מתאימה לטיפול בנתונים מהתפלגות נורמלית על החלקים שלא כוללים ערכים חסרים. החלטנו שעבור הפיצ'רים 'PageValues', 'num_of_admin_pages' ו-'num_of_product_pages' נוציא חריגים בשיטה שונה. תחילה הצגנו עבור שלושתם stripplot שיציג בצורה ויזואלית את החריגים בכל אחד מהם (ראה נספח 10). לאחר מכן, מצאנו לכל אחד מהם "threshold" שמוגדר לפי numpy.nanpercentile והוצאנו את כל הערכים שמעל אחוזון מסוים. לאחר מספר ניסיונות החלטנו שנשתמש באחוזון ה-99.93 עבור כל עמודה, שכן, אחוזון זה מקסם את מדד ה-AUC עבור המודלים.

מילוי ערכים חסרים- כבר בשלב האקספלורציה ראינו שקיימים ערכים חסרים.
במקום התצפיות החסרות בפיצ'רים נומריים, הכנסנו את החציון מסט הנתונים של ה-train (מכיוון שלא כל הנתונים מתפלגים נורמאלית בחרנו להשתמש בחציון), ובפיצ'רים קטגוריאליים מילאנו את התצפיות הריקות בקטגוריה הכי שכיחה ב-train עבור כל פיצ'ר. בנוסף, כפי שגילינו בשלב האקספלורציה, עמודה 'D' היא בעלת כמות גדולה של ערכים חסרים ומצד שני, בעלת קורלציה גבוהה למדי עם ה-labels. לאחר שניסינו מספר דרכים שונות למילוי הערכים החסרים ללא ערך מוסף משמעותי למודל, החלטנו לטפל בערכים החסרים של עמודה זו באופן שונה- בעזרת KNNimputer, אשר בוחר עבור כל תצפית שמקבלת nan בעמודה D את הממוצע של k התצפיות הקרובות ביותר אליה במרחב. בעזרת שיטה זו יצרנו imputer ש"התאמן" על ה-train data (ללא הלייבלים, כדי למנוע overfitting) והשתמשנו בו כדי לחזות את הערכים החסרים בעמודה 'D', גם על ה-test וגם על ה-train. שיטה זו עדיפה במקרה זה, לעומת השיטות לעיל, מפני שיש לנו מעט דגימות יחסית בסט ה-train שלנו עבור עמודה D. שלב זה בוצע לאחר שהפכנו את המשתנים הקטגוריאליים למשתני דאמי.
נרמול הנתונים: בשלב החקירה, ראינו שלכל פיצ'ר קנה מידה שונה, ולכן יש לנרמל את הנתונים לאותו קנה המידה, מפני שעבור מודלים מסוימים (כגון, KNN, בו נשתמש בהמשך), פיצ'רים בעלי סקאלת ערכים שונה ייגרמו להטיה בתוצאות המודל והשפעה לא פרופורציונאלית לחשיבותם על התוצאות. יתרה מכך, נורמליזציה חשובה מכיוון שעבור חלק מהשיטות להורדת מימדים (כגון, PCA, שבהמשך נפרט מדוע בחרנו לא להשתמש בו) נעשה שימוש בשונות של כל פיצ'ר. באם הפיצ'רים לא באותו קנה מידה, הדבר עלול לגרום להטיה בתוצאות. בסופו של דבר, בחרנו לנרמל את העמודות הנומריות לפי שיטת MinMax: נרמול לפי ערכי ה-min וה-max של סט הנתונים. נרמול זה מביא את הערכים לסקאלה של בין 0 ל-1, וביצענו אותו לאחר שהסרנו ערכים חריגים בכדי למנוע הטיה בסקאלות של הפיצ'רים.
המרת פיצ'רים קטגוריאליים למשתני דאמי- בחרנו להשתמש בפונקציה Get_dummies של פנדה, על מנת להתמודד עם הפיצ'רים שאינם נומריים ואינם בינאריים. הפונקציה מפצלת כל פיצ'ר קטגוריאלי לעמודות לפי מספר הערכים שמכילה העמודה המקורית. כאשר כל תצפית תקבל את הערכים 1 או 0 בפיצ'ר המתאים בעמודות המפוצלות, בהתאם לערך הקיים בעמודה המקורית.
בניית פיצ'רים חדשים- החלטנו לייצר את הפיצ'ר "Weekend*closeness_to_holiday" שמורכב ממכפלת עמודת 'Weekend' ב-'closeness_to_holiday', מתוך מחשבה שאולי לסופי שבוע שקרובים לחגים מסוימים תהיה השפעה על כמות הרוכשים. כמו כן, יצרנו את הפיצ'ר "total_num_of_pages" המורכב מחיבור של הפיצ'רים 'num_of_admin_pages', 'num_of_product_pages', 'num_of_info_pages' על מנת לאחד את כמות הדפים שמשתמש נכנס אליהם.
מימדיות הבעיה – התחלנו את הבעיה עם 23 פיצ'רים שונים. לאחר שימוש בפונקציית Get_ dummies, הוספת ומחיקת פיצ'רים, קיבלנו 53 פיצ'רים. כדי לבדוק האם יש לנו מימדיות גדולה מידי, הרצנו את שני המודלים המורכבים שלנו, אחרי הורדת מימדים ב-feature selection, ואכן, קיבלנו שיפור במדד ה-AUC. לכן, בחרנו להשתמש רק ב-21 פיצ'רים ב-random-forest וב-13 פיצ'רים עבור מודל של עץ החלטה יחיד.
מימדיות גדולה עלולה ליצור בעיה מהסיבות הבאות: ראשית, מספר מימדים רב מידי עלול להקטין את צפיפות התצפיות ולגרום לעלייה בשונות המודל עד כדי overfitting (קללת המימדים). שנית, פיצ'רים רבים יותר גורמים לזמן חישוב ארוך יותר וסיבוכיות גבוהה יותר. כמו כן, קשה להבין מי מבין הפיצ'רים בעלי השפעה רבה יותר לעומת פיצ'רים אחרים בדאטה, וכך תהליך הקלסיפיקציה נהיה מורכב יותר, וכמובן ישנו סיכוי לפיצ'רים קורלטיביים אשר אינם מוסיפים מידע.
זיהוי והקטנת ממדיות הבעיה- בשלב העיבוד המקדים ניסינו לבחון מספר שיטות שונות של הורדת מימדיות:
-קורלציה- כפי שהצגנו בשלב האקספלורציה והורדנו את אחת העמודות הקורלטיביות (הרחבה בתת נושא "מחיקת פיצ'רים").
-PCA- ניסינו לבחון כמה פיצ'רים יבחרו בסף של 95% מהשונות המוסברת, וקיבלנו ש34 פיצ'רים

מספיקים. לאחר מחשבה, החלטנו לא להשתמש ב-PCA מתוך הרצון להיות מסוגלים להסביר את תוצאות המודלים בהמשך על ידי אלגוריתמים מבוססי עצי החלטה.

חלק שלישי- הרצת מודלים

כדי למצוא את הפרמטרים האופטימליים לכל מסווג, נעזרנו בפונקציית GridSearchCV – בכדי לבחון קומבינציות שונות של פרמטרים תוך מקסום על מדד ה-AUC (test).

Logistic Regression- ההיפר פרמטרים שנבחרו:

{'C': 0.1, 'max_iter': 150, 'penalty': 'l1', 'solver': 'liblinear'}

KNN- היפר הפרמטרים שנבחרו -

ההיפר פרמטרים שנבחרו: {'metric': 'minkowski', 'n_neighbors': 46, 'p': 1, 'weights': 'distance'}.

Decision Tree- בחרנו להשתמש במודל זה אף על פי שהוא כנראה לא ייתן לנו את ה-AUC האופטימאלי, מתוך רצון להבנה של התחזיות כך שנוכל להפיק גם מסקנות עסקיות, שכן עץ החלטה הוא מודל בתחום כריית נתונים ותורם להצגה ויזואלית של החלטות (ראה נספח 16).

תחילה בחרנו את ההיפר פרמטר אלפא האופטימלי באמצעות פונקציה cost_complexity_purning_path המחשבת את נתיב הגיזום עבור גיזום מינימלי בעלויות מורכבות. היפר פרמטר זה משמש לגיזום מינימלי תחת הטרייד אוף של עלות-מורכבות המודל. האלפא הכי טוב שהתקבל הוא 0.0006. בהמשך, בחרנו להשתמש בפונקציית GridSearchCV בהינתן האלפא האופטימלי מפני שהניסיון לחשב מספר רב של אלפות, במקביל לשאר ההיפר-פרמטרים, לקח זמן ריצה רב מידי.

היפר פרמטרים שנבחנו הם: ['splitter':['best', 'random'],

['criterion':['gini', 'entropy' , 'log_loss'], וערכים שונים ל'max_depth' ול-'min_samples_split'.

ההיפר פרמטרים שנבחרו:

{'criterion': 'entropy', 'max_depth': 5, 'min_samples_split': 2, 'splitter': 'best'}

במודל זה החלטנו לצמצם את מספר הפיצ'רים באמצעות feature importance (ראה נספח 14). בנוסף, הגרף מציג שהמודל משתמש רק ב-13 פיצ'רים, המעיד על הורדה רבה של מימדים.

לבסוף, הצגנו עבור מודל זה confusion matrix, עליה נרחיב בהערכת מודל, ופלט של עץ החלטה עבור 13 הפיצ'רים החשובים, המסייע בהבנה של הסיווג.

Random Forest- עבור מודל זה בחנו את ההיפר פרמטרים הבאים –

criterion': 'entropy', 'max_features': None, 'min_samples_leaf': 10, 'min_samples_split': 5, 'n_estimators': 250.

גם עבור מודל זה החלטנו לצמצם את מספר הפיצ'רים באמצעות feature importance, בחרנו להשתמש ב21 פיצ'רים מפני שהם מסבירים מעל 98 אחוז מהשונות של המודל ונתנו לנו ציון AUC גבוה יותר על סט ה-validation.

Adaptive Boosting- לא השתמשנו במודל זה בסוף מפני ש-random forest קיבל תוצאות טובות יותר.

חלק רביעי- הערכת המודלים

Confusion matrix- בנינו confusion matrix עבור Decision Tree. במטריצה זו השורה העליונה מייצגת תצפיות עם label=0 (סשנים שלא הסתיימו ברכישה), השורה התחתונה מייצגת תצפיות עם label=1 (סשנים שהסתיימו ברכישה), העמודה השמאלית מייצגת תצפיות שהתחזית לגביהן היא אי-רכישה (label=0) והעמודה הימנית מייצגת תצפיות שהתחזית לגביהן היא רכישה (label=1). כך שהתאים השמאלי העליון והימני התחתון מייצגים תחזיות נכונות (TP,TN), והיתר מייצגים תחזיות שגויות. לפי המטריצה, מרבית הדגימות נמצאות בתא TN- דגימות שסווגו כנכונות עבור אי רכישה. תוצאה זו אינה מפתיעה, שכן מרבית התצפיות בסט הנתונים מייצגות לקוחות שגלשו ולא רכשו, כפי שראינו בשלב האקספלורציה, ולכן את מרבית התצפיות המודל מצליח לסווג

בצורה נכונה. התא השני בגודלו הוא התא הימני התחתון (TP) מאותן סיבות שפורטו קודם. אחריו בגודלו הוא התא השמאלי התחתון (FN), סביר להניח שהמודל התאמן יותר על דגימות עם label=0 ולכן נוטה לסווג לפיו.
ישנם מדדים נוספים שניתן להסיק ממטריצה זו. למשל: מדד precision שמייצג חיזוי נכון של רכישה, הוא גבוה (85%). בנוסף, מדד ה-specificity, שמייצג את יכולות המודל לחזות נכונה סשנים שלא הסתיימו ברכישה, גבוה יותר ממדד ה-sensitivity, שמייצג את יכולת המודל לחזות נכונה סשנים שהסתיימו ברכישה. המשמעות היא פחות "אזעקות שווא", כלומר יהיה חיזוי טוב יותר של אי-רכישה באתר לאחר שגיאה.
K-Fold Cross Validation+ROC- כל מודל, עם היפר פרמטרים שנבחרו עבורו, נבחן על ROC-AUC בפונקציית K-fold, כך שמדד הAUC מחושב לפי ה-test של כל fold, וכך נבנה עבור כל מודל פלט ROC. במהלך בניית הפלט, מדדי ה-mean_fpr_v, mean_tpr_v, mean_auc_v נשמרו, כך שהגרף יכיל את ה-ROC הממוצע של כל מודל בשביל השוואה בין המודלים. המודל הטוב ביותר לפי מדד AUC הינו Random Forest עם AUC=0.931.
Overfitting- כדי להחליט האם קיימת התאמת יתר במודלים, בדקנו את ההפרש בין ערך AUC הממוצע על ה-test של כל Fold לבין ערך AUC הממוצע על ה-train של כל Fold. קיבלנו הפרשים קטנים ולא משמעותיים, ועל כן סביר להניח כי לא קיים overfitting במודלים שלנו.
יכולת ההכללה של המודל ניתנת לשיפור בעזרת מספר דרכים. ראשית, בשלב pre processing, ניתן למלא ערכים חסרים ב-test באופן שונה, בצורה כזו שניקח בחשבון את ההתפלגות ממנה הגיע כל פיצ'ר, כך שנוכל להשלים ערכים חסרים לפי התפלגות זו. כמו כן, בבחירת היפר-פרמטרים, יכולנו לבחון סט רחב יותר של היפר-פרמטרים, במקום להשתמש בחלק מערכי ברירת המחדל של המודלים, או עבור הפרמטרים האופטימליים שנבחרו יכולנו לבחון שוב עבור טווח רחב יותר של היפר פרמטרים (כמו שעשינו בDecision Tree). לא עשינו זאת עקב מגבלות זמן ריצה.

<u>חלק חמישי- ביצוע פרדיקציה</u>- ביצענו תחזית לקובץ ה-test בעזרת random forest עם הפרמטרים והפיצ'רים (21 פיצ'רים סך הכל) אשר מקסמו עבורנו את ה-AUC test.

<u>סיכום</u>
מהות הפרויקט הינה בניית מודל לצורך סיווג בינארי של נתונים. בעזרת החומר הנלמד וחקירה רבה, נעשה שימוש בכלים שונים כדי להגיע למודל המסווג בצורה הטובה ביותר, תוך ניסיון למיקסום מדד ה-AUC. בכדי להגיע למטרה זו, ניסינו מגוון ניסיונות הרצה אשר כללו מודלים שונים, קומבינציות שונות של היפר-פרמטרים עבור כל מודל, השלמות ערכים חסרים בצורות שונות ועוד.
בתחילת הפרויקט ביצענו חשיפה של הנתונים בצורה שתאפשר לנו להפיק מסקנות וללמוד על הנתונים לקראת שלב העיבוד המקדים. במהלך העיבוד המקדים טיפלנו בערכים החריגים, צמצמנו את כמות הערכים הייחודיים בחלק מהפיצ'רים, נרמלנו את סט הנתונים ועבדנו רבות על השלמת הערכים החסרים בצורה שתטיב עם המודלים ככל האפשר. מקרה מעניין ומאתגר היה ההתמודדות עם השלמת הערכים החסרים עבור עמודה D, במהלך ניסיונותינו חיפשנו מגוון שיטות שונות להשלמת ערכים אלו בצורה שתועיל למשימת החיזוי של המודלים, בלי לגרום להטיה או overfitting מיותרים. בשלב הבא, בכדי שנוכל לקבל את המודלים הטובים ביותר נעזרנו ב-GridSearchCV בכדי לנסות קומבינציות שונות של היפר-פרמטרים, תוך אופטימיזציה של מדד ה-AUC. לאחר מכן, הערכנו את המודלים בעזרת K-fold (כאשר לk=3), וכן עבור decision tree ביצענו confusion matrix שהציג סיווג לא רע של TN ושל TP. מהגרף המציג את ROC של כל מודל, הסקנו כי רוב המודלים שבחרנו נותנים ערך AUC יחסית גבוה. המודל הנבחר בסופו של דבר היה random forest, מפני שהפיק את ערך ה-AUC הגבוה ביותר ועל כן, נבחר על מנת לחזות את נתוני ה-test.

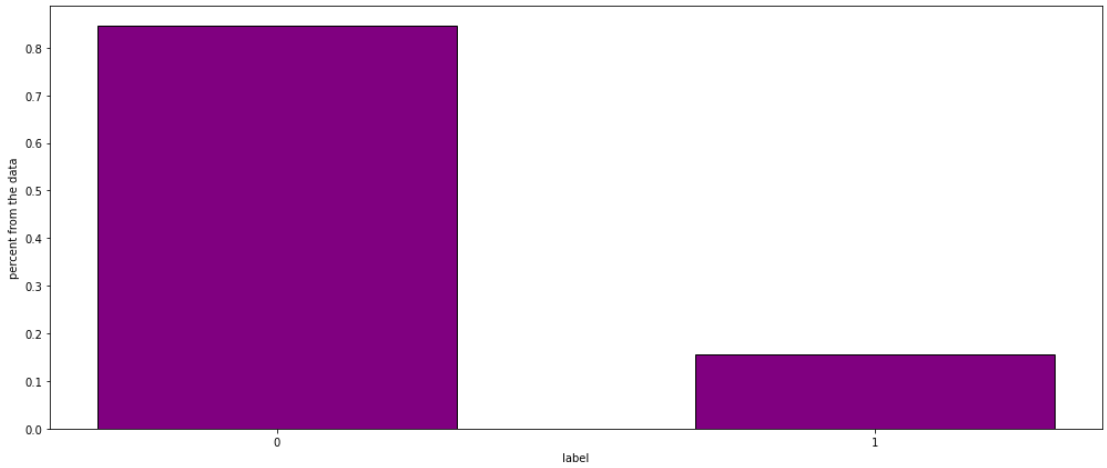
נספחים

<u>אחריות כל שותף ותרומתו לעבודה</u>

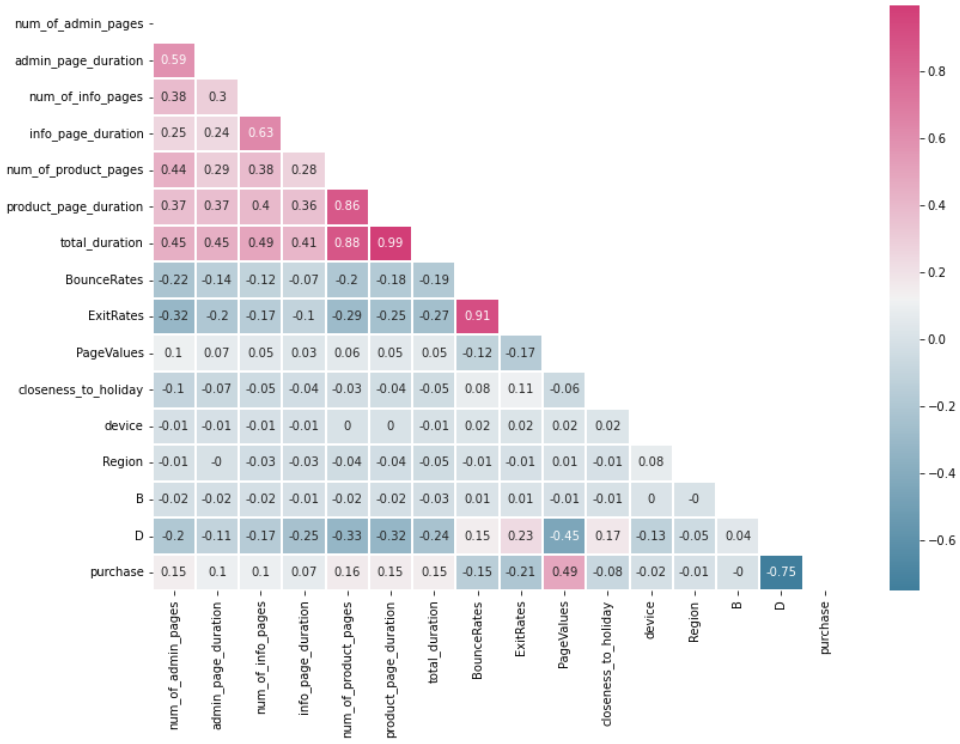
בפרויקט זה שנינו פעלנו בשיתוף פעולה לאורך כל התהליך.
בשלב האקספלורציה שנינו ביצענו ויזואליזציות של הדאטה וחקירה ראשונית שלה. שם הבנו שיש צורך לפצל את הדאטה לנומרית ולקטגוריאלית, וסיוון ביצעה פיצול זה.
בשלב pre- processing כל אחד קיבל תחום אחריות משלו. שגיא היה אחראי על הוצאת חריגים מחיקת פיצ'רים וצמצום ערכים "מיוחדים" לכל פיצ'ר, וסיוון הייתה אחראית על נרמול נתונים וכן המרת פיצ'רים קטגוריאליים לבינאריים ואת מילוי הערכים החסרים ביצענו ביחד. בשלב זה הרבה מהעבודה בוצעה ביחד אף על פי שהייתה חלוקה ברמת האחריות לביצוע.
שלבי הרצת המודלים והערכתם היו שלבים משמעותיים, שכן כל אחד ניסה להריץ מודל וכל פעם שיפר את ההרצה על ידי שינוי בקוד והוספת היפר פרמטר. סיוון הריצה רגרסיה לוגיסטית ו- Random Forest , ושגיא הריץ Adaptive Boosting, KNN. ו-Random Forest. לאחר שהבנו שהדרישה היא Random Forest או Adaptive Boosting, בחרנו בRandom Forest שנתן AUC טוב יותר, ולכן שגיא הריץ את מודל עצי החלטה יחד עם תצוגה ויזואלית של עצי החלטה והוסיף confusion matrix ו- kfold-cv-plot להערכת טיב המודלים.

<u>הוויזואליזציה</u>:

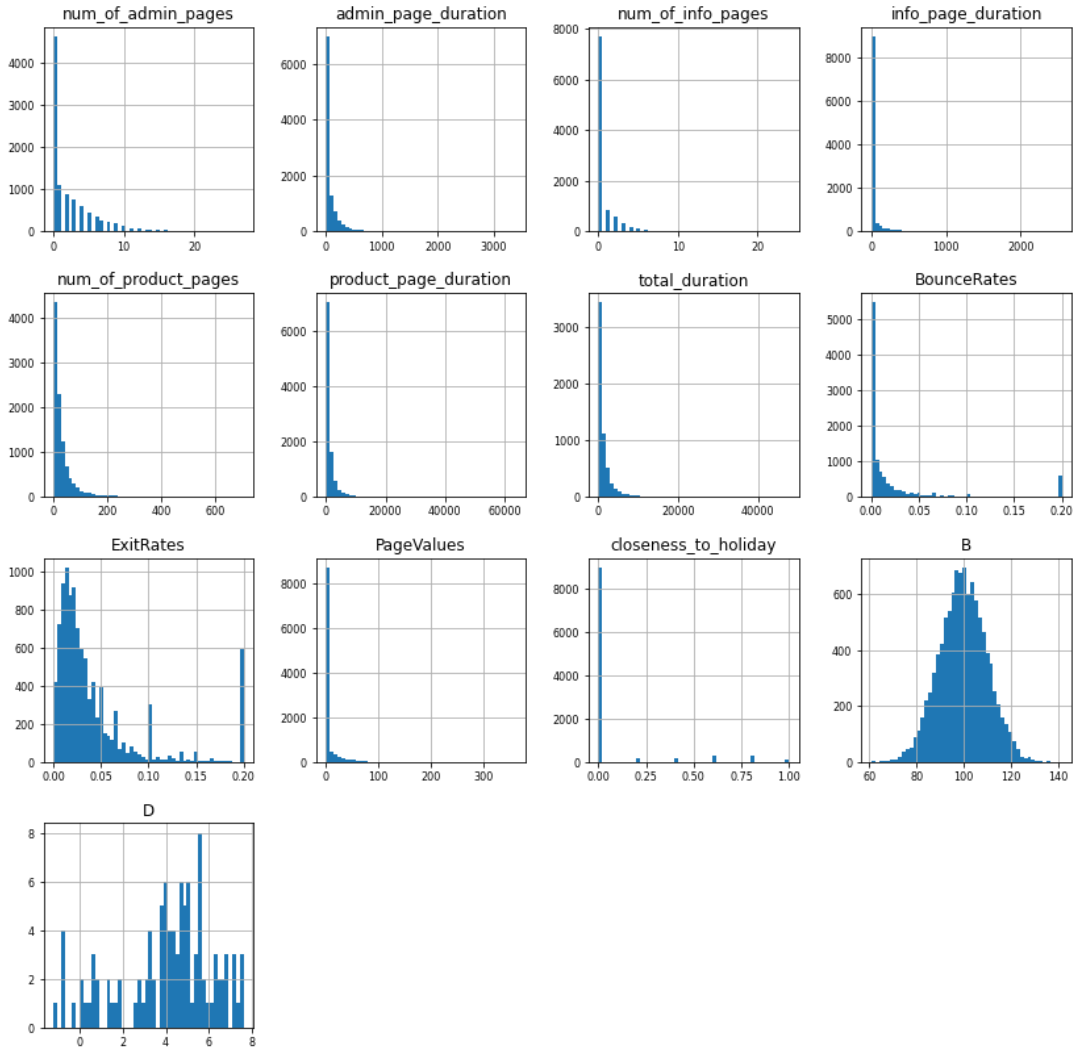
נספח 1: הצגת חלוקת ה-labels בדאטה:



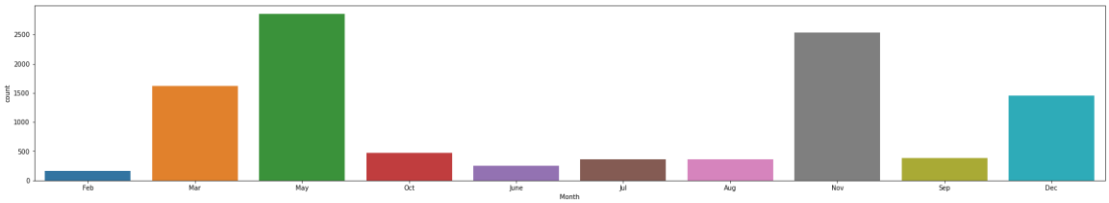
נספח 2:
Heatmap:

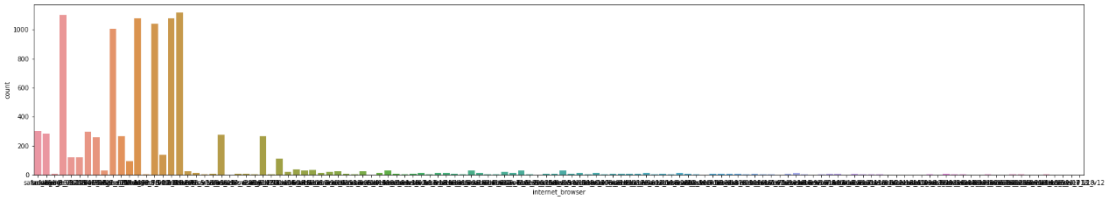


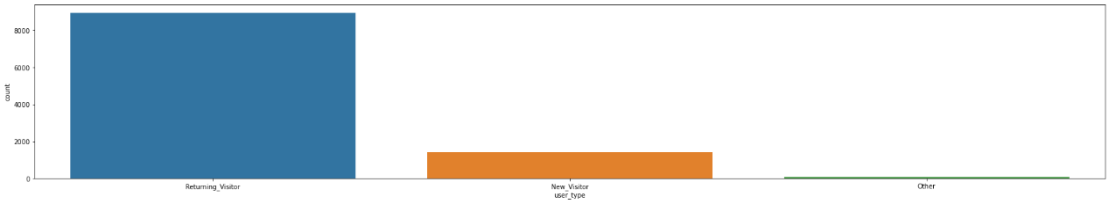
נספח 3: היסטוגרמות- התפלגות הפיצ'רים הנומריים:

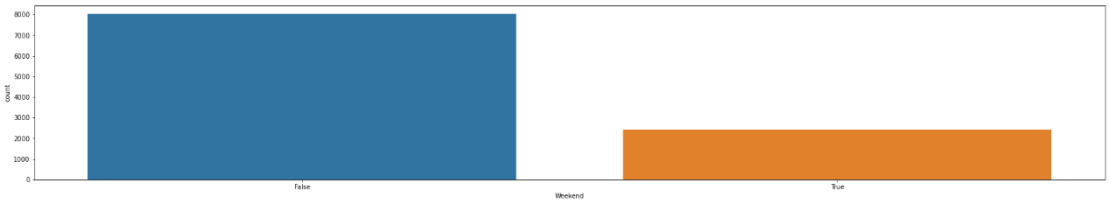


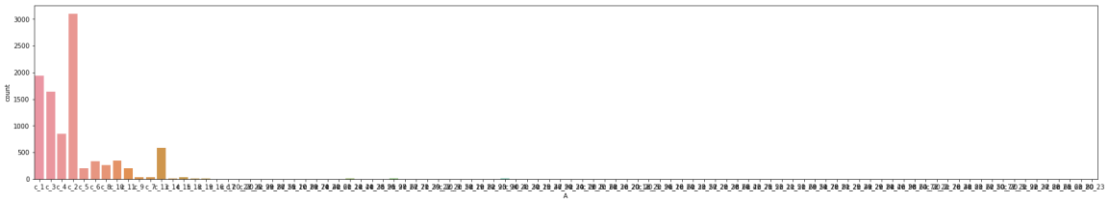
נספח 4: התפלגות הפיצ'רים הקטגוריאליים

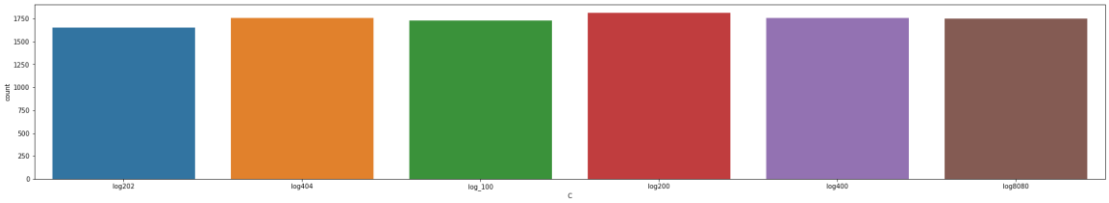


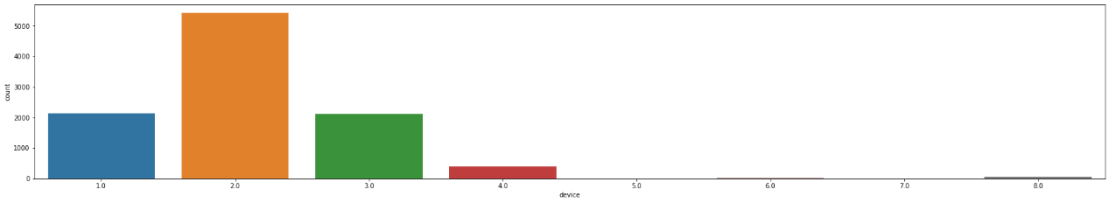


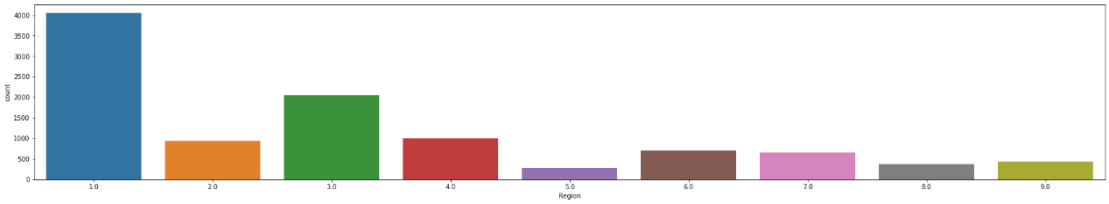

count
8000
6000
4000
2000
0
Returning_Visitor
New_Visitor
Other
user_type


count
8000
7000
6000
5000
4000
3000
2000
1000
0
False
True
Weekend

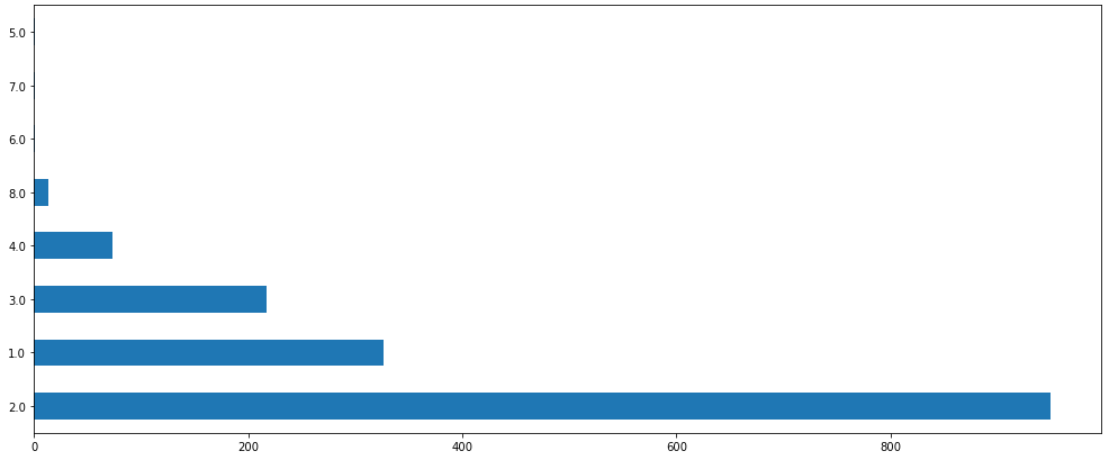

count
3000
2500
2000
1500
1000
500
0
A


count
1750
1500
1250
1000
750
500
250
0
log202
log404
log_100
log200
log400
log8080
C

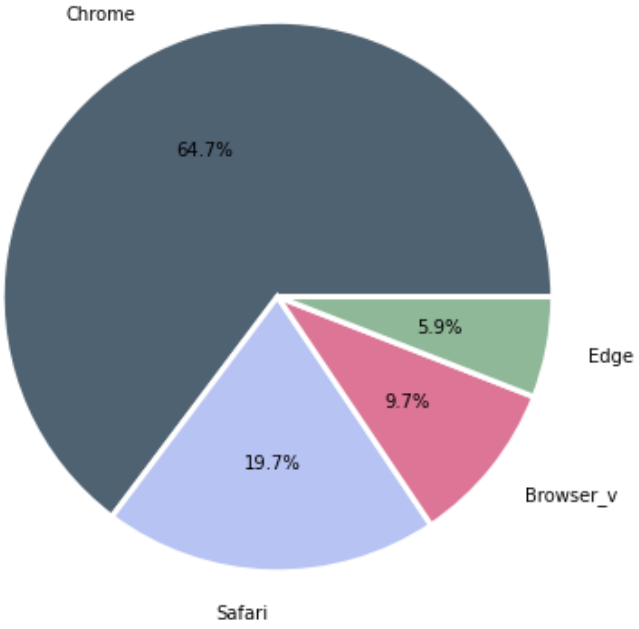

count
5000
4000
3000
2000
1000
0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
device


count
4000
3500
3000
2500
2000
1500
1000
500
0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
Region

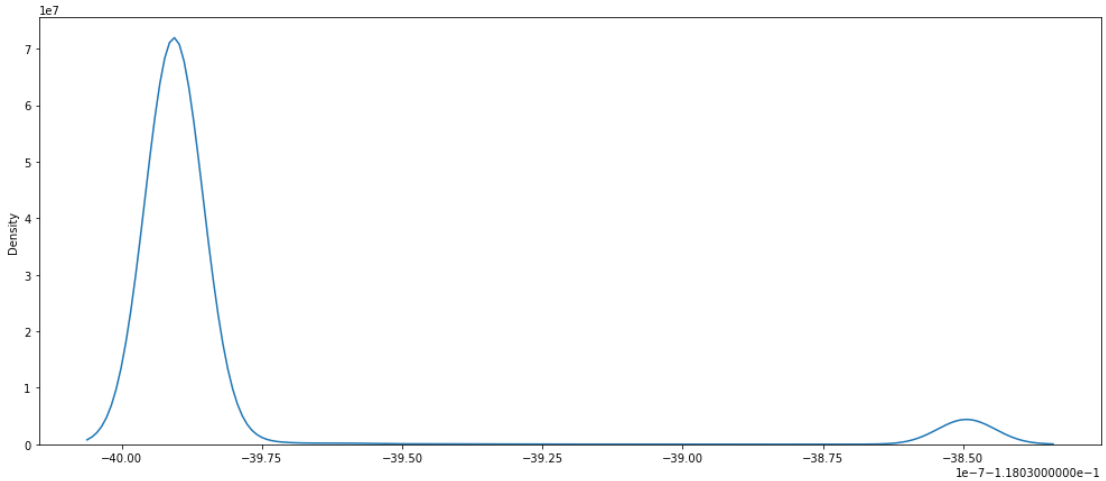
נספח 5: איזה מכשיר רוכש הכי הרבה:



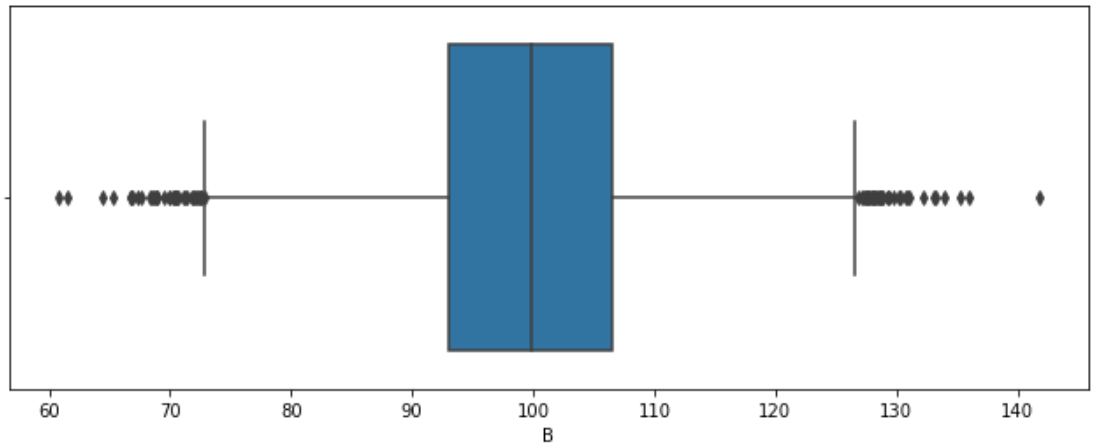
נספח 6: חלוקת הנתונים לפי דפדפנים:



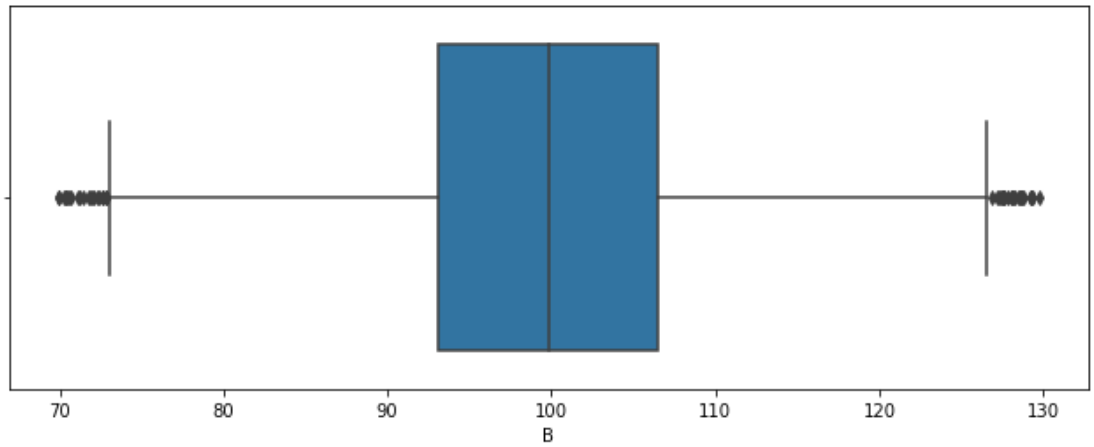
נספח 7: התפלגות 'ExitRates':



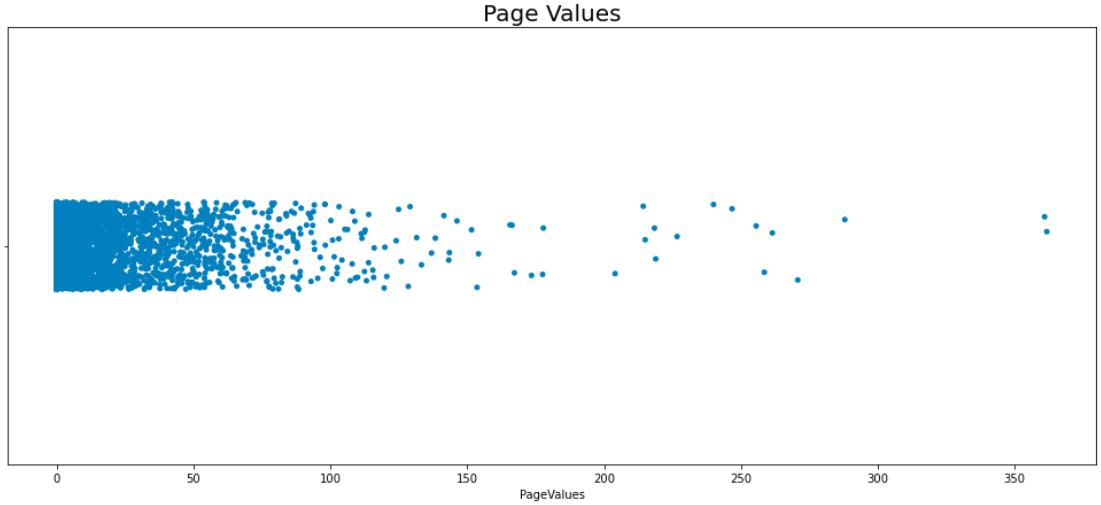
נספח 8: BoxPlot ל-'B' לפני הוצאת חריגים:

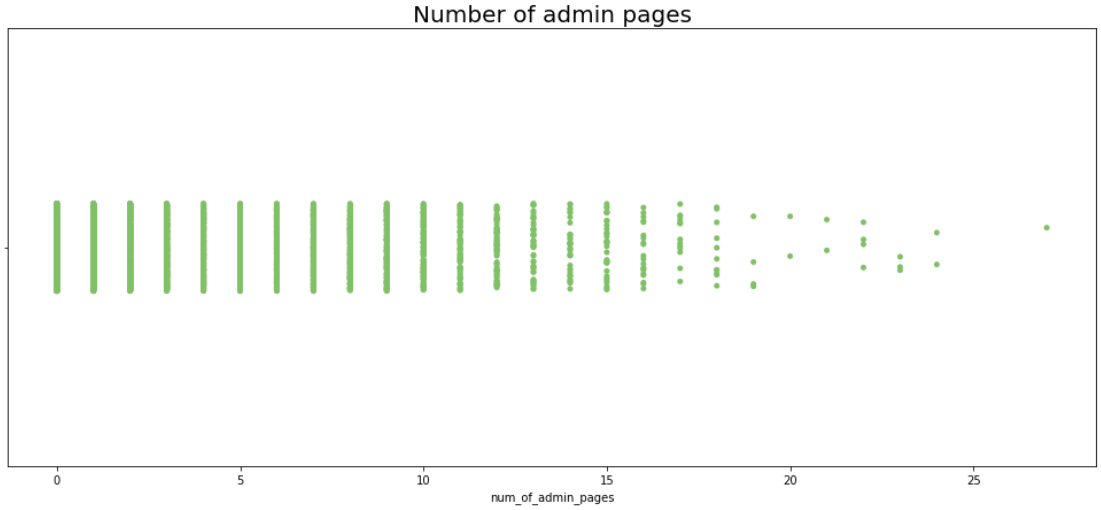


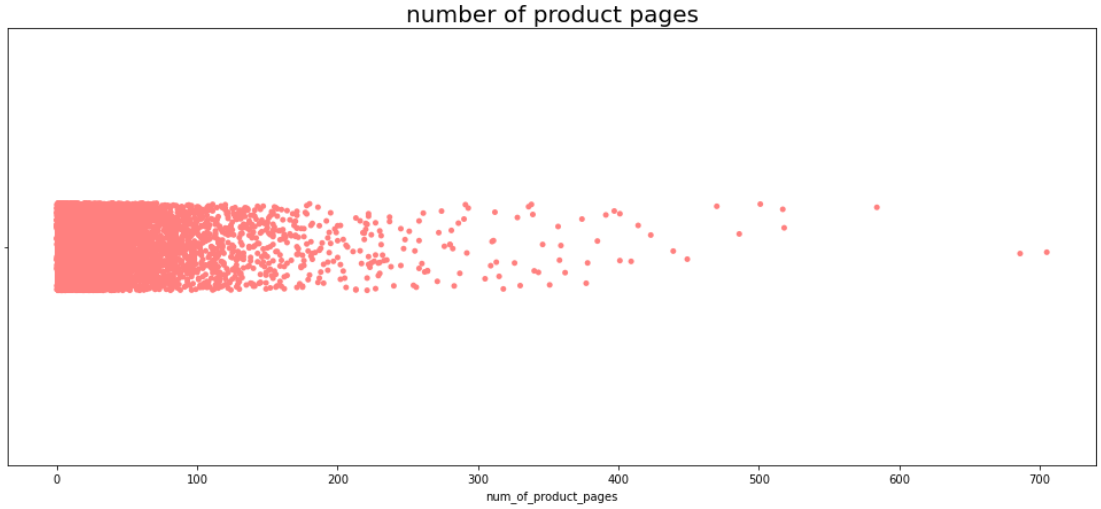
נספח 9: BoxPlot ל-'B' אחרי הוצאת חריגים:



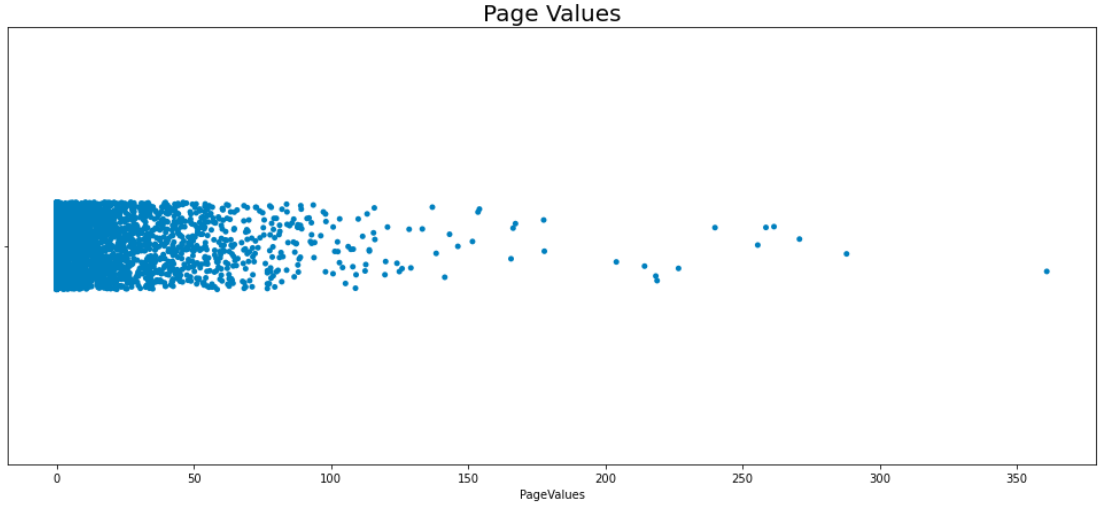
נספח 10: תרשים פיזור ל: 'PageValues', 'Num of admin pages' ו- 'number of product pages' לפני הוצאת חריגים:

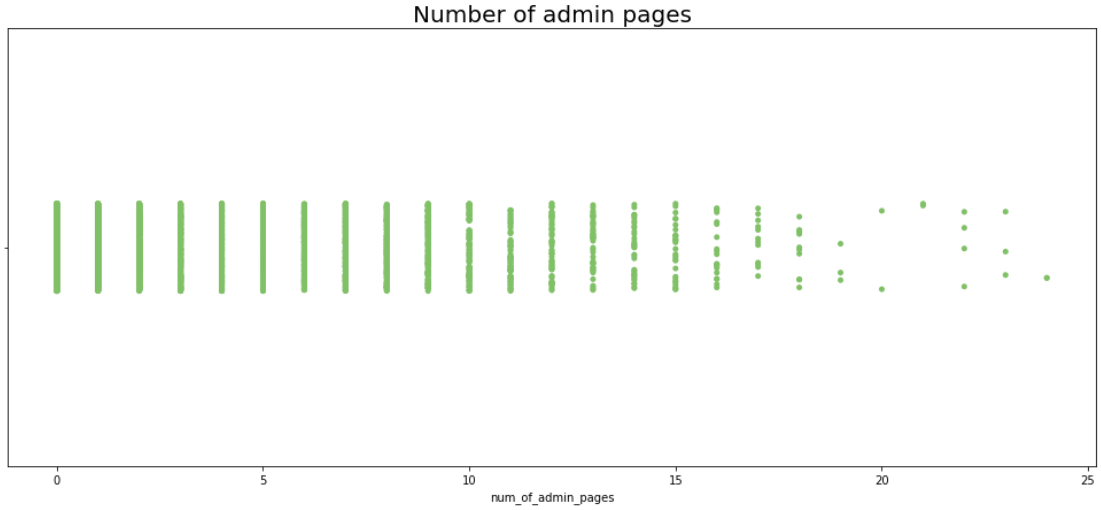




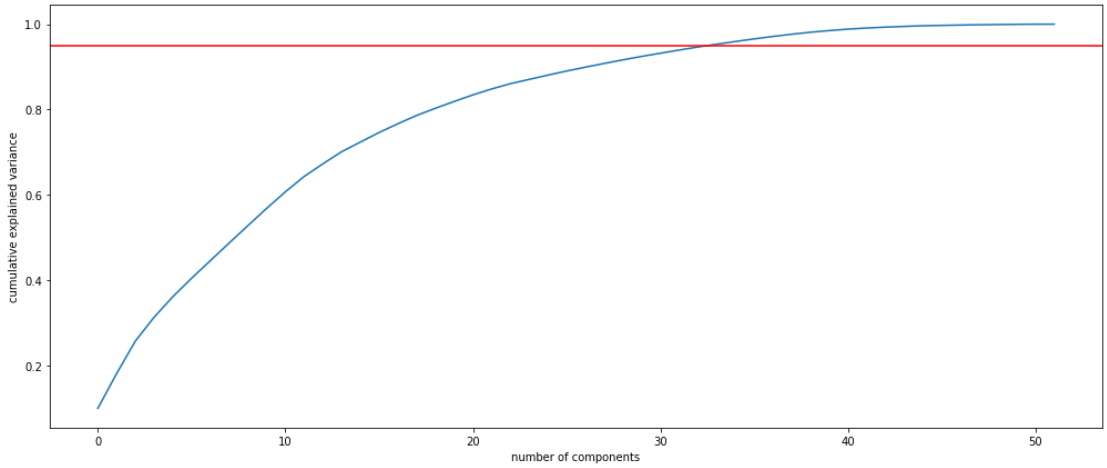


נספח 11: תרשים פיזור ל: 'PageValues', 'Num of admin pages' ו- 'number of product pages' אחרי הוצאת חריגים:



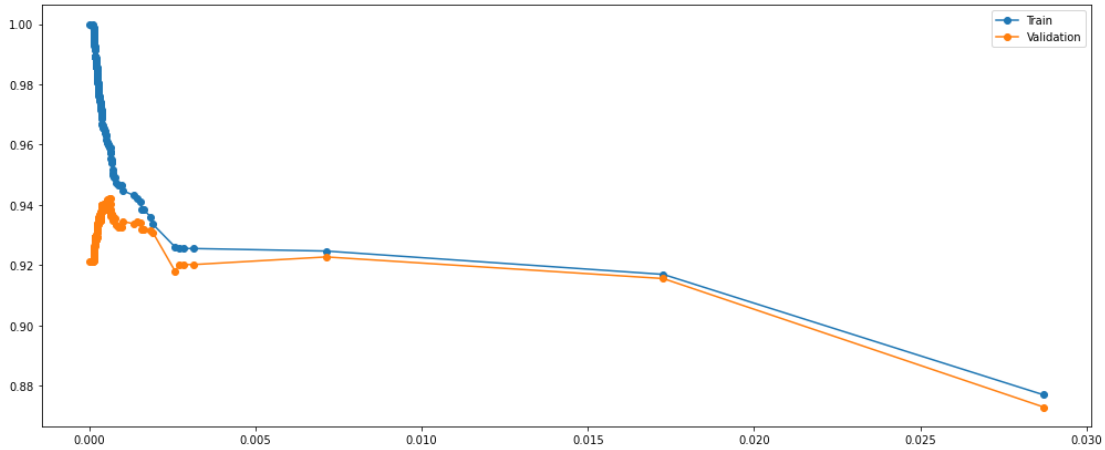


נספח 12: PCA:

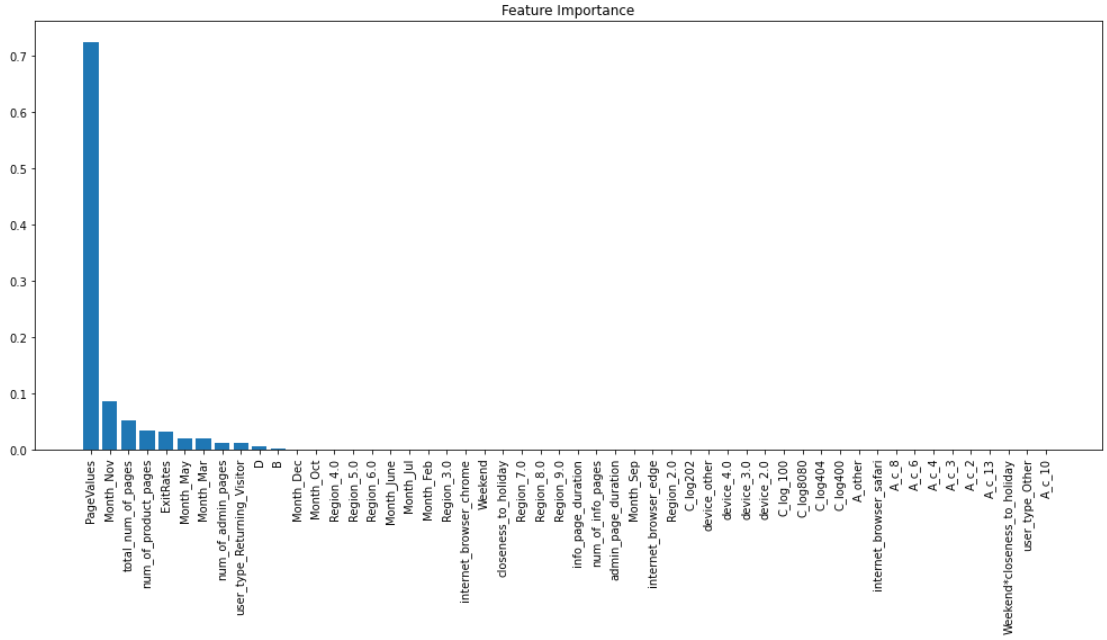


<u>פלטים עבור מודל Decision Tree:</u>

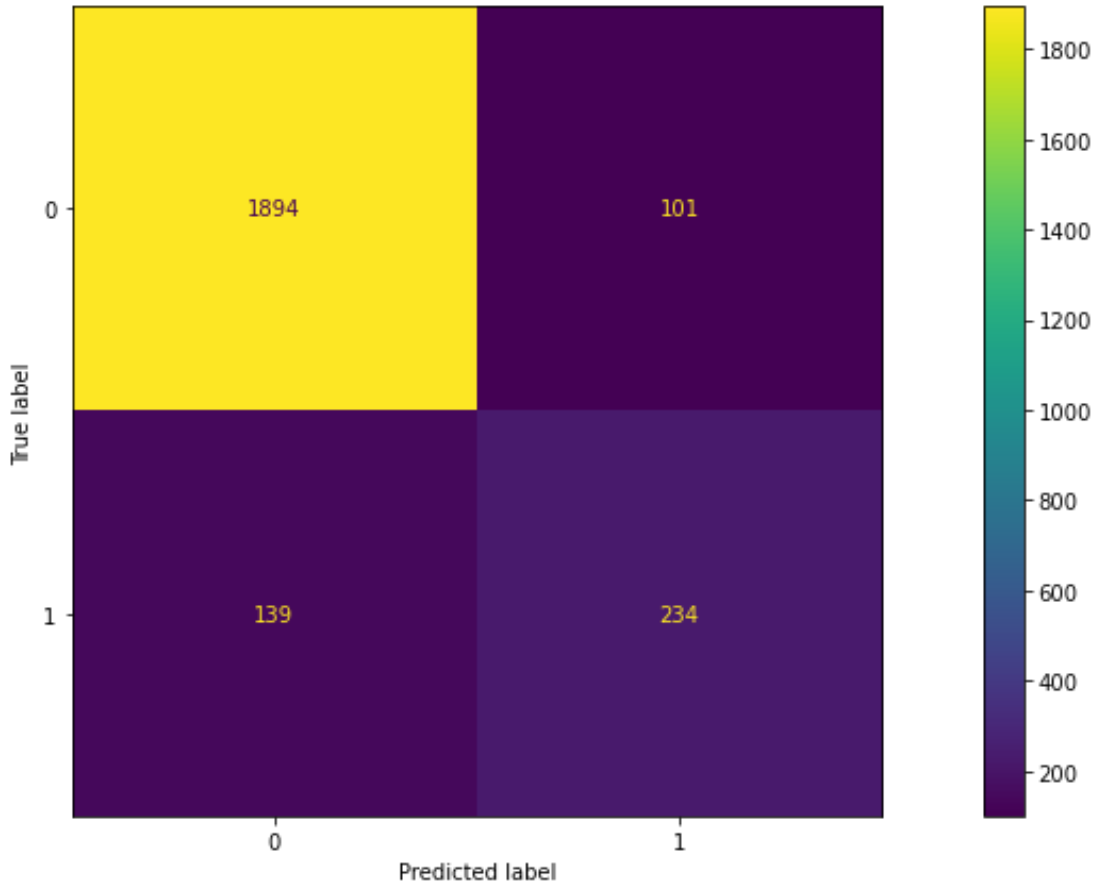
נספח 13: גרף עבור אלפות שונות:



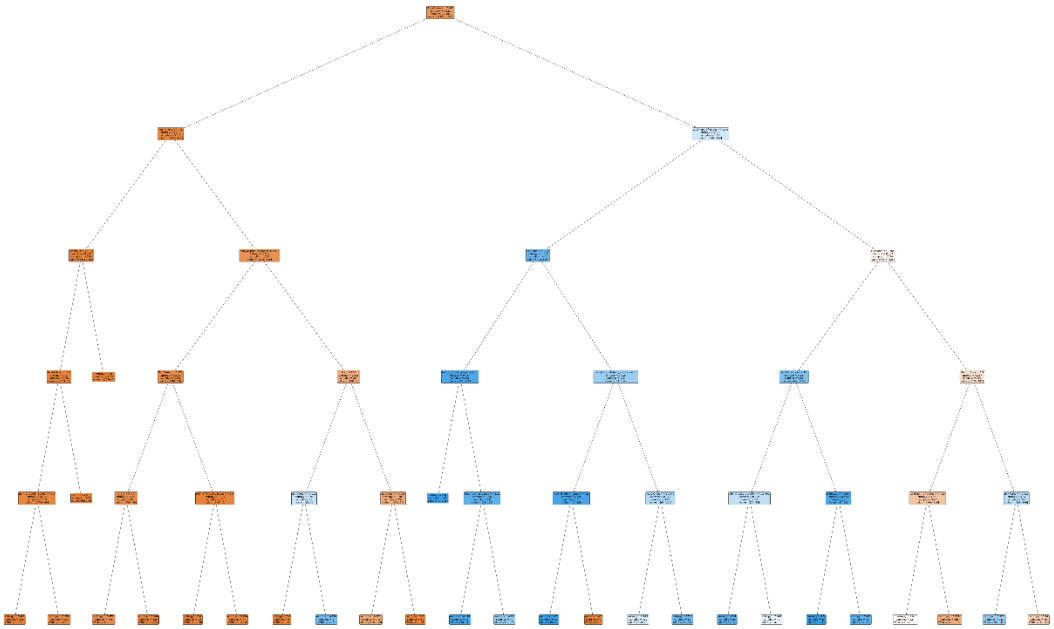
נספח 14: Feature importance:



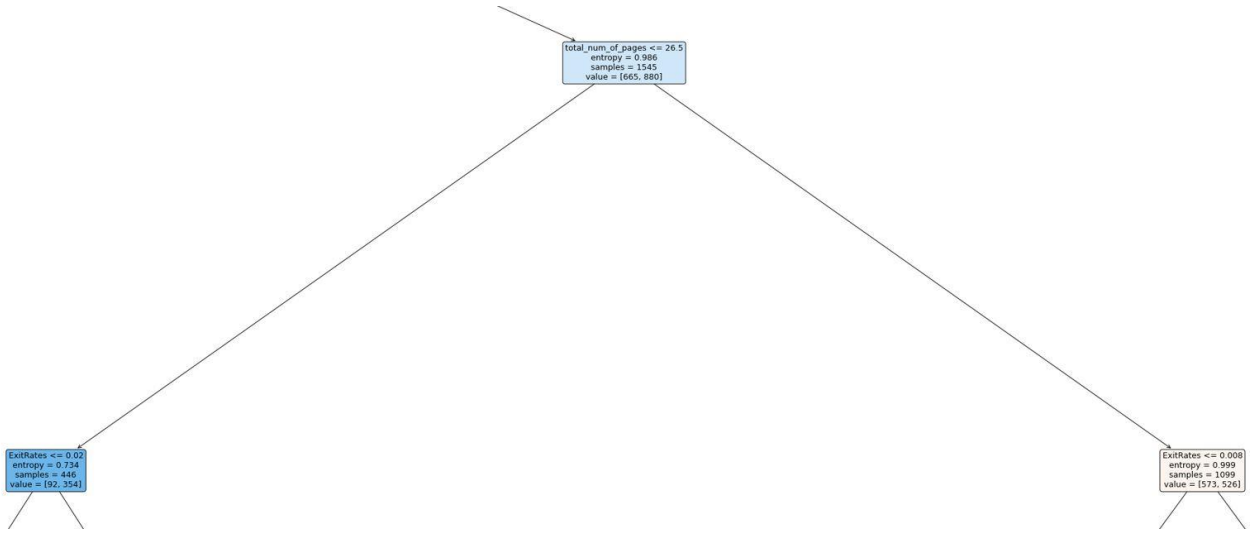
נספח 15:Confusion matrix:



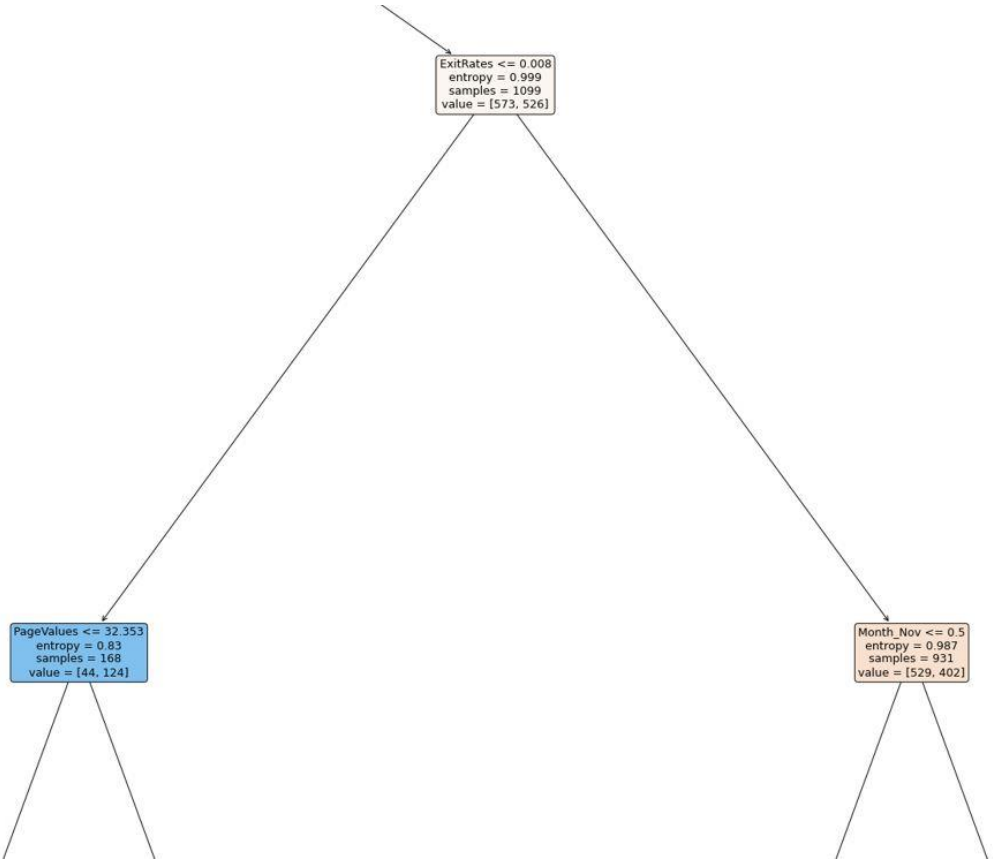
נספח 16: פלט של העץ לאחר הורדת ממדים:



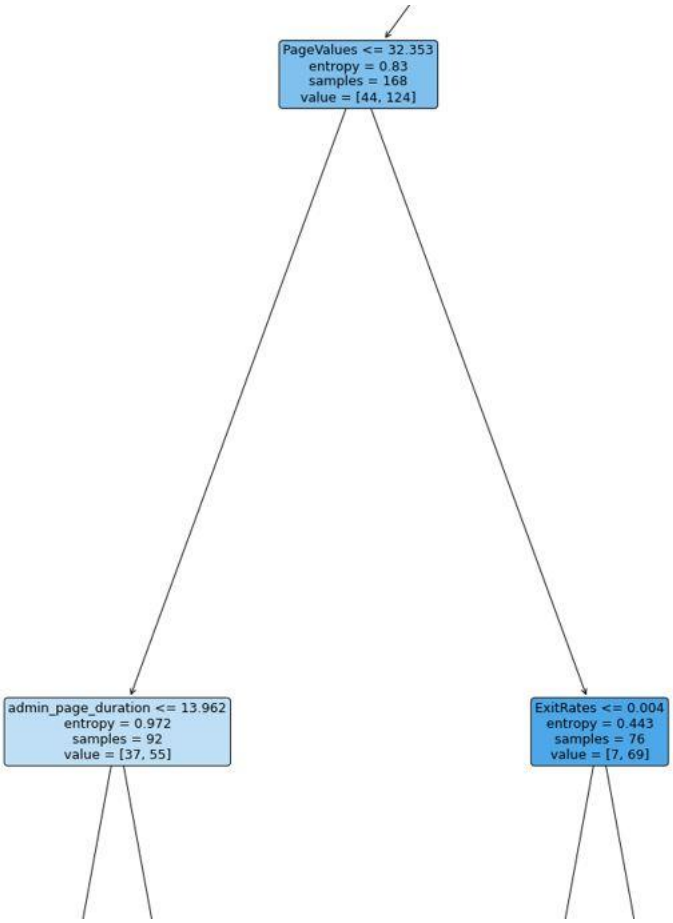
מסלול אחד של עץ החלטה:
נספח 16.1:



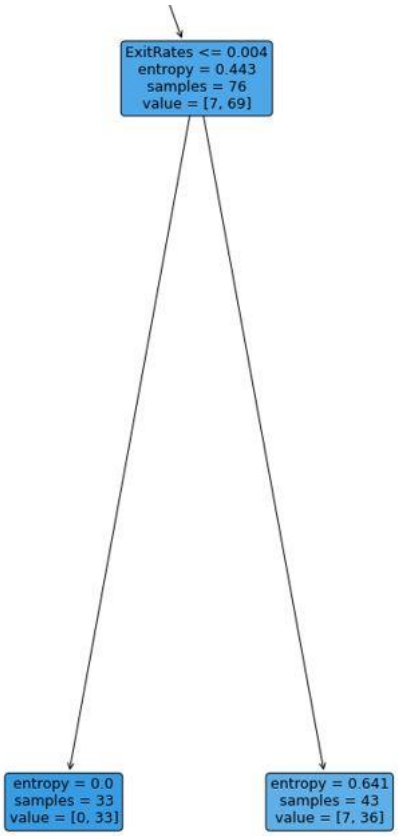
נספח 16.2:


ExitRates <= 0.008
entropy = 0.999
samples = 1099
value = [573, 526]
PageValues <= 32.353
entropy = 0.83
samples = 168
value = [44, 124]
Month_Nov <= 0.5
entropy = 0.987
samples = 931
value = [529, 402]

נספח 16.3:


PageValues <= 32.353
entropy = 0.83
samples = 168
value = [44, 124]
admin_page_duration <= 13.962
entropy = 0.972
samples = 92
value = [37, 55]
ExitRates <= 0.004
entropy = 0.443
samples = 76
value = [7, 69]

נספח 16.4:


ExitRates <= 0.004
entropy = 0.443
samples = 76
value = [7, 69]
entropy = 0.0
samples = 33
value = [0, 33]
entropy = 0.641
samples = 43
value = [7, 36]

נספח 17: ROC עבור כל המודלים: