פרויקט סיכום – מבוא ללמידת מכונה

דנה יכיני, 206571291

רתם אקר, 206015240

תקציר מנהלות:

בפרויקט זה רצינו לבנות מודלים שונים על מנת לחזות סיכוי לביצוע רכישה מקוונת (e-commerce). לצורך כך, ערכנו השוואה בין 4 מודלים מפורסמים בעולם של למידת מכונה: רגרסיה לוגיסטית, KNN, עץ החלטה ו-Random Forest. לאחר השוואת הביצועים, גילינו שהמודל הטוב ביותר מבין הארבעה הוא ה-Random Forest ולכן ביצענו את התחזיות שלנו על פיו.

שלב העיבוד המקדים:

- ניתוח כללי של התפלגויות
- קורלציות
- השלמת ערכים ריקים

לפני בניית המודלים, ביצענו מספר טרנספורמציות על הנתונים על מנת שיהיה לנו נוח לעבוד איתם.

ראשית, בדקנו אילו משתנים הם מספריים ואילו לא, ואת הערכים הלא מספריים המרנו לערכים מספריים. שינוי זה כלל השמטת מחרוזות, חילוץ מספר מתוך מחרוזת ויצירת משתני dummy כעמודות חדשות. המהלך הזה בוצע על מנת שנוכל לבנות מודל רגרסיה לוגיסטית. נפרט בהמשך הדו"ח על העמודות החדשות שיצרנו.

שנית, בדקנו את התפלגות הנתונים בעמודות השונות (נספח 2.1) ואת הקורלציה בין עמודות בעלות מכנים משותפים (למשל, עמודות שקשורות לתאריכים, נספח 2.2). על פי הקורלציות וההתפלגויות השונות, בחרנו את דרך השלמת הנתונים הריקים.

השלמת הערכים הריקים בוצעה ביחס לקובץ האימון. במקרים של ערכים בדידים (למשל, חודשים), השלמנו את הנתונים בעזרת הערך הנפוץ ביותר, במקרים רציפים השתמשנו בחציון ובמקרים בהם ראינו קורלציה חזקה בשלב הקודם, השתמשנו ביחס שקיים בין העמודות.

פירוט העמודות:

טרנספורמציות-

מכיוון שרצינו לעבוד עם רגרסיה לוגיסטית, היה חשוב לנו לוודא שכל העמודות מכילות ערכים נומריים בלבד. על כל העמודות שהכילו ערכים נומריים מלכתחילה לא ביצענו טרנספורמציה כוללת, אלא רק השלמנו ערכים ריקים.

העמודות info_page_duration ו-product_page_duration הכילו את מספר הדקות כמחרוזת שבסופה המילה minutes - חתכנו את המחרוזת כדי שתכיל רק את המספר, והמרנו ל-float.

את עמודת month, שהכילה את החודש כמחרוזת, בחרנו להמיר למספר תואם למספר החודש. אפשרות נוספת הייתה הוספת משתני dummy לכל חודש, אבל זה היה מוביל לעליה גדולה במימדיות. כמו כן, קיים קשר של רצף בין חודשי השנה והמרה למספר יכולה לשקף את קשר זה אם הוא רלוונטי לרגרסיה.

העמודה weekend הכילה ערכי True ו-False, שבחנו להמיר ל-1 ו-0 בהתאמה כדי שיוכלו לשמש ברגרסיה.

העמודה user_type הכילה 3 ערכים: Returning, New ו-Other. את עמודה זאת הורדנו ויצרנו במקומה שתי עמודות חדשות, is_returning ו-is_new, שמכילות 0 או 1 בהתאם לערך שהיה בעמודה המקורית. הערך other מיוצג על ידי 0 בשתי העמודות. במקרה זה מדובר בהוספת 2 עמודות חדשות בלבד, ולכן העלאת המימדיות במקרה זה היא הרבה יותר סבירה.

העמודה internet_browser הכילה מחרוזות שמתחילות בסוג browser: כרום, ספארי וכד', ואז פירוט של הגרסה הספציפית. אין קשר של רצף בין הדפדפנים השונים, ולכן המרנו למשתני dummy. החלטנו שגרסת הדפדפן היא כנראה פחות רלוונטית, ולנסות לייצר משתני dummy לכל הגרסאות היה מוביל לעלייה רצינית במימדיות, ולכן החלפנו את העמודה המקורית ב-4 עמודות בעלות ערכי 0 ו-1: is_safari, is_chrome, is_browser ו-is_edge.

בעמודה האנונימית A היו מחרוזות בפורמט c_x או c_x_y, כאשר X ו-Y הם מספרים. אין לנו דרך לדעת מה הערכים מייצגים, אך מכיוון שהחלטנו להמיר הכל לערכים נומריים כדי לבצע רגרסיה לוגיסטית, החלטנו להתייחס לx_y כ x.y, ולהמיר ל-float. על סמך כמות הערכים השונים, המרה למשתני dummy לא נראתה הגיונית. בנוסף, גם ללא ידע על מה העמודה מייצגת, סביר להניח שהערכים c_x_y ו-c_x_z יהיו קרובים יותר זה לזה מאשר c_y ו-c_z, במקרה הראשון הם כנראה תתי קטגוריה של אותה הקטגוריה x. המרה ל-float תשמור על דמיון זה.

עמודה אנונימית C הכילה מחרוזות בפורמט log_x, כאשר x הוא לוג מחשב ידוע (404...). אין קשר של רצף בין לוגים של מחשב, ולכן בחנו במקרה זה להשתמש dummies: עמודות is_log<x> עבור כל אחד מהלוגים בסט הנתונים, המכילות ערכי 0 או 1 בהתאם לערך המקורי בעמודה C.

בשלב זה כל העמודות הומרו לנומריות, ועברנו להשלמת ערכים ריקים.

בעמודת הקטגוריאליות- Region , is_log<x>, is_returning\new, device, is_<browser_catagory>, Month, בדקנו ב-bar plot (נספח 2.3) את התפלגות הערכים, והשתמשנו בערך הנפוץ ביותר כערך הדיפולטי לערכים ריקים.

לעמודת D הייתה קרולאציה חזקה יחסית עם הקנייה, ורוב הערכים בה היו ריקים, ולכן כדי לא להטות אותה העדפנו להשתמש ב-0 כערך דיפולטי להשלמה.

בקטגוריות של דפי admin, info ו-product ראינו שיש קורלציה בין מספר העמודים שמשתמש ביקר בהם ובין הזמן שהמשתמש ביקר בדפים בקטגוריה זו, באופן לא מפתיע מאוד. לכן החלטנו להשתמש ביחס כדי להשלים ערכים חסרים. עבור כל אחת מהקטגוריות, התחלנו בלהגדיר את הערך של מספר העמודים לפי ערך החציון של העומדה בכל השורות בהן שני הערכים היו ריקים. אחרי שעשינו זאת, ידענו בוודאות שבכל שורה לפחות אחד מהערכים מוגדר. חישבנו את החציון של היחס בין מספר העמודים לזמן, והשתמשנו בערך הזה ובערך בעמודה האחרת כדי להשלים ערכים ריקים בשתי העמודות.

אחרי שלב זה, השלמנו ערכים ריקים לtotal_duration כסכום של זמן השהייה בעמודים משלושת הקטגוריות בשלב הקודם.

את BounceRates, ExitRates ו-A השלמנו לפי החציון. את B, שמתפלג בקירוב נורמלי, השלמנו לפי התוחלת (ממוצע).

כדי להשלים ערכים ריקים בcloseness_to_holiday, חישבנו את הערך הממוצע לכל חודש, והשלמנו את הערכים בהתאם למספר החודש.

טיפול ב outliers:

בעזרת תרשימי boxplot (נספח 2.4) מצאנו outlier בעמודת ה-total_duration: שורה אחת הייתה עם ערך total_duration גבוה במיוחד ולכן הורדנו אותה מהנתונים.

בחירת מודלים:

מרבית המודלים שהשתמשנו בהם לא דורשים הנחות מיוחדות. יוצא הדופן הוא מודל הרגרסיה הלוגיסטית: משום שרגרסיה לוגיסטית מתבסס על רגרסיה לינארית, המודל מניח שניתן להפריד בין הקבוצות בצורה לינארית.

מבין שלושת המודלים הפשוטים, מודל ה-KNN לא מניח כלל הנחות ולכן נוח להשתמש בו כאשר ההתפלגויות של הנתונים כל כך שונות זו מזו. מודל הרגרסיה הלוגיסטית מניח הנחה שקל יותר לקבל מאשר המודל של Naive Bayes, במיוחד לאור העובדה שראינו שרוב הפיצ'רים לא מתפלגים נורמלית. מסיבות אלו בחרנו להשתמש במודלים הנ"ל.

מבין ארבעת המודלים המורכבים, עניינה אותנו ההשוואה בין עץ החלטה לבין Random Forest. מכיוון ש-Random Forest הוא אוסף גדול של עצי החלטה, ההשוואה בין הביצועים של עץ יחיד לעומת יער היא בלתי

נמנעת. מודלים שמבוססים על עצי החלטה לא דורשים הורדת מימד או הנחת הנחות כלשהן על הנתונים, מה שמאפשר גמישות רבה יותר בבניית המודל.

הורדת מימד ובחירת היפר פרמטרים:

הורדת מימד:

לצורך הורדת המימד השתמשנו בפונקציה המובנית של sklearn להורדת מימד, SequentialFeatureSelector, מתוך המודול feature_selection. הפונקציה הנ"ל היא מימוש של אלגוריתם ה-forward-backward selection שלמדנו בכיתה. העדפנו להשתמש באלגוריתם forward selection משום שיש לנו כמות גדולה של פיצ'רים אך לרבים מהם אין קורלציה חזקה לקנייה ולכן אלגוריתם דוגמת PCA יכול ליצור קורלציות שלא קיימות בנתונים. בנוסף, אלגוריתם של forward selection קל יותר לניתוח הממצאים והבנתם. על מנת לקבוע מהו מספר הפיצ'רים שנרצה להשתמש בהם במודלים הפשוטים שלנו, הרצנו לולאת for עם פונקצית ה-SequentialFeatureSelector על מספר שונה של פיצ'רים, החל מ-2 פיצ'רים ועד 11 (כולל). בדקנו את ציון ה-accuracy של כל בחירה על ידי אימון רגרסיה לוגיסטית על הפרמטרים שנבחרו על ידי הפונקציה, והפעלת הפונקציה accuracy_score על סט ולידציה שפוצל מראש מסט האימון. מצאנו נקודת מינימום לוקאלית של ה-accuracy ב-10 פיצ'רים, אך לפניה ה-accuracy היה יציב (נספח 2.5). מסיבה זו, אימנו את המודלים שלנו על תשעת הפיצ'רים הטובים ביותר כפי שנבחרו על ידי הפונקציה, והם:

'BounceRates', 'PageValues', 'device', 'A', 'D', 'is_returning', 'is_new', 'is_edge', 'is_log400'

היפר פרמטרים:

לאורך הפרויקט נתקלנו במספר מקומות שבהם בחרנו היפר פרמטרים.

k-fold cross validation: בחרנו לחלק ל-20 שכבות של ולידציה משום שיש לנו כ-10000 שורות בנתונים וחלוקה ל-20 קבוצות תשאיר לנו בכל פעם כ-500 דוגמאות רנדומליות לטסט, כלומר 5% מהנתונים. באופן הזה אנחנו לא מקריבות הרבה מהנתונים המצומצמים שלנו אך עדיין יכולות לבחון את המודל על דוגמאות שהוא לא התאמן עליהן.

רגרסיה לוגיסטית: ההיפר פרמטר היחיד ששינינו ברגרסיה הלוגיסטית הוא max_iter. כמות הפיצ'רים שלנו יחסית גדולה ו-100 איטרציות (הערך הדיפולטי של max_iter) לא הספיקו לנו, ולכן העלנו את כמות האיטרציות ל-500.

KNN: ההיפר פרמטר החשוב באלגוריתם KNN הוא כמובן כמות השכנים. בדומה לבחירת כמות הפיצ'רים, רצנו בלולאת for על מספר שכנים אפשרי לכל דוגמה ובדקנו את ציון ה-accuracy לאחר אימון של אלגוריתם knn וחיזוי על סט ולידציה שהופרד מראש מסט האימון. ראינו שהחל מ-11 שכנים העליה ב-accuracy מאוד מתונה ("מרפק", נספח 2.5), ולכן אימנו את המודל על 11 שכנים.

במודלים המורכבים, השתמשנו בפונקצית GridSearchCV שמנסה את כל השילובים האפשריים של הפרמטרים שהוצעו ומחזירה את השילוב שמספק את הציון הגבוה ביותר בהתאם לשיטת הניקוד שנבחרה. בחרנו בשיטת ניקוד של roc_auc משום שזו השיטה הנפוצה לניקוד תחזיות קלסיפיקציה בינארית, כמו חיזוי רכישה.

השתמשנו בחיפוש על שלושה היפר פרמטרים מרכזיים: max_depth, min_sample_leaf ו-min_sample_split.

ביצענו את החיפוש על עומק מקסימלי בין שתי רמות ל-15 רמות, על מינימום דוגמאות בעלה בין דוגמה יחידה ל-4 דוגמאות בעלה ועל מספר הדוגמאות המינימלי לפיצול בעץ בין 2 דוגמאות ל-6. את החיפושים האלה ביצענו גם על עץ ההחלטה (נספח 2.6) וגם על ה-random forest, לצורך אחידות בין המודלים והשוואה בין הביצועים שלהם. מעניין היה לראות שה-GridSearchCV בחר ערכים שונים לפרמטר max_depth בין עץ בודד לבין יער, למרות שהבחירה הייתה מאותו טווח ערכים.

במודל של random forest ישנה אפשרות לבחור את מספר העצים ביער. ניסינו להתייחס להיפר הפרמטר הזה, אך ראינו שזה מעלה את זמן הריצה של הפרויקט בצורה מאוד משמעותית ולכן החלטנו להישאר עם כמות העצים הדיפולטית.

תוצאות המודלים:

לצורך השוואת התוצאות בין המודלים השונים השתמשנו ב-ROC_AUC (נספח 2.7). התוצאות הממוצעות של המודלים היו כדלקמן:

- רגרסיה לוגיסטית: 0.88, סטיית תקן ±0.03
- k שכנים קרובים: 0.87, סטיית תקן ±0.03
- עץ החלטה: 0.92, סטיית תקן ±0.02
- יער רנדומי: 0.93, סטיית תקן ±0.02

כפי שניתן לראות, random forest קיבל את הציון הגבוה ביותר ולכן הוא המודל שנבחר לחיזוי. קיים הבדל מובהק בין הביצועים של המודלים הפשוטים לעומת המודלים המורכבים, אך בין המודלים המורכבים כמעט ואין הבדל בתוצאות עם היפר הפרמטרים שנבחרו.

שמנו לב למספר פיצ'רים מעניינים שה-random forest בחר (נספח 2.6):

הפיצ'ר PageValue הוא באופן בולט מאוד הפיצ'ר התורם ביותר, כאשר הפיצ'ר השני בחשיבותו הוא BounceRate. שניהם נבחרו גם על ידי אלגוריתם ה-forward selection. באופן הגיוני, ה-PageValue מחושב ביחס להאם העמוד הוביל לרכישה בממוצע, ו-BounceRate מייצג יציאה מהאתר, ולכן טבעי ששניהם יתרמו בצורה חזקה לחישוב סיכויי הקנייה. בנוסף לכך, פיצ'ר תורם נוסף היה num_of_product_pages. מכיוון שרכישה היא של מוצרים, הגיוני לראות שמספר עמודי ה-product תורמים יותר לחיזוי ממספר עמודי admin, שהתרומה שלו קטנה יותר, וממספר עמודי ה-info שכמעט לא תורם כלל. כמו כן ראינו תרומה של זמן השהייה הכולל באתר, והחודש בו הכניסה לאתר התבצעה.

בעזרת confusion matrix (נספח 2.8), ראינו שהמודל מצליח לחזות בצורה טובה באופן משמעותי מתי לא תתבצע רכישה. לעומת זאת, המודל מתקשה לחזור בצורה טובה מתי כן תתבצע רכישה ועדיף באחוזים בודדים על ניחוש אקראי (כ-55% דיוק). תוצאה זו אינה מפתיעה בהתחשב בכך שמרבית הנתונים מתארים מצב שבו לא התבצעה רכישה. על מנת לשפר את הדיוק עבור מקרים בהם מתבצעת רכישה, ניתן לייצר דוגמאות נוספות למקרים בהם התבצעה רכישה בעזרת אלגוריתם PCA ולאמן את המודל מחדש על נתונים אלה.

סיכום:

לאורך הפרויקט בחנו 4 מודלים שונים: KNN, רגרסיה לוגיסטית, עץ החלטה ו-Random Forest. ביצענו טרנספורמציות כדי שנוכל לבנות רגרסיה לוגיסטית על סמך הנתונים ולנתח אותם. ביצענו feature selection כדי לצמצם את המימדיות, ובכך להקטין זמני ריצה וגם להקטין את הסיכון ל-overfitting. אחר כך בחרנו היפר-פרמטרים לכל מודל על סמך ה accuracy של האפשרויות השונות, ואימנו את המודלים עם הפרמטרים שהובילו לציון הגבוה ביותר. השתמשנו ב-k-fold כדי לבדוק את המודלים השונים ולהשוות ביניהם, בכך גם בדקנו האם המודל ב-overfitting. לאחר השוואת התוצאות, בחרנו במודל הטוב מבין הארבעה, שהוא ה-random forest, והשתמשנו בו בשביל לחזות את סיכויי הרכישה על הנתונים ב-test.

נספחים

1 - חלוקת אחריות

2.1-2.8 - גרפים

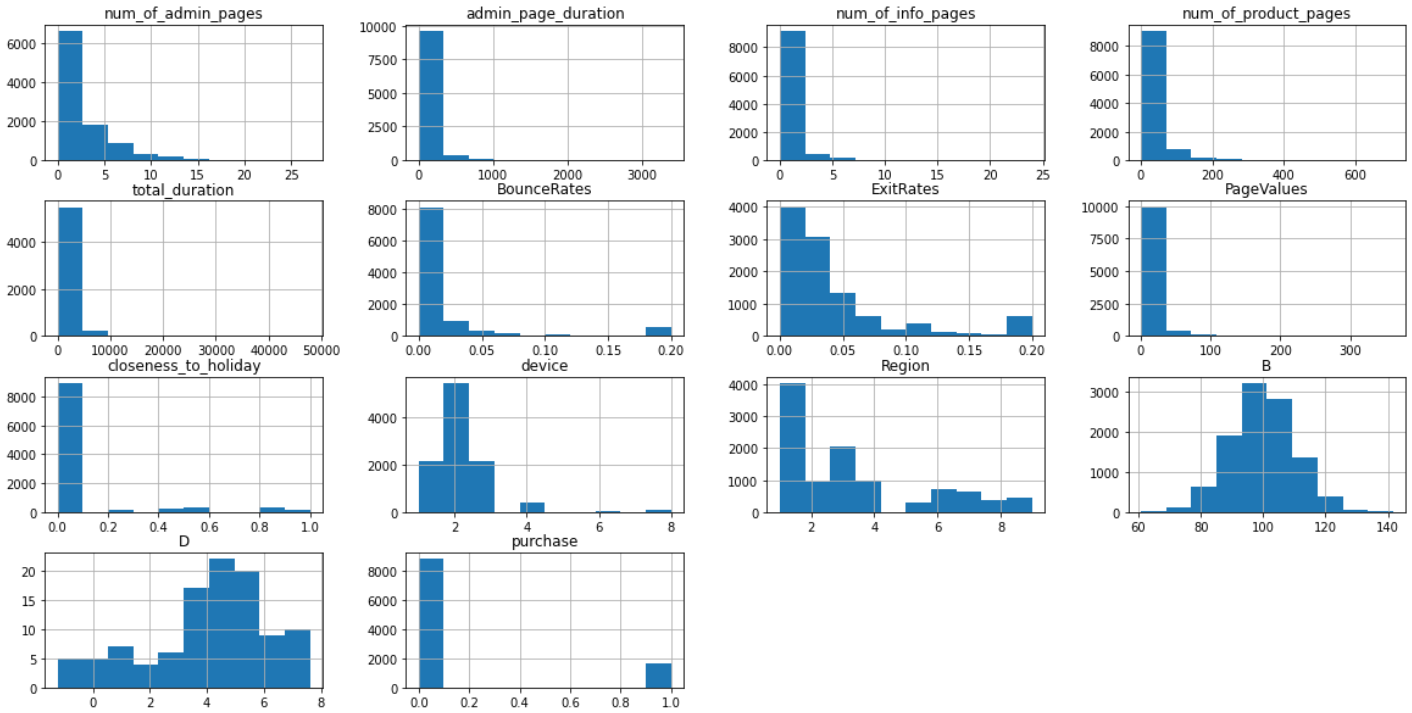
נספח 1 - חלוקת אחריות

העבודה על הפרויקט נעשתה בצורה משותפת בפגישות פיזיות ובזום באופן מלא, לא היה חלק שאחת מאיתנו עשתה לבדה.

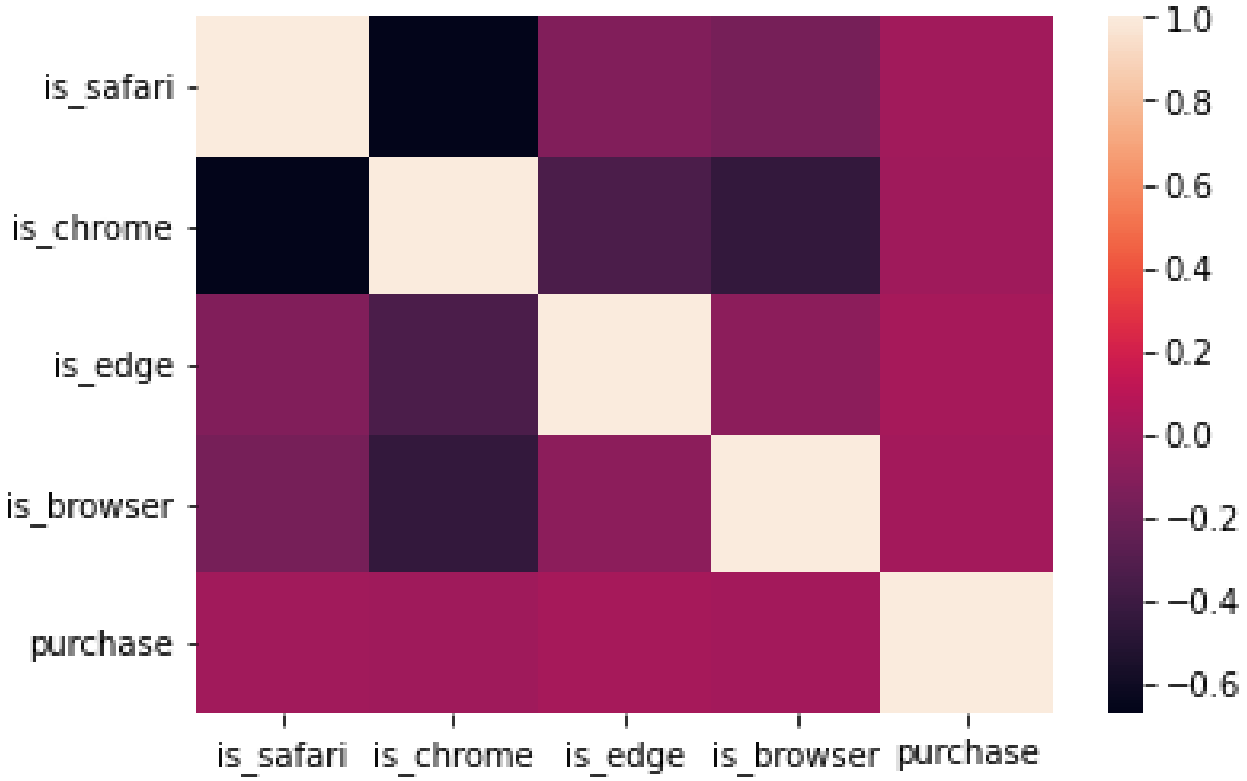
לאורך החודשיים האחרונים נפגשנו פעם בשבוע בספריה ופעם בשבוע לפחות בזום, וכך הגענו לפרויקט הנ"ל.

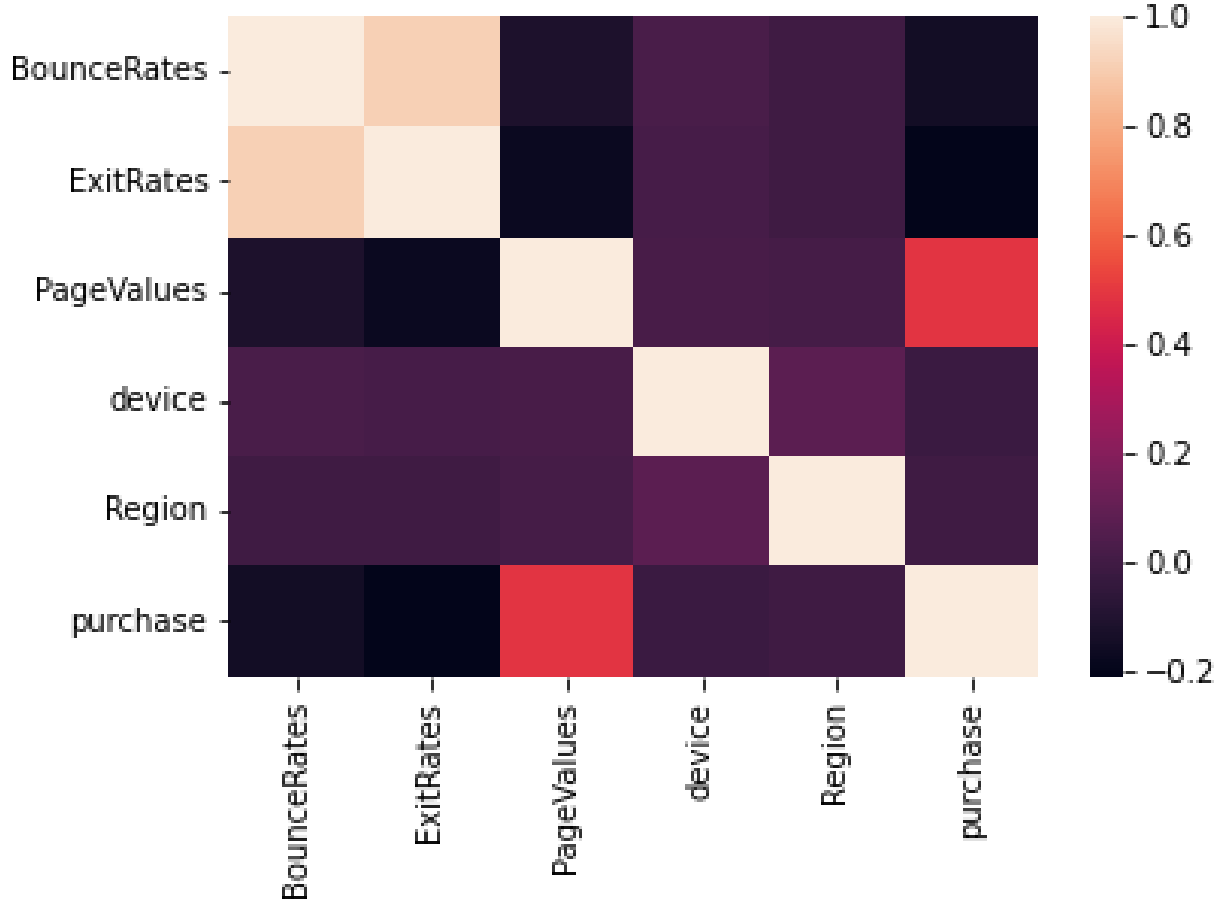
נספח 2 - גרפים

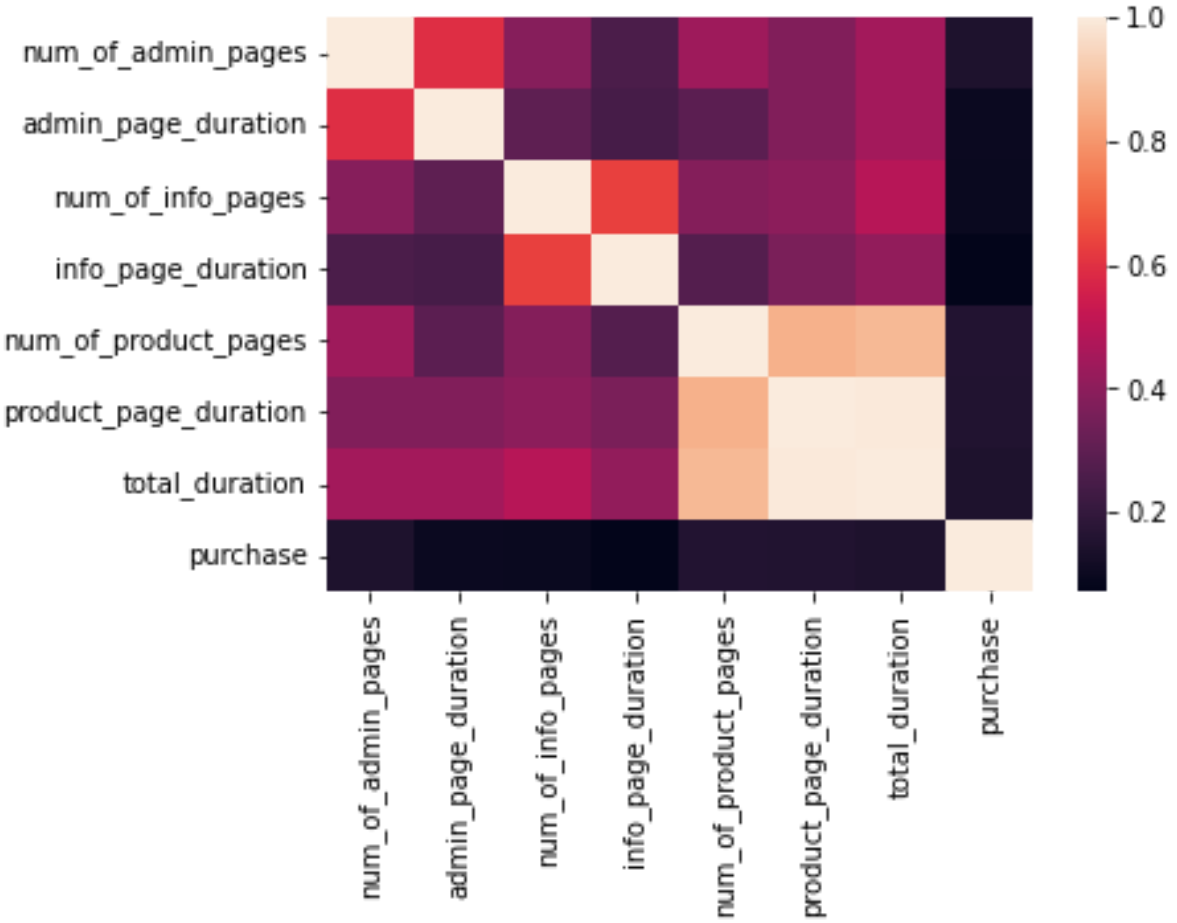
2.1 היסטוגרמות

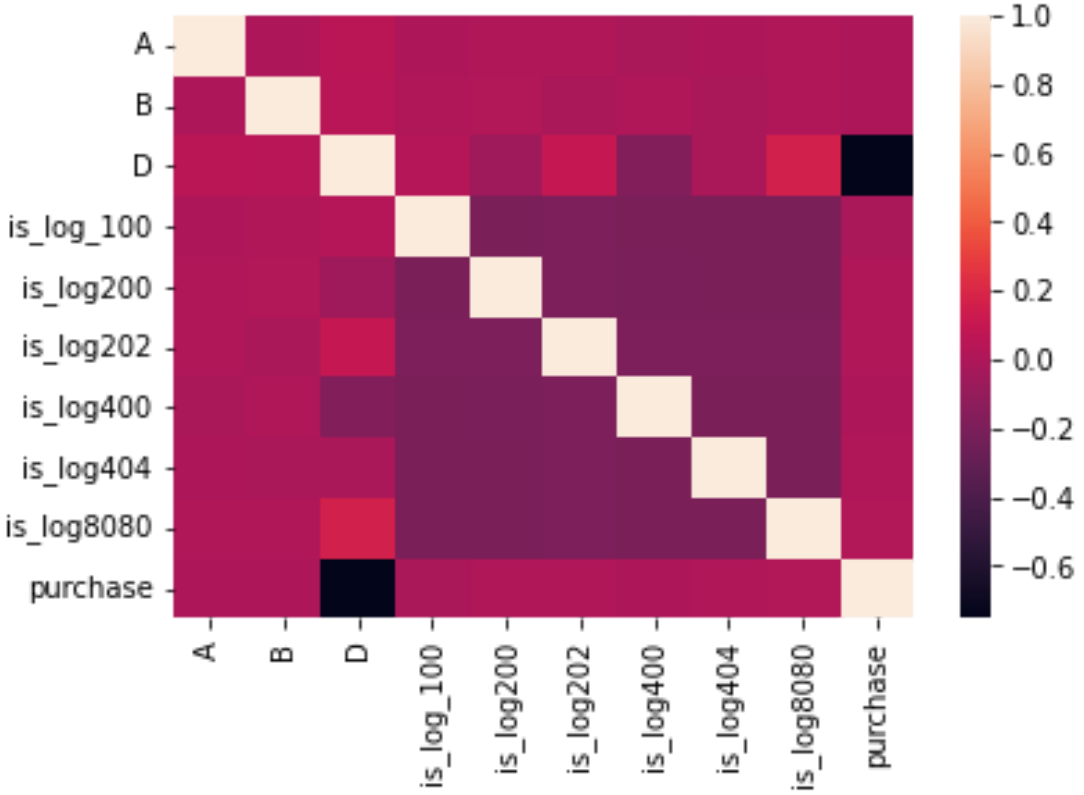


2.2 קורלציות

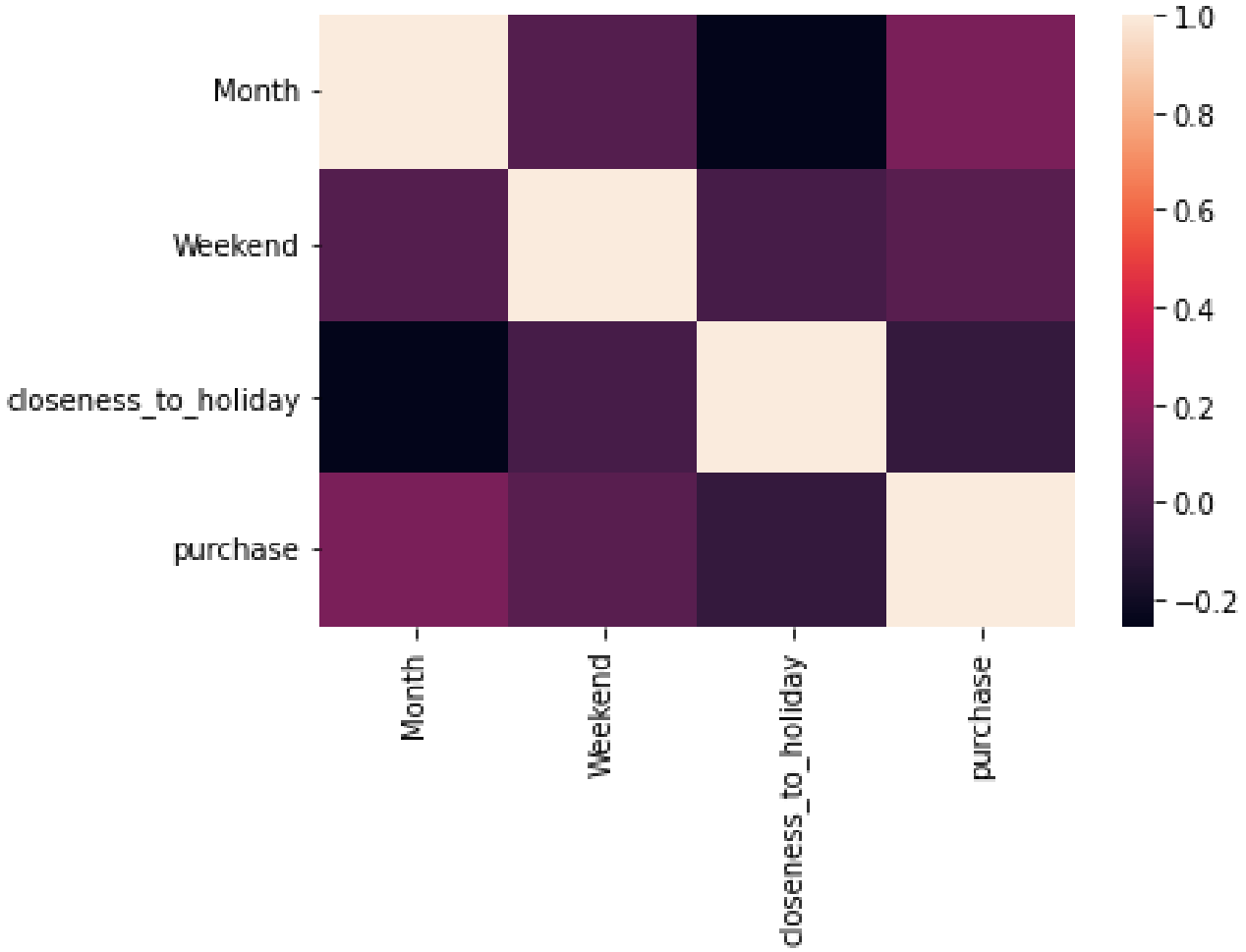

is_safari
is_chrome
is_edge
is_browser
purchase
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6


BounceRates
ExitRates
PageValues
device
Region
purchase
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

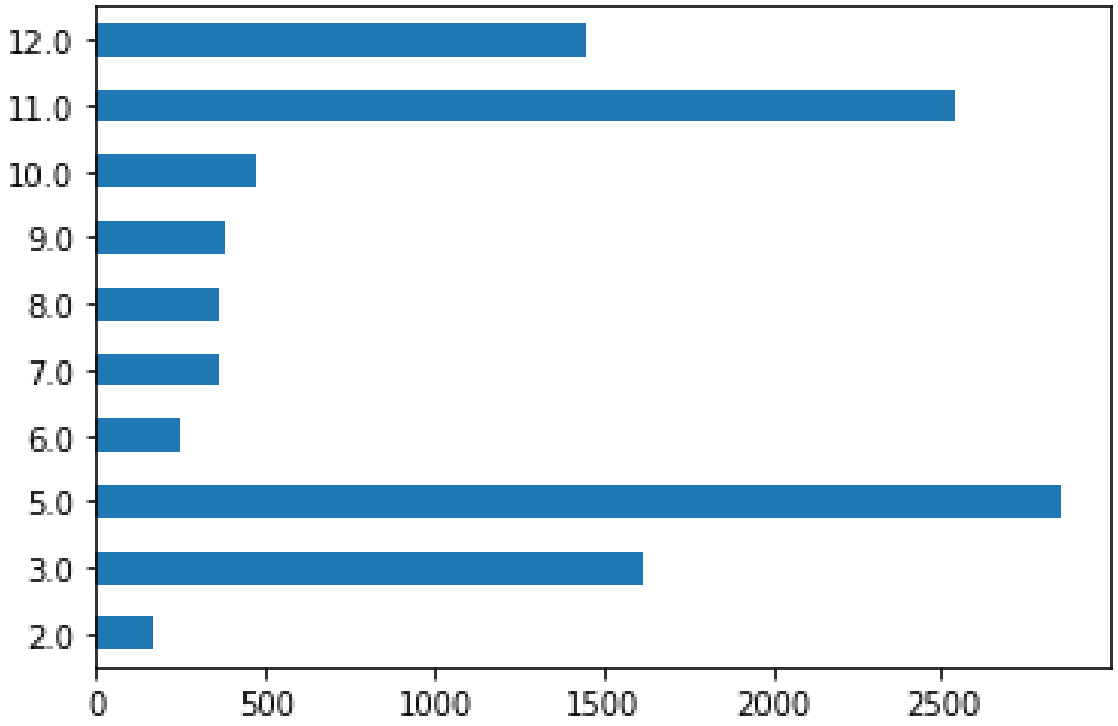

num_of_admin_pages
admin_page_duration
num_of_info_pages
info_page_duration
num_of_product_pages
product_page_duration
total_duration
purchase
1.0
0.8
0.6
0.4
0.2

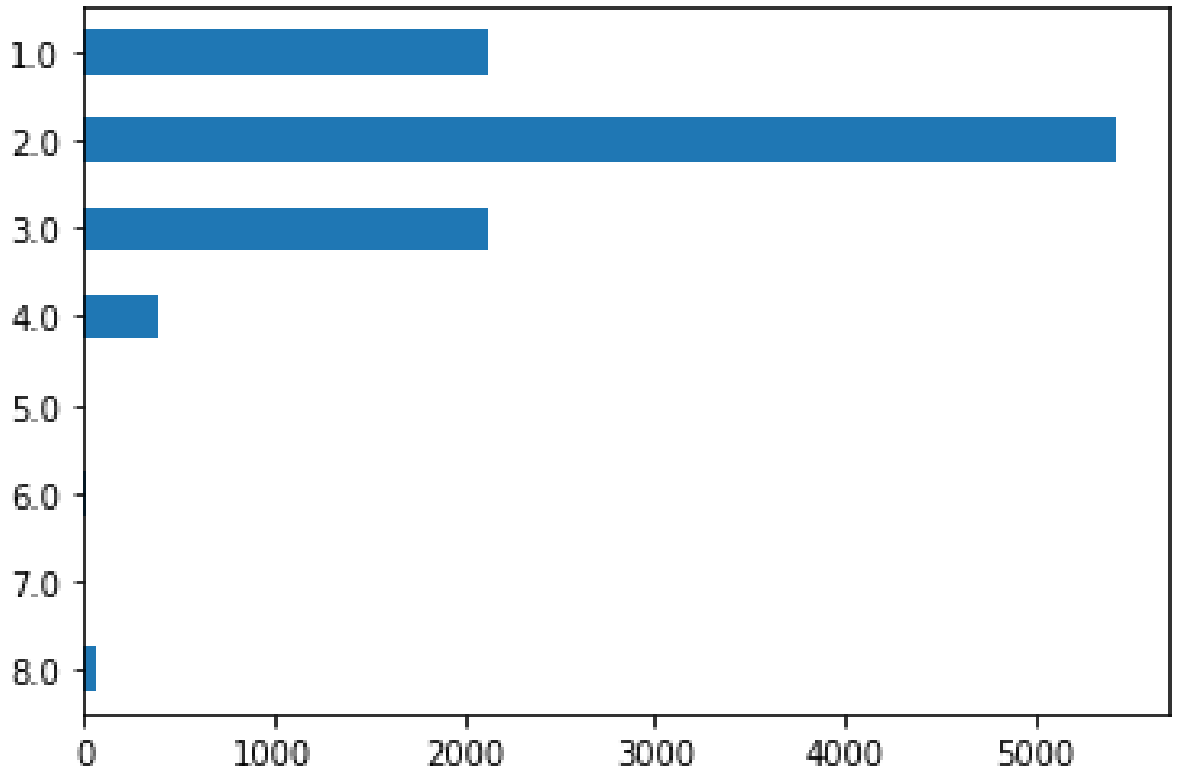

A
B
D
is_log_100
is_log200
is_log202
is_log400
is_log404
is_log8080
purchase
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

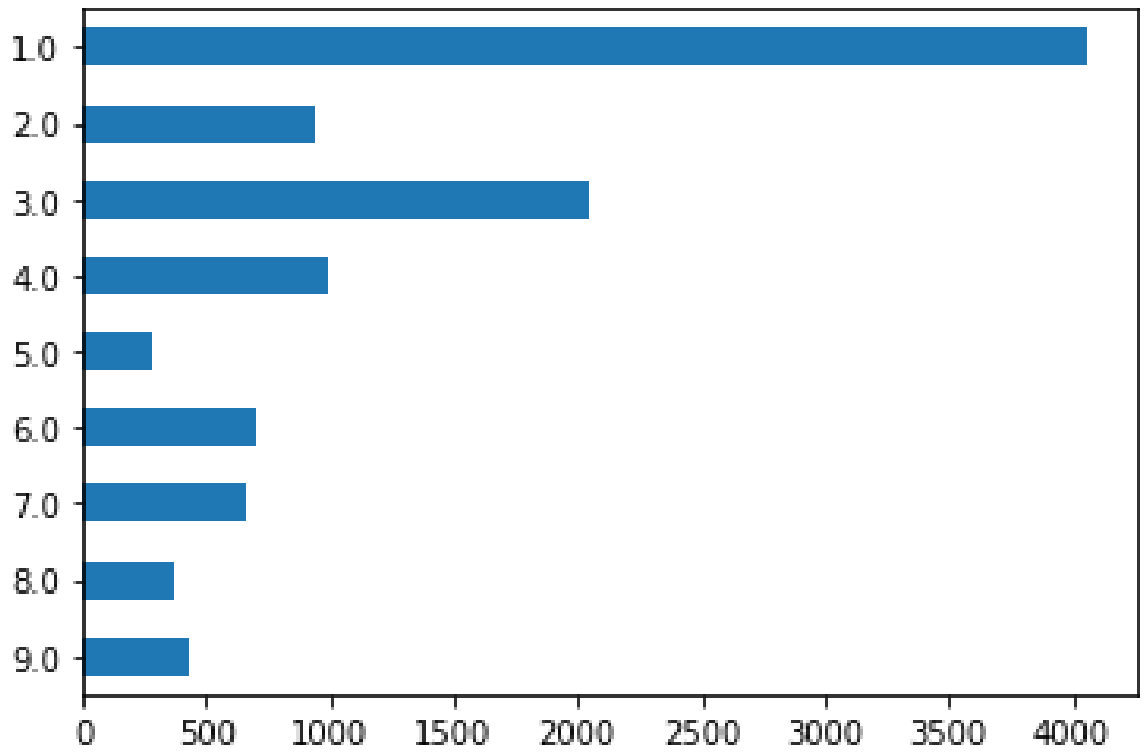

is_returning
is_new
purchase
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

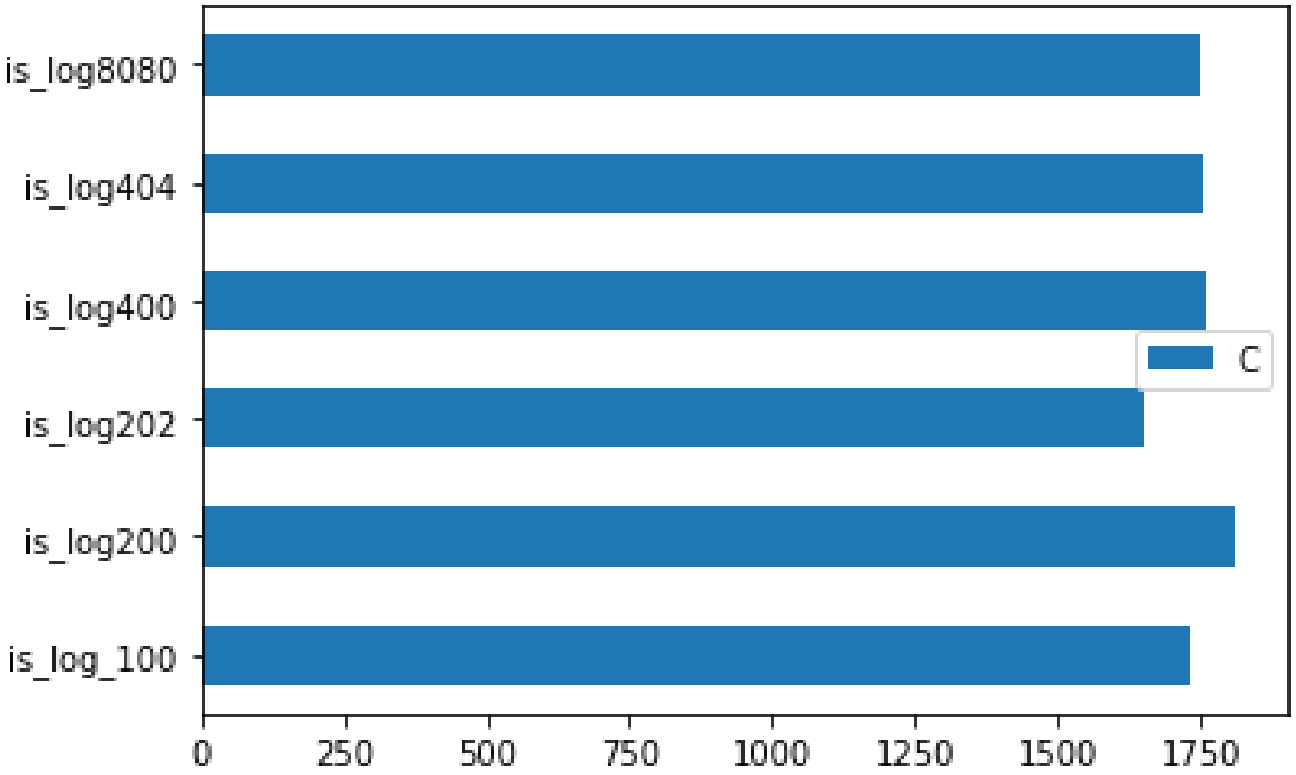

Month
Weekend
closeness_to_holiday
purchase
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

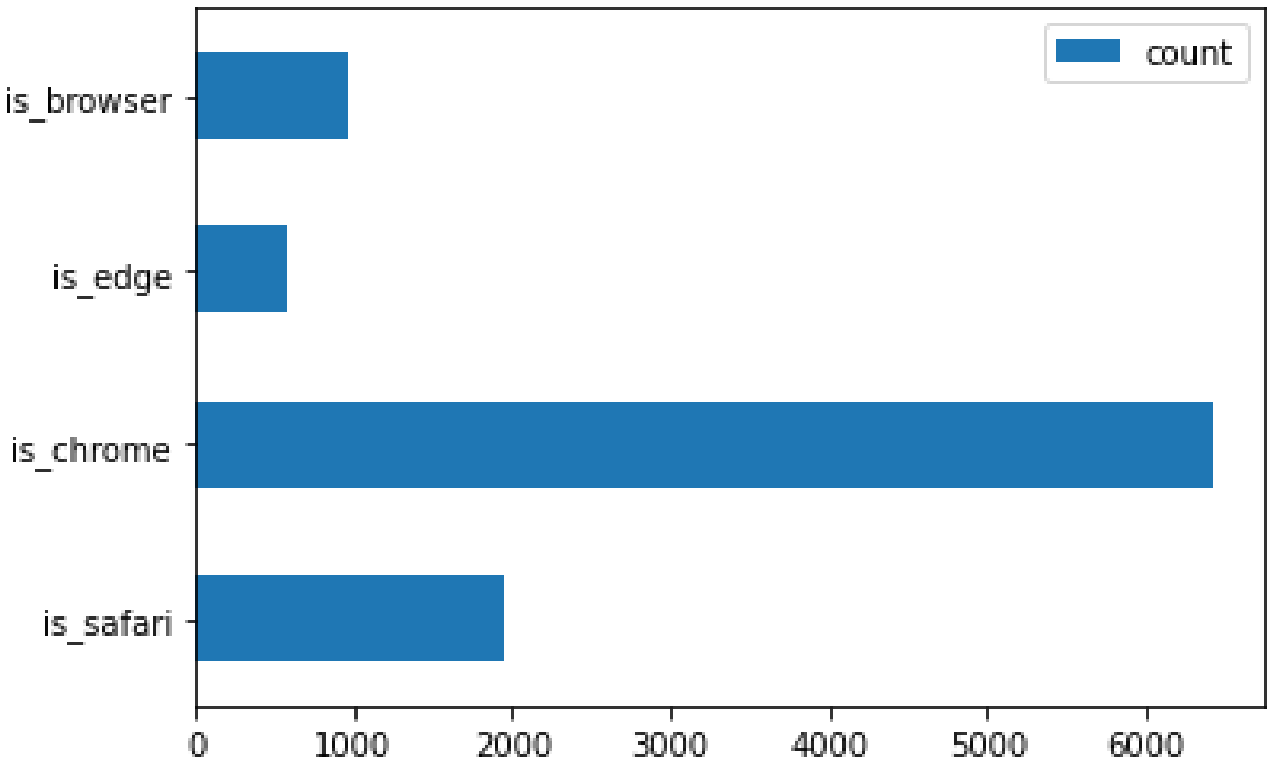
2.3 גרפי עמודות לצורך מילוי ערכים ריקים

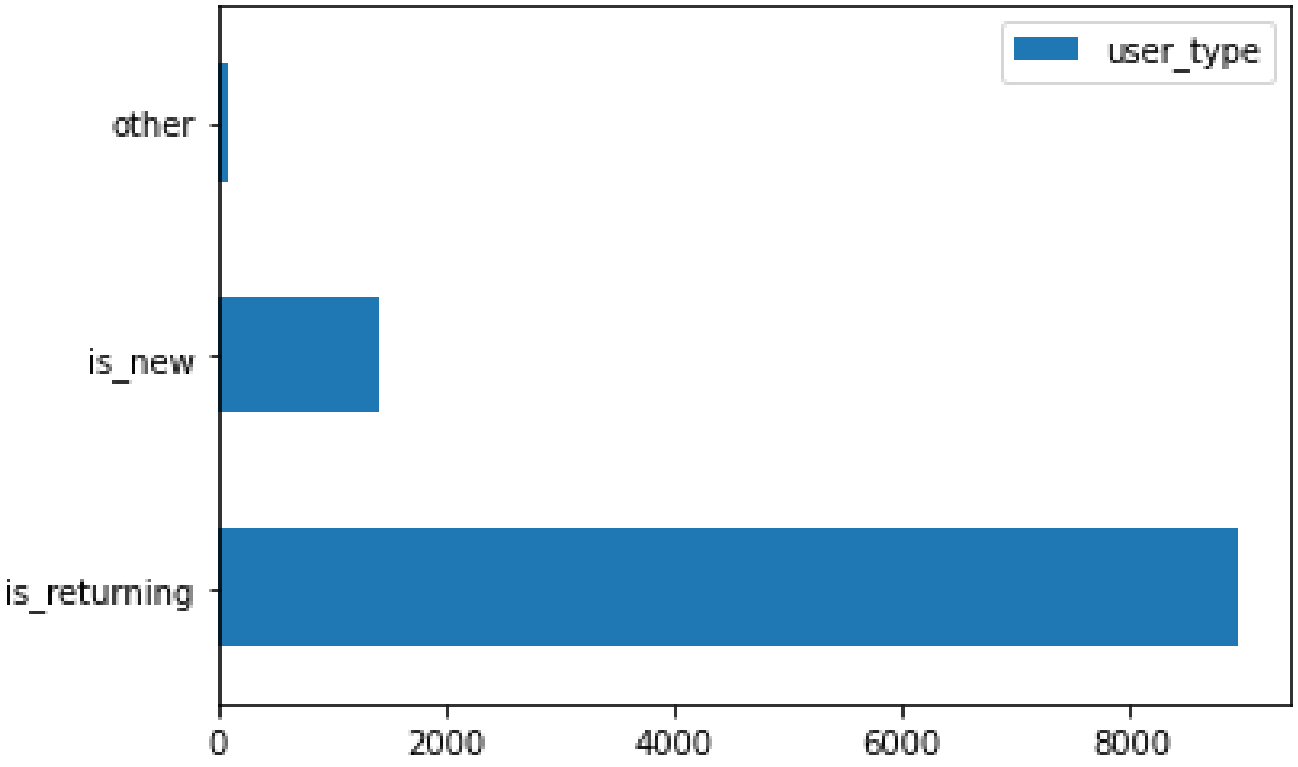




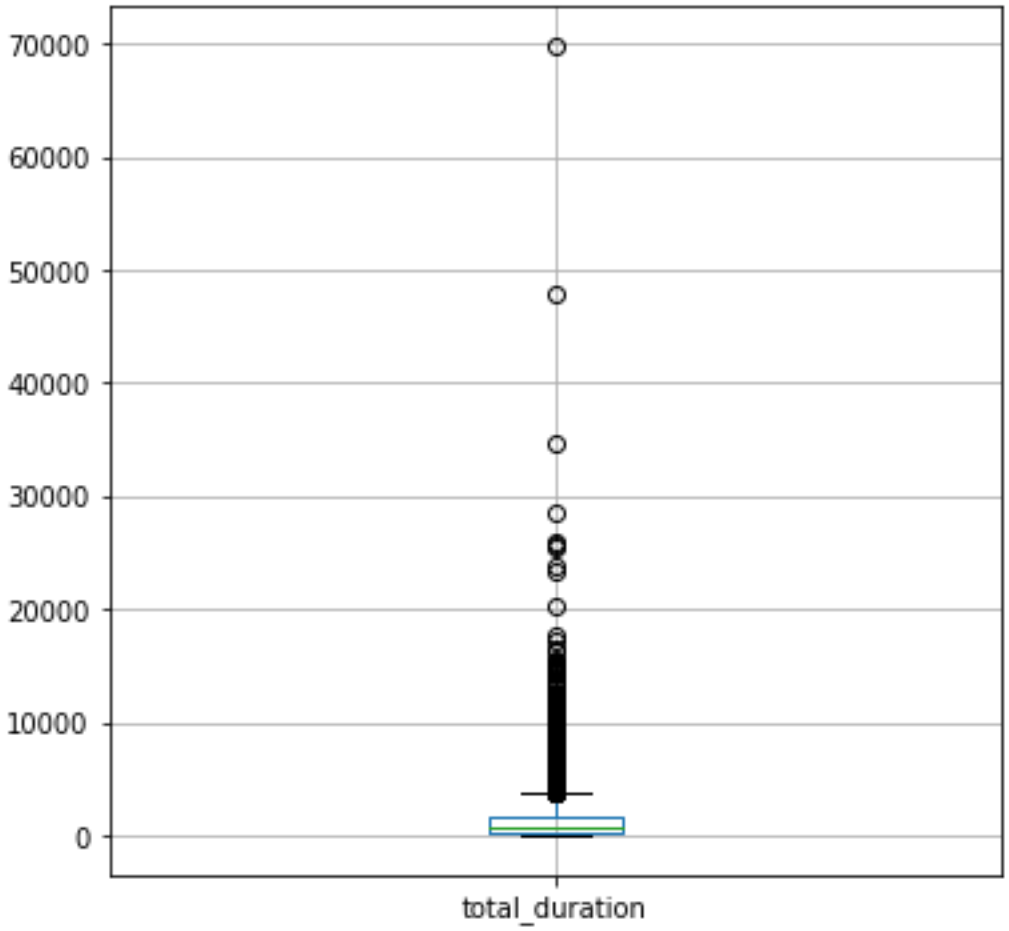

1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
0
500
1000
1500
2000
2500
3000
3500
4000

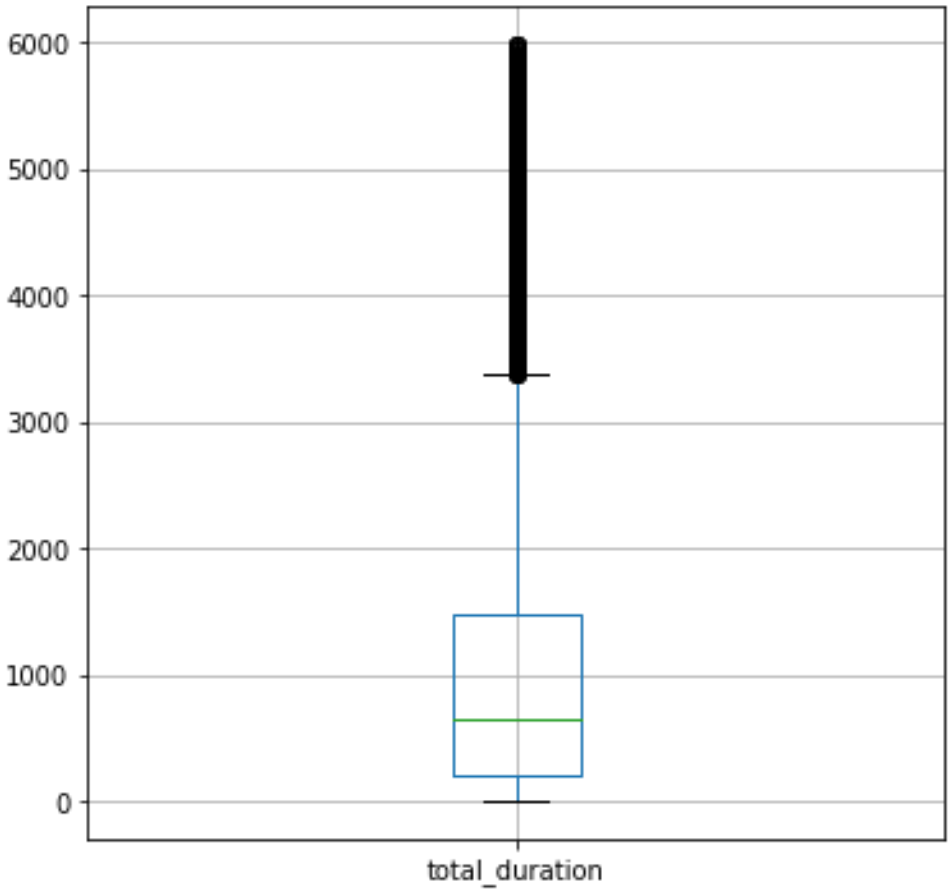

is_log8080
is_log404
is_log400
is_log202
is_log200
is_log_100
C
0
250
500
750
1000
1250
1500
1750

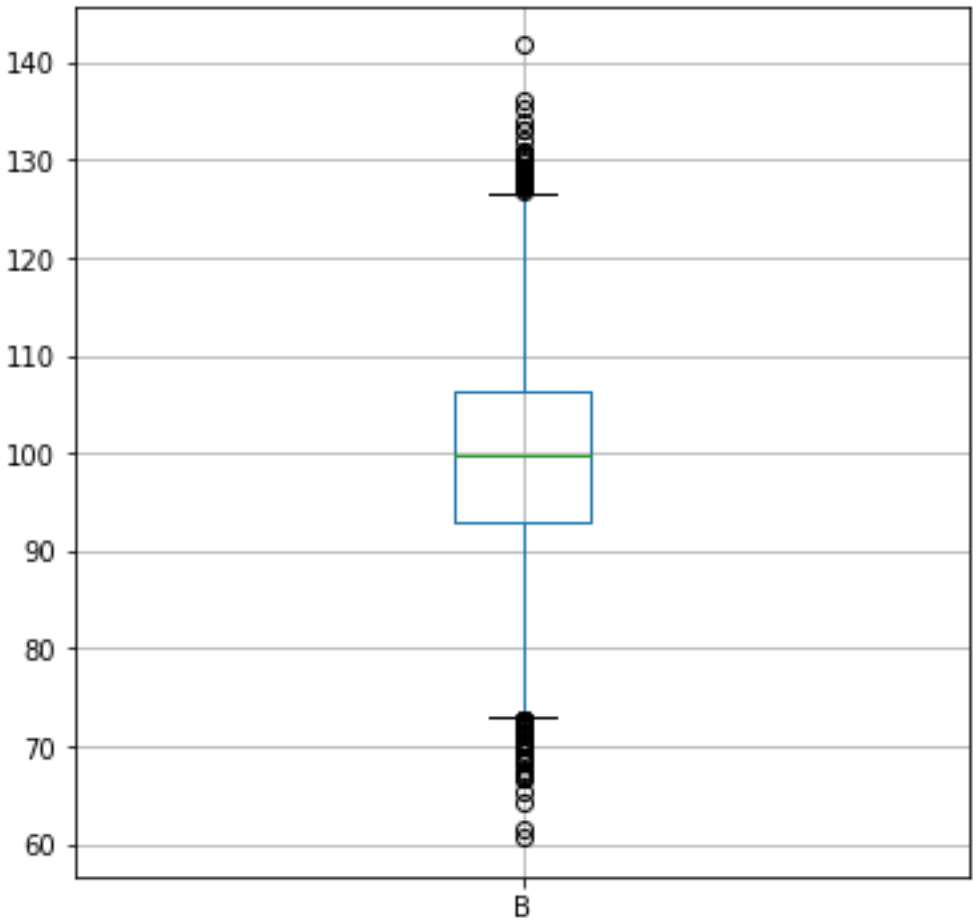

count
is_browser
is_edge
is_chrome
is_safari
0
1000
2000
3000
4000
5000
6000

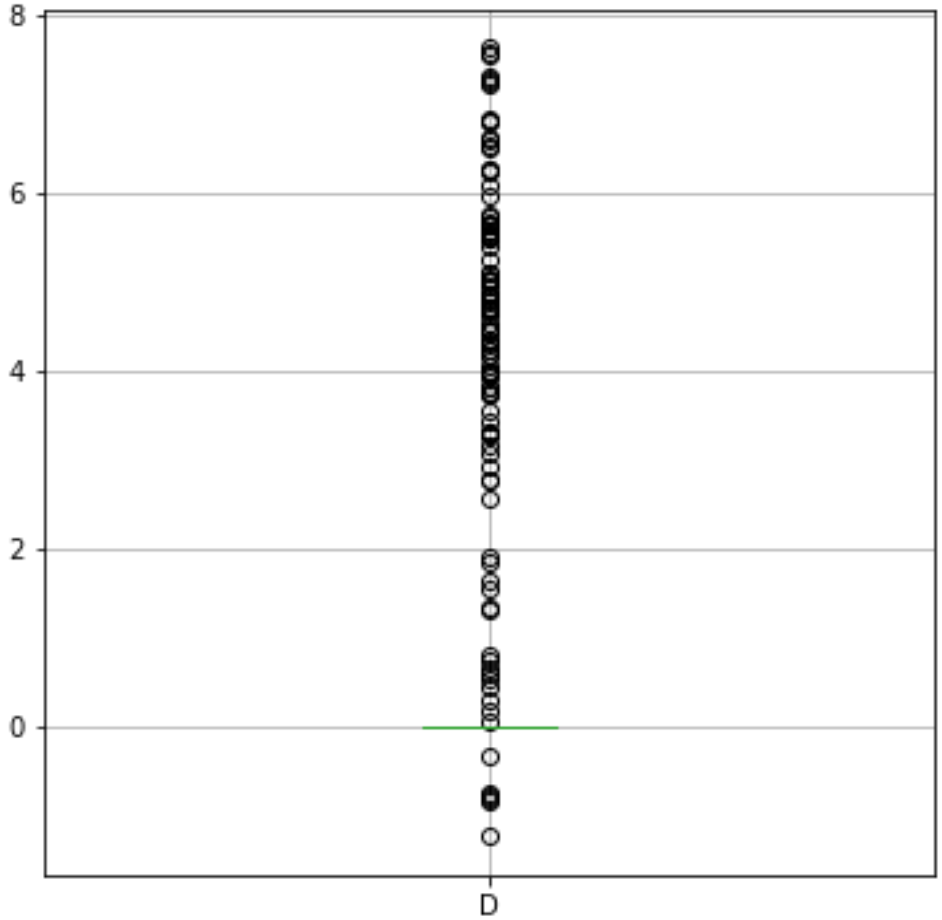

user_type
other
is_new
is_returning
0
2000
4000
6000
8000

boxplot 24

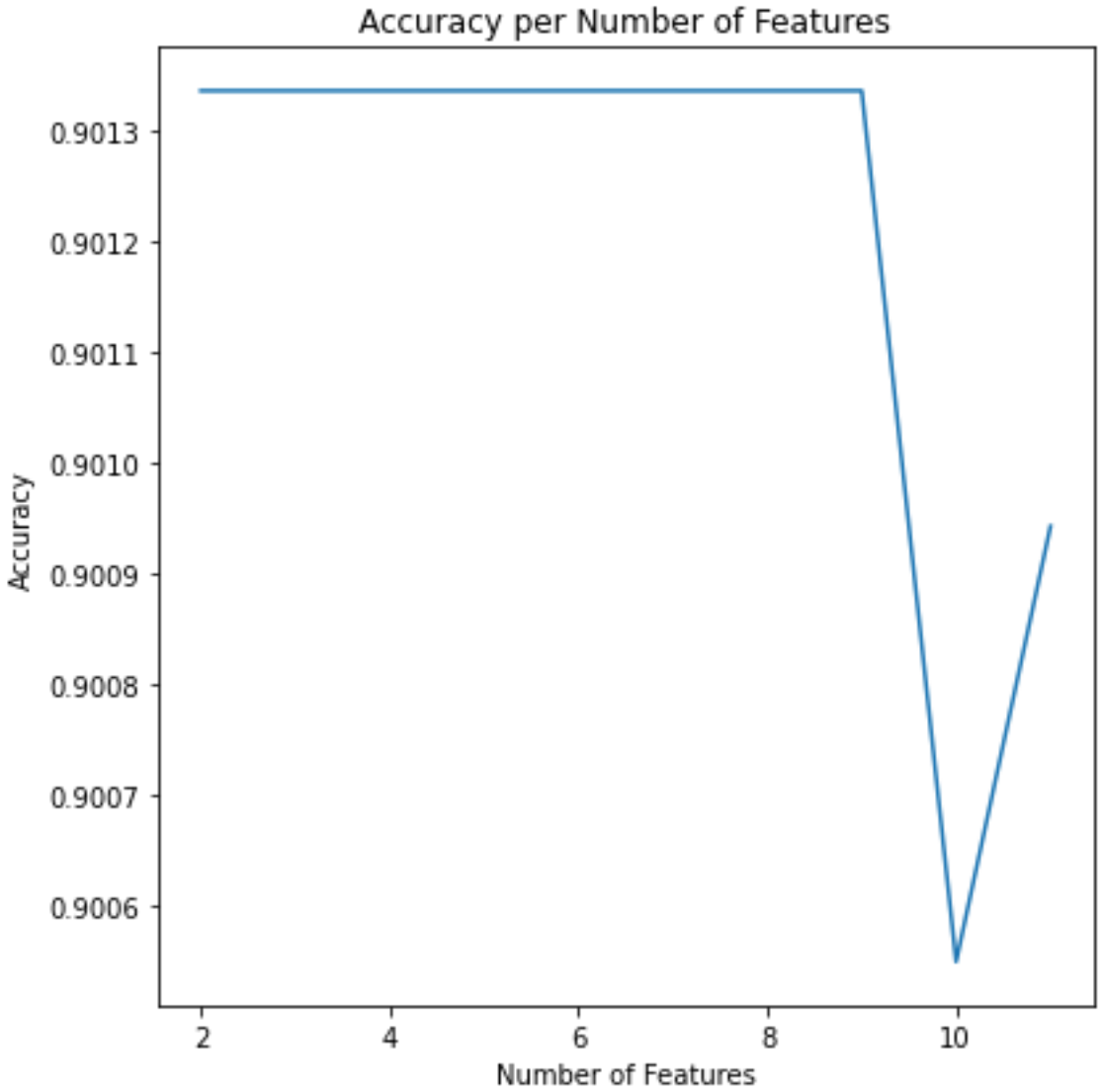

70000
60000
50000
40000
30000
20000
10000
0
total_duration

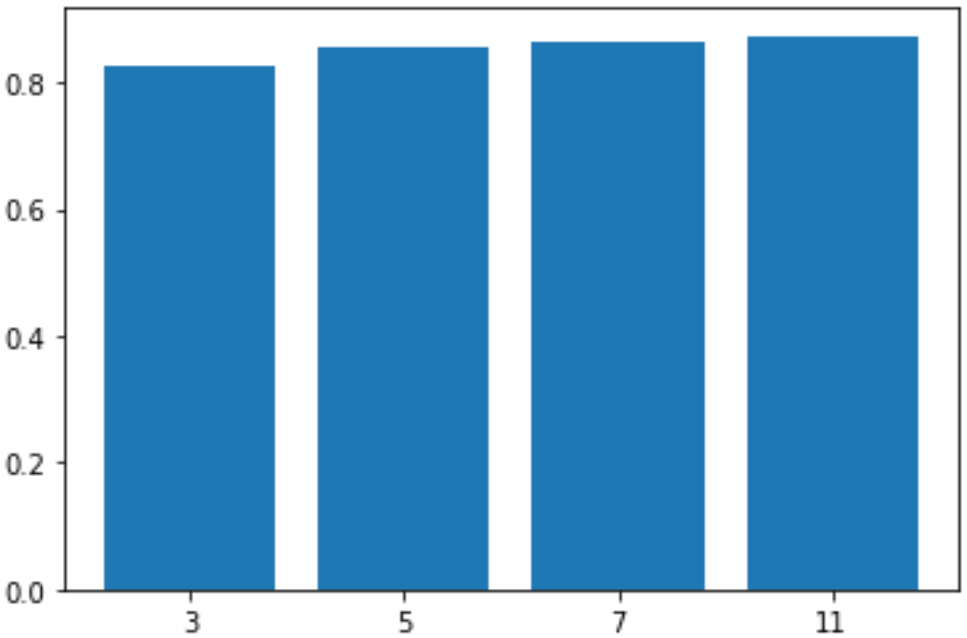

6000
5000
4000
3000
2000
1000
0
total_duration


140
130
120
110
100
90
80
70
60
B

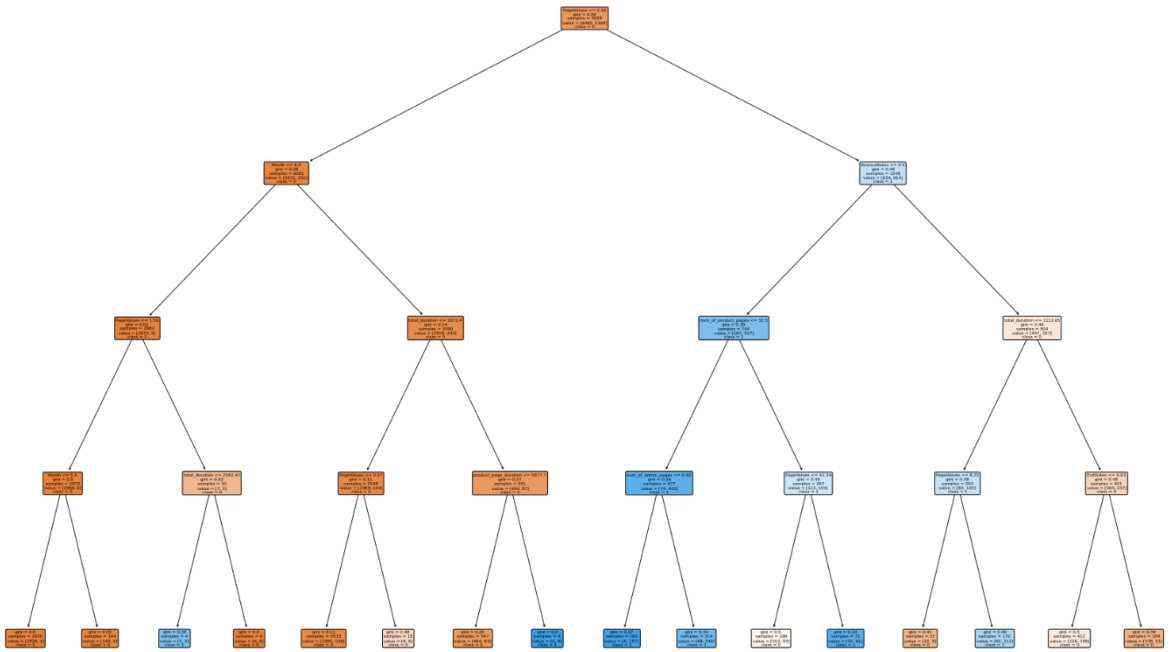

8
6
4
2
0
D

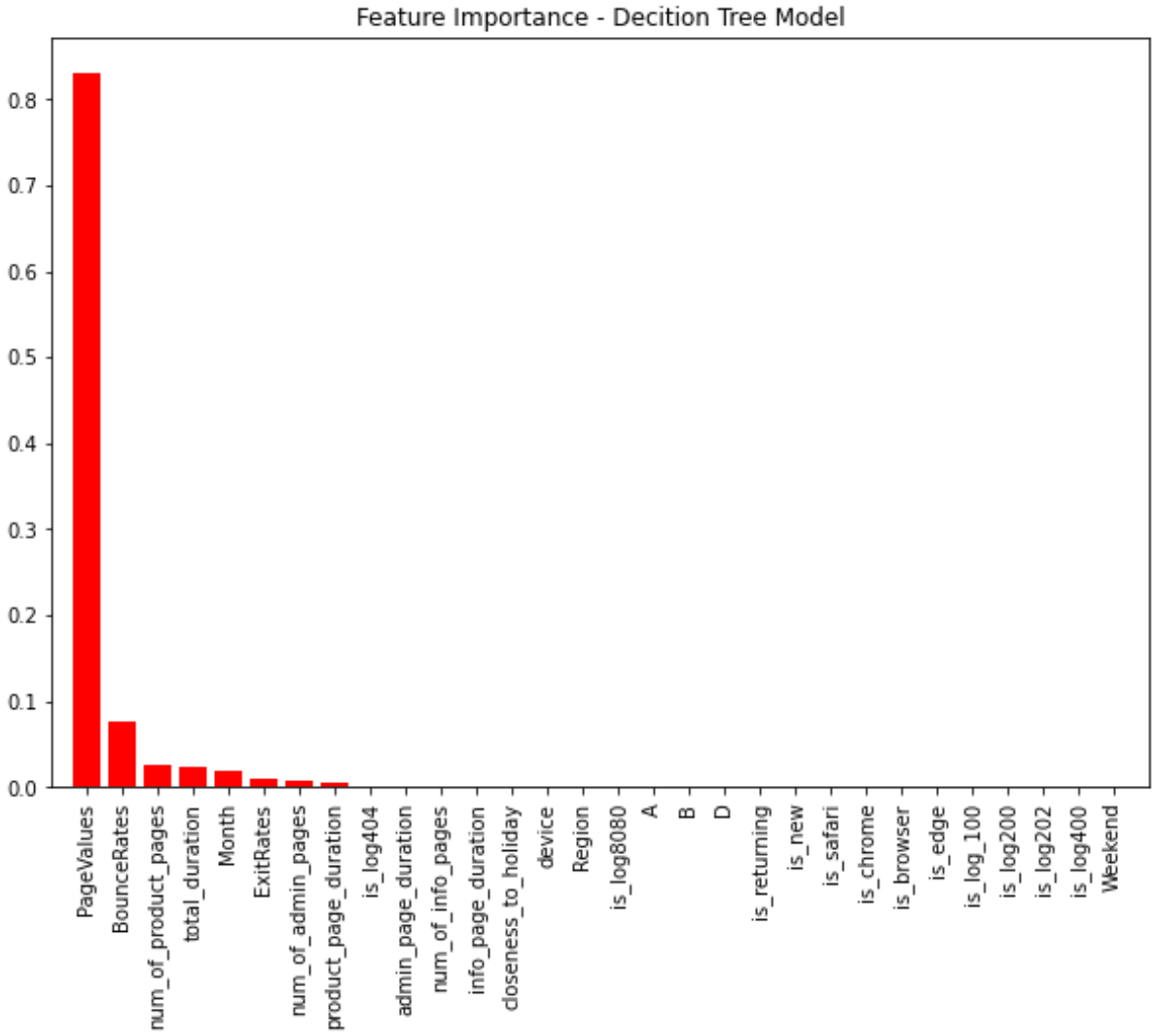
2.5 בחירת פיצ'רים והיפר פרמטרים למודלים פשוטים



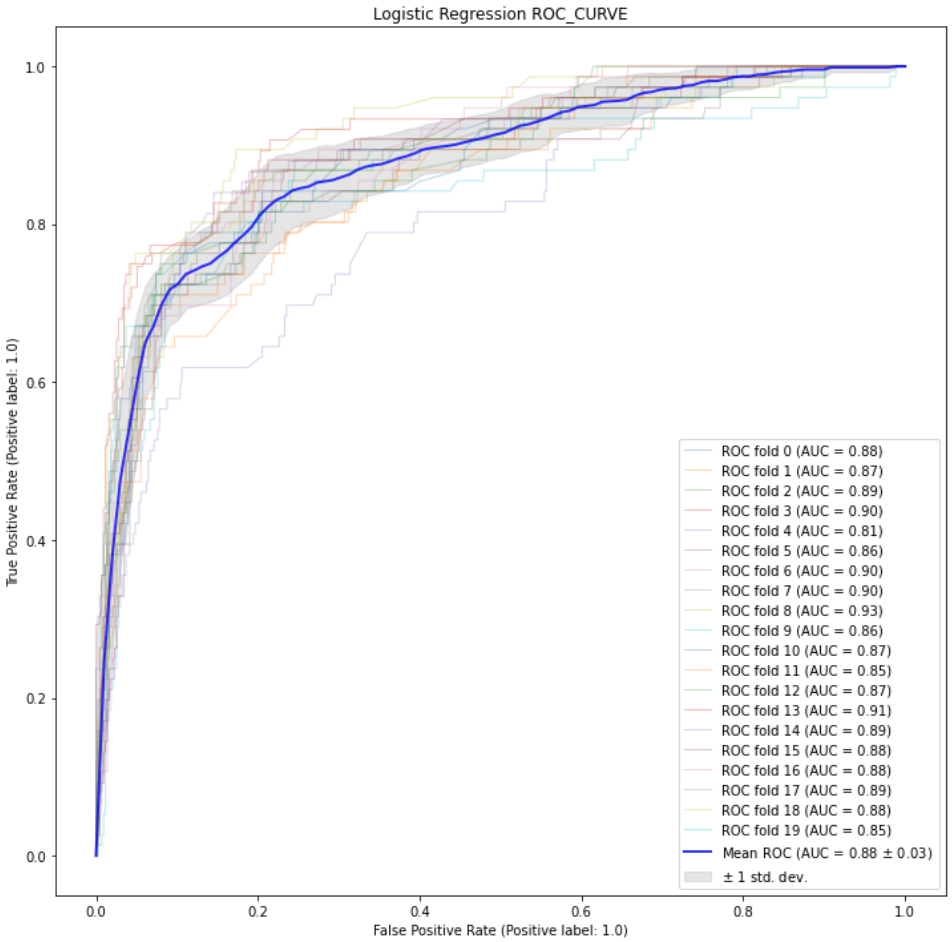


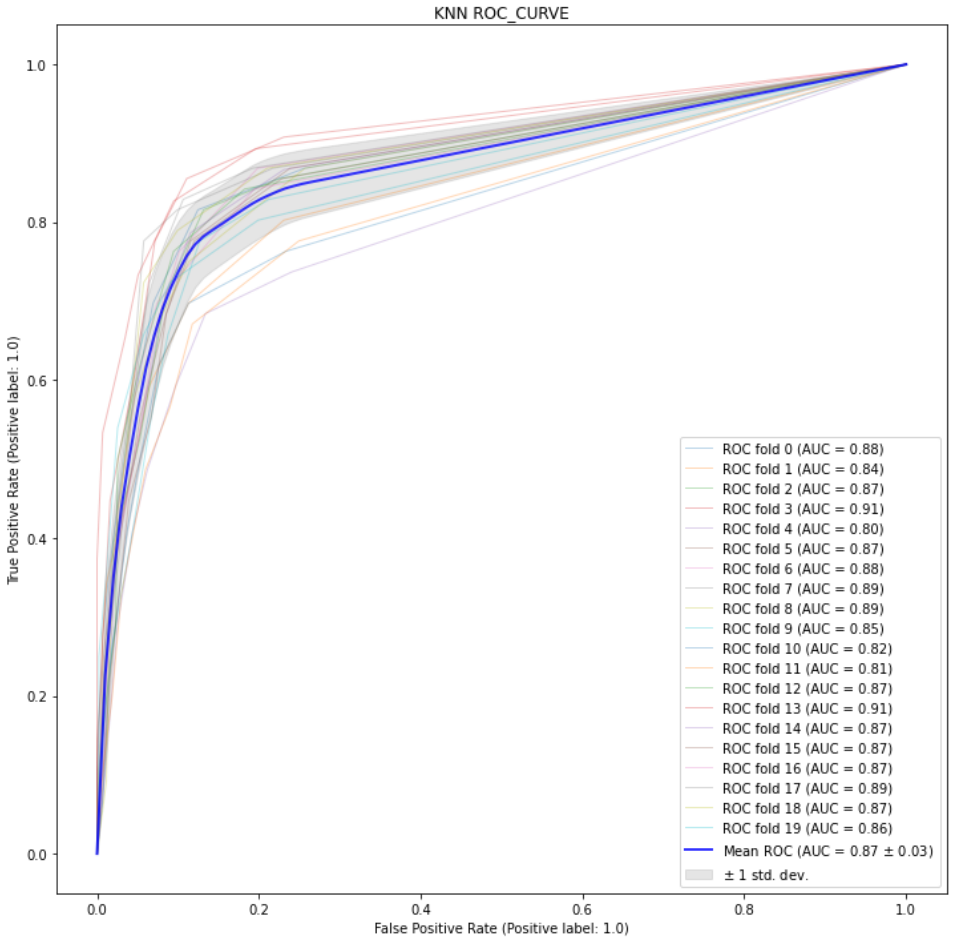
2.6 מודלים מורכבים

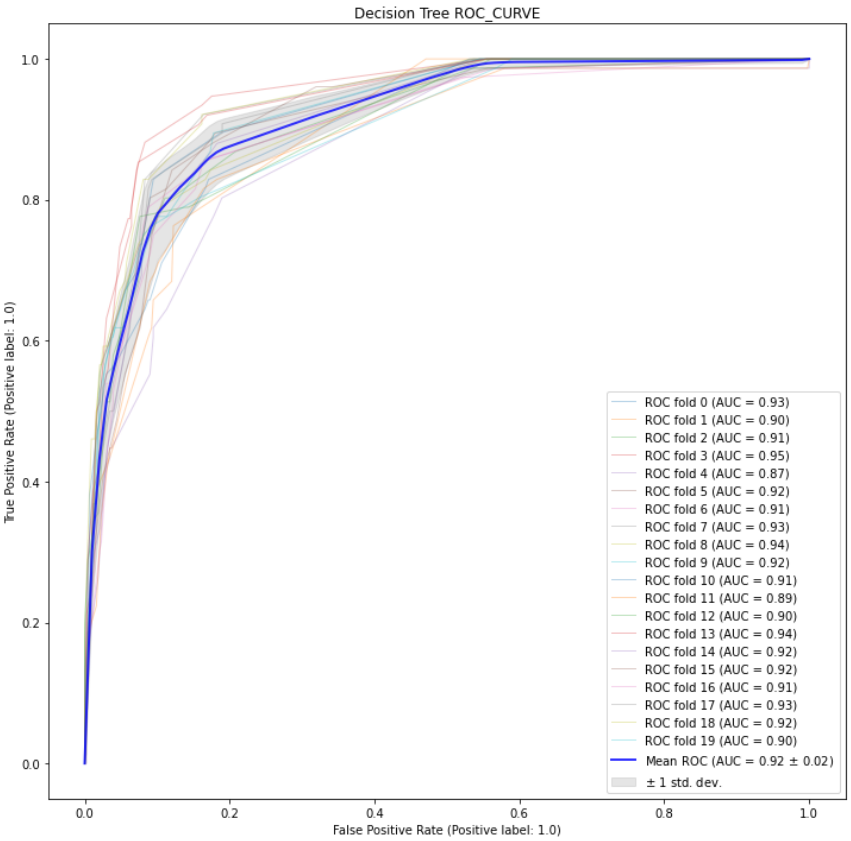


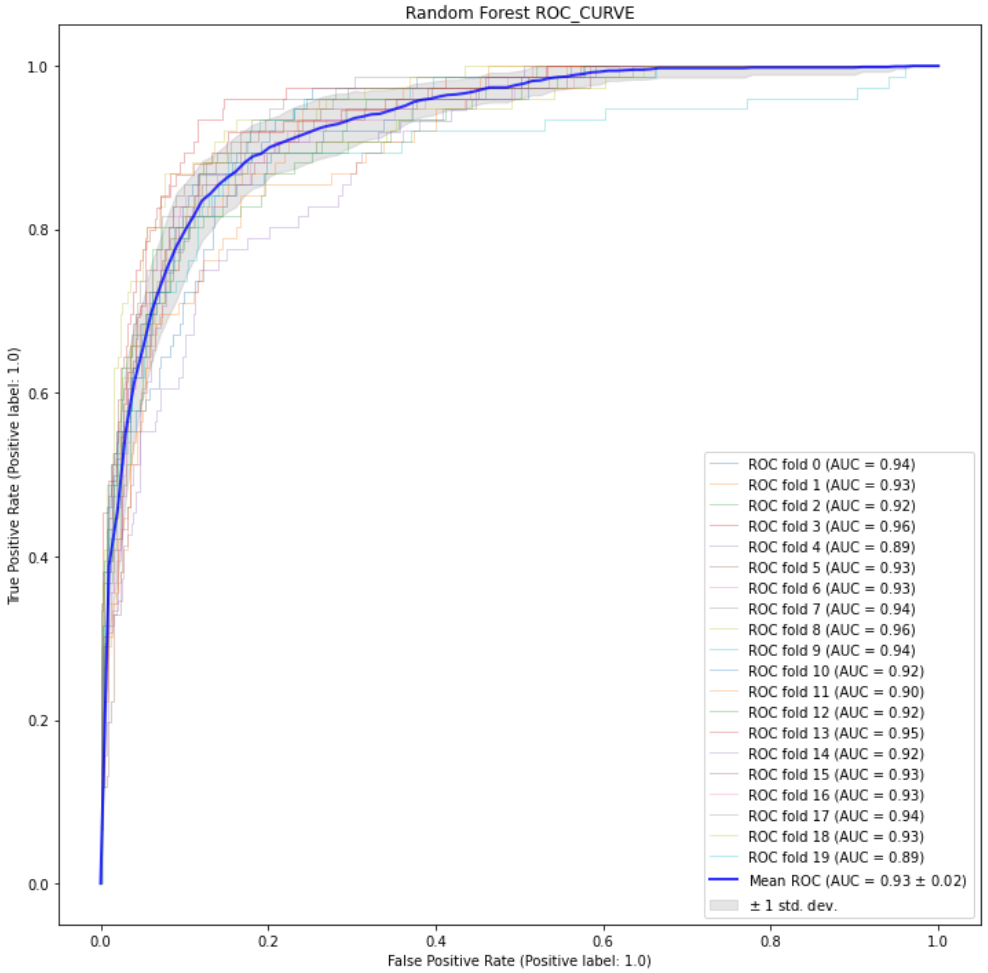


2.7 Roc_curve לכל מודל






Decision Tree ROC_CURVE
True Positive Rate (Positive label: 1.0)
False Positive Rate (Positive label: 1.0)
0.0
0.2
0.4
0.6
0.8
1.0
ROC fold 0 (AUC = 0.93)
ROC fold 1 (AUC = 0.90)
ROC fold 2 (AUC = 0.91)
ROC fold 3 (AUC = 0.95)
ROC fold 4 (AUC = 0.87)
ROC fold 5 (AUC = 0.92)
ROC fold 6 (AUC = 0.91)
ROC fold 7 (AUC = 0.93)
ROC fold 8 (AUC = 0.94)
ROC fold 9 (AUC = 0.92)
ROC fold 10 (AUC = 0.91)
ROC fold 11 (AUC = 0.89)
ROC fold 12 (AUC = 0.90)
ROC fold 13 (AUC = 0.94)
ROC fold 14 (AUC = 0.92)
ROC fold 15 (AUC = 0.92)
ROC fold 16 (AUC = 0.91)
ROC fold 17 (AUC = 0.93)
ROC fold 18 (AUC = 0.92)
ROC fold 19 (AUC = 0.90)
Mean ROC (AUC = 0.92 ± 0.02)
± 1 std. dev.


Random Forest ROC_CURVE
True Positive Rate (Positive label: 1.0)
False Positive Rate (Positive label: 1.0)
0.0
0.2
0.4
0.6
0.8
1.0
ROC fold 0 (AUC = 0.94)
ROC fold 1 (AUC = 0.93)
ROC fold 2 (AUC = 0.92)
ROC fold 3 (AUC = 0.96)
ROC fold 4 (AUC = 0.89)
ROC fold 5 (AUC = 0.93)
ROC fold 6 (AUC = 0.93)
ROC fold 7 (AUC = 0.94)
ROC fold 8 (AUC = 0.96)
ROC fold 9 (AUC = 0.94)
ROC fold 10 (AUC = 0.92)
ROC fold 11 (AUC = 0.90)
ROC fold 12 (AUC = 0.92)
ROC fold 13 (AUC = 0.95)
ROC fold 14 (AUC = 0.92)
ROC fold 15 (AUC = 0.93)
ROC fold 16 (AUC = 0.93)
ROC fold 17 (AUC = 0.94)
ROC fold 18 (AUC = 0.93)
ROC fold 19 (AUC = 0.89)
Mean ROC (AUC = 0.93 ± 0.02)
± 1 std. dev.

2.8 confusion matrix לפי random forest