הפקולטה להנדסה
ע"ש איבי ואלדר פליישמן
אוניברסיטת תל אביב

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

פרויקט בקורס "מבוא ללמידת מכונה"

דוח מסכם

מרצה: מר דור בנק

מתרגל: אילן וסילבסקי

מגישות – קבוצה 10:

רוני אביבי

יעל הרמן ליזנר

תקציר מנהלים:

בפרויקט זה נעסוק בבעיית Binary Classification בה נסווג רשומות לשתי קטגוריות - האם משתמש צפוי לרכוש (1) או לא (0), על סמך 21 פיצ'רים בסט נתונים נתון. נבדוק 4 מודלים: SVM, Logistic regression, KNN, RandomForest. מטרת הפרויקט היא לבצע חיזוי מדויק ככל הניתן, נבדוק זאת באמצעות מדד ה- AUC אשר אותו נשאף למקסם.

תחילה, ביצענו חקירה של הנתונים, אשר בעיקרה ויזואלית ובחלקה כללה התאמות מינימליות של הנתונים לצורך עיבוד מיטבי בהמשך. מחקירה זו הפקנו מסקנות ותובנות רבות בעלות ערך רב על סט הנתונים. לאחר מכאן ביצענו עיבוד מקדים ל-Train set, מילאנו את הערכים החסרים, ביצענו הוצאת חריגים, ביצענו התאמות נדרשות (טרנספומציית לוג, הסרת פיצ'רים קורלטיבים) ונרמלנו את הנתונים. בנוסף, יצרנו פיצ'רים חדשים : "Browser_new", "A_new". את הפיצ'רים הקטגוריאליים בעלי ערכים אשר הינם מחרוזות, המרנו למספר פיצ'רים בוליאנים באמצעות הפונקציה Get Dummies. על מנת להוריד ממדיות לבעיה השתמשנו ב-PCA.

לאחר הרצת המודלים, בשלב האחרון של הפרויקט ביצענו הערכה לטיב המודלים. זו התבססה על שיטת Cross validation תוך שימוש ב-K-fold בו הרצנו את המודלים אשר נבנו בפרויקט. בכל פעם, ההרצה בוצעה על Train ו- Validation אחרים מתוך סט הנתונים. טרם הרצת המודלים, ביצענו חיפוש אחר השיטות האופטימליות לנרמול וההיפר פרמטרים הטובים ביותר עבור כל מודל שיביאו למדד ה-AUC הגבוה ביותר. עבור כל הרצה של מודל עם הפרמטרים המיטביים שנמצאו הצגנו עקומת ROC. כמו כן, עבור מודל SVM בנינו Confusion Metrix. לבסוף, המודל הטוב ביותר הנבחר בפרויקט הינו SVM עם מדד AUC של 0.918.

חלק ראשון - אקספלורציה:

בשלב זה בוצעו חיתוכים וניתוחים ויזואליים על הדאטה על מנת לאפיין את הנתונים. במהלך האקספלורציה ננקטו השלבים הבאים: לאחר טעינת הספריות הרלוונטיות וטעינת הדאטה ל-DataFrame ראינו כי מדובר בבעיה בעלת 21 פיצ'רים ו- 10479 דגימות, וכי ה-labels מתפלגים באופן שאינו מאוזן ולייבל "0" קיים הרבה יותר מלייבל "1" (15.5% לעומת 84.5%). כלומר, אנחנו נמצאים במצב של imbalanced data מה שישפיע במידה מסוימת על התחזיות בהמשך, שכן ייתכן שהמודלים ידעו להתמודד בצורה טובה יותר עם רשומות שאמורות לקבל label 0 אך לא עם כאלו שיקבלו label 1.

מתוך הפיצ'רים, 13 מכילים ערכים מספריים (כולל פיצ'ר אחד בוליאני) ו-8 פיצ'רים מכילים ערכים קטגוריאליים. לאחר שמצאנו פיצ'רים בעלי מספר נמוך של קטגוריות ייחודיות (חשודים כקטגוריאליים), כתבנו פונקציה שמשנה פיצ'רים ('device', 'Region') שהם בעלי ערכים מספריים לקטגוריאליים כיוון שאין חשיבות לערך המספרי שלהם, משנה את ערכי החודשים בפיצ'ר Month למחרוזות של ספרות (לצורך הנוחות) ומשנה את הערכים הבוליאניים של הפיצ'ר 'Weekend' ל-1 ו-0. בנוסף, הסרנו את המחרוזת "minutes" מהפיצ'רים info_page_duration ומ- product_page_duration בכדי שנוכל להתייחס למשתנים אלו כנומריים. לאחר השינויים התקבלו 14 פיצ'רים נומריים (כולל פיצ'ר אחד בוליאני) ו-7 פיצ'רים קטגוריאליים.

לאחר מכן, הצגנו טבלה המציגה עבור כל פיצ'ר נתונים סטטיסטיים - מצאנו כי קיים הבדל בטווח

הערכים שלהם ולכן ישנה חשיבות לביצוע נרמול בהמשך. ולבסוף ספרנו כמה ערכים חסרים קיימים בכל פיצ'ר וראינו כי בפיצ'ר D קיימים 98.9% ערכים חסרים – בחרנו להסיר אותו בהמשך.

<u>ניתוח הדאטה של הפיצ'רים הנומריים (int & float):</u> ראשית, שרטטנו היסטוגרמה לכל פיצ'ר נומרי על מנת לצפות בצורה בהירה בהתפלגותו וכן להתחיל לזהות ערכים חריגים הקיימים בעמודה. ראינו כי רוב ההתפלגויות מוטות חיובית וכי הפיצ'ר B מתפלג נורמלית. לבסוף בחנו את הקורלציות שבין כל פיצ'ר לפיצ'רים אחרים באמצעות heatmap של מטריצת הקורלציות. ראינו שהפיצ'רים:
'product_page_duration' ו- 'total_duration', 'BounceRates' ו- 'ExitRates',
'num_of_product_pages' ו- 'total_duration', 'num_of_product_pages' ו-
'product_page_duration' בעלי קורלציה של 0.99, 0.91, 0.88, 0.86 בהתאמה ושרטטנו קורלציות אלה על מנת לקבל התרשמות ויזואלית. בנוסף בדקנו קורלציה של הפיצ'רים עם ה- labels – ראינו כי לפיצ'ר D הקורלציה הגבוהה ביותר אך בהתחשב בעובדה כי הוא בעל 98.9% ערכים חסרים, הקורלציה מבוססת על מספר דוגמאות קטן, מטעה וחסרת משמעות אמיתית.

<u>ניתוח הדאטה של הפיצ'רים הקטגוריאליים (str):</u> עבור כל פיצ'ר הצגנו היסטוגרמה המראה את התפלגותו וערכיו. בבדיקה מקדימה ראינו שהפיצ'רים 'A' ו- 'internet_browser' בעלי ערכים ייחודיים רבים, ולכן בחרנו לשרטט אותם בנפרד עם threshold של 0.005 על מנת לסנן outliers ולהוריד מימדים בשביל פונקציית get_dummies שתגיע בהמשך. ראינו כי בפיצ'ר 'internet_browser' ההתפלגות של כל browser כללי (chrome, safari, edge) יחסית דומה ונרצה לאחד ביניהם בהמשך.

<u>חלק שני – עיבוד מקדים:</u>

העיבוד המקדים התבצע על סט הנתונים של ה-Train וה-Test בצורה זהה הכוללת את השלבים הבאים:
- <u>חלוקת הדאטה לtrain set, validation set</u> - ביחס של 30% validation, 70% train.
- <u>ערכים חסרים</u> – עבור פיצ'רים בעלי התפלגויות מוטות חיוביות בחרנו למלא NA עם החציון שכן הוא הכי מתאים במקרה זה. עבור "total_duration", הסקנו מהאקספולרציה כי הוא סוכם את 3 משתני ה-duration האחרים ולכן השלמנו אותו באמצעות סכימתם. עבור פיצ'ר 'B' המתפלג נורמלית השלמנו עם הממוצע ועבור שאר הפיצ'רים השלמנו באמצעות הערך הנפוץ / ערך רנדומי. את כלל החישובים בחלק זה ביססנו על ה-train set. כמו כן בשלב זה מחקנו את פיצ'ר D שכן הוא ברובו ערכים חסרים ויכול להטות את המודל.
- <u>יצירת פיצ'רים חדשים</u> - בנינו שתי פונקציות: 'create_browser_new' , 'create_A_new' אשר יוצרות פיצ'רים חדשים במקום 'internet_browser' ו- 'A'. הפונקציה הראשונה מאחדת את שלושת סוגי ה-browsers העיקריים, וכל browser שלא שייך ל-safari, chrom, או edge יסווג כ-"other". הפונקציה השנייה מסווגת כל ערך במשתנה "A" ששכיחותו קטנה מ-0.005 כ"other" ובכך מקטינה את מספר הערכים הייחודיים בפיצ'ר.
- <u>מחיקת פיצ'רים</u> - בחרנו למחוק פיצ'רים בעלי קורלציה גבוהה מ-0.9 מהסיבה שהם לא מלמדים אותנו מידע חדש ורק מעלים מימדיות ושונות. הפיצ'רים שמחקנו הם 'product_page_duration' (שהיה בקורלציה גבוהה של 0.99 עם הפיצ'ר 'total_duration') ו- 'BounceRates' (שהיה בקורלציה גבוהה של 0.91 עם הפיצ'ר 'ExitRates').

- <u>טרנספורמציית Log לדאטה</u> - מאחר שגילינו בשלב הראשון של האקספולרציה כי רוב הפיצ'רים שלנו מתפלגים בצורה מוטה (skewed distribution) בחרנו לבצע Log transformation לפיצ'רים, שמחליף כל משתנה x ב-log(x) ובכך גורם להתפלגות של כל פיצ'ר להתקרב לנורמלית. נרצה שההתפלגות של הפיצ'רים תהיה פחות מוטה מהסיבות שהטרנפורמציה הופכת דפוסים בדאטה ליותר ברורים ומקלה הנחת מסקנות סטטיסטיות של הדאטה.
- <u>הסרת חריגים עבור פיצ'רים נומריים</u> - בתהליך האקספלורציה ראינו בהיסטוגרמות ואז ביתר פירוט ב-Boxplots ששרטטנו שלחלק מהפיצ'רים ישנם ערכים חריגים. על אף זאת, זה מנהג רע להסיר data points בשביל לייצר better fitting model או תוצאות מובהקות יותר סטטיסטית. הפיצ'רים בהם זיהנו ערכים חריגים הם: 'num_of_info_pages', 'info_page_duration', 'num_of_product_pages' 'total_duration', 'PageValues', 'B', 'ExitRates' . קיבלנו החלטה לא להסיר שורות עם ערכים חריגים אלא לבצע clipping. בשיטה זו הגדרנו גבול עליון - אחוזון 0.95 וגבול תחתון - אחוזון 0.05 והגבלנו את הערכים באותם פיצ'רים להיות בין הגבולות הללו ע"י שימוש בפונקציה מובנית clip. ערכים שהסיגו גבולות אלה, הומרו לערך הגבוה/נמוך ביותר האפשרי.
החלטנו להשתמש ב-clipping במקום הסרת שורות של ערכים חריגים מכמה סיבות:
1. הדאטה שלנו לא גדול ולכן ידינו לא קלה על ההדק. העדפתנו הייתה למחוק כמה שפחות דוגמאות.
2. ה-outliers לא היו מאוד קיצוניים ולכן הגיוני להשתמש ב-clipping ולהגבילם מאשר למחוק אותם.
3. בחרנו לבצע לדאטה Log transformation. אחת המטרות בטרנספורמציה זו היא צמצום ההשפעה של חריגים על הדאטה. הרעיון הוא שביצוע log על הדאטה יכול לשפר את הסימטריה של הדאטה.

-<u>נרמול הנתונים</u> - כפי שראינו בשלב האקספלורציה, קנה המידה של הפיצ'רים השונים הינו בטווחים שונים. נרצה לנטרל השפעה זו על ידי ביצוע נרמול והעברת כל הפיצ'רים לקנה מידה אחיד. פעולת הנרמול חשובה עבור שיטת PCA בה נשתמש בהמשך, בה השונות של הפיצ'ר היא באופן יחסי לשונות של שאר הפיצ'רים, במידה ולא ננרמל נוכל לקבל תוצאות שגויות. בחרנו לנרמל בשיטת RobustScalar שמותאמת יותר לדאטה בעל skewed distribution בגלל שהשיטה מבצעת טרנספורמציה לדאטה בהתבסס על החציון. לאחר נרמול הנתונים המרנו את המשתנים הקטגוריאליים לבינארים באמצעות הפונקציה המובנית get_dummies. פונקציה זו מפצלת כל פיצ'ר קטגוריאלי למספר עמודות כמספר הערכים הייחודיים בפיצ'ר, כאשר כל תצפית מקבלת ערך בינארי 0 או 1 בעמודה הרלוונטית לפי ערכה.

<u>Feature Selection - ממדיות הבעיה</u>:
תחילה ממדיות הבעיה הייתה 21 פיצ'רים. לאחר שימוש בפונקציה Get_dummies והפיצ'רים החדשים שהוספנו קיבלנו 64 פיצ'רים. בהתאם לכלל האצבע האומר שמספר התצפיות הנדרשות הינו מספר הפיצ'רים בריבוע, כלומר 4096, וברשותנו סט נתונים של 10478 (לאחר חלוקה ל-train ו-validation 7335 בסט ה-train), נרצה להקטין את ממדיות הבעיה ולקבל כמות מינימלית של פיצ'רים המכילים מקסימום מידע.
חסרונות בבעיה בעלת ממדיות גדולה:
ככל שכמות הפיצ'רים יותר גדולה, השונות גדלה ועולה הסיכון לoverfitting ולפגיעה ביכולת החיזוי ב-test. כמו כן, נוצר קושי בהבנת התוצאה - ריבוי פיצ'רים מקשה על ההבנה של מי מהפיצ'רים משפיע יותר באופן יחסי. בנוסף לאלה, ריבוי פיצ'רים גורר סיבוכיות גבוהה וזמן חישוב ארוך.
נתמודד עם הממדיות הגדולה בעזרת PCA ובעזרת הסרת פיצ'רים בעלי אחוז נתונים חסרים גדול או

בעלי קורלציה גבוהה (עליהם פירטנו לעיל ב"מחיקת פיצ'רים" וב-"ערכים חסרים").
הקטנת ממדיות הבעיה: PCA - מצאנו כי על מנת להסביר 95% מהשונות מספיק להשתמש בכ-33 פיצ'רים.

חלק שלישי – הרצת המודלים:

על מנת למצוא את המודל האופטימלי נעשתה בדיקה בשני שלבים: בחירת ההיפר פרמטרים הטובים ביותר למודל - בעזרת פונקציית GridSearchCV. ההיפר פרמטרים שנבחרו הם אלו שנתנו למודל את ה-AUC הגבוה ביותר.

Logistic Regression: היפר הפרמטרים שנבחנו: **1)** Penalty - ערך מוחלט וערך מוחלט ריבועי. חשוב לבחון את פונקציית הקנס לאור העובדה שפונקציה זו מבצעת רגולריזציה ע"י מניעת overfitting של המודל בכך שמגדילה את הקנס ככל שמורכבות המודל גדלה. **2)** C - ערכים קטנים מובילים לרגולריזציה קשה. **3)** solver - אלגוריתם לבעיית האופטימיזציה. **4)** class_weight = ['balanced'] - מצב balanced מתאים למצב של imbalanced data.

היפר פרמטרים שנבחרו: (C= 0.1, class_weight= 'balanced', penalty= 'l1', solver= 'liblinear')

K-NN Classifier Algorithm: היפר הפרמטרים שנבחנו: **1)** n_neighbors – מספר השכנים. **2)** weights - בפונקציה זו משתמשים בפרדיקציה. המשמעות של הפרמטרים היא: Uniform= כל הנקודות בכל שכונה שוקלות באותו המשקל לעומת distance= משקל הנקודות מחושב ע"י ה-inverse של מרחקן. **3)** p - כאשר p=1 הדבר שקול לשימוש ב-manhattan distance וכאשר p=2 הדבר שקול לשימוש ב-euclidean distance. **4)** סוגי metric (מטריקה) שונים.

היפר פרמטרים שנבחרו: (metric= 'minkowski', n_neighbors= 100, p= 2, weights= 'distance')

SVM: Support Vector Machine הוא מודל שמייצר מסווג אשר יוצר מרווח גדול ככל האפשר בינו לבין הדוגמאות הקרובות לו בשתי הקטגוריות. היפר הפרמטרים אשר נבחנו הם: **1)** C- פרמטר הרגולריזציה. **2)** kernel- סוג הגרעין לשימוש באלגוריתם. **3)** gamma – kernel coefficient.

היפר פרמטרים שנבחרו: (C= 15, class_weight= 'balanced', gamma= 0.01, kernel= 'rbf', probability= True)

Random Forest Classifier: מודל הבנוי מעצי החלטה. המודל משתמש ב-Bootstrap ובהתחשבות בקבוצות רנדומליות של פיצ'רים במהלך החלוקה לעלים. היפר הפרמטרים שנבחנו: **1)** n_estimators- מספר העצים ביער. **2)** criterion- הפונקציה שמודדת את איכות החלוקה. **3)** max_depth- העומק המקסימלי של כל עץ. **4)** random_state – שולט ברנדומליזציה של ה-boostrapping של הדוגמאות איתן המודל בונה את העצים. בחרנו לבחון random seed=0, שנחשב ערך פופולרי לפרמטר. **5)** class_weight = ['balanced'] - מתאים למצב של imbalanced data.

היפר פרמטרים שנבחרו: (class_weight= 'balanced', criterion= 'entropy', max_depth= 15, n_estimators= 350, random_state= 0)

חלק רביעי – הערכת המודלים:

Confusion Matrix - ב-confusion matrix שבנינו עבור מודל SVM נמצא מדד accuracy של 0.78:
תא ימני עליון FP - אחוז חיזוי הערך 1, כשבפועל הערך האמיתי היה 0 - 11%.
תא ימני תחתון TP - אחוז חיזויי הערך 1, כשבפועל הערך האמיתי היה 1 – 13%.
תא שמאלי עליון TN - אחוז חיזויי הערך 0 ,כשבפועל הערך האמיתי היה 0 – 74%.
תא שמאלי תחתון FN - אחוז חיזויי הערך 0 ,כשבפועל הערך האמיתי היה 1 – 2.7%.

K-fold Cross Validation- ראשית, באמצעות pipeline מסודר, עיבדנו מחדש בתהליך ה-preprocessing את סט ה-train המלא אשר כולל את הדאטה המלא והמאוחד (שפוצל ל-train ו-validation בחלק בקודם). בחנו כל מודל עם ההיפר פרמטרים שנבחרו תוך שימוש בשיטת ה-AUC ובפונקציית K-fold. חילקנו את הדאטה ל-10 חלקים (K=10). בכל איטרציה נתאמן על 9 חלקים והסט העשירי ישמש לצורך ולידציה.
בדיקת overfitting- על מנת להחליט האם קיים overfitting בדקנו עבור כל מודל את ההבדל בין ציון ה-AUC של ה-train לזה של סט ה-validation (לפי שחושבו ב-K-fold Cross Validation). עבור כל הבדל שגדול מ-0.1, המודל מוגדר כ-overfitted. כלל המודלים אינם יצאו overfitted.
עם ציון AUC על סט ה-validation של 0.918 החלטנו לבחור במודל SVM לביצוע הפרדיקציה.

חלק חמישי – ביצוע פרדיקציה:

המודל בעל התוצאות הטובות ביותר הינו SVM עם ההיפר פרמטרים: (C= 15, class_weight= 'balanced', gamma= 0.01, kernel= 'rbf', probability= True), נירמול לפי RobustScaler והקטנת ממד בשיטת PCA. באמצעות pipeline מסודר, ביצענו את כלל תהליך ה-preprocessing על סט ה-train המלא ועל סט ה-test. אימנו את מודל ה-SVM הנבחר על כל סט ה-Train ולבסוף ביצענו פרדיקציה על סט ה-Test ושמרנו את תחזיות הסתברות הקלסיפיקציה לקבלת הערך 1 לקובץ csv.

סיכום:

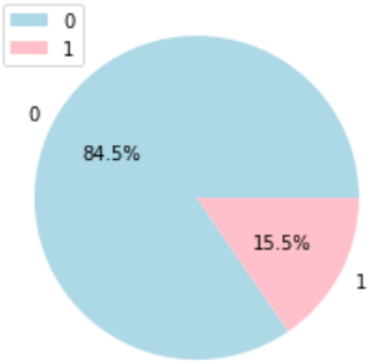
פרויקט זה עסק בבעיית קלסיפיקציה בינארית לביצוע רכישה אינטרנטית. בפרויקט זה ביצענו שימוש בכלים שונים והבאנו לידי ביטוי מעשי את החומר הנלמד בקורס. המטרה העיקרית של הפרויקט הייתה למקסם את מדד ה-AUC באמצעות בחינת מודלים שונים, שיטות נרמול, הורדת מימדים. השלבים העיקריים על פיהם פעלנו במהלך הפרוייקט:
1. Data exploration- בחינה ראשונית של סט הנתונים והסקת מסקנות לצורך עיבוד מקדים.
2. Preprocessing- בשלב זה ביצענו טרנספורמציה לדאטה, מילאנו ערכים חסרים, הוספנו פיצ'רים חדשים והסרנו פיצ'רים קורלטיביים, הורדנו חריגים, נרמלנו את הנתונים, הורדנו את ממד הבעיה ועוד.
3. Modeling- בנינו 4 מודלים ומצאנו את הפרמטרים המיטביים עבורם.
4. Evaluation- בוצע באמצעות K-fold עם הערך k=10 ובעזרת מדד AUC.

לאחר בחינת כלל המודלים בשילובים השונים, המודל שהביא אותנו לתוצאת ה-AUC הגבוהה ביותר עבור סט הנתונים הינו מודל SVM עם מדד AUC של 0.918.

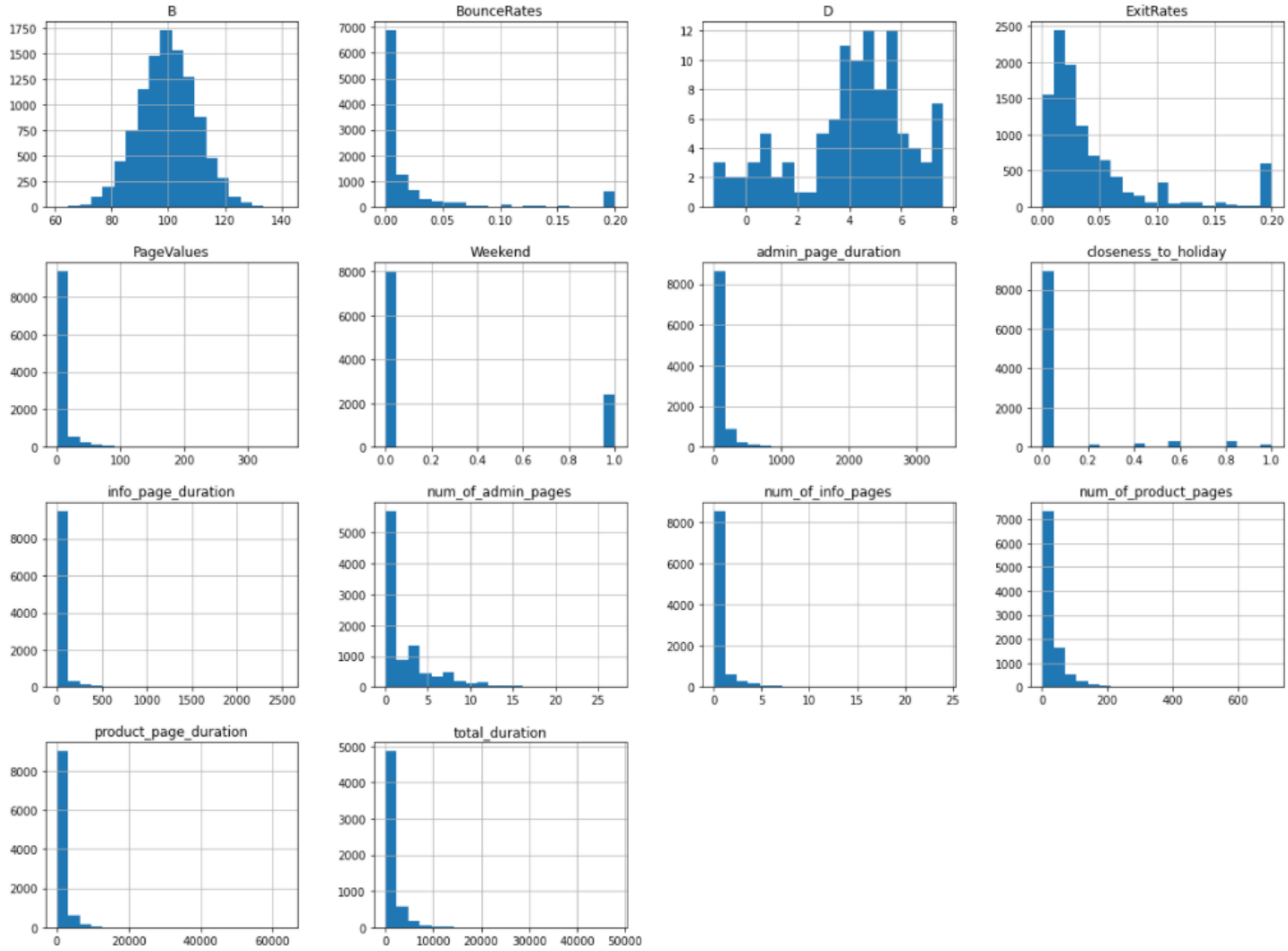
נספחים

נספח א' – Data Exploration

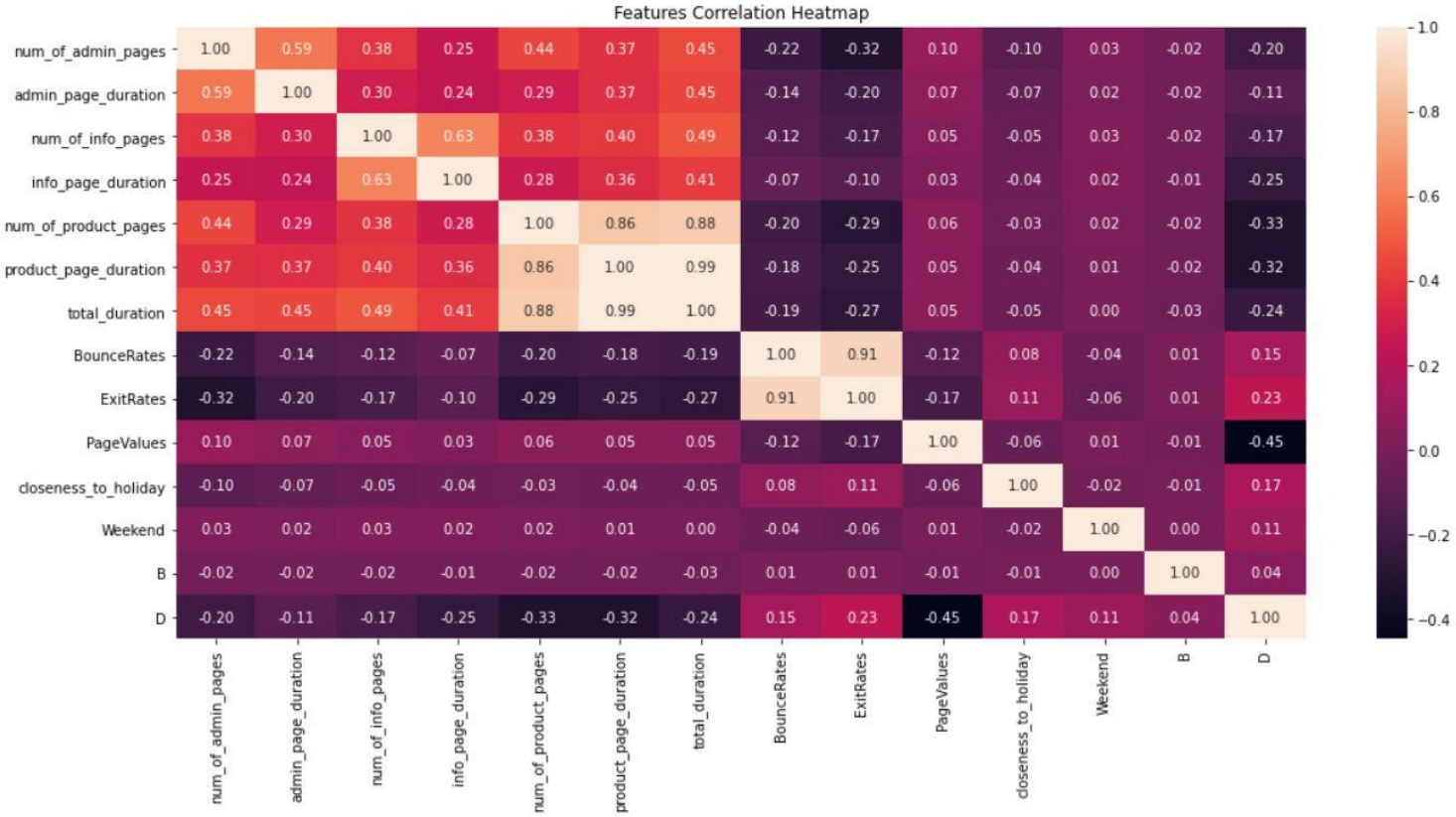
התפלגות הלייבלים:



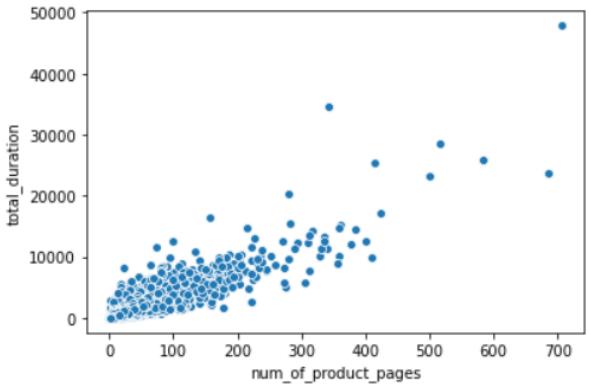
התפלגות משתנים נומריים:

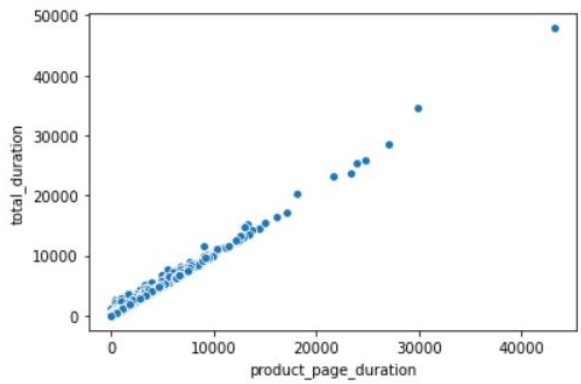


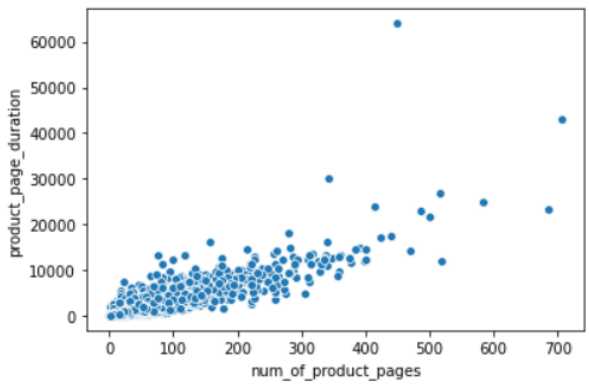
Heatmap - מטריצת קורלציה:

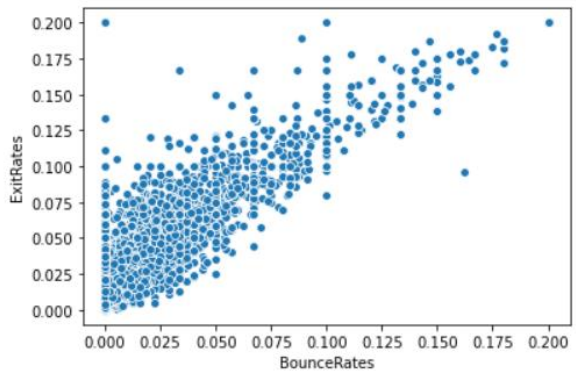


שרטוט קורלציות של משתנים קורלטיביים:

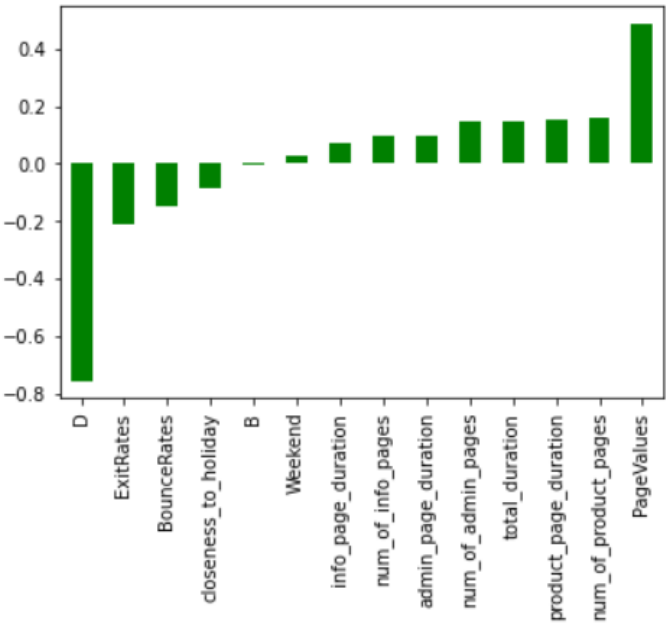








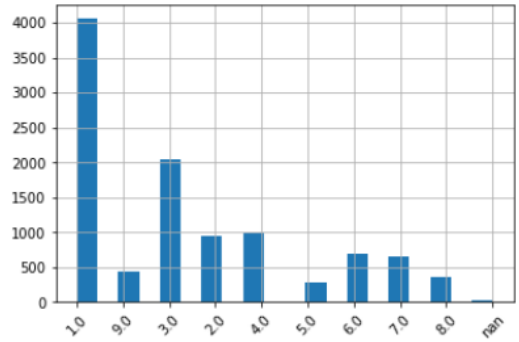
קורלציות של המשתנים הנומריים עם הלייבל:



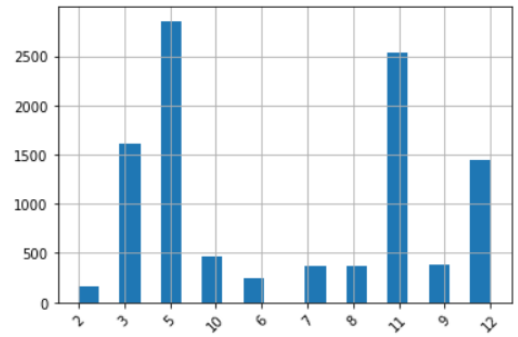
	Corr_with_label
PageValues	0.486404
num_of_product_pages	0.157167
product_page_duration	0.152133
total_duration	0.145429
num_of_admin_pages	0.145048
admin_page_duration	0.097504
num_of_info_pages	0.095563
info_page_duration	0.070309
Weekend	0.028725
B	-0.003981
closeness_to_holiday	-0.083926
BounceRates	-0.150683
ExitRates	-0.207804
D	-0.753238

משתנים קטגוריאלים:

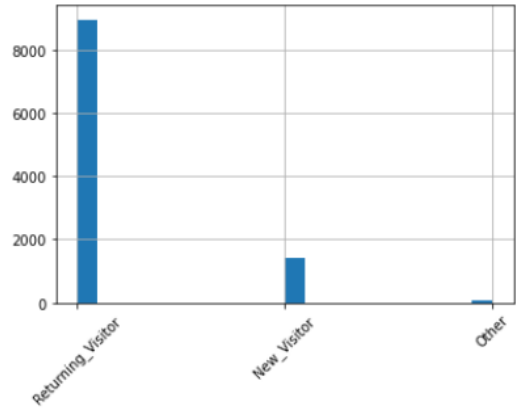
Region:



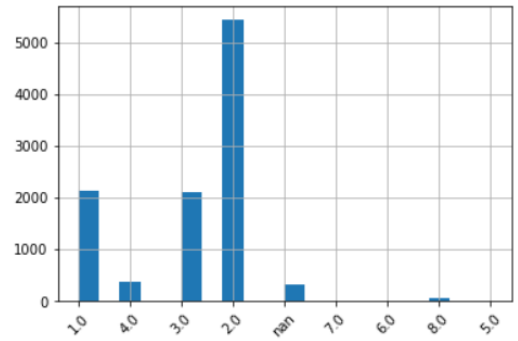
Month:



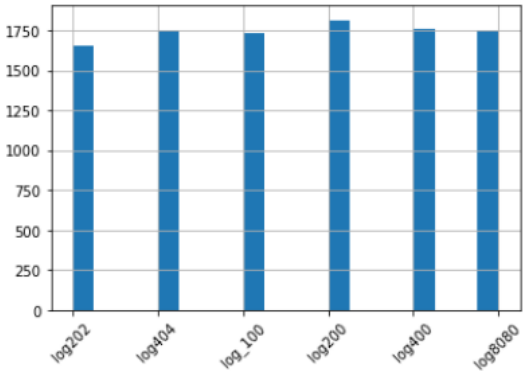
user_type:

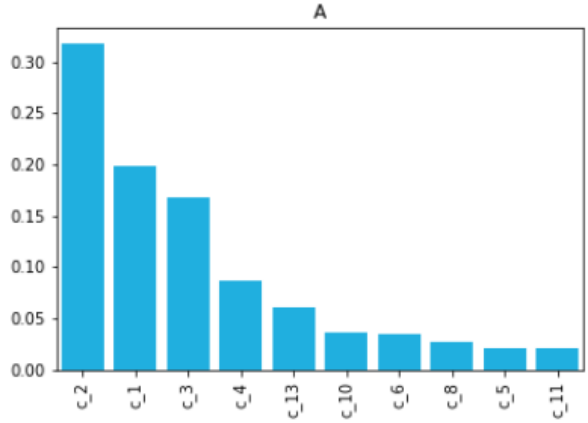


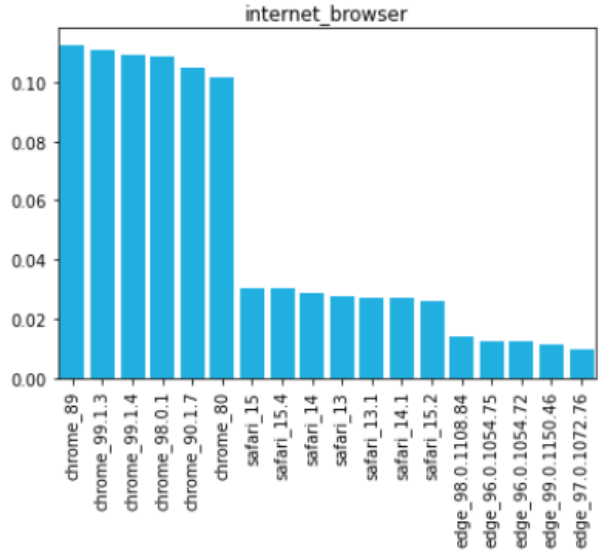
device:



C:

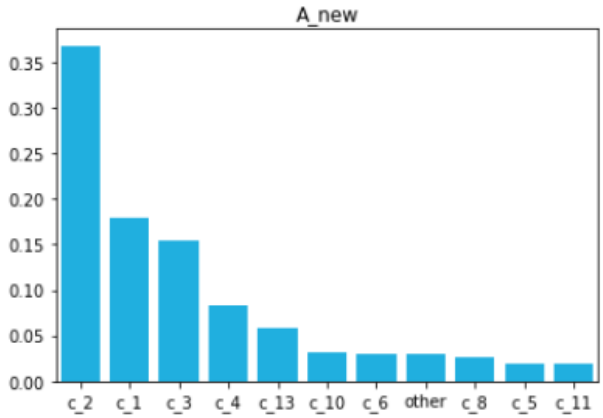


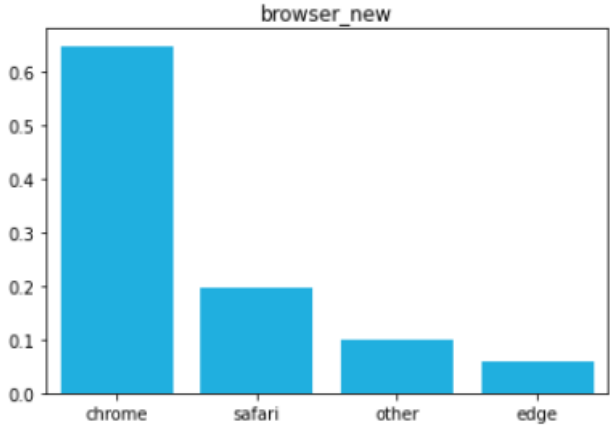




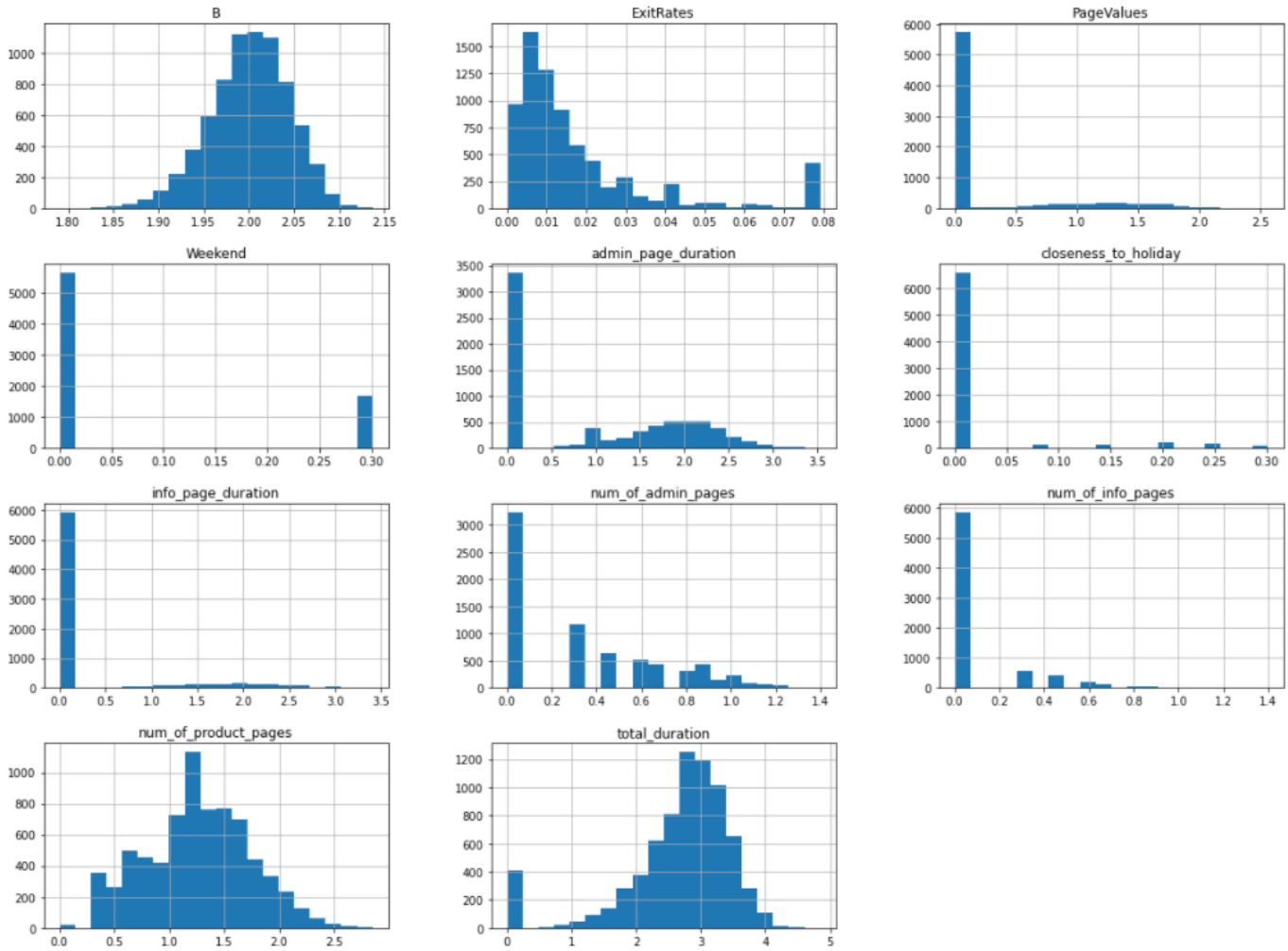
נספח ב' – Preprocessing

יצירת פיצ'רים חדשים:

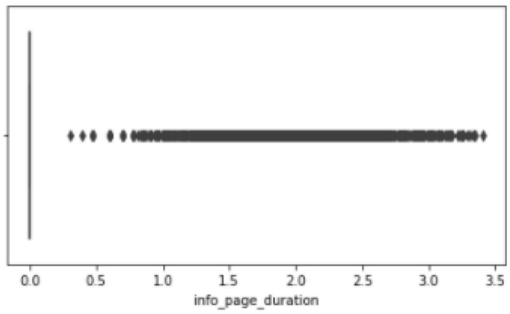


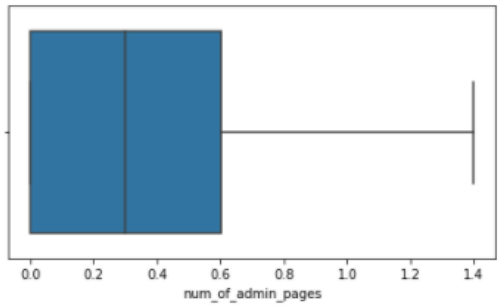


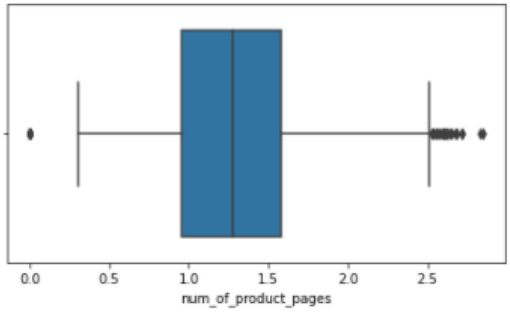
פיצ'רים לאחר טרנספורמציית Log:

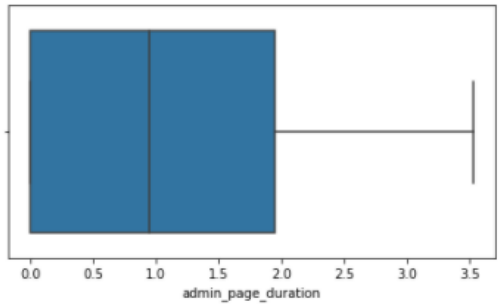


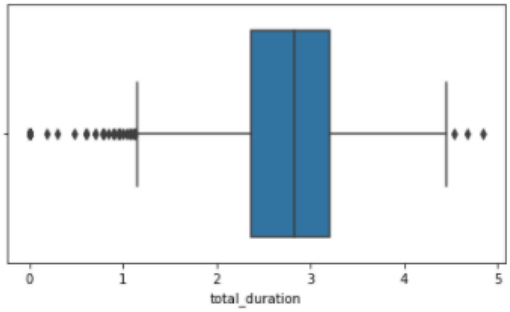
:BOXPLOTS

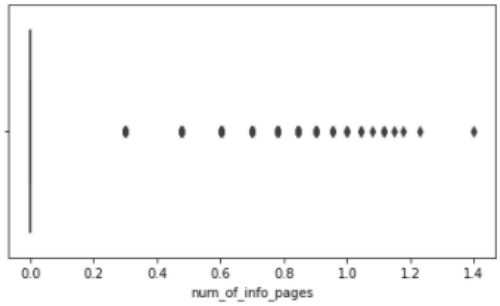

0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
info_page_duration

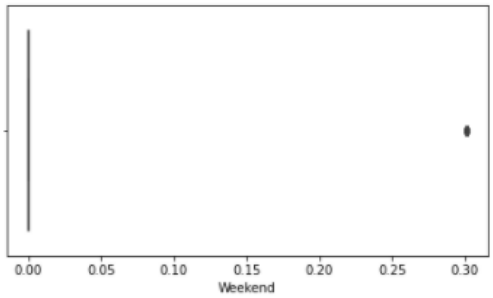

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
num_of_admin_pages

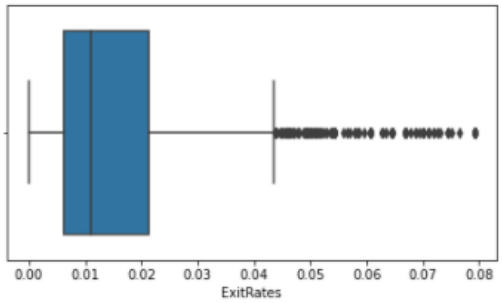

0.0
0.5
1.0
1.5
2.0
2.5
num_of_product_pages

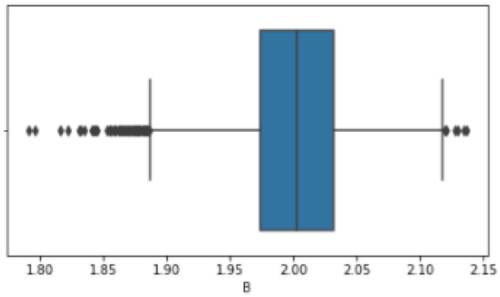

0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
admin_page_duration

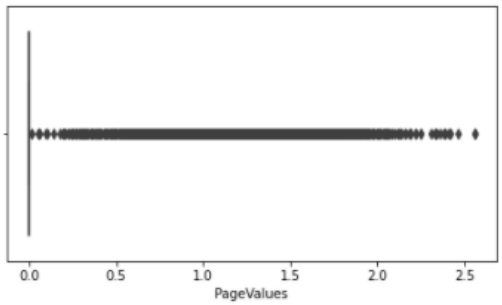

0
1
2
3
4
5
total_duration

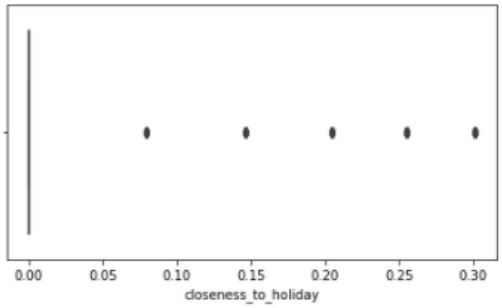

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
num_of_info_pages


0.00
0.05
0.10
0.15
0.20
0.25
0.30
Weekend


0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
ExitRates

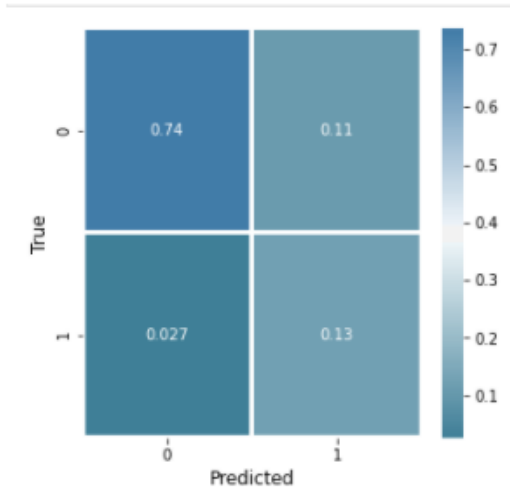

1.80
1.85
1.90
1.95
2.00
2.05
2.10
2.15
B


0.0
0.5
1.0
1.5
2.0
2.5
PageValues


0.00
0.05
0.10
0.15
0.20
0.25
0.30
closeness_to_holiday

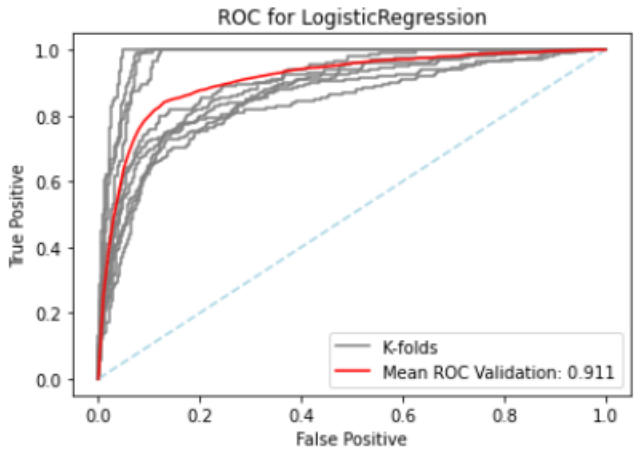
נספח ג' – Model Evaluation

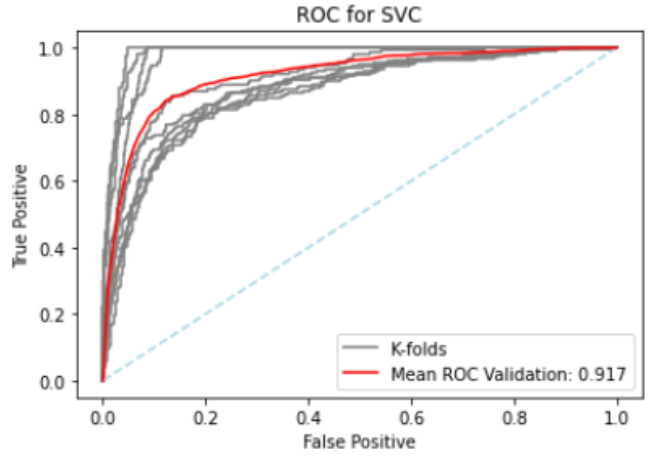
Confusion Matrix:




```
weighted accuracy: 0.7787356321839081
```

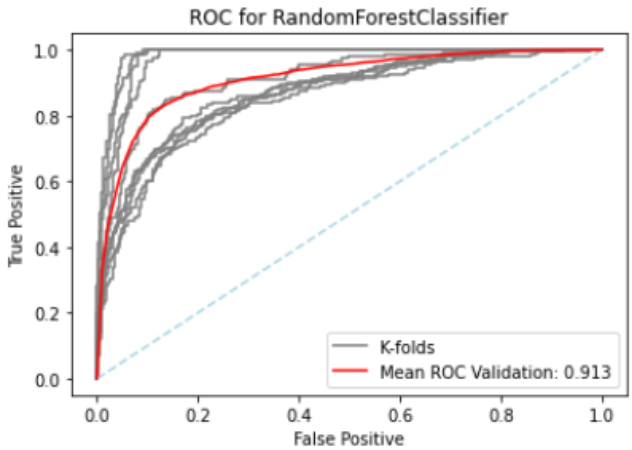
ROC למודלים:



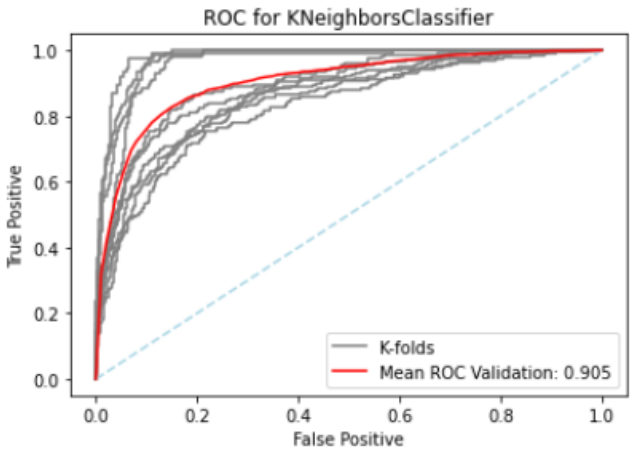



```
Validation AUC: 0.9110801378855078
Train AUC: 0.9212264109324741
Difference: 0.010146273046966359
The model is not overfitting, Great!
```

```
Validation AUC: 0.9171059309081614
Train AUC: 0.9372362439535806
Difference: 0.020130313045419235
The model is not overfitting, Great!
```




```
Validation AUC: 0.9128527384971825
Train AUC: 0.995
Difference: 0.08214726150281748
The model is not overfitting, Great!
```




```
Validation AUC: 0.9052579432661244
Train AUC: 0.995
Difference: 0.08974205673387559
The model is not overfitting, Great!
```

<u>נספח ד' – הסבר על אחריות כל שותפה לפרוייקט</u>

את כלל הפרוייקט ביצענו ביחד. לא עבדנו עליו בנפרד, וכלל הפגישות היו משותפות וארוכות. התהליך כלל ניסיון וטעייה, מחקר רב, לבטים וקבלת החלטות מעניינות.
בשלב ה-Preprocessing, רוני שמה דגש על מחקר במערכי השיעורים והתרגולים וברחבי האינטרנט ויעל שמה דגש על כתיבת ועדכון הקוד, אך גם בשלב הזה העבודה הייתה משולבת ומשותפת.

בשלב הסופי של הפרוייקט, יעל ביצעה את הטיובים האחרונים בקוד ורוני התמקדה בהכנת מסמך ה-PDF הראשוני, ובסיום, עברנו על המסמך ועל הקוד ביחד וביצענו את השינויים הנדרשים.