מבוא ללמידת מכונה
פרויקט מסכם

מרצה : ד״ר עמיחי פינסקי

מגישים : קבוצה 21

זהר פויר 208930297

גיא ליפשיץ 313328981

רותם לוי 316381862

תוכן עניינים

1. <u>תקציר מנהלים</u>

פרויקט זה עוסק בבעיית קלסיפיקציה בינארית. מטרתו לבנות מודל סיווג מדויק ככל הניתן. הפרויקט מחולק לשני חלקים עיקריים.

בחלקו הראשון מתבצעת אקספלורציה ועיבוד של סט הנתונים. חלק זה כולל חקירת הנתונים ההתחלתיים והתאמתם למודל סיווג על ידי שינוי טיפוסים, סיווג משתנה רציף או קטגוריאלי, השלמת חסרים, טיפול בחריגים, הורדת מימד ונרמול.

בחלקו השני של הפרויקט מתבצעת בניית מודל סיווג, אימון והערכת הביצוע.

לאורך הפרויקט נעזרנו בויזואליזציות ובמקורות מידע חיצוניים. הנ"ל מופיעים תחת נספחים וביבליוגרפיה.

מסמך זה מציג בפירוט את כלל שלבי הפרויקט.

2. <u>חלק ראשון - אקספלורציה</u>

בשלב הראשון לפרויקט חקרנו את סט הנתונים על מנת להכיר את הפקטורים הפוטנציאלים אשר ישמשו אותנו במודל. להלן הממצאים (ויזואליזציות **בנספח 1**):

2.1. גודל סט הנתונים - 10,479 רשומות.
2.2. מימד סט הנתונים - 18 עמודות בעלות תוכן ידוע מראש + 4 עמודות עם תוכן לא ידוע.
2.3. טיפוסי הדאטה – int,float,object.
2.4. מספר רשומות בעלות ערך null בכל עמודה.
2.5. התפלגות הפקטורים.
2.6. איזון משתנה המטרה purchase.

לאחר בחינת ההתפלגויות של המשתנים עלו מספר נקודות לבחינה בהמשך. עבור משתנים נומריים, זיהינו מספר פקטורים בעלי מספר סופי של קטגוריות, ומכאן מועמדים למעבר לקטגוריאלי. המשתנים הם: Weekend, Region, device, closness_to_holiday.

משתנים מטיפוס Object נדרשים לשנות טיפוס לint/float ולקבל סיווג כנומריים (רציפים) או קטגוריאלים. לאחר בחינת ההיסטוגרמות החלטנו לשקול את המשתנים user_type, C, Month כקטגוריאלים ואת המשתנים internet_broswer, A, info_page_duration, product_page_duration כמשתנים נומריים.

3. <u>חלק שני - עיבוד מקדים</u>

3.1. טיפול בנתונים חסרים

נתונים חסרים משבשים את תהליך הסיווג, שכן מקשים על למידת הפקטורים והבנת השפעתם על הסיווג. הגישות לטיפול בחסרים הן (**ביבליוגרפיה 1**):

3.1.1. מחיקת רשומות בעלות מספר רב של נתונים חסרים.
3.1.2. מחיקת פקטורים (עמודות) בעלי מספר רב של נתונים חסרים.
3.1.3. השלמת חסרים על ידי הערך השכיח ביותר, ממוצע או חציון.

בפרויקט ביצענו את שלושת האופציות הנ"ל לאורך שלב העיבוד המקדים ובהתאמה לפקטור.

ראשית, בדקנו אם ניתן למחוק רשומות / עמודות שלמות. מצאנו כי המספר המקסימלי של פקטורים חסרים ברשומה הינו 11, המותיר 12 פקטורים הניתנים לשימוש. החלטנו שזה מספר גבוה דיו ולכן לא מחקנו אף רשומה. לעומת זאת, בבחינת ערכים חסרים בפקטורים מצאנו כי פקטור D מכיל 10,370 ערכי null (כמעט 100% מהרשומות) ולכן הסרנו אותו מסט הנתונים.

בהמשך הפרויקט ביצענו השלמת חסרים לפני הצורך. פירוט בהמשך.

3.2. מימד סט הנתונים

מימדיות הבעיה נקבעת על ידי מספר הפקטורים במודל הסיווג. מימדיות גדולה מדי עלולה לגרור מספר בעיות:

3.2.1. שגיאת מודל גדולה (למשל בעקבות overfitting).
3.2.2. זמן ריצה איטי ועלויות חישוב גבוהות.
3.2.3. קושי בויזואליזציה.

טיפול במימדיות גדולה נעשה על ידי הורדת פקטורים. הדרך הראשונה בה ניסינו להסיר פקטורים היא על ידי מציאת קורלציה גבוהה בין פקטורים והסרה של אחד מהם.

בנספח 2 ניתן לראות את מטריצות קורלציה בין משתנים נומריים ובין משתנים מטיפוס Object. עבור פקטורים עם קורלציה גבוה, הסרנו מסט הנתונים את הפקטור בעל המספר הגבוה יותר של נתונים חסרים. התוצאות:

3.2.4. ExitRates, BounceRates (קורלציה 0.91) - הסרנו את ExitRates.
3.2.5. num_of_product_page, total_duration (קורלציה 0.87) - הסרנו את total_durations.
3.2.6. product_page_duration , Month (קורלציה 0.85) - הסרנו את num_of_product_page.

נוסף על בדיקת קורלציה, הסרנו 2 פקטורים נוספים:

3.2.7. שמנו לב שפקטור 'B' נמצא בקורלציה אפסית עם כל שאר המשתנים. החלטנו להסיר משתנה זה מעובדה זו ובנוסף לאבחנה כי ההתפלגות של הפקטור נורמלית באופן מדויק. נראה כי הפקטור הינו "רעש" ואינו בעל משמעות גבוהה.
3.2.8. פקטור 'id' חסר משמעות אמיתית ולכן בחרנו להסיר גם אותו.

בשלב זה לא יכולנו לבצע הורדת מימד בשיטת PCA (בגלל טיפוס הדאטה, נתונים חסרים ואי נרמול). נטפל בכך ונבצע PCA בהמשך.

3.3. טיפול במשתנים קטגוריאליים

ככלל, פקטור יכול להיות קטגוריאלי או נומרי.

קטגוריאלי - פקטור בעל מספר סופי של קטגוריות. ניתן להמיר משתנה לקטגוריאלי על ידי הרחבת עמודה למספר עמודות בינאריות בשיטת one hot encoding. לדוגמה: עמודה המתארת "מין" וכוללת קטגוריות ["זכר", "נקבה"] תפוצל ל-2 עמודות נפרדות בשמות הקטגוריות, וכל רשומה תסומן 1/0. רשומה שהיתה בעבר "זכר" בעמודת "מין" תסומן 1 בעמודת "זכר" ו-0 בעמודת "נקבה".

נומרי - פקטור בעל מספר אינסופי של קטגוריות (רציף או מספר גבוהה מאוד של קטגוריות). ניתן להמיר משתנה מטיפוס Object לנומרי על ידי המרת הקטגוריות בעמודה למספרים בעלי ערך. אופציה זו בדרך כלל תתאים עבור נתונים שיש ביניהם היררכיה. לדוגמה: עבור עמודה המתארת "רמת נסיון" הכוללת קטגוריות "מתחיל", "בינוני" ו-"מתקדם", נמיר כל קטגוריה למספר באופן הבא- "מתחיל"=1 , "בינוני"=2 ו-"מתקדם"=3.

עבור פקטורים בדידים (יכולים להיות מסוג object, int, float) נצטרך להחליט האם יהיו קטגוריאליים או נומריים.

ראשית, בנינו את פונקציית one_hot_encoding הממירה פקטור לקטגוריאלי כמתואר לעיל. לאחר מכן בחנו את הפקטורים שזיהינו בשלב האקספלורציה.

3.3.1. פקטורים מטיפוס נומרי שיש לשקול המרתם לקטגוריאלים - מצאנו כי עבור כל הפקטורים שבחנו כי מספר סופי וקטן של קטגוריות. השלמנו ערכים חסרים עם הערך השכיח ביותר והחלנו עליהם את פונקציית one_hot_encoding. הפקטורים הם - Weekend, Region, device, closeness_to_holiday.

3.3.2. פקטורים מטיפוס Object שיש לשקול המרתם לקטגוריאלים - בדומה לסעיף הקודם, הפקטורים הם - user_type, C, Month.

3.4. טיפולים נוספים בסט הנתונים

3.4.1. בחינת שורות כפולות והסרתן – לא מצאנו שורות כפולות בסט הנתונים.

3.4.2. טיפול במשתנים נותרים מטיפוס Object – נותרו 3 משתנים מטיפוס Object שלא נראו מתאימים להיות קטגוריאלים ויש לבחון המרתם לנומריים.

3.4.2.1. משתנה internet_browser - לאחר בחינת הקטגוריות, מצאנו כי אפשר לקבץ קטגוריות דומות (גרסאות שונות לאותם דפדפנים). כך צימצנו ל-4 קטגוריות, השלמנו ערכים חסרים עם הערך השכיח ביותר והפכנו את המשתנה לקטגוריאלי.

3.4.2.2. משתנה info_page_duration - מצאנו כי מדובר בstring המכיל את כמות הדקות ואת המילה 'minutes'. בדקנו שאכן מדובר בדקות עבור כל הרשומות והמרנו לטיפוס float המכיל את כמות הדקות בלבד. כך משתנה זה הפך לנומרי.

3.4.2.3. משתנה A - נמצאו קטגוריות רבות. מכיוון שאין מידע מקדים אודות הפקטור, בחנו את כמות הרשומות בכל קטגוריה בתרשים Bar (**נספח 3**). הנחנו כי ניתן לאחד קבוצות דומות. איחדנו, השלמנו חסרים עם הערך השכיח והמרנו את הפקטור לקטגוריאלי. השיקולים באיחוד: בחרנו לאחד את 'c'', 'c_2', 'c_1_3 מכיוון שניתן היה לראות כי אלו הקטגוריות הגדולות ביותר. לאחר מכן איחדנו את כל הקטגוריות מהצורה 'c_20_x' בעקבות הדמיון בשם ולקבוצת האיחוד האחרונה הכנסנו את כל שאר הקטגוריות.

3.5. טיפול בחריגים (**ביבליוגרפיה 2**)

התחלנו עם ויזואליזציה עבור חריגים אצל משתנים נומריים בעזרת Box-Plot (**נספח 4**). בתרשימים ניתן לראות את ערכי הרבעונים (אחוזון 25%, 50%, 75%) וחריגים. ניתן לזהות חריגים על ידי צורתן כנקודות עגולות מעל ומתחת קצוות הגרף. זיהנו כי לכל המשתנים הנומריים יש ערכים חריגים. הערכים החריגים מפריעים לניתוח הדאטה שכן עלולים להיות מטעים מהיותם שגויים או בעלי סבירות נמוכה מדי מכדי שניקח אותם בחשבון. בחרנו למחוק רשומות בהן יש ערכים חריגים. בחנו שתי אפשרויות לאיתור חריגים ומחיקתם - על בסיס סטיית תקן (ערך הרחוק מהממוצע ב-3 סטיות תקן ומעלה יחשב כחריג) או על בסיס IQR (ערכים קיצוניים ברבעון עליון ורבעון תחתון יחשבו חריגים). באפשרות על בסיס IQR מחצית הרשומות זוהו כמכילות חריגים ואילו באפשרות על בסיס סטיית תקן 14% מהרשומות זוהו ככאלה. בחרנו באופציה השניה שכן הראשונה היתה כללית מדי וביטלה רשומות רבות מדי.

3.6. טיפול בנתונים חסרים שנותרו

השלמנו את שאר הערכים החסרים בשיטת החציון. בחרנו בשיטה זו מכיוון שהיא הכי פחות רגישה לקצוות הדאטה.

3.7. נרמול סט הנתונים

לנרמול הנתונים חשיבות גבוהה ובקשר ישיר למודל הסיווג. אלגוריתמים מבוססי gradient descent יגיעו לאופטימום באיטיות כאשר הנתונים לא מנורמלים. עבור אלגוריתמים המשתמשים במרחק אוקלידי (כמו KNN), אי נרמול הפקטורים עלול לגרור העדפה לפקטורים שלא לצורך (פקטורים בעלי מרחק גדול יותר יקבלו עדיפות על פני אחרים). שיטת PCA להורדת מימד המבוססת על שונות הפקטורים תיתן עדיפות שעלולה להיות שגויה לפקטורים לא מנורמלים, על בסיס טווח השונות שלהם.

בדקנו את נרמול הנתונים על ידי פונקציית describe של פייתון המציגה סטיית תקן וממוצע לכל פקטור. בפקטור מנורמל נצפה לראות ממוצע קרוב ל-0 וסטיית תקן קרובה ל-1. ראינו כי הפקטורים אינם מנורמלים.

בחנו 2 שיטות לנרמול הנתונים (**ביבליוגרפיה 3**):

3.7.1. נורמליזציה - נרמול בשיטת מינימום+מקסימום על פי הנוסחה:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

3.7.2. סטנדרטיזציה - נרמול הנתונים על ידי המרתם וריכוזם סביב הממוצע עם סטיית תקן אחידה.

לאחר ביצוע שתי השיטות, מצאנו תוצאות טובות יותר בשיטת הנורמליזציה ואותה החלנו על סט הנתונים.

3.8. ביצוע PCA להורדת מימד

בהמשך לסעיף 3.2, נוריד מימדיות הבעיה בעזרת שיטת PCA. השיטה נותנת ניקוד לכל פקטור, לפי "כמות" השונות המוסברת על ידו ובסדר חשיבות יורד. הפקטורים שיחד מצליחים להסביר 95% מהשונות נשארים בסט הנתונים ואלו שלא יורדים. סט הנתונים עומד בתנאי דרישה למינימום N^2 דגימות עבור N פקטורים.

4. <u>חלק שלישי - הרצת המודלים</u>

עבור כלל המודלים בחלק זה, יש להריץ grid search על מנת לבצע אופטימיזציה על ערכי הפרמטרים הדיפולטים של המודל (**ביבליוגרפיה 4**). לאחר בחינה של מספר ערכים עבור כל אחד מהפרמטרים שנבחר, פלט הפונקציה ייתן לנו את הערכים האופטימליים לטובת סט הנתונים שלנו. ישנם מספר פרמטרים בכל מודל שאותם בחרנו להותיר בערך הדיפולטי שלהם.

4.1. החלת מודל בסיסי על סט הנתונים

ראשית, יש להחיל על סט הנתונים מודל בסיסי לבחירתנו. ניתנה לנו הבחירה בין המודלים KNN לבין Logistic Regression. בחרנו במודל KNN כיוון שסט הנתונים שלנו גדול ויכול להתאים למודל (ישנם שכנים רבים להשוות איתם). בנוסף, מצאנו יותר עניין בשיטה זו וסיקרן אותנו לראות אותה בפעולה.

על מנת להחיל את המודל על סט הנתונים, ביצענו אופטימיזציה על משתני המודל הדיפולטים. בחנו את: n_neighbors, weights, metric, leaf_size.

למשל, לטובת בחירת n_neighbors, בחנו מספר מערכי מספרים כדי להבין מהו באמת מספר השכנים האופטימלי. ביצענו לולאה על מערך רשימות שמכיל רשימות מספרים בנות 5 מספרים, ההבדל בין המערכים מתבטא בגובה המספרים בכל מערך. כך ביצענו את ה- grid search עד שקיבלנו מספר מקסימלי עבור n_neighbors שלא משתנה גם עבור סט מספרים עם מספרים גבוהים יותר.

4.2. החלת שלושה מודלים מתקדמים על סט הנתונים

בדומה למודל KNN, ביצענו אופטימיזציה על משתני המודל הדיפולטים עבור כל מודל על ידי הצעת אלטרנטיבות לפרמטרים מסוימים. האחרים נשארו בערך הדיפולטיבי.

4.2.1. מודל ANN - הפרמטרים שנבחנו: hidden_layer_size, activation, early_stopping, batch_size, learning_rate_init, max_iter. למשל, את פרמטר max_ הגדלנו ל- 1500 כדי שלא להגביל את ריצת המודל.

4.2.2. מודל Decision Tree - הפרמטרים שנבחנו: splitter, min_impurity_split, criterion, min_impurity_decrease.

4.2.3. מודל Adaptive Boosting - הפרמטרים שנבחנו: learning_rate, n_estimators.

5. <u>חלק רביעי - הערכת המודלים</u>

5.1. Confusion Matrix (**נספח 5**) - בחרנו להציג את המטריצה עבור המודל Adaptive Boosting. תוצאות המטריצה מעידות על טיב המודל. ראשית ניתן לראות כי רוב הדגימות הן מסוג TN, כלומר סווגו נכון. עבור דגימות Positive, ניתן לראות כי יש כפי שתיים יותר דגימות שסווגו TP מאשר FP ולכן גם פה הסיווג נעשה בצורה טובה.

5.2. Cross Validation (**נספח 6**) - חילקנו את סעיף זה לשני החלקים. האחד - פונקציית cross_validation שמשמשת אותנו לביצוע התהליך ולבניית עקומת ROC. באמצעות הפונקציה אנו עוברים לחלק השני- הזנה של כלל המודלים וביצוע חלוקה לחמישה חלקים (k folds) זאת כיוון שמדובר בברירת המחדל של הפונקציה.

עבור כל מודל, הודפס גרף ה- ROC ומדד ה- AUC שעל פיו נמדד טיב המודל. הגרף הודפס על בסיס חישוב ראשוני של מדד ה- AUC הממוצע על פני כל חמשת הפולדים. במהלך בניית הפלט, חישבנו את המדדים tpr, fpr, auc_mean עבור כל מודל ועל בסיס מדדים אלה בנינו את הגרף שמכיל ROC ממוצע עבור כל מודל, על מנת שנוכל להשוות באופן סופי בין כל המודלים.

המודל MLPclassifier קיבל את ערך ה- AUC הגבוה ביותר ועל כן בחרנו בו לביצוע הפרדיקציה של סט ה- test

5.3. Overfitting (**ביבליוגרפיה 5, נספח 7**) – בדקנו התאמת-יתר בעזרת הפרשי auc בין ה- train_set ל- test_set. צריך להיות פער כלשהוא בין מדדי ה- auc. באם מדובר בפער גדול מדי, נדע כי אנחנו נמצאים במצב של overfitting. אם אין פער נדע כי מדובר ב- underfitting. בחרנו שרירותית 0.05 כרף לקביעת התאמת יתר (מחמיר מספיק). מהבדיקה מצאנו שהמודלים Desicion Tree ו- KNN במצב Overfitted. מספר "ענפים" גדול מדי בעץ החלטה ופרמטר k קטן ביחס לסט הנתונים יכולים להיות הגורם להתאמת היתר. מכיוון ששני המודלים לא קיבלו את ציון ה-AUC הגבוה ביותר, לא ביצענו שינויים לשיפורם.

6. <u>חלק חמישי - ביצוע פרדיקציה</u>

שלבי העיבוד המקדים על הנתונים, בוצעו גם על סט נתוני ה- test. כתוצאה משלב הערכת המודלים, בחרנו להשתמש במודל (MLP) ANN. כפי שציינו, מודל זה קיבל את תוצאת המדד AUC הגבוהה ביותר. בעזרתו, ייצרנו קובץ שמכיל את התחזיות עבור סט נתוני ה- test. קובץ זה מצורף לפרויקט.

7. <u>סיכום</u>

בפרויקט, עסקנו בקלסיפיקציה של בעיה בינארית, במטרה לסווג רשומות של סט בוחן לשתי קטגוריות - רכישה / אי רכישה. במהלך הפרויקט התנסנו בפועל בחומר הקורס, מה שהניב מספר מסקנות.

המסקנה הראשונה והעיקרית - העבודה המשמעותית ביותר הינה ניתוח הנתונים ועיבודם. בשלב זה היה עלינו להשקיע את מירב הזמן והמאמצים, שכן ניתוח ועיבוד נכון של הנתונים הוא זה שיקבע את טיב המודל, היכולת והדיוק שלו לנבא תוצאות ובעקבות זאת סיוע בקבלת ההחלטות שלנו במהלך הפרויקט.

בהמשך, בחנו מספר גדול יחסית של מודלים על מנת לבצע בחירה מושכלת במודל שנותן את מדד ה- AUC הגבוה ביותר. בשלב של הערכת המודלים הללו, נחשפנו לאיך מדדים שונים שלמדנו מתקשרים למעשה לשוני במידע אותו הם מציגים. בשלב זה גם בחנו אילו מודלים הם "overfitted" וגם שיערנו מדוע.

מחקר זה לימד אותנו לא מעט על המודלים, גילינו כי עבור מודל knn- מספר k נמוך לעומת סט הנתונים יכול להוביל אותנו למצב של overfitting ובעבור Decision Tree נראה כי ככל שמספר "הענפים" או ההסתעפויות גדול יותר כך נגדיל את הסיכון להתאמת יתר של המודל.

לבסוף, החלטנו להשתמש במודל ANN על מנת לבצע את הפרדיקציה על סט test.

ביבליוגרפיה

1. טיפול בערכים ריקים

https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e

2. שיטות טיפול בחריגים

https://medium.com/analytics-vidhya/how-to-remove-outliers-for-machine-learning-24620c4657e8

https://www.analyticsvidhya.com/blog/2021/05/detecting-and-treating-outliers-treating-the-odd-one-out/

3. שיטות נרמול נתונים

https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

4. הסבר על שיטת grid search

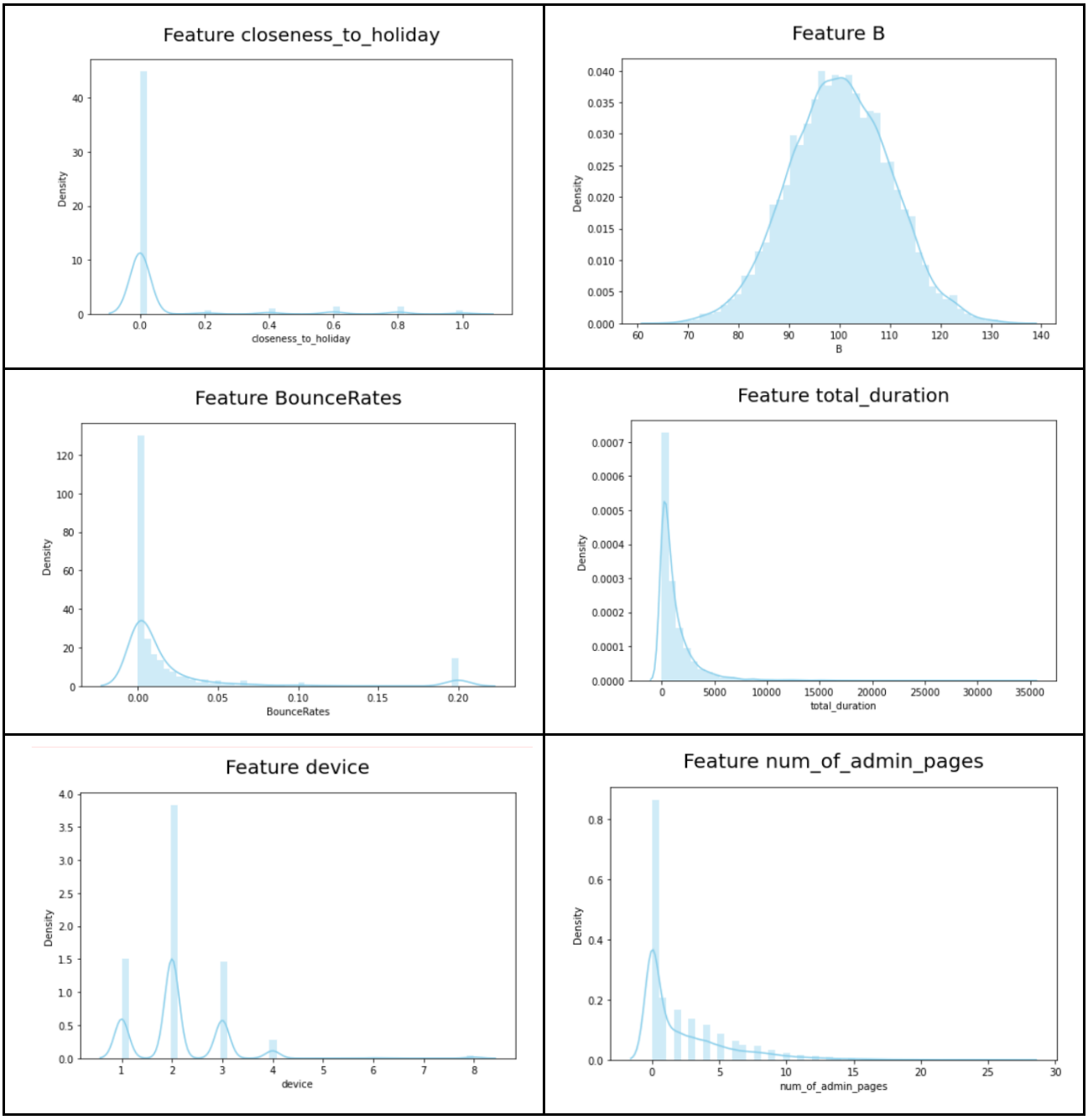
https://elutins.medium.com/grid-searching-in-machine-learning-quick-explanation-and-python-implementation-550552200596

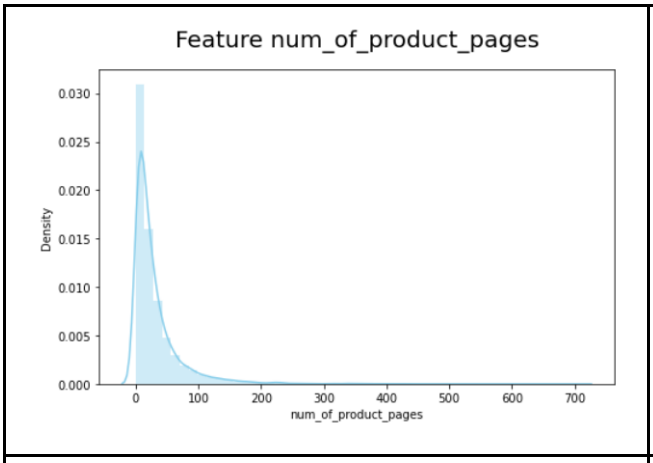
5. הסבר על overfitting באמצעות auc

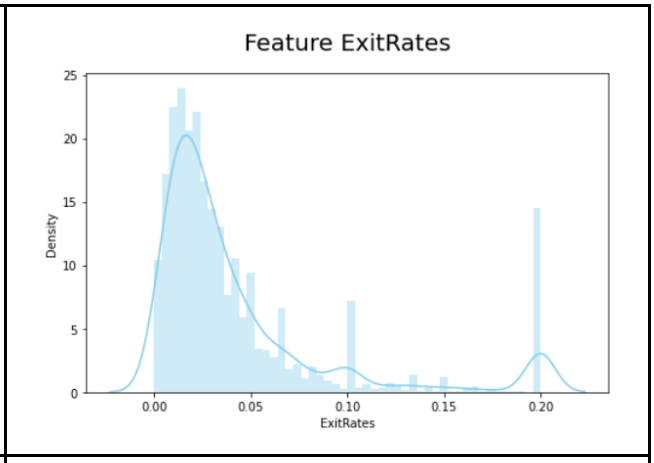
https://stats.stackexchange.com/questions/389865/how-to-distinguish-overfitting-and-underfitting-from-the-roc-auc-curve

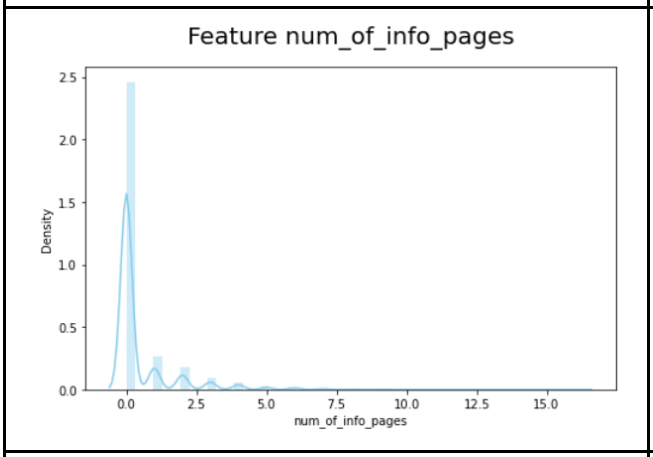
נספח 1 - ויזואליזציה עבור חלק ראשון, אקספלורציה

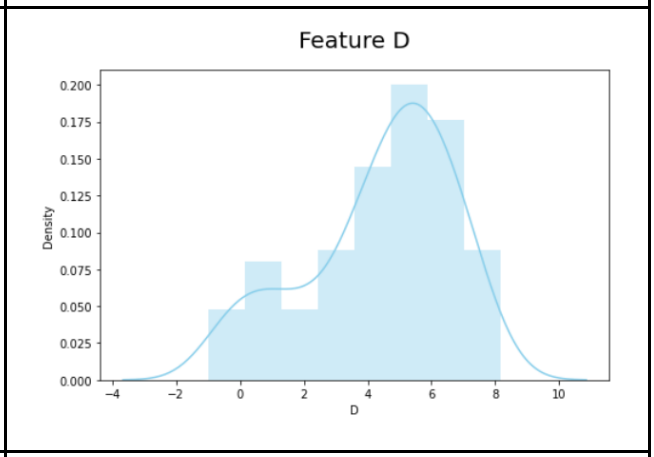
1. התפלגות פקטורים מטיפוסים int/float

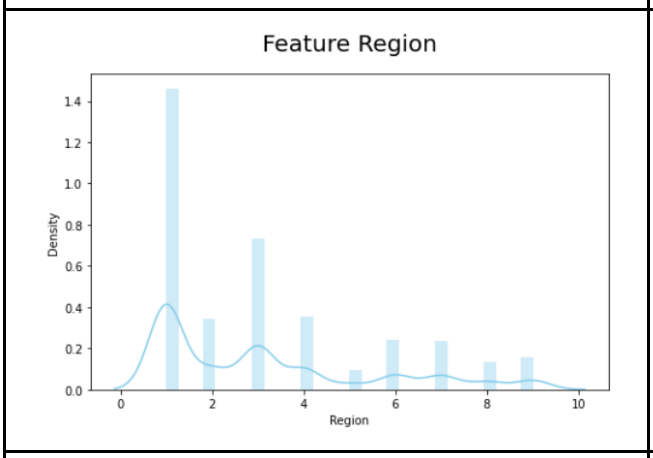


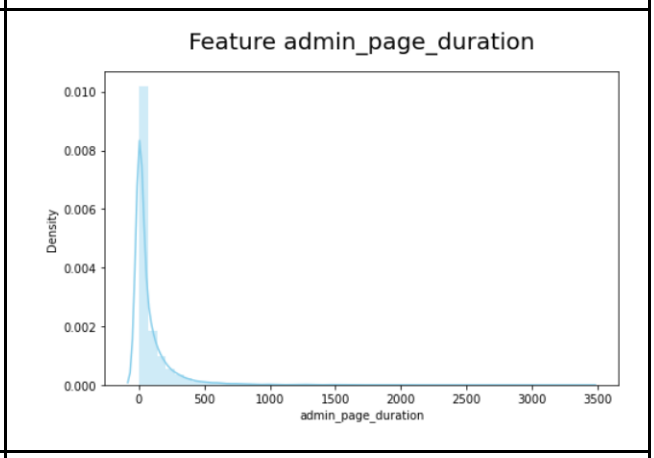

Feature num_of_product_pages
Density
num_of_product_pages

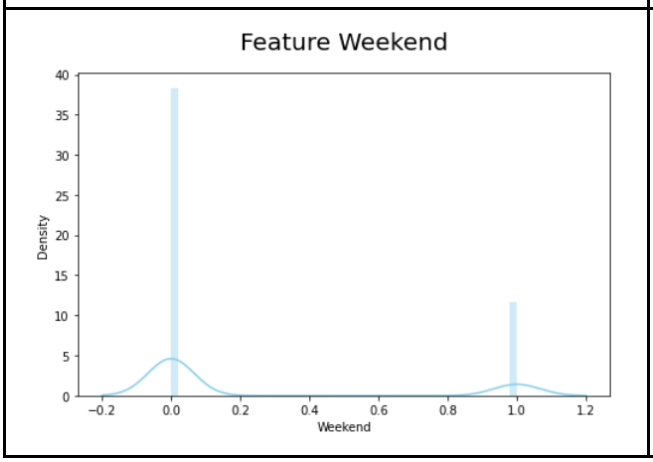

Feature ExitRates
Density
ExitRates

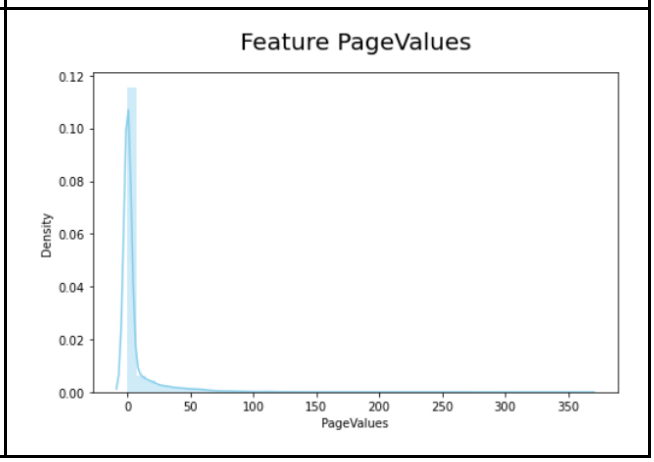

Feature num_of_info_pages
Density
num_of_info_pages


Feature D
Density
D

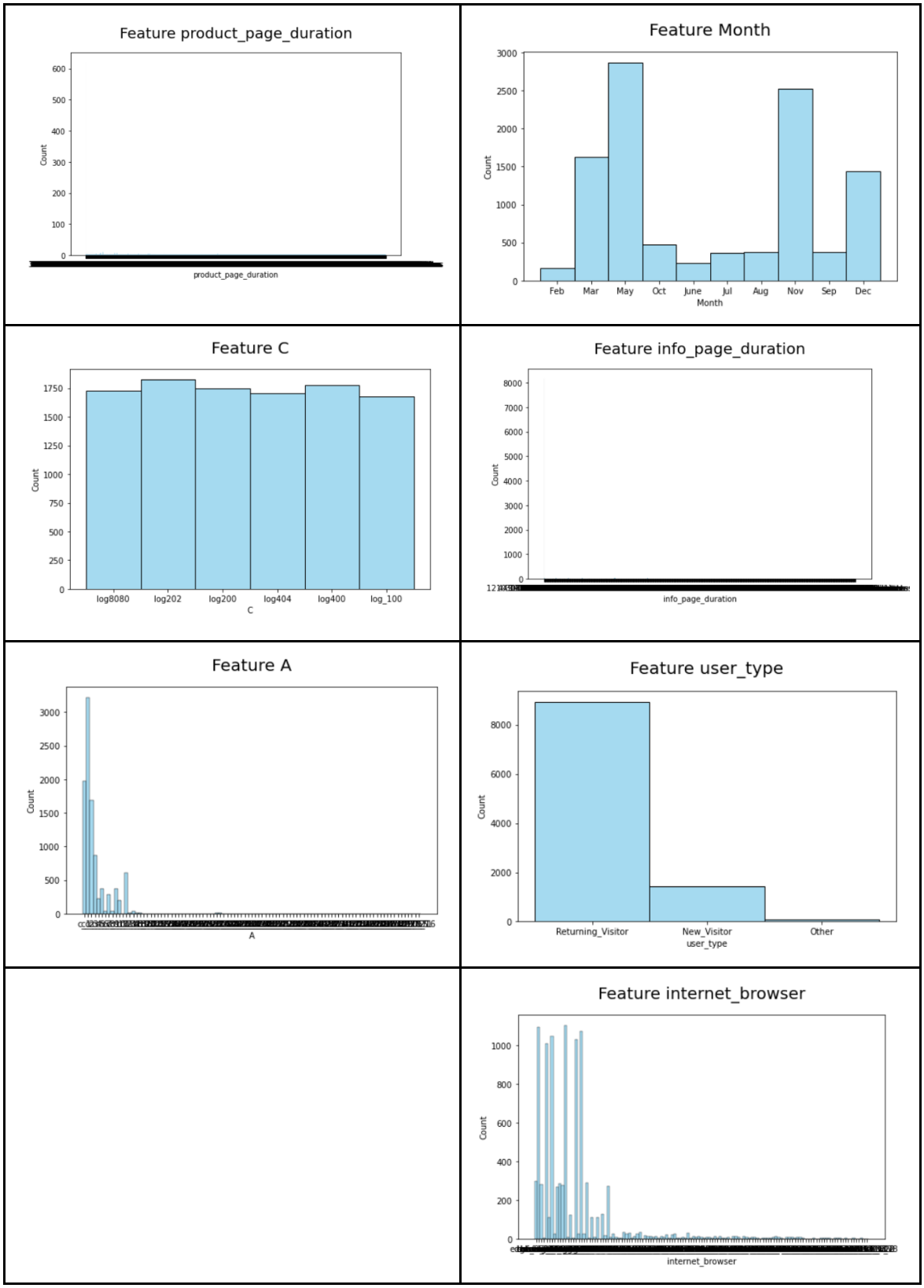

Feature Region
Density
Region


Feature admin_page_duration
Density
admin_page_duration

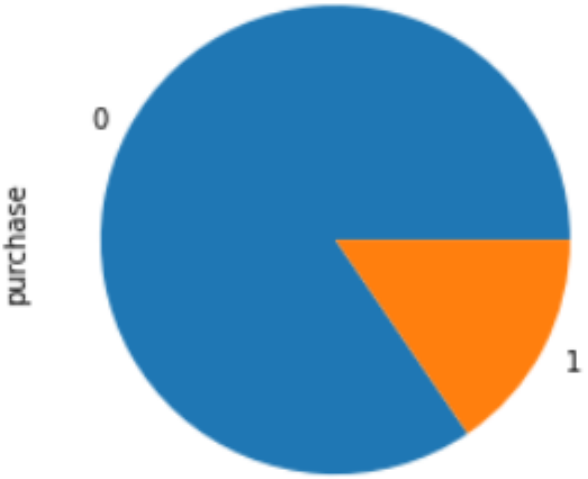

Feature Weekend
Density
Weekend


Feature PageValues
Density
PageValues

2. התפלגות פקטורים מטיפוס Object

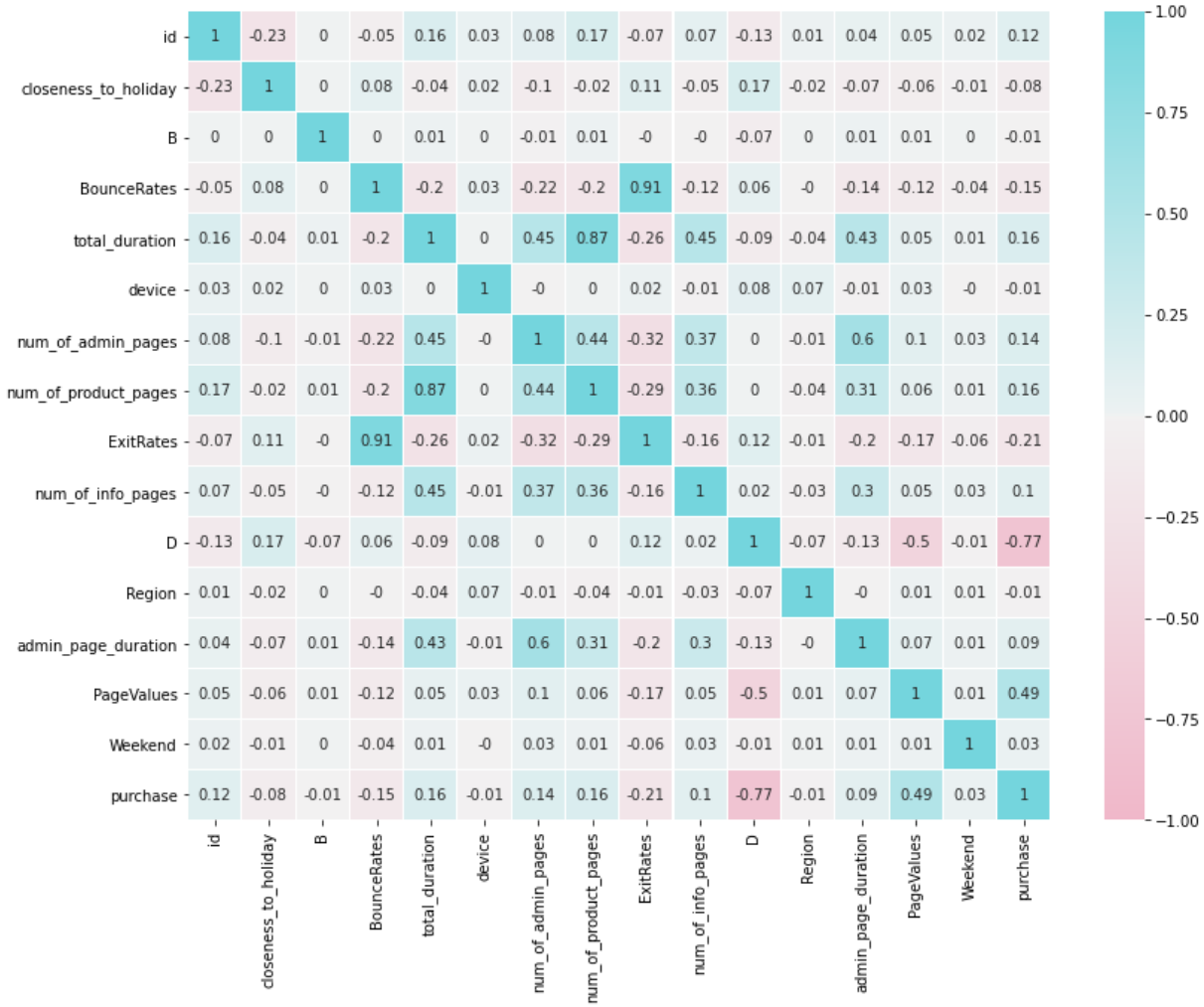


3. איזון משתנה המטרה purchase.


0
1
purchase

נספח 2 - שלב העיבוד המקדים, מטריצות קורלציה בין פקטורים

1. קורלציה בין משתנים נומריים

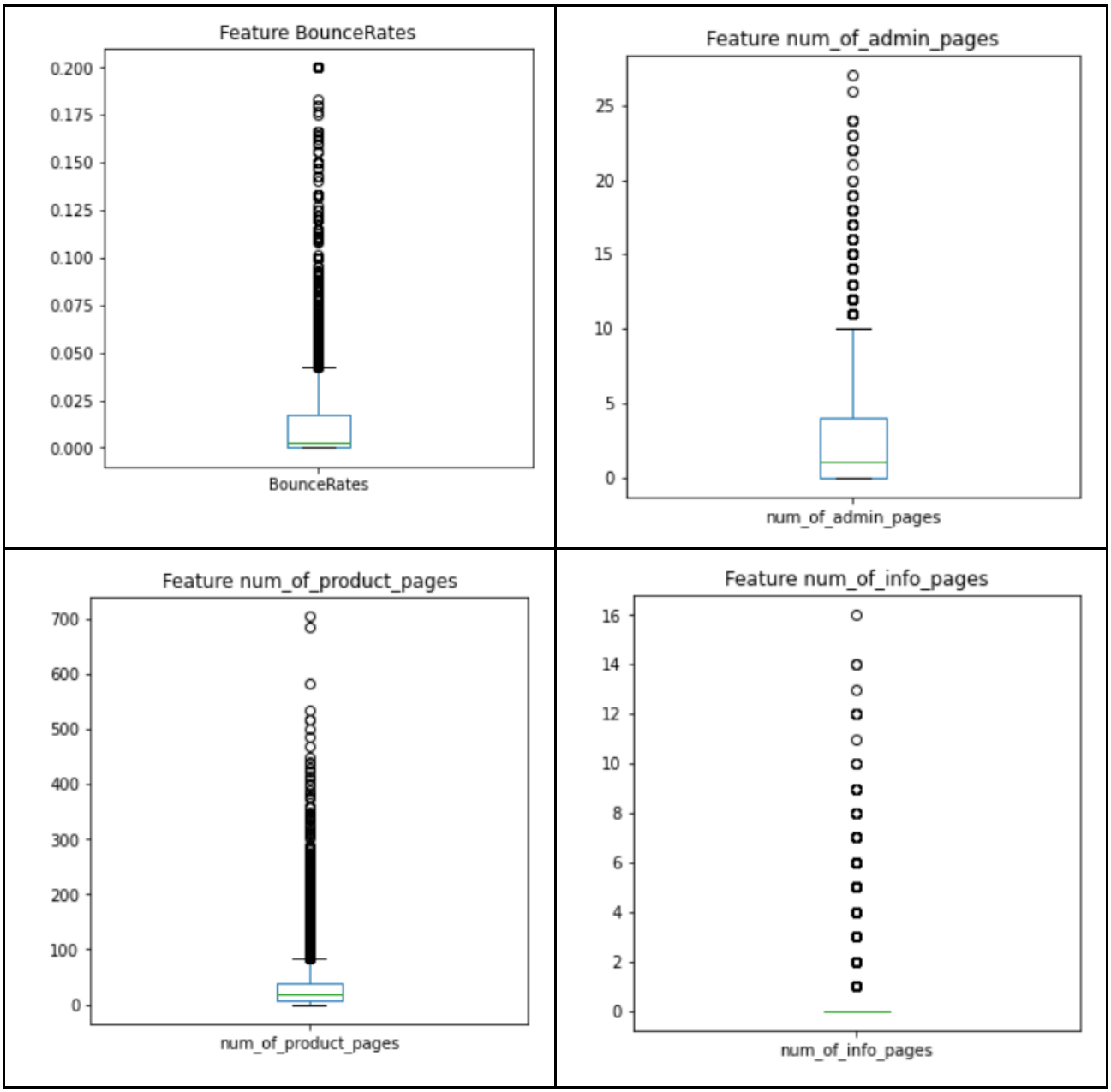


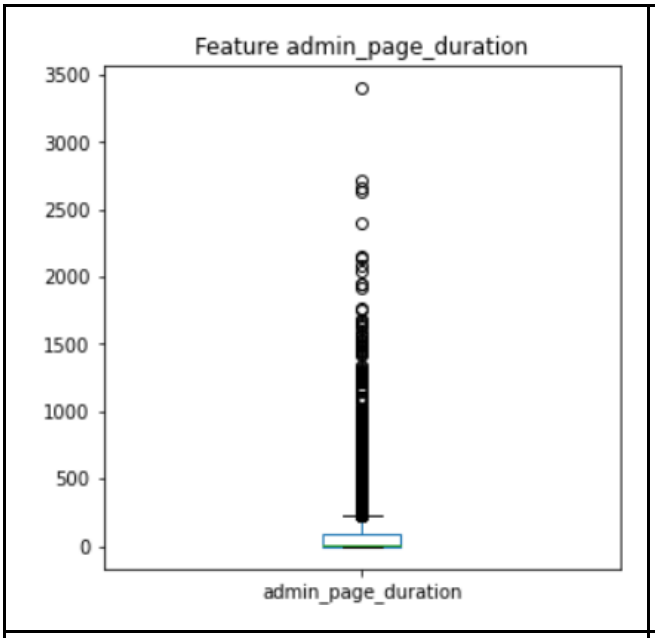
2. קורלציה בין משתנים מטיפוס Object

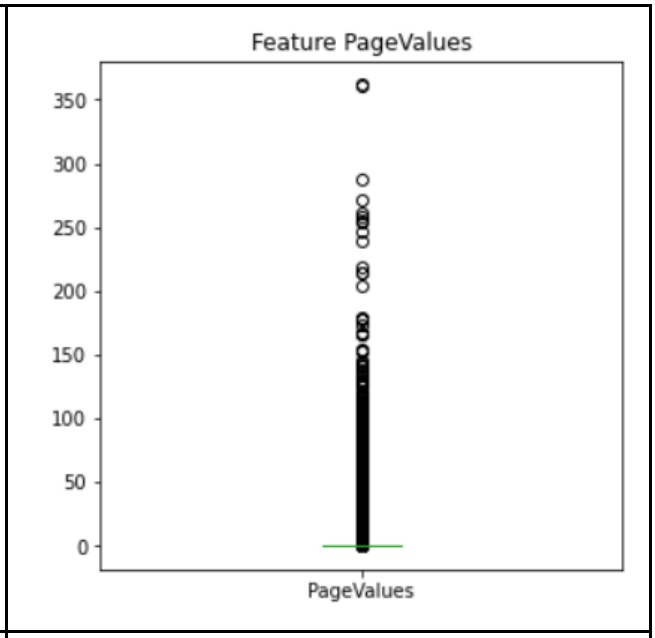

Month
product_page_duration
info_page_duration
C
user_type
A
internet_browser
1 0.85 0.15 -0.36 -0.12 -0.28 -0.31
0.85 1 0.26 -0.31 -0.23 -0.35 -0.38
0.15 0.26 1 -0.23 -0.4 -0.39 -0.38
-0.36 -0.31 -0.23 1 -0.26 -0.22 -0.23
-0.12 -0.23 -0.4 -0.26 1 0.04 0
-0.28 -0.35 -0.39 -0.22 0.04 1 0.17
-0.31 -0.38 -0.38 -0.23 0 0.17 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

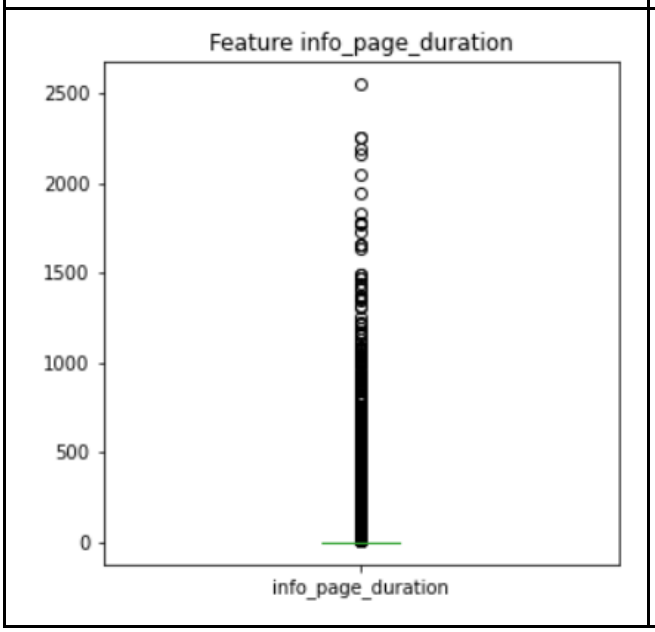
נספח 3 - בחינת פקטור A

נספח 4 - **Box Plot** למציאת חריגים אצל משתנים נומריים

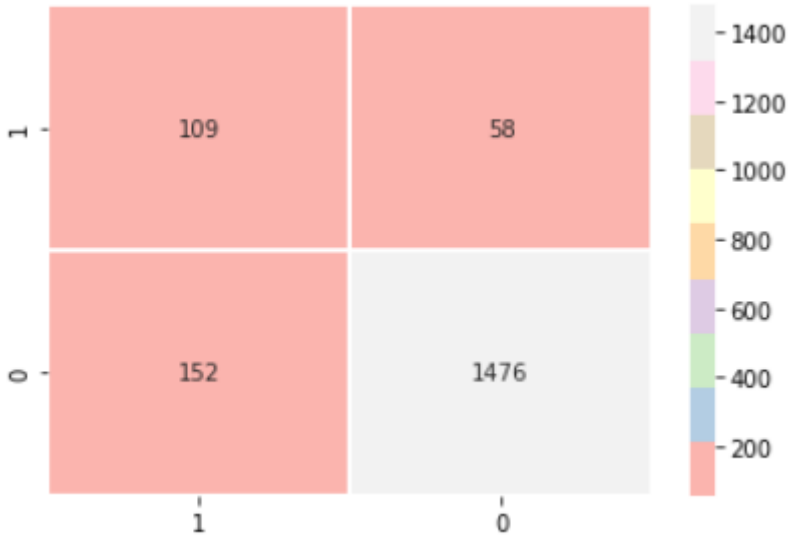



Feature admin_page_duration
3500
3000
2500
2000
1500
1000
500
0
admin_page_duration


Feature PageValues
350
300
250
200
150
100
50
0
PageValues

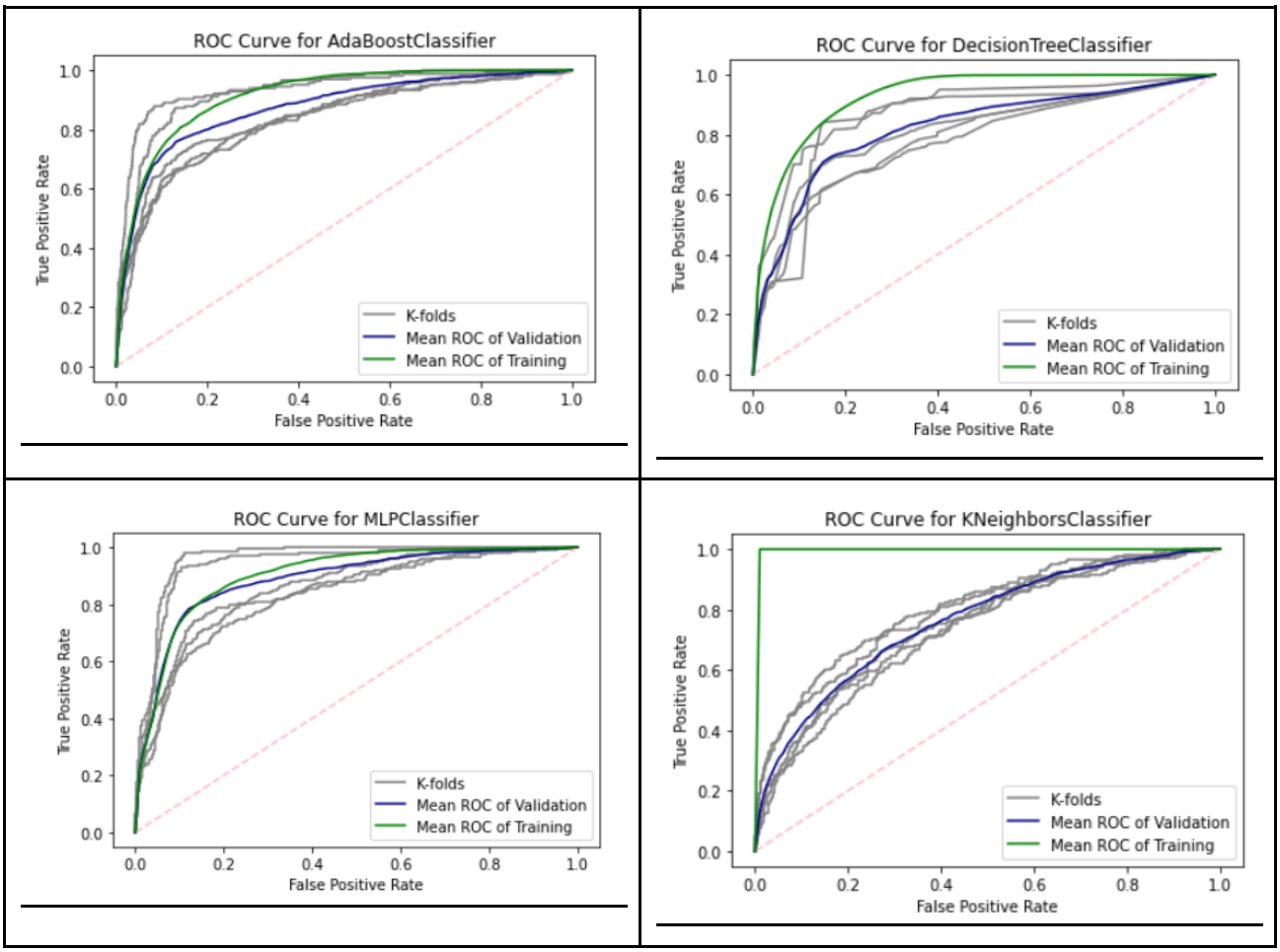

Feature info_page_duration
2500
2000
1500
1000
500
0
info_page_duration

נספח 5 - Confusion Matrix עבור מסווג Adaptive Boosting



נספח 6 - ROC עבור כל המסווגים

```
The best classifier is MLPClassifier(activation='logistic', hidden_layer_sizes=(20, 20, 20),
                max_iter=1500, n_iter_no_change=5) with AUC score of 0.8873679327562802
```



נספח 7 - תוצאת בדיקת overfitting

```
Classifier: AdaBoostClassifier(learning_rate=0.5, n_estimators=75)
AUC score difference between training and validation: 0.041
The model is not overfitted

Classifier: DecisionTreeClassifier(min_impurity_decrease=1e-05, min_samples_leaf=4,
                       min_samples_split=200)
AUC score difference between training and validation: 0.107
The model is overfitted

Classifier: MLPClassifier(activation='logistic', hidden_layer_sizes=(20, 20, 20),
              max_iter=1500, n_iter_no_change=5)
AUC score difference between training and validation: 0.016
The model is not overfitted

Classifier: KNeighborsClassifier(metric='euclidean', n_jobs=-1, n_neighbors=170,
                     weights='distance')
AUC score difference between training and validation: 0.232
The model is overfitted
```