אוניברסיטת תל אביב

הפקולטה להנדסה

מדעים דיגיטליים להייטק

מבוא ללמידת מכונה

מרצה: דור בנק

מתרגל: אילן וסילבסקי

קבוצה 6

מגישים: תמר טל (209011881), מתן גזית (204866925)

תוכן עניינים

תקציר מנהלים

מטרת פרויקט זה היא סיווג בינארי לרכישה\אי רכישה של תצפיות משתמשים שונים. במהלך הפרויקט, ניתחנו את הנתונים על ידי ויזואליזציות שונות, מהן הסקנו מסקנות שונות לגבי הנתונים, לדוגמא התפלגות הנתונים, טווח ערכים, כמות ערכים חסרים ועוד. לאחר ניתוח הנתונים, עיבדנו את הדאטא וחילקנו אותו למסגרות דאטא שונות, וביצענו PCA להורדת ממדים ופישוט המודלים. בפרויקט זה ניתן למצוא מימוש של ארבעה מודלים שונים (GNB, Logistic Regression, Random Forest, SVM). בחנו את טיב המודלים באמצעות ציון AUC והערכות מודלים שונות, ולבסוף בחרנו במודל SVM לביצוע פרדיקציה לקובץ ה test.

פרויקט עזר לנו לממש באופן פרקטי את הידע שצברנו במהלך הקורס, ובזכותו אנו מבינים את חומר הקורס בצורה הרבה יותר טובה ורחבה.

אקספלורציה

ניתוח ראשוני (Exploration: Basic Analysis)

שלבי העבודה:

1. ייבאנו את כל הספריות הנדרשות בשביל לבצע את ניתוח הנתונים
2. בנינו שתי פונקציות שיאפשרו לנו להבין את הנתונים בצורה טובה יותר
   a. Plot_boxen: פונקציה שמטרתה לשרטט את ההתפלגות מחולקת ל purchase/no purchase באמצעות גרף box plot. בחרנו בגרף זה כי הוא מאפשר לנו לראות את ה outliers בצורה נוחה. בנוסף באמצעות הגרף, יכולנו לראות את ההבדל בין רכישה לאי רכישה שהיווה לנו אינדקציה להשפעת הפיצ'ר על עמודת purchase.
   b. Compute_stats: פונקציה שמציגה נתונים בסיסיים לגבי הפיצ'רים: הערך הנפוץ ביותר, החציון, הממוצע, threshold, מספר ערכי ה null, אחוז ערכי ה null, קורלציה עם עמודת purchase. באמצעות פונקציה זו, יכולנו לנתח בקלות את כל אחד מה פיצ'רים.
3. עשינו סקירה כללית של הנתונים וביצענו סדר בסיסי, הסתכלנו על הגודל של הנתונים, הפכנו את כל האותיות לקטנות, ובדקנו כמה ערכי null קיימים.
4. בעקבות הסקירה הכללית שמנו לב כי הדאטא מורכבת מערבוב של נתונים קטגוריאליים ונומרים בעלי הרבה ערכים חסרים. ובשלב הבא נפרט מה עשינו עם כל אחד מהפיצ'רים.

ניתוח מעמיק (Exploration: Individual Feature Analysis):

עבור כל אחד מה פיצ'רים השתמשנו בשתי הפונקציות לעיל, וכעת נפרט מה למדנו על כל פיצ'ר. ניסינו כבר משלב זה להבין כיצד נוכל לטפל ב outliers ובערכי null. לשם טיפול ב outliers השתמשנו בערך 3sigma (ממוצע פלוס שלוש סטיות תקן) ובנוסף לקחנו את מרחב ביטחון, ולכן כל ערך שגדול מ 3sigma פלוס מרחב הביטחון שלנו נחשב כ outlier והוסר. לשם טיפול בערכי ה nulls בחרנו האם להחליף אותם בערך השכיח ביותר, ב – 0 או בחציון.

1. **Num of Admin Pages –** הפיצ'ר מתפלג בצורה אקספוננציאלית (גרף 1) ולכן נצטרך לעשות לו נורמליזציה. כל ערך הגדול מ 15 ייחשב כ outlier ויימחק. Nulls יוחלפו עם 1 (החציון) או עם 0 מאחר שהוא הערך השכיח ביותר.
2. **Admin Page Duration -** הפיצ'ר מתפלג בצורה אקספוננציאלית (גרף 2) ולכן נצטרך לעשות לו נורמליזציה. יש כמות גדולה של outliers במיוחד במקרה של no purchase. כל ערך הגדול מ 1500 ייחשב כ outlier ויימחק. Nulls יוחלפו עם 8 (החציון) או עם 0 מאחר שהוא הערך השכיח ביותר.

3. **Num of Info Pages -** יש כמות גדולה של outliers במיוחד במקרה של no purchase. כל ערך הגדול מ 12 ייחשב כ outlier ויימחק. Nulls יוחלפו עם הממוצע של מספר ה info pages לכל משתמש.
4. **Info Page Duration -** יש כמות גדולה של outliers במיוחד במקרה של no purchase. כל ערך הגדול מ 1500 ייחשב כ outlier ויימחק. אחוז ה nulls קטן באופן יחסי ולכן נחליף אותם עם 0.
5. **Num of Product Pages –** אי אפשר להחליף את ערכי ה nulls עם 0 מכיוון שזה לא הערך השכיח ביותר וגם לא החציון. נחליף את ערכי ה nulls עם הערך השכיח ביותר או עם הממוצע.
6. **Product Page Duration -** יש כמות גדולה של outliers במיוחד במקרה של no purchase. כל ערך הגדול מ 20,000 ייחשב כ outlier ויימחק.
7. **Total Duration -** יש כמות גדולה של outliers במיוחד במקרה של no purchase. כל ערך הגדול מ 20,000 ייחשב כ outlier ויימחק. אחוז ערכי ה nulls גבוה מאוד לכן נחפש פיצ'ר בעל קורלציה גבוהה עם total duration כדי להחליף אותו.
8. **Bounce Rates –** פיצ'ר זה נקבע על ידי חישוב מספר הקפיצות בין עמוד לעמוד. קפיצה מתרחשת כאשר משתמש נכנס לעמוד מסוים באתר, ויוצא ממנו מבלי לבקר בעמוד נוסף או מבלי ללחוץ על שום דבר בעמוד. ההתפלגות נראית שונה עבור purchase ועבור no purchase (גרף 8) וזה רומז לנו שה פיצ'ר הנוכחי יכול להיות בעל חשיבות. החלפה של nulls עם 0 יכולה להיות בעייתית מכיוון שהיא פוגעת בהתפלגות.
9. **Exit Rates –** פיצ'ר זה מראה את מספר המשתמשים שיצאו מהאתר מסך כל המשתמשים. ההתפלגות היא גאוסיאנית עם כמה outliers בעיקר במקרה של purchase ולכן לא נוריד את ה outliers בפיצ'ר זה. ההתפלגות נראית שונה עבור purchase ועבור no purchase (גרף 9) וזה רומז לנו שהפיצ'ר הנוכחי יכול להיות בעל חשיבות. החלפה של nulls עם 0 יכולה להיות בעייתית מכיוון שהיא פוגעת בהתפלגות. **<u>בהשוואה לפיצ'ר הקודם אז כל bounce הוא גם exit אבל לא כל exit הוא bounce, ולכן החלטנו להוריד את הפיצ'ר הנוכחי ולהשתמש רק ב bounce rates.</u>**
10. **Page Values –** פיצ'ר זה מראה את ממוצע הפעמים עבור דף ממנו בוצע מעבר לדף רכישה. ערך זה אמור לתת אינדקציה טובה לאיזה דפים באתר תורמים יותר לביצוע רכישה. ההתפלגות נראית שונה מאד עבור purchase ועבור no purchase (גרף 10) וזה רומז לנו שהפיצ'ר הנוכחי יכול להיות בעל חשיבות ואינדיקציה מעולה להסתברות לקניה. החלפה של nulls עם 0 יכולה להיות בעייתית מכיוון שהיא פוגעת בהתפלגות ומטה את התוצאות לכיוון no purchase.
11. **Closeness to Holiday –** פיצ'ר קטגוריאלי בעל 6 ערכים. אין outliers. 0.0 הוא הערך השכיח ביותר (משמעותו, ביצוע קנייה ביום החג). יש 496 nulls שצריך להחליט מה לעשות איתם.
12. **Month –** פיצ'ר קטגוריאלי. צריך לעשות לו encoding. ינואר ואפריל חסרים.
13. **Device –** יש 8 מכשירים שונים. מכשיר 2.0 הוא המכשיר השכיח ביותר. אפשר להחליף nulls עם מכשיר 2.0 (השכיח ביותר). למרות שזה משתנה מספרי יש להתייחס אליו כמשתנה קטגוריאלי ולבצע encoding.
14. **Internet Browser –** פיצ'ר קטגוריאלי. יש 126 דפדפנים שונים. נבצע encoding ונפצל את הדפדפנים לפי chrome/not chrome .
15. **Region –** יש 9 איזורים שונים. איזור 1.0 הוא השכיח ביותר. אפשר להחליף את ה nulls עם איזור 1.0 (השכיח ביותר). למרות שזה משתנה מספרי יש להתייחס אליו כמשתנה קטגוריאלי ולבצע encoding.
16. **User Type -** פיצ'ר קטגוריאלי. Returning_visitor הוא הערך השכיח ביותר. נבצע encoding ונפצל את המשתמשים לפי Returning_visitor/New_visitor. Nulls יוחלפו ב other.
17. **Weekend -** פיצ'ר קטגוריאלי. נבצע encoding ויוחלף עם עמודה של 1 עבור True ו 0 עבור False.
18. **A -** פיצ'ר קטגוריאלי. יש לו 96 ערכים. נחליף את nulls לערך השכיח ביותר. נבצע encoding.
19. **B –** משתנה זה מתפלג נורמלית והממוצע שלו נע סביב הערך מאה. כאשר בוחנים את ההבדלים בין התפלגות מקרי הרכישה ומקרי אי הרכישה נראה כי ההתפלגות כמעט זהה (גרף 17), מכך ניתן להסיק שחשיבות פיצ'ר זה היא נמוכה כי אינה מעידה רבות על ההבדלים בין שני המקרים.

20. **C** – זהו משתנה קטגוריאלי בעל שישה ערכים אשר משמעותו אינו ברורה לנו (גרף 18). נרצה להבין בשלב מאוחר יותר האם תורם למודל.
21. **D** –פיצ'ר זה מכיל בעיקר ערכיי null ולכן הסקנו כי ניתן יהיה לוותר עליו, הוא אינו תורם מידע משמעותי ללמידת המודל.
22. **Purchase** – לפיצ'ר זה אין ערכי null, כלומר יש תווית לכל שורת מידע. ניתן לראות מהגרף (גרף 19) כי 84.5% מהמקרים מסתיימים באי רכישה, זה מעיד על מקרים מועטים שנגמרים ברכישה וזהו פריט מידע שיש לקחת בחשבון בעת ביצוע המודל. הסקנו כי נצטרך שה accuracy אליו נגיע יהיה גדול מ-84.5% .

לאחר ניתוח כל הפיצ'רים, בנינו מפת חום שמראה את הקורלציה בין כל הפיצ'רים הנומרים (גרף 20). מהמפה הסקנו כי region,B הם בעלי הקורלציה הנמוכה ביותר עם purchase ולעומת זאת page values, exit rates הם בעלי הקורלציה הגבוהה ביותר עם purchase. בנוסף, גילינו כי total duration בקורלציה גבוהה עם product page duration (0.99) ולכן החלטנו שניתן לוותר על total duration. כמו כן, מפת החום חיזקה את ההחלטה שלנו בסעיף 9, כי ראינו ש exit rates ו bounce rates בעלי קורלציה מאד גבוהה ולכן החלטנו להוריד את exit rates באופן סופי.

עיבוד (Processing)

בחלק הבא עיבדנו את הדאטא. ייצרנו מספר data frames כאשר בכל אחד מהם ביצענו עיבוד שונה לדאטא הכולל. המטרה הסופית הייתה להגיע ל final data שבו כל הפיצ'רים (הנומרים והקטגוריאליים) עברו עיבוד לפי טבלה בה סיכמנו את כל הפעולות שאנו רוצים לבצע לפיצ'רים (גרף 26). כעת נפרט על כל אחד מה data frames השונים.

1. **Df1 –** עיבוד בסיסי בו השתמשנו רק בפיצ'רים נומריים. החלפנו את כל ערכי ה nulls ל 0, ולא הסרנו outliers.
2. **Df2 –** עיבוד בסיסי בו השתמשנו רק בפיצ'רים נומריים. החלפנו את כל ערכי ה nulls לערכים הכתובים בטבלה המסכמת (גרף 26) , והסרנו outliers גם לפי הערכים בטבלה המסכמת.
3. **Df3 –** encoding user_type
4. **Df4 –** encoding months. חילקנו את החודשים ל4 קבוצות לפי עונות (summer,fall,spring,winter) כאשר בכל עונה יש 3 חודשים (חוץ מ spring,winter שבהם יש 2 חודשים). לאחר מכאן הסרנו את ערכי ה nulls וביצענו encoding לעונות.
5. **Df5-** encoding internet_browser, weekend, device, region, C, closeness_to_holiday. בפיצ'ר של הדפדפן צמצמנו את כל הערכים לשתי קטגוריות, שימוש בדפדפן כרום או בדפדפן אחר.
6. **DfA –** encoding A. כיוון שלא הבנו עד הסוף את משמעות הפיצ'ר, יצרנו גרף עוגה ומפת חום (גרף 21). החלטנו להוריד ערכים המופיעים פחות מ 199 פעמים, מאחר שלא היו הרבה ערכים כאלו והנחנו שזה לא ישפיע על טיב המודל. לאחר מכן, יצרנו את מפת החום בה ניתן לראות שהערכים של A אינם בעלי השפעה רבה על purchase.
7. **Df6 –** איחוד כל הפיצ'רים הקטגוריאליים (לאחר encoding) ללא פיצ'רים נומריים.
8. **Df7 –** final data. איחוד כל הפיצ'רים הקטגוריאליים (לאחר encoding) יחד עם הפיצ'רים הנומריים. מסוכם בגרף 26.

הקטנת מימדים (PCA and Scaling)

כדי לפשט את אימון המודלים ומניעת over-fitting, הורדנו מימדים בעזרת שיטת PCA. התחלנו בחלוקת הדאטא ל training ול validation. לאחר ניסיון ראשוני של PCA, הבנו שעליו לנרמל את הדאטא. ניסינו לעשות נורמליזציה בעזרת שני scalers שונים (standardscaler, powertransformer), כדי לראות מי מהם ייתן תוצאות טובות יותר בעת הרצת המודלים. לאחר נרמול הנתונים, ביצענו PCA עם דרישה למספר קופמננטות שמסבירות 99% מהשונות. כתוצאה מכך, כמות הפיצ'רים ירדה מ 55 ל 45-46.

הרצת מודלים (Modeling)

על מנת למדוד את טיב המודלים, השתמשנו במדד AUC עבור validation ועבור training. בסוף כל מודל, אפשר למצוא טבלה המסכמת את ציוני ה AUC עבור data frames שונים. המודלים אותם בחרנו לממש הם:

1. **Gaussian Naïve Bayes** – זהו המודל הראשון שמימשנו. מטרתנו הייתה לקבל מושג לגבי ציון ה AUC, ולהבין האם נדרשת עבודה נוספת על עיבוד הנתונים. כפי שניתן לראות בטבלה המסכמת של המודל (גרף 27) הציון הגבוה ביותר אליו הצלחנו להגיע הוא 0.8734 באמצעות final data using PCA בשימוש ב powertransformer scaling. בעקבות כך, מכאן והלאה נשתמש בנורמליזציה שנעשתה עם ה scaler הנ"ל.
2. **Logistic Regression** – כדי לממש מודל זה יש לבצע תחילה נורמליזציה של הנתונים. לכן, ניתן למצוא פונקציה המשתמשת ב standardscaler לביצוע נורמליזציה לפני הרצת המודל (נשים לב שהנתונים שעברו PCA כבר עברו scaling ולכן אין צורך לממש עבורם את הפונקציה). ניתן לראות בטבלה (גרף 28) כי הציון הגבוה ביותר אליו הגענו הוא 0.8921 עבור final data using PCA.
3. **Random Forest** – הרצנו את מודל זה פעמיים על final data using PCA, פעם עם ערכים דיפולטיים ופעם בשימוש GridSearch. פונקציית GridSearch סייעה לנו למצוא את הפרמטרים שיביאו לתוצאה הטובה ביותר. אפשר לראות את כל ההיפר פרמטרים של random forest מסומנים בהערה בקוד, אך בפועל הרצנו את הפונקציה עם ההיפר פרמטרים המשמעותיים. ניתן לראות בטבלת הסיכום (גרף 29) כי חל שיפור לאחר שימוש ב GridSearch והגענו לציון של 0.9024.
4. **SVM** - הרצנו את מודל זה פעמיים על final data using PCA, פעם עם ערכים דיפולטיים ופעם בשימוש GridSearch. אפשר לראות את כל ההיפר פרמטרים של SVM מסומנים בהערה בקוד, אך בפועל הרצנו את הפונקציה עם ההיפר פרמטרים המשמעותיים. ניתן לראות בטבלת הסיכום (גרף 30) כי לא חל שיפור לאחר שימוש ב GridSearch והציון הטוב ביותר אליו הגענו 0.8802.

בסוף חלק זה, ניתן למצוא טבלת המסכמת את ערכי ה AUC הטובים ביותר מכל מודל שהרצנו (גרף 31).

הערכת המודלים (Model Evaluation)

ביצענו כמה סוגים שונים של הערכות וגרפים על מנת לנתח את טיב המודלים שהרצנו.

1. **Confusion Matrix –** מדד זה איפשר לנו להבין איפה המודל טועה ואיפה המודל צודק, ולטובת איזה צד הוא נוטה. ביצענו את ההערכה הנ"ל על כל ארבעת המודלים (גרף 22), אך נפרט כן רק על תוצאות מודל ה SVM (כי זה המודל שבחרנו בסופו של לדבר להריץ על ה test). גילינו שמתוך 2026 מקרים בסט הולידיציה, ישנם 89.09% של סיווג נכון.
2. **Roc Plot –** ייצרנו גרף יחיד המציג את כל עקומות ה ROC של ארבעת המודלים בצבעים שונים (גרף 23). ה-ROC משמש כתחליף לכמות גדולה של confusion matrices, וכדי לדעת איזה מודל הוא המוצלח ביותר, חישבנו את ציון ה AUC של כל מודל, והוספנו לגרף. אפשר לראות שציוני ה AUC של המודלים יחסית קרובים אך הציונים הגבוהים ביותר היו של Random Forest ורגרסיה לוגיסטית.
3. **K-fold Method –** בהערכה זו הצגנו שוב את עקומות ROC של ארבעת המודלים (גרף 24) באמצעות cross validation בשיטת k-fold. בגרף הצגנו את התוצאה של כל קיפול, וגם את התוצאה הממוצעת של הקיפולים עבור מדד AUC.
4. **Learning Curve –** אחת הדרכים לבדוק האם מודל הוא over-fitted או under-fitted היא באמצעות שרטוט learning curve עבור כל מודל (גרף 25). בסופו של דבר, זה היה מדד ההערכה החשוב ביותר עבורנו ולפי התוצאות שלו קיבלנו את ההחלטה הסופית באיזה מודל להשתמש. גילינו שמודל ה Random Forest שאימנו הוא over-fitted מאחר שלא חש שיפור בגרף בעת הוספת training examples. הייתה לנו התלבטות בין

בחירה במודל רגרסיה לוגיסטית לבין מודל SVM, זאת מכיוון ששניהם נראו good-fitted כי המרחק בין הגרפים הצטמטם ככל שהתווספו דוגמאות מה training. עם זאת, ראינו בגרף של ה performance שהמודל של SVM משתפר עם כל איטרציה ולכן בסופו של דבר בחרנו להשתמש במודל זה.

ביצוע פרדיקציה (Prediction – Final code that does it all)

בשלב סופי זה יצרנו פייפליין אשר מאגד את כל קטעי הקוד (מטעינת הנתונים וביצוע עיבוד ועד ביצוע חיזוי). ביצענו עיבוד של קובץ ה training (ללא חלוקה לtrain ו valid) וגם של קובץ ה test. במהלך ההרצה על קובץ ה test, ראינו כי חסרים ממדים לעומת קובץ ה training, לאחר מחקר מעמיק, הבנו שישנם ערכי פיצ'ר 'a' שלא נמצאים בקובץ ה test, אך כן נמצאים בקובץ ה training. החלפנו לכן את ה encoding המלא ב encoding רק של חמשת הערכים הכי שכיחים של 'a' בקובץ האימון.

בסוף הפייפליין נוצר קובץ CSV של תוצאות החיזוי, וניתן לראות כי בנוסף הדפסנו את גודל ה training data, גודל ה test data, את מספר הפיצ'רים שנשארו אחרי PCA, ואת ציון ה AUC (0.9389) של מודל ה SVM לאחר הרצתו על ה training (גרף 32).

לסיכום

בפרויקט זה התבקשנו לסווג בצורה בינארית תצפיות שונות של משתמשים בעלות פיצ'רים שונים המעידים על קיום רכישה או אי רכישה.
תחילה ביצענו ניתוח מקדים לכלל הפיצ'רים שבו למדנו איך כל פיצ'ר מתנהג ומתפלג. עבור כל פיצ'ר, הצגנו גרפים שונים ונתונים סטטיסיטיים אשר אפשרו לנו לייצר טבלה שבה קיבלנו החלטה כיצד לעבד את הנתונים.

לאחר הניתוח הראשוני, יכולנו להתחיל לעבד את הדאטא. במהלך שלב העיבוד, יצרנו מסגרות דאטא שונות, בהן ניסינו "לשחק" עם הנתונים עד קבלת דאטא סופית. הדברים בהם התמקדנו בשלב זה הם: טיפול ב outliers, בערכי null, וביצוע encoding למשתנים קטגוריאליים. בסיום שלב זה, הגענו לדאטא סופית בשם df7 שבה נמצאים כל הפיצ'רים הקטגוריאליים והנומריים.

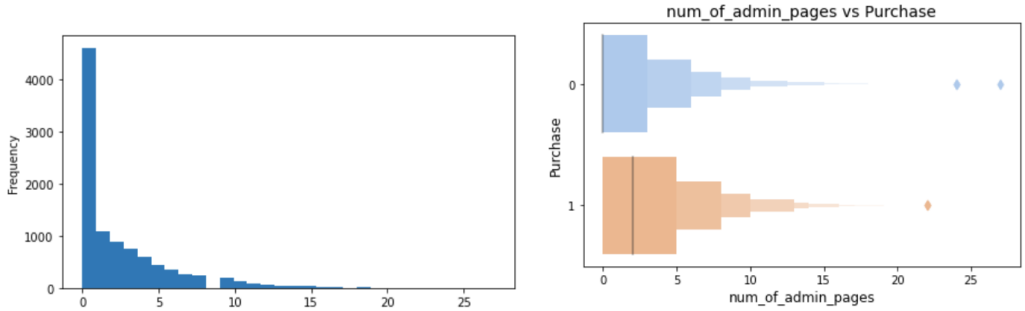
בשלב הבא, ביצענו PCA להורדת מימדים ואכן צמצמנו כ 10 מימדים. לאחר מכן, בחרנו ארבעה מודלים אותם אימנו על המסגרות הדאטא השונות. ארבעת המודלים הם: **GNB, Logistic Regression, Random Forest, SVM.**
בכדי להעריך את טיב המודלים, בחנו את ציון ה AUC שלהם בהרצות השונות. עבור SVM ו Random Forest השתמשנו בפונקציית GridSearch אשר אפשרה לנו למצוא את ההיפר פרמטרים הטובים ביותר עבור כל מודל. בסוף שלב זה יצרנו טבלה סופית המאגדת את ציוני ה AUC הטובים ביותר עבור כל מודל (גרף 31). בשלב זה, המודל עם הציון הגבוה ביותר היה Random Forest, וחשבנו שהוא יהיה המודל הנבחר.

כדי לוודא שהמודל שבחרנו הוא אכן מודל מוצלח, ביצענו הערכות מודלים שונות, ביניהן confusion matrix, ROC plot, K-fold Method, Learning curve. המדד אשר אפשר לנו להכריע בין המודלים השונים, היה מדד learning curve, באמצעותו גילינו שמודל Random Forest עשוי להיות over-fitted על אף ציונו הגבוה, ולכן בחרנו במודל SVM שנראה good-fitted ובעל ציון AUC גבוה גם כן.

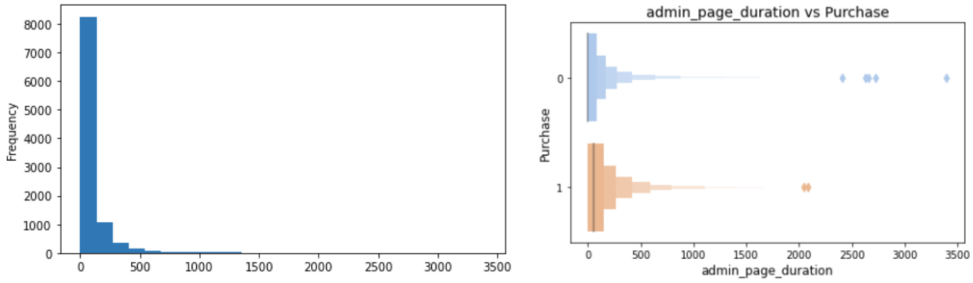
כל מה שנותר היה לאמן את המודל SVM על קובץ ה training (ללא חלוקה ל validation ול train), ולאחר מכן להריץ את המודל על קובץ ה test ולקבל פרדיקציה לרכישה או אי רכישה אותה ייצאנו לקובץ CSV נפרד.

נספחים

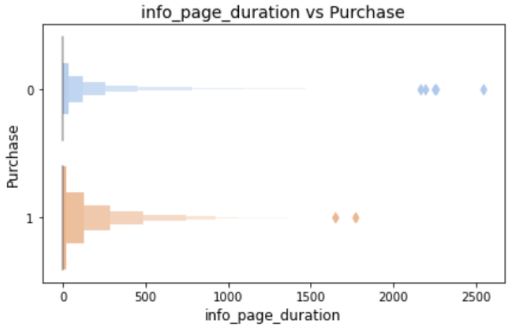
Num Of Info Pages Analysis - 1 גרף



Admin Page Duration Analysis - 2 גרף

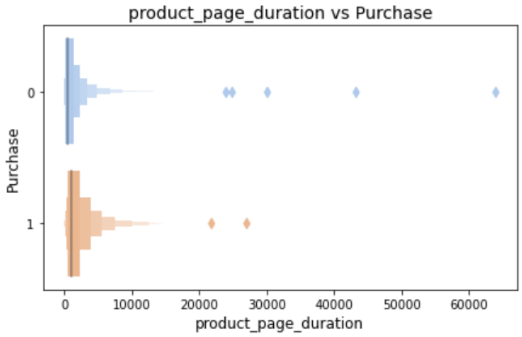


Info pages duration Analysis– 4 גרף

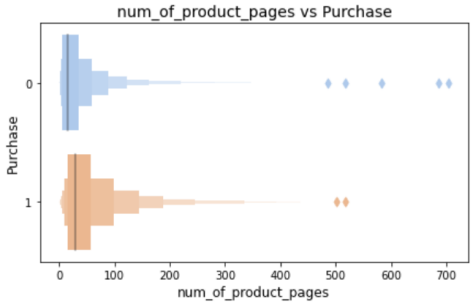


Num of info pages Analysis – 3 גרף

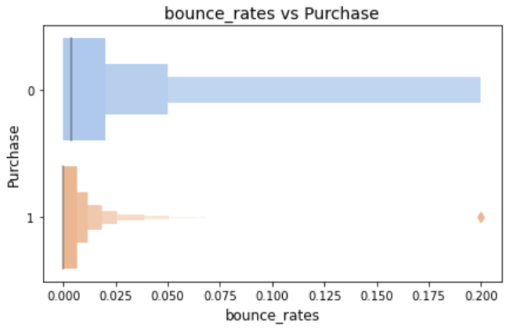
Product page duration Analysis– 6 גרף



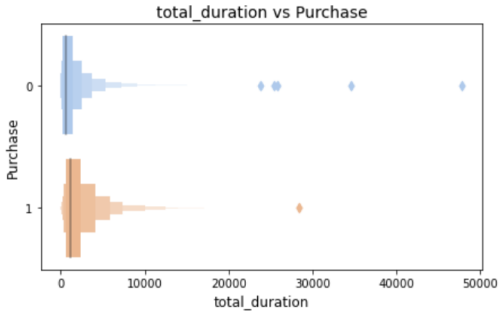
Num of product pages Analysis – 5 גרף



Bounce rates Analysis– 8 גרף

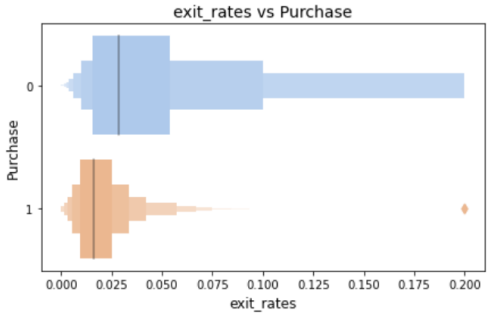


Total duration Analysis– 7 גרף

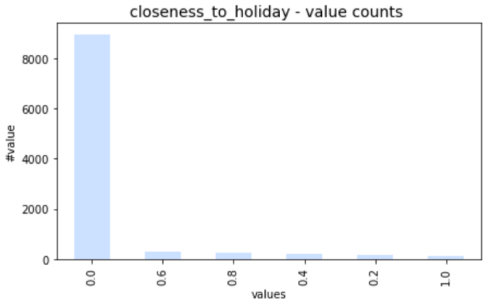


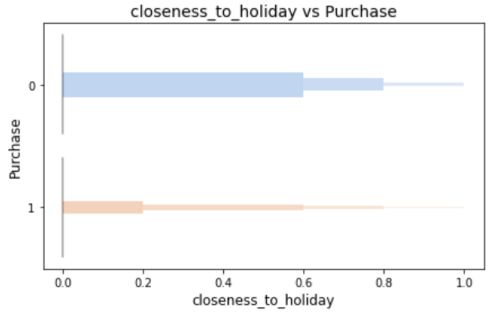
Page values Analysis– 10 גרף

Exit rates Analysis– 9 גרף

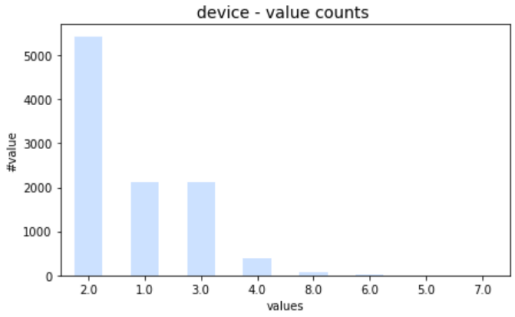


Closeness to holiday Analysis– 11 גרף

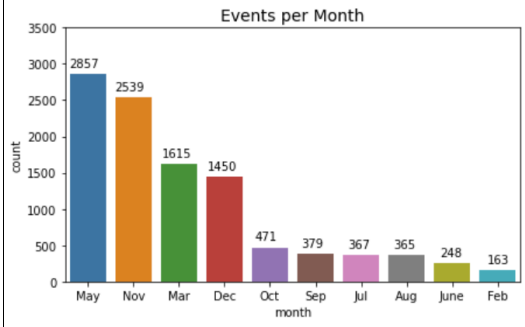




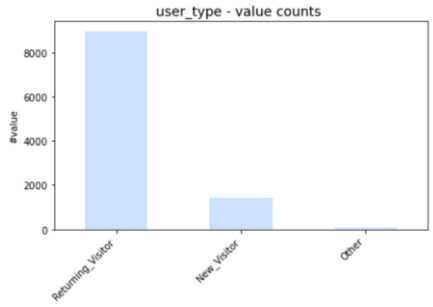
Device Analysis –13 גרף



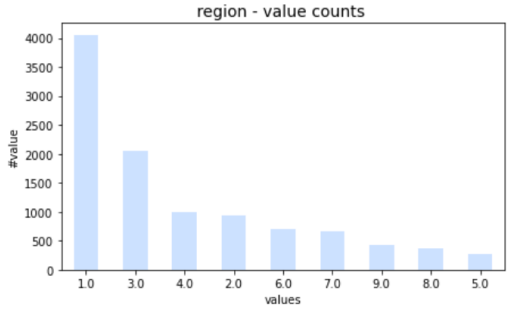
Month Analysis –12 גרף



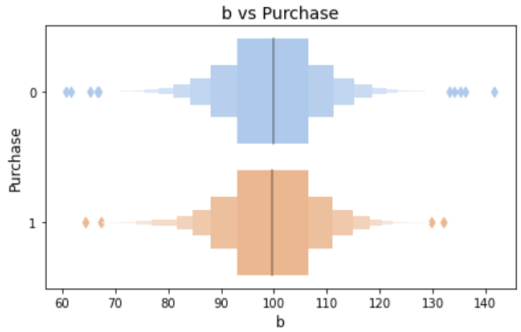
User type Analysis –15 גרף



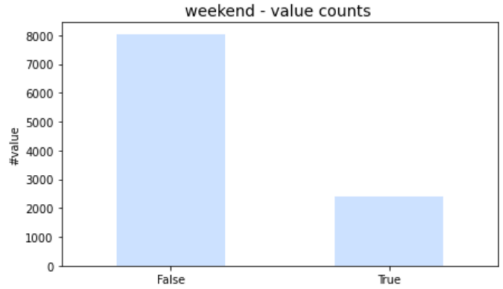
Region Analysis –14 גרף



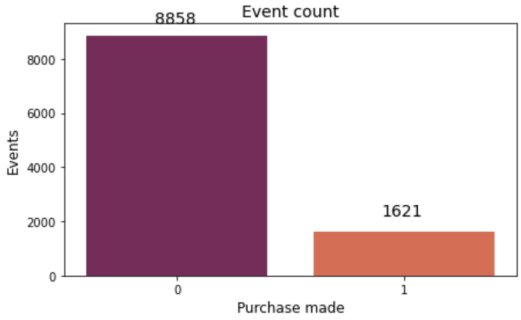
B Analysis– 17 גרף



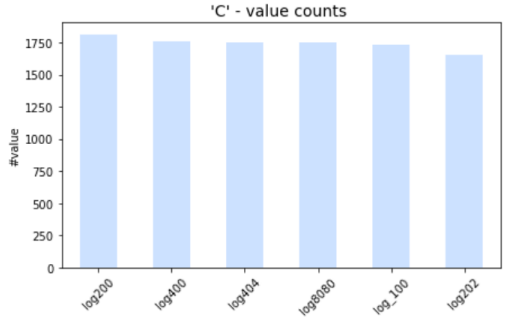
Weekend Analysis– 16 גרף



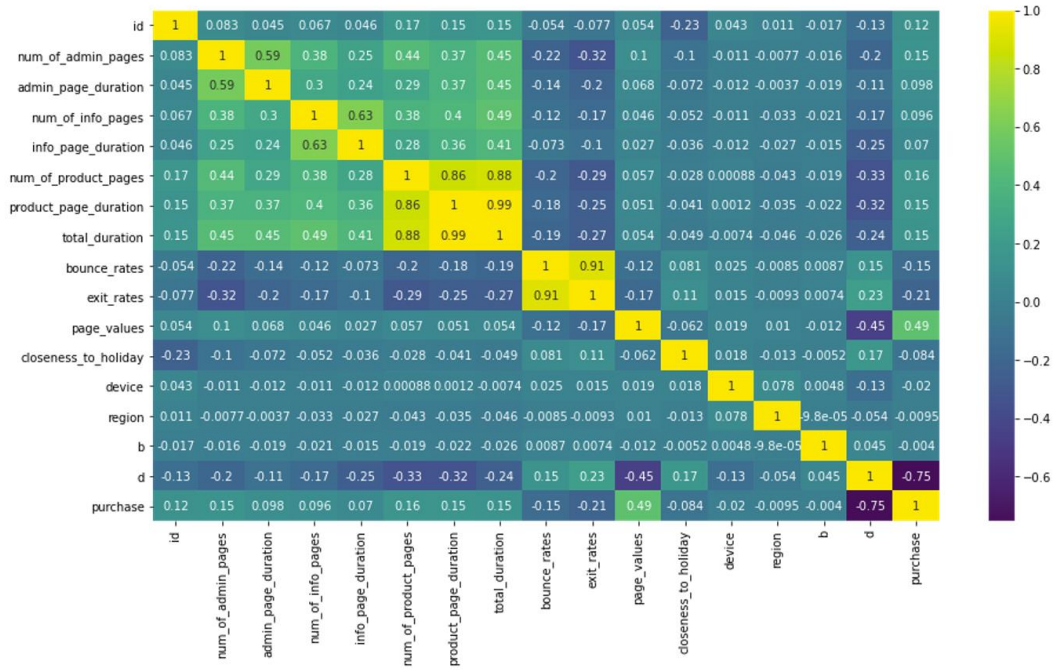
Purchase Analysis– 19 גרף



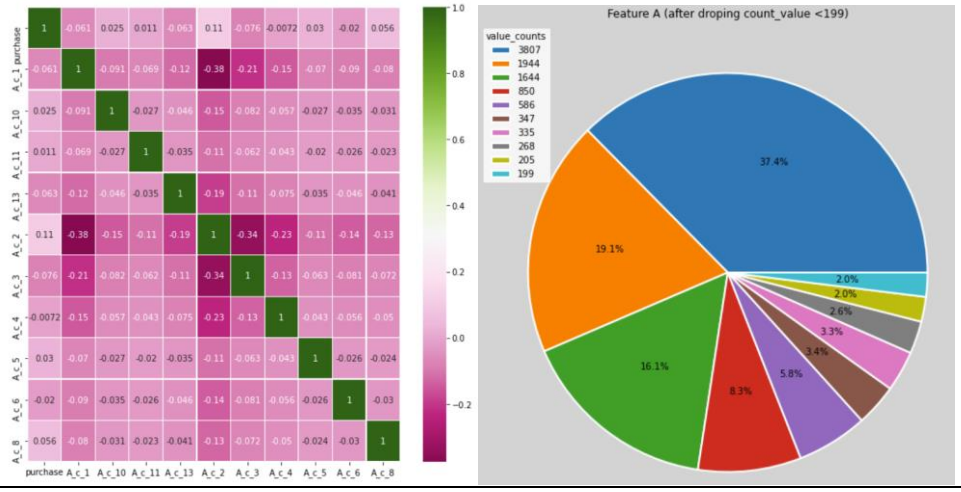
C Analysis– 18 גרף



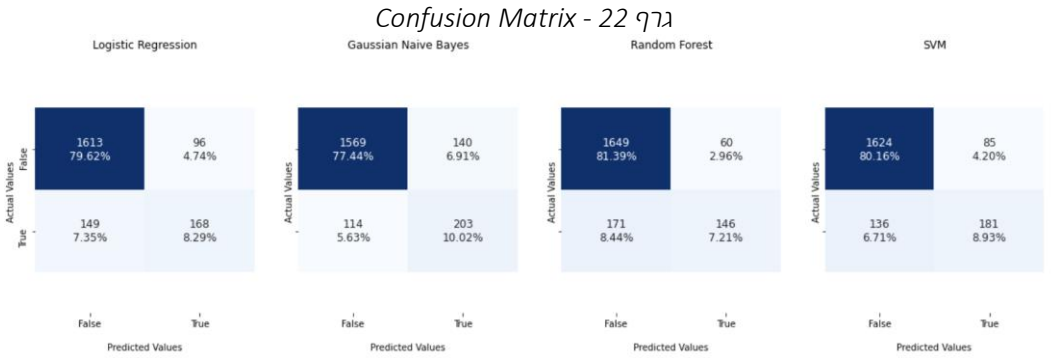
Heatmap of numerical features – 20 גרף



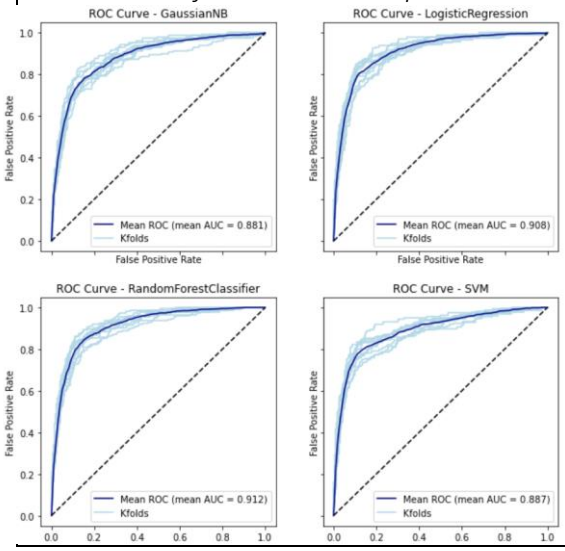
A Analysis - 21 גרף



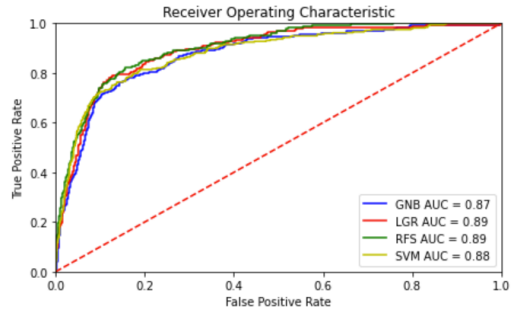
Confusion Matrix - 22 גרף



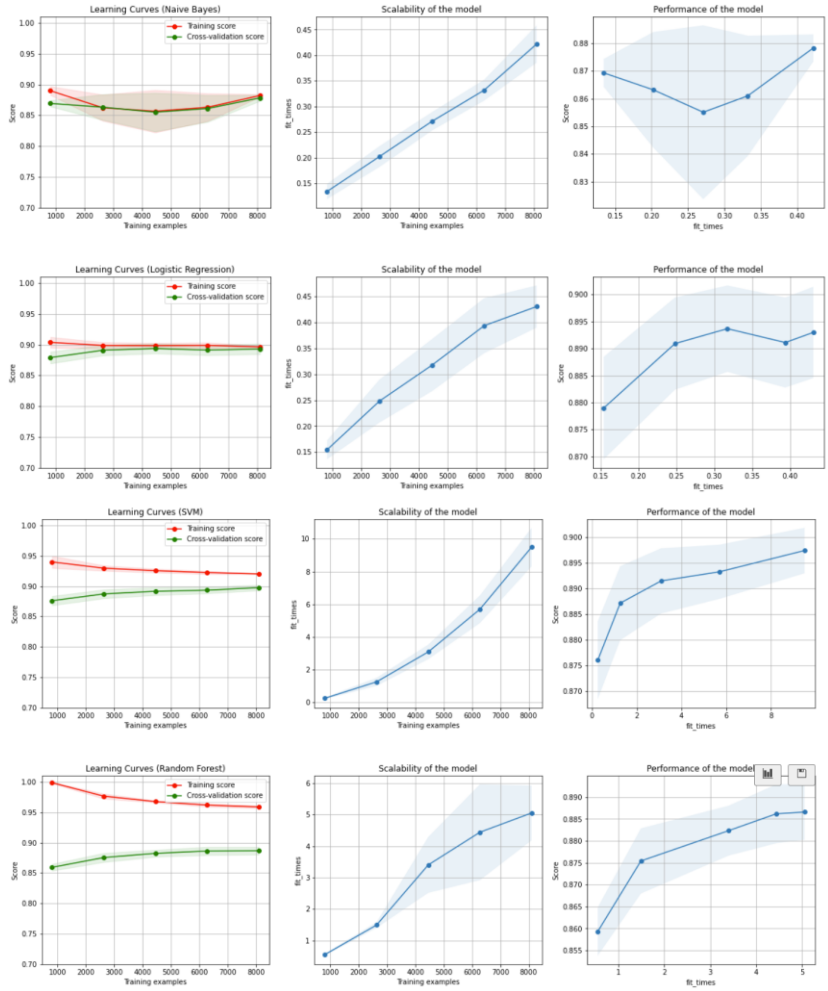
K-fold Method– 24 גרף



ROC plot– 23 גרף



Num Of Info Pages Analysis - 25 גרף



processing table for final data – 26 גרף

	Feature name	Type	Comments
0	id	int64	Drop
1	num_of_admin_pages	float64	Replace nulls with 0; remove outliers>15
2	admin_page_duration	float64	Replace nulls with 0; remove outliers>1500
3	num_of_info_pages	float64	Replace nulls with 0; remove outliers>12
4	info_page_duration	object	remove minutes; convert to float; replace nulls with 0; replace outliers with most frequent value
5	num_of_product_pages	float64	Replace nulls with mean; replace outliers with threshold
6	product_page_duration	object	remove minutes; convert to float; replace nulls with mean; drop outliers
7	total_duration	float64	dropping feature since it has a very high correlation (0.99) with product_page_duration
8	bounce_rates	float64	Replace nulls with 0; not removing outliers because the outlier number is very small and they appear only when a purchase is made
9	exit_rates	float64	Dropping feature because bounces are only recorded if a user exits directly from the page they entered while exit rates are recorded regardless of a user's prior activity on website. Therefore, all bounces are exits but not all exits are bounces.
10	page_values	float64	Replace nulls with mean; drop outliers
11	closeness_to_holiday	float64	No outliers; replace nulls with most frequent value; encode
12	month	object	Dividing feature into 4 seasons to reduce number of features before encoding; encoding; dropping nulls
13	device	float64	Replace nulls with most frequent value; encode
14	internet_browser	object	Encode; Convert to Chrome / non Chrome; replace nulls with 0
15	region	float64	Replace nulls with most frequent value; encode
16	user_type	object	Encode; replace nulls with "other"
17	weekend	object	Encode
18	a	object	Replace nulls to most frequent value; encode
19	b	float64	Drop (not informative)
20	c	object	Drop nulls; Encode
21	d	float64	Drop (too many nulls)

Log Reg model– 28 גרף

	Features	Data frame	Model	Validation AUC
0	Basic processing (Numeric only)	df1	Logistic Regression	0.876
1	Numeric only (without NAs & outliers)	df2	Logistic Regression	0.8853
2	Categorical only	df6	Logistic Regression	0.6686
3	Final Data	df7	Logistic Regression	0.8742
4	Final Data using PCA	df7	Logistic Regression	0.8921

Gaussian Naive Bayes model – 27 גרף

	Features	Data frame	Model	Validation AUC
0	Basic processing (Numeric only)	df1	Naive Bayes	0.8281
1	Numeric only (without NAs & outliers)	df2	Naive Bayes	0.8469
2	Categorical only	df6	Naive Bayes	0.642
3	Final Data	df7	Naive Bayes	0.7931
4	Final Data using PCA (PowerTransformer Scaling)	df7	Naive Bayes	0.8734
5	Final Data using PCA (StandardScaler)	df7	Naive Bayes	0.8045

SVM model -30 גרף

	Features	Data frame	Model	Validation AUC
0	Final Data using PCA	df7	Support Vectors Machine	0.8802
1	Final Data using PCA after GridSearch	df7	Support Vectors Machine	0.8763

Random Forest model – 29 גרף

	Features	Data frame	Model	Validation AUC
0	Final Data using PCA	df7	Random Forest	0.8998
1	Final Data using PCA after GridSearch	df7	Random Forest	0.9024

Pipeline print – 32 גרף

```
data size  (10479, 23)
data size after feature engineering  (10479, 55)
test size  (1851, 22)
test size after feature engineering  (1851, 55)
number of components in training data which preserve at least 99% of the variance: 45
number of components in test data which preserve at least 99% of the variance: 45
Support Vectors Machine - df7-final data:
Training AUC: 0.9389
```

Summary table – 31 גרף

	Features	Model	Validation AUC
0	Final Data with PCA	Guassian Naive Bayes	0.8734
1	Final Data with PCA	Log Reg	0.8921
2	Final Data with PCA	Random Forest	0.9024
3	Final Data with PCA	SVM	0.8802

נספח 33 – אחריות ותרומה לעבודה

תרומתנו לעבודה הייתה שווה, נפגשנו לעבוד יחדיו על כל קטעי הקוד והמחקר הנדרש בכדי ליצור פרויקט זה. את כל נסיונות ההרצה ביצענו יחדיו ובנוסף את דו"ח הסיכום כתבנו בסוף העבודה על הקוד.

אנו רוצים להודות לסגל הקורס שחיבר בינינו, הפכנו לחברים טובים יותר מאז תחילת הפרויקט (:

נספח 34 – תיעוד מהשטח