מבוא ללמידת מכונה - פרויקט מסכם

מרצה: עמיחי פיינסקי
מתרגלת: תמר אמיר

מגישים:
אלמוג אסרף 313200511 | דניאל פידטילוק 322558867 | ניר לבנון 313160715

תוכן עניינים

תקציר מנהלים **2**

הנחות **2**

ניתוח נתונים **3**

אקספלורציה 3

עיבוד מקדים 3

הרצת מודלים והערכתם 4

סיכום **6**

נספחים **7**

תקציר מנהלים

במסגרת עבודה זו קיבלנו מאגר נתונים העוסק ברכישות באתר אינטרנט (E-Commerce), בעל 10,497 דגימות ו-21 משתנים מסבירים. ראשית, ביצענו עבודת חקר במטרה לזהות אילו מן הגורמים המסבירים הם בעלי יכולת גבוהה יותר לחזות את פועלו של הלקוח, ואילו אינם רלוונטיים. עבודה זו כללה יצירת "תעודת זהות" לכל אחד מן הגורמים המסבירים, הכוללת התייחסות למידע על קונטקסט הפעולה של הגורם, אחוז הערכים החסרים מתוך כלל הדגימות, אחוז הערכים הייחודיים, התפלגות הנתונים, בדיקת קשר בין הנתונים לבין תוצאת הרכישה, גרף נקודות לגורמים מסבירים עם קורלציה גבוהה, גרף Boxplot ומספר חריגים לפי IQR.

כתוצאה מלמידת כל אחד מן הגורמים המסבירים, החלטנו לבצע עיבוד מקדים על החומר, וזאת במטרה לאפשר לימוד מודל מוצלח לפיו. העיבוד המקדים כלל תחילה ניקוי ביטויים "מפריעים" והמרתם ל-Float, והן מילוי ערכים חסרים בהתאם להיותם נומריים או קטגוריאליים. לאחר מכן, הסרנו גורם מסביר אשר 98.96% מערכיו היו חסרים וצמצמנו זוגות של גורמים מסבירים כאשר זיהינו קורלציה גבוהה בין התנהגותם לתוצאה, במטרה לצמצם את מימדי הבעיה. כמו כן, המרנו גורמים מסבירים קטגוריאליים לבינאריים במטרה למנוע הטייה ולאפשר את הרצת המודלים, ביצענו סטנדרטיזציה של הנתונים כדי לצמצם השפעה של ערכים חריגים, יצרנו עותק של הנתונים והרצנו עליו PCA השומר על 95% מהשונות המקורית, ולבסוף גם הסרנו תוצאות קיצוניות בעלות סטיית תקן גדולה מ-5.

על בסיס ה-data שהתקבל, בנינו שמונה מסווגים בהתבסס על ארבעה סוגים של מודלים: Logistic Regression, ANN, Decision Tree ו-Random Forest, כאשר בכל פעם אימנו כל סוג של מודל על ה-dataset הרגיל ועל ה-PCA. כדי למצוא היפר-פרמטרים מוצלחים ככל הניתן לטובת כל מודל, בנינו כל אחד מהם בעזרת GridSearchCV. עם סיום האימון יצרנו Confusion Matrix מדגמית באמצעות מודל הרגרסיה הלוגיסטית שלנו, וגילינו כי קיימת סבירות גבוהה יותר לקבל FN מאשר TP. בהמשך, בנינו שמונה תרשימי ROC-AUC של כל אחד מהמודלים השונים, כאשר השתמשנו ב-K-Fold וחילקנו את התצפיות ל-10 קבוצות, ששימשו אותנו לטובת אימון המודלים השונים ומציאת תוחלת התוצאות. לאחר מכן ביצענו אימון נוסף של המודלים השונים, חישבנו את ציון החיזוי שלהם על האימון (Train) והווידוא (Validation) ובעזרתם בנינו את מדד ה-Difference, המעיד על overfitting. בהתאם, פסלנו את המודלים אשר מדד ה-Difference עבורם היה גדול מ-3%. מתוך המודלים שנשארו, בחרנו את המודל בעל ציון ה-Validation הגבוה ביותר, אשר במקרה שלנו התברר להיות **רשת נוירונים (ANN) שאומנה על ה-dataset הרגיל**.

לבסוף, השתמשנו ברשת הנוירונים הנבחרת כדי לבצע חיזוי באמצעות מאגר תצפיות המבחן (Test) שסופק לנו, וקיבלנו תחזית של 9% רכישות מתוך הפעולות הקיימות בו, זאת בהשוואה לתצפיות שקיבלנו כאימון, בהן 15% מהתצפיות הסתיימו ברכישה. לפיכך הסקנו כי המודל שלנו אינו סובל מ-overfitting, וזאת מכיוון שהוא מגיע לתוצאות דומות בכלל השלבים - האימון, הוידוא והמבחן.

הנחות

- כלל ההחלטות שהתקבלו - התבססו על קבצי ה-train וה-test המסוימים שהיו בידינו. ייתכן כי על קובץ test אחר, יש לבצע התאמות אחרות. דוגמה לכך, היא הסרת הגורם המסביר "device_5.0" בעקבות ההתמודדות עם משתנים קטגוריאליים. להרחבה, ראו סעיף 6 תחת עיבוד מקדים.

- עבור קבלת פלט זהה עבור כל הרצה, בוצע קיבוע seed בפונקציות השונות.

ניתוח נתונים

אקספלורציה

מטרת האקספלורציה הינה הצגת **המידע** (להלן: data) **הקיים**, והצפת פעולות רלוונטיות לביצוע בשלב העיבוד המקדים - לדוגמה, בהיבטי מילוי ערכים חסרים, הסרת גורמים מסבירים לקוניים וכו'.

תחילה, נציג נתונים סיכומיים (ערכים וויזואליציה) עבור ה-data כולו. הודות לכך, נסיק מסקנות השוואתיות וקשרים בין הפי'צרים השונים.
לאחר זאת, נזקק מסקנות קונקרטיות עבור כל גורם מסביר על ידי הצגת "**תעודת זהות" אישית** לגורם המסביר, אשר כוללת:

1. תמצית הגורם המסביר
   a. מידע סיכומי על הגורם המסביר
   b. אחוז הערכים החסרים - נבחן את רלוונטיות הגורם המסביר ונגזור משמעויות עבור מילוי הערכים החסרים במידת הצורך.
   c. אחוז הערכים הייחודים - נבחן אם הגורם המסביר אכן רציף או דווקא קטגוריאלי במהותו.
2. התפלגות הנתונים - נבחן אם ישנה התפלגות מוכרת לנתונים, ונשער את השפעת נרמול הנתונים.
3. התפלגות הנתונים בחלוקה לתוצאת הרכישה (0,1) - נבחן אם יש קשר ראשוני-אינטואיטיבי בין ערכי הגורם המסביר לבין תוצאת הרכישה.
4. התפלגות הערכי החסרים בחלוקה לתוצאת הרכישה (0,1) - נבחן אם הערכי החסרים מתנהגים בדומה לשאר הנתונים הקיימים, או שמא בעלי השפעה ברורה על תוצאת הרכישה, כך שיש משמעות להיותם חסרים.
5. גרף נקודות (Scatter) בין עמודות עם קורלציה גבוהה - נסמן את הגורמים המסבירים התואמים עבור הורדת מימד אפשרית (עבור גורם מסביר רציף).
6. גרף Boxplot ומספר החריגים לפי IQR - נבחן את אחוז הערכים החריגים ומשמעותם, עבור הסרתם בהמשך (עבור גורם מסביר רציף). נבחר ב-IQR, מאחר שזהו מדד מקובל, וחלק מהנתונים מתפלגים נורמלית.

לדוגמה, ראו נספח א'.

עיבוד מקדים

כלל הפעולות שהתבצעו בחלק זה, התבססו באופן ישיר על מסקנות האקספלורציה. עבור כל שלב, עלו מספר שיטות רלוונטיות להתמודדות (כלל הפונקציות נותרו בקוד, אך לא הופעלו בהגשה הסופית). להלן יפורט רציונל השיטות שנבחרו:

1. **ניקוי ביטויים "מפריעים" והמרה ל-float -** בהתאם לחלק א', התגלה כי עמודות "info_page_duration", "product_page_duration", ו-"A" מכילות ביטויים לצד ערכים מספריים, ולכן בתחילה הוגדרו כגורמים מסבירים קטגוריאליים (מסוג object), אך התפלגותם מתאימה לגורם מסביר נומרי רציף. בהתאם, "ננקה" את "הביטויים המפריעים", ונמיר את ה-data_type שלהם ל-float.
2. **חילוץ המידע הרלוונטי מהגורם המסביר "internet_browser"** - מחקר האקספלורציה התגלה כי גורם מסביר זה מכיל 125 ערכים ייחודים, כאשר הגורם למספר גדול זה היא גרסת ה-browser. לדעתנו, ההכללה תהיה טובה יותר אם נצמצם את מספר הקטגוריות בגורם מסביר זה, ולכן עבור כל תצפית נותר שם הגרסה בלבד, כך שסך הכל ישנן 4 קטגוריות.
3. **מילוי ערכים חסרים**

לאחר מספר ניסיונות, הוחלט למלא את הערכים החסרים בגורמים מסבירים השונים לפי סיווגם - כגורמים מסבירים נומריים או קטגוריאליים:

   a. גורמים מסבירים נומריים (למעט גורם מסביר "total duration") - הערכים החסרים מולאו לפי עשרת השכנים הקרובים ביותר.
   b. גורם מסביר "total duration" - כשם הגורם המסביר, משמעותו היא סכימה של שאר הגורמים המסבירים המייצגים duration. לכן, נמלא את הערכים החסרים על ידי סכום הגורם המסבירים הרלוונטיים.
   c. גורמים מסבירים קטגוריאליים - הערכים החסרים מולאו לפי הערך התדיר ביותר.

4. **הסרת גורמים מסבירים**
   a. <u>גורם מסביר "D"</u> - גורם מסביר רציף בעל 98.96% ערכים חסרים. עם זאת, ישנה חלוקה יחסית ברורה עבור תוצאת הרכישה (0 או 1). לבסוף, הוכרע כי מילוי הערכים החסרים הרב עלול לגרום להטיה של הנתונים, ולכן נוריד את הגורם המסביר זה.
   b. <u>גורמים מסבירים עם קורלציה גבוהה</u> - באקספלורציה התגלה כי ישנם שני זוגות של פי'צרים קולרטיביים, ולאחר העיבוד המקדים שבוצע עד כה, נוספו זוגות נוספים. כדי להוריד את מימד הבעיה, הוחלט להוריד את הגורמים המסבירים אשר הקורלציה ביניהם גדולה מ-0.8 (בערך מוחלט).
   הגרמים המסבירים שהוסרו הם: 'ExitRates', 'num_of_product_pages', 'total_duration'.
5. **התאמת ה-datatype** (לגורמים המסבירים שנותרו) - עבור וידוא.
6. **התמודדות עם גורמים מסבירים קטגוריאליים**
   a. כלל הגורמים המסבירים אלה הומרו לגורמים מסבירים בינאריים. יצוין כי אנו ערים לכך כי פעולה זו מגדילה את מימד הבעיה, אך לדעתנו עשויה למנוע הטיה (במקום המרה לערכים מספריים, לדוגמה).
   b. נשים לב, כי עבור התאמה בין הגורמים המסבירים הקיימים בסט ה-train ובסט ה-test, גורם מסביר "device_5.0" הוסר מסט ה-train, מאחר שאינו קיים ב-test.
7. **ביצוע Standardization & Scaling** - בעת בחירת השיטה הרלוונטית, הובאו בחשבון שני שיקולים מרכזיים - האחד התאמה להתפלגות הקיימת עבור חלק מהגורמים המסבירים, והשני שיטה אשר פחות רגישה לערכים חריגים. באקספלורציה נראה כי ישנן עמודות המתפלגות נורמליות, וידוע כי סטנדרטיזציה רגישה פחות לערכים חריגים ← ולכן נבחר להשתמש בשיטה זו.
8. **PCA** - הוחלט לבצע PCA אשר שומר 95% מהשונות המקורית (הוחלט לא לקבוע מספר קשיח של גורמים מסבירים, מאחר שחששנו לפגוע בהתאמת המודל והכללה בהמשך). כדי לבחון את תרומת שלב זה, ביצענו את השיטה על עותק של ה-data המקורי, כך שבהמשך - המודלים שנבחרו הורצו והוערכו עבור ה-data עם ובלי PCA.
9. **הסרת Outliers** - בהתאם להצגה בשלב האקספלורציה, נראה כי בכל גורם מסביר ישנם ערכים חריגים. בהמשך לסטנדרטיזציה, הערכים החריגים הוסרו על ידי z_score, לפי 5 סטיות תקן (יצוין כי בוצע ניסיון גם עבור 3 ס"ת, אך נותר מספר יחסית קטן של רשומות, שעלול למנוע למידה רצויה).

יצוין כי כל אחד משלבי העיבוד המקדים (למעט הסרת Outliers) , בוצע במקביל על סט ה-train, וגם על סט ה-test, כאשר ההחלה על ה-test בוצעה על בסיס הערכים ב-train.

<u>הרצת מודלים והערכתם</u>

עם סיום העיבוד המקדים, **בוצעה חלוקת train_test_split** בקנה מידה של 75% train ו-25% validation, על כל אחד משני ה-dataset-ים שלנו - הרגיל וה-PCA. על כל אחד מהם הורץ **GridSearchCV** על ארבעה מודלים: **Logistic Regression, ANN, Decision Tree ו-Random Forest**, עבור מציאת הפרמטרים הטובים ביותר.
במסגרת כך, יצוינו ההערות הבאות:

1. כנלמד בקורס, הובא בחשבון מראש כי מודל Decision Tree הינו מודל בעל יכולת הכללה פחות טובה, בהשוואה לשאר המודלים.
2. בין היתר, בוצע ניסיון גם להריץ את ה-dataset-ים על מודל ה-SVM, אך נתקלנו בתוצאות לא טובות עם היפר-הפרמטרים שבדקנו, כך שהוחלט להתעלם ממודל זה.
3. בקוד מפורטים הפרמטרים הסופיים שנבחנו במסגרת הרצת GridSearchCV. במסגרת העבודה, נבחנו מספר אפשרויות נוספות לפרמטרים הנבחנים.

על בסיס פלט ה-GridSearchCV, נבנו המודלים עם הפרמטרים הטובים ביותר שנמצאו.
לאחר זאת, **חושבו המדדים ROC ו-AUC**, דרך פונקציית K-Fold-Plot (עבור k=10, כמקובל), עבור כל אחד מהמודלים הנ"ל. פלט הפונקציה הינו תרשים אחוד הכולל שמונה גרפים שונים, שני גרפים לכל סוג מודל, כך שאחד מציג תוצאה של ה-dataset רגיל והשני של ה-dataset לאחר PCA.
מהתוצאות שהתקבלו ניתן לומר כי עבור כל אחד המודלים, מדד ה-AUC היה **גבוה יותר עבור ה-dataset הרגיל**, בהשוואה ל-dataset לאחר PCA. בתוך כך, המודל בעל המדד הגבוה ביותר הינו **0.918** עבור **המודל Random Forest**, ולאחריו המודל Logistic Regression עם 0.909, המודל ANN עם 0.903, ולבסוף Decision Tree עם 0.731.

בעקבות כך, כבר בנקודה זו ניתן לשער בסבירות בינונית-גבוהה כי המודל האידיאלי יתקבל עבור ה-dataset הרגיל, מבין שלושת המודלים בעלי AUC>0.9. אך, למען הסר ספק, נמשיך לבחון את כל שמונת המודלים, מאחר שטרם בוצעה התייחסות לאפשרות ה-overfitting.

לאחר זאת, נבחן אילו מודלים מביאים ל-**overfitting** דרך פונקציית Score_By_K_fold. לטובת כך, הוגדר המדד Difference המייצג בעינינו את הפער בין ה-train לבין ה-validate, או במילים אחרות, מעיד על overfitting. להלן יפורט אופן ביצוע המדד ומשמעותו:

1. חלוקת ה-data ל-train ו-validate בעזרת פונקצית k-fold.
2. אימון המודל הנבחן על ה-train שהתקבל.
3. חישוב ה-score עבור ה-train.
4. חישוב ה-score עבור ה-validate.
5. חישוב המדד **Difference** - היחס בין ההפרש בין ה-score של ה-train וה-validate לבין ציון ה-train, כלומר $\frac{abs(TrainScore - ValidateScore)}{TrainScore}$.

חישוב זה התבצע עבור 10 חלוקות שונות של ה-data (בעזרת k-fold), כאשר עבור כל חלוקה חושבו הציונים עבור ה-train וה-validate, ולאחר מכן בוצע מיצוע של הציונים, ובהתאם - חישוב מדד ה-Difference.
(יצוין כי עבור הורדת סיבוכיות זמן הריצה של הקוד, היה ניתן לשלב את חישוב מדד זה במסגרת פונקציית K-Fold-Plot, אך עבור פישוט, הועדף לפצל בין שתי הפונקציות).

עבור <u>בחירת המודל הרצוי</u>, הוגדרו **שני התנאים** הבאים:

1. מדד ה-Difference קטן מ-3%.
2. מבין המודלים שנבחרו, נבחר את המודל בעל ה-score הגבוה ביותר עבור ה-validate.

התנאי הראשון פסל את המודלים Decisions Tree ו-RandomForest, גם ל-dataset הרגיל וגם ל-PCA.
מבין המודלים התקינים, התנאי השני הביא לבחירתו של **<u>המודל רשת הנוירונים (ANN) על ה-dataset הרגיל</u>**, שהשיג ציון של 89.1% על ה-Train ו- 88.8% על ה-Valditaion.

כפועל יוצא, בוצע אימון על כלל הנתונים של קובץ ה-train (ללא PCA, כפי שהתקבלו לאחר העיבוד המקדים בשלב 2), והתבצעה פרדיקציה על קובץ ה-test המתאים (שוב, לאחר העיבוד המקדים).

לצד כל זאת, כנדרש, נבנתה Confusion Matrix מדגמית עבור המודל Logistic Regression שאומן על ה-dataset הרגיל. התקבלה תוצאה של TN 1410, FN 179, FP 27, TP 85, ודיוק של 91% בהרצה על ה-validate. מעניין לראות כי אנו נמצאים יותר ב-FN יותר מאשר ב-TP, כלומר - אנחנו מפספסים אחוז גדול של רכישות בפועל.

כלל התוצאות בכל אחד מן השלבים, מופיע כפלט ייעודי במחברת המצורפת.

סיכום

לאורך שלבי העבודה השונים, החל מהפרדיקציה, דרך העיבוד המקדים, ועד הרצת המודלים והערכתם, נתקלנו בסוגיות ובאתגרים להתמודדות - אופן הצגת המידע בצורה היעילה והברורה ביותר, החלת הפעולות המתאימות ביותר על ה-dataset-ים שברשותינו, בחירת המודלים וקביעת התנאים לבחירת המודל הטוב ביותר. במסגרת כך, לאורך הדרך חזרנו לשלבים העבודה השונים והתאמנו את ההחלטות שהתקבלו על מנת לקבל את המודל המיטבי עבור הנתונים הקיימים.

בתוך כך, אתגר משמעותי שניצב בפנינו הוא ה-overfitting. בשלבים הראשונים, ניסינו להקטין את בעיה זו בעזרת הסרת מסבירים משתנים ותצפיות בעזרת שיטות שנלמדו, ולאחר זאת, נעזרנו בשיטת K Fold Cross Validation וכן בעזרת מדד ה-overfitting שהוגדר (Difference). במסגרת זאת יצויין כי לפי כלל המודלים שנבחנו, שיטת PCA אינה תרמה לקבלת מדדים גבוהים יותר ו/או הקטנת ה-overfitting.

בשורה התחתונה, התקבל כי המודל המתאים ביותר לבעיה הינו **המודל רשת הנוירונים (ANN) על ה-dataset הרגיל**, ללא עיבוד עם שיטת PCA. להלן פרטיו:

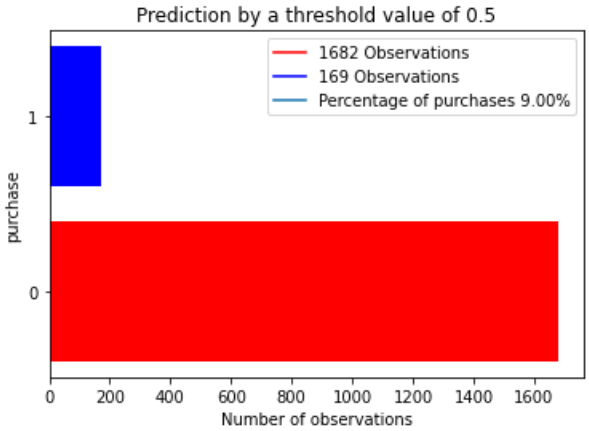
- הפרמטרים הטובים ביותר:

```
{'activation': 'logistic', 'alpha': 0.1, 'batch_size': 'auto',
'early_stopping': False, 'hidden_layer_sizes': (50, 50), 'learning_rate':
'constant', 'learning_rate_init': 0.1, 'max_iter': 500, 'power_t': 0.5,
'random_state': 0, 'solver': 'adam', 'tol': 0.0001, 'warm_start': False}
```

- ציון ה-AUC: 0.903 (מדורג שלישי מבין ארבעת המודל ללא עיבוד PCA)
- מדד ה-Difference: כ- 0.3% (<3%).
- מדד ה-Validate Score הממוצע: 88.8%.

כמפורט לעיל, יצוין כי מודל זה לא 'זכה' במקום הראשונים במדדים הראשונים שנבחנו, אלא דווקא המודלים Logistic Regression ו-Random Forest הביאו לתוצאות טובות יותר. בסופו של דבר, מודל Random Forest נפסל מפאת מדד ה-Difference (כ-10%), ומודל Logistic Regression אינו נבחר מאשר שקיבל ציון ממוצע נמוך יותר עבור ה-validate (כ-87%).

בהתאם, בוצע אימון על ה-data על בסיס המודל הנבחר, וחיזוי להסתברות שתתבצע רכישה. על בסיס הפלט ההסתברויות, קבענו כי ערך הסף לרכישה הינו 0.5, כך שתצפית שערכה קטן מ-0.5 תסווג כ-0 (אין רכישה), ותצפית שערכה שווה או גדול ל-0.5 , תסווג כ-1 (תתבצע רכישה). להלן גרף המתאר את תוצאות חישוב זה:



לפיכך, נסיק כי רק כ-9% מכלל העסקאות הקיימות, יהיו מוצלחות.

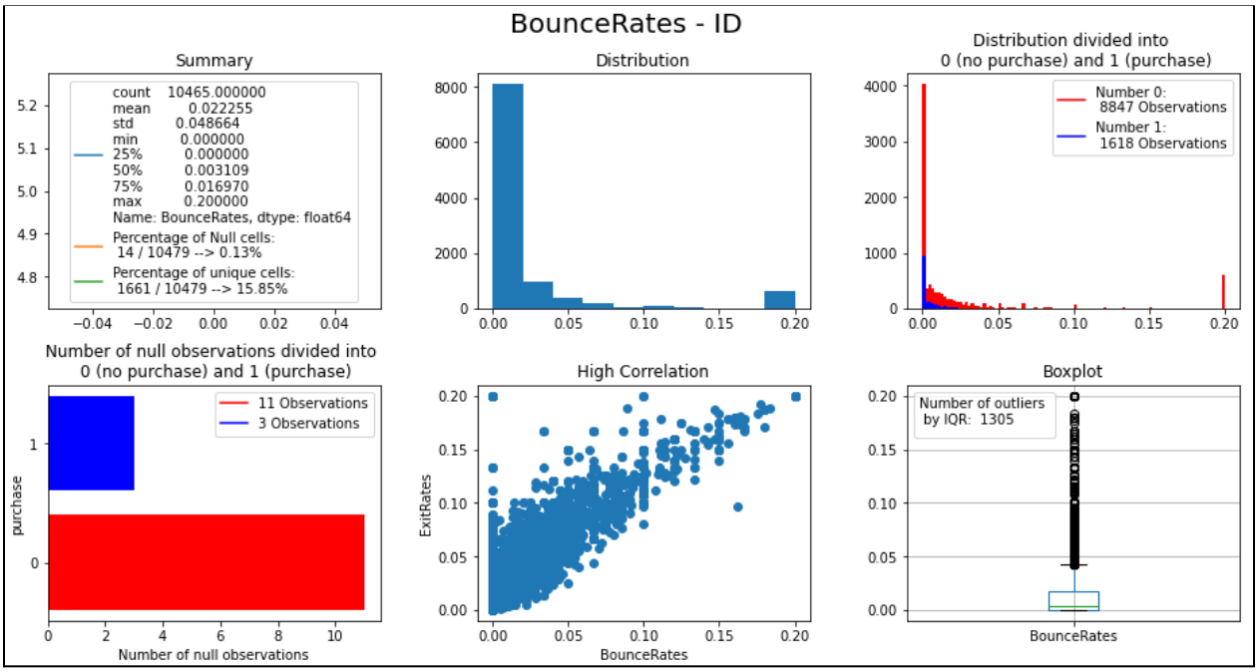
ב"עיבוד המקדים" ראינו בסט נתונים "train" כי מכלל העסקאות, רק 15% מתוכן היו מוצלחות. נתון זה יחסית קרוב לתחזית שביצענו על סט הנתונים "test". כלומר, נוכל להסיק כי המודל הסופי שנבחר אכן אינו overfitted, היות שהוא מצליח להתמודד בצורה טובה ודומה גם עם סט אימון וגם עם סט test.

נספחים

<u>נספח א'</u>

מטרת הויזואליזציה הזו הינה הצגת המידע הרלוונטי עבור גורם מסביר בצורה תמציתית ובמקום יחיד.

תעודת זהות אישית עבור גורם מסביר רציף - לדוגמה **BounceRates**



תעודת זהות אישית עבור גורם מסביר קטגוריאלי - **Closeness_to_holiday**