מבוא ללמידת מכונה – פרויקט סיום

בר סובל, ת.ז. - 206574584

דין צור, ת.ז. - 209493782

09/06/2022

תקציר מנהלים

בפרויקט זה עבדנו עם דאטה אודות סשנים של משתמשים באתר קניות באינטרנט. מטרתנו הייתה לבנות מערכת המנבאת את הסיכוי של משתמש מסוים לבצע רכישה בזמן הגלישה באתר. הדאטה שקיבלנו היה צריך לעבור עיבוד מקדים על מנת שנוכל להריץ עליו מודלים. השלמנו ערכים חסרים, מחקנו ערכים חריגים, ניתחנו את ההתנהגות של כל פיצ'ר בנפרד, הסרנו פיצ'רים מיותרים, יצרנו פיצ'ר חדש. כמו כן, נרמלנו את הדאטה, הורדנו מימדים בשביל לשמור על ה-Bias-Variance tradeoff. יתרה מכך, הרצנו 4 מודלים שונים – מצאנו לכל מודל את ההיפר-פרמטרים המיטביים, הרצנו K-Fold בשביל לבחור את המודל הרצוי ולהבין את פערי הביצועים של המודלים ולנסות להימנע ככל הניתן מ-overfitting.

לבסוף, המודל הנבחר הינו Random Forest לו התוצאות הטובות ביותר. ביצענו תחזיות על סט ה-test ואת כל הנאמר לעיל הכנסו לפונקציית Pipeline המקבצת את כל התהליך.

חלק ראשון – אקספלורציה

בחלק הראשון של משימת התכנות התמקדנו בלנסות להבין את הנתונים ולחקור את הדאטה. ראשית, סקרנו את הנתונים במבט מעל, וראינו שיש לנו 23 פיצ'רים (הכוללים עמודת ID שהיא לא אינפורמטיבית וחיונית להמשך הפרויקט ולכן נסיר אותה, ועמודת הלייבל) ו-10,479 תצפיות. לאחר מכן, בדקנו את הסוג של כל פיצ'ר, וכמה ערכים חסרים ולא חסרים קיימים בעמודתו. בנק' ההתחלה, היו לנו 13 פיצ'רים מסוג float64, 2 מסוג int64 ועוד 8 מסוג object. לשם הנוחות וכדי לקבל כיוונים להמשך העבודה, נעזרנו בקובץ שמגיע עם הנחיות הפרויקט ומסביר על כל פיצ'ר. יצרנו רשימות לכל סוג מהנ"ל ובהם הכנסנו את שמות העמודות ברשימות המתאימות. למשל, ברשימת 'bool_features' הכנסנו את הפיצ'ר 'Weekend'.

המשכנו עם ויזואלציה הממחישה את מס' הערכים החסרים של כל הפיצ'רים (נספח 1). ראינו למשל, שבעמודת D יש יותר מ-80% ערכים חסרים. לכן, החלטנו להסיר את הפיצ'ר הזה.

הסתכלנו על ההתנהגות הקורלטיבית בין הפיצ'רים ואף חקרנו איפה יש קורלציות הגבוהות מ-0.5 (נספח 2). נספחים 3, 4 ו-5 מציגים גרפי פיזור של זוגות של פיצ'רים ביניהם יש קורלציה גבוה. מיד לאחר מכן, הסתכלנו על ההיסטוגרמות של הפיצ'רים המספריים, בעיקר כדי להבין את ההתפלגות שלהם (נספח 6).

בסוף החלק של האקספלורציה, ביצענו השוואה בין התצפיות בהן הלייבל הוא '0' לתצפיות בהן הוא '1', תוך הסתכלות על הממוצע המתקבל בכל פיצ'ר במצבים אלו. ראינו בפיצ'רים 'device', 'Region' ו-'B' שהממוצעים כמעט זהים. כתוצאה מכך, חקרנו אותם יותר לעומק, ובעזרת ויזואליזציות (נספחים 7-9), ראינו שהם מתנהגים בצורה דומה בין אם המשתמש קנה או לא. למרות שלא נראה שיהיה להם השפעה על חיזוי הלייבל ולאחר התלבטויות, החלטנו בשלב זה להשאיר את הפיצ'רים הנ"ל ולבחון את חשיבותם בהמשך הפרויקט.

חלק שני – עיבוד מקדים

לכל אורך החלק הנ"ל, התייחסנו לכל פיצ'ר בנפרד: ביצענו חקירה נוספת, השלמת ערכים חסרים, הסרת ערכים חריגים, עריכות נקודתיות (לדוגמה, להסיר תווים לא רלוונטיים) והחלת העיבוד המקדים הרלוונטי לסט ה-test. את שלב העיבוד המקדים התחלנו ביצירת מס' פונקציות שונות למילוי ערכים חסרים: מילוי לפי הערך הממוצע בעמודה, החציון בעמודה, הערך השכיח בעמודה, החציון של הערכים לפני ואחרי כל ערך חסר, מילוי עמודה אחת בהסתמך על ערכים של עמודה אחרת ויצירת ערך (קטגוריאלי) חדש בשם 'unknown'.

להלן הניתוח שביצענו על הפיצ'רים:

- 'num_of_admin_pages' & 'admin_page_duration' – בהסתמך על נספח 6, ראינו שהערך הנפוץ ביותר ב-'num_of_admin_pages' מופיע בכ-45% מהפעמים. החלטנו למלא כל ערך חסר בפיצ'ר זה על ידי החציון של התצפית שקודמת לו והתצפית שבאה אחריו. כך, נקבל מידע מדויק יותר על ה-train. לאחר שמילאנו את הערכים החסרים ב-train, השתמשנו בחציון של כלל הערכים ומילאנו בעזרתו את הערכים החסרים בעמודה זו ב-test. החלטנו למלא את הערכים החסרים בעמודת

'admin_page_duration' ב-train על סמך הערכים של עמודת 'num_of_admin_pages' בעקבות הקורלציה הגבוהה שקיימת ביניהן (נספחים 2 ו-3) כדי לקבל ערכים מדויקים יותר ב-train. את הערכים החסרים ב-test של עמודה זו מילאנו על סמך הממוצע ב-train. לבסוף, הסרנו ערכים חריגים (Outliers) מהפיצ'רים הנ"ל בהסתמך על ה-scatter plot שלהם (נספח 3).

- 'num_of_info_pages' & 'info_page_duration' – הערך הנפוץ ביותר ב-'num_of_info_pages' מופיע ביותר מ-80% מהתצפיות (נספח 6). לכן, מילאנו את הערכים החסרים של עמודה זאת בעזרתו. הערכים בעמודת 'info_page_duration' הופיעו כ-'XXX minutes'. ראשית, הסרנו את המלל כך שהערכים יהיו רק מספרים וכך יכולנו להמיר את ערכי העמודה לסוג float. גם בעמודה זו מילאנו את הערכים החסרים על סמך הערך הנפוץ (נספח 10). מצאנו שבין הפיצ'רים הנ"ל קיימת קורלציה גבוה גם כן, ולכן הצגנו על בסיסם scatter plot (נספח 11) שבעזרתו החלטנו איך לטפל בנתונים החריגים.
- 'num_of_product_pages' & product_page_duration' – לפי נספח 6, לא מצאנו ערך שכיח בצורה מוחלטת ב-'num_of_product_pages'. כתוצאה מכך, החלטנו למלא כל ערך חסר בפיצ'ר זה על ידי החציון של התצפית שקודמת לו והתצפית שבאה אחריו על מנת להגיע לדיוק גבוה יותר בסט ה-train. לאחר שמילאנו את הערכים החסרים ב-train, השתמשנו בחציון של כלל הערכים ובעזרתו מילאנו את הערכים החסרים בעמודה זו ב-test. בעמודת 'product_page_duration' הערכים הופיעו כ-'XXX minutes'. הסרנו את הטקסט המיותר כך שהערכים יהיו רק מספרים. כתוצאה מכך, יכולנו להמיר את ערכי העמודה לסוג float. מילאנו את הערכים החסרים בעמודת 'product_page_duration' ב-train על סמך הערכים של עמודת 'num_of_product_pages' בעקבות הקורלציה הגבוהה שקיימת ביניהן (נספחים 2 ו-12). את הערכים החסרים ב-test של עמודה זו מילאנו על סמך הממוצע ב-train. לבסוף, הסרנו ערכים חריגים מהפיצ'רים הנ"ל בהסתמך על ה-scatter plot שלהם (נספח 12).
- 'total_duration' – בדקנו האם ישנם הבדלים בערכי הפיצ'רים האחרים, כאשר ערכי הפיצ'ר הנ"ל הם ריקים וכאשר הם אינם ריקים. ראינו שהממוצעים בעמודות האחרות דומים בשני המצבים. זאת הסיבה שבחרנו להשלים בעמודה זו את הערכים החסרים על סמך הממוצע של הערכים הקיימים ובעזרת היסטוגרמת צפיפות (נספח 13) החלטנו אילו ערכים נחשבים כחריגים אותם מחקנו.
- 'BounceRates' & 'ExitRates' – הערכים בפיצ'רים הנ"ל נעים בין 0 ל-0.2. מכיוון שבשניהם היו מעט ערכים חסרים, שינינו אותם לערכים הנפוצים ביותר. גם כאן, בהסתמך על גרף הפיזור (נספח 5) מחקנו נתונים חריגים. בהמשך הפרויקט, נשקול ליצור פיצ'ר חדש שהוא קומבינציה לינארית של הפיצ'רים הנ"ל בעקבות הקורלציה הגבוהה ביניהם (נספח 2).
- 'PageValues & 'closeness_to_holiday' – מצד אחד, ראינו שהערכים בעמודת 'PageValues' נעים בין 0 ל-361. הערך שהופיע ביותר משלושת רבעי מהתצפיות היה 0. מנגד, פיצ'ר זה מדבר על ערך כספי ולכן לא החשבנו את הערכים השונים כחריגים. מילאנו את הערכים החסרים בעזרת הערך הנפוץ ביותר. גם בעמודת 'closenss_to_holiday' הערך '0' הופיע מעל 75% מהפעמים. בעמודה זו הערכים נעים במרחקים קבועים בין 0 ל-1 ולכן החלטנו לא להוריד תצפיות. כתוצאה מכך, גם בעמודה זה הערך הנפוץ ביותר היה הדרך למלא את הערכים החסרים.
- 'Month' – מכיוון שהיו מעט ערכים חסרים בפיצ'ר זה, השתמשנו בערך הנפוץ כדי למלא אותם (נספח 14). לאחר מכן, מכיוון שמדובר בפיצ'ר קטגוריאלי נאלצנו ליצור dummy variables.
- 'device' – יצרנו ערך חדש – 'unknown' איתו מילאנו את הערכים החסרים בעמודה. מכיוון שמדובר בפיצ'ר קטגוריאלי, יצרנו dummy variables.
- 'internet_browser' – גם בפיצ'ר קטגוריאלי זה שינינו את הערכים החסרים לערך 'unknown'. יתר על כן, החלטנו לקבץ את כל סוגי הערכים (כלל גרסאות הדפדפנים) למס' מצומצם של ערכים (לפי סוג הדפדפן). בגלל הסוג הקטגוריאלי של העמודה יצרנו dummy variables.
- 'Region' – גם בעמודה זו ראינו שיש מעט ערכים חסרים. השתמשנו בערך הנפוץ כדי למלא את הערכים החסרים (נספח 15). לאחר מכן, יצרנו dummy variables.
- 'user_type' – 85% מהערכים בפיצ'ר קטגוריאלי זה היו 'Returning_visitor' ולכן החלטנו להשתמש בו כדי למלא את הערכים החסרים, בטרם יצרנו dummy variables.
- 'Weekend' – בדומה לפיצ'רים אחרים, גם בפיצ'ר זה מילאנו את הערכים החסרים בעזרת הערך הנפוץ. לאחר מכן המרנו את סוג העמודה מבוליאני למספרי.
- 'A' – תחילה, מילאנו את הערכים החסרים בערך חדש – 'unknown'. מיד לאחר מכן, הבחנו שלערך 'c_20' יש גרסאות רבות, אז החלטנו להפוך אותן ל-'c_20'. בעמודה זו, התלבטנו לא מעט מהם ערכים חריגים. קבענו threshold של 30, כך שערכים שהופיעו ב-train פחות מה-threshold יימחקו. מכיוון

שאין הלימה מוחלטת בין הערכים ב-train לאלו שב-test, הגדרנו שהערכים החסרים ב-test ישונו ל-'unknown'. כמו שביצענו בסוף העבודה על כל פיצ'ר קטגוריאלי, גם בפיצ'ר זה יצרנו dummy variables.

- 'B' – לפי נספח 6, ערכי עמודה זו הינם היחידים שמתפלגים בצורה שדומה להתפלגות נורמלית. כתוצאה מכך, מילאנו את הערכים החסרים בממוצע של ערכי העמודה. בשלב הבא, השתמשנו ב-boxplot כדי לאתר ולהסיר ערכים חריגים – ערכים מעל/מתחת לטווח הבין-רבעוני (נספח 16).
- 'C' – מהויזואליציה להתפלגות הערכים בפיצ'ר זה לפי ערכי הלייבל שביצענו (נספח 17), הסקנו שהערכים התפלגו בצורה כמעט שיוויונית ללא קשר אם המשתמש ביצע רכישה או לא. לכן, במחשבה ראשונה בחרנו למלא בצורה שיוויונית את הערכים החסרים בצורה אחידה על סמך ערכי הפיצ'ר. לאחר שהשקענו מחשבה נוספת, החלטנו שהעמודה לא תשפיע על חיזוי הלייבל, ועל כן החלטנו להסירה.

לאחר שסיימנו לעבור על כלל העמודות, השלב הבא היה ליצור פיצ'ר חדש בשם 'MixedRates', שהינו קומבינציה לינארית של 'ExitRates' ושל 'BounceRates'. זאת מכיוון שהקורלציה בין הפיצ'רים הללו הייתה ב-0.9. מכיוון של-'ExitRates' יש קורלציה גבוהה יותר (בערך מוחלט) עם הלייבל, המשקולת שלו בפיצ'ר החדש נקבעה על ידינו ל-0.6, כאשר המשקולת של 'BounceRates' הינה 0.4.

בסך הכל, הפיצ'רים שהסרנו במהלך שלב זה הם 'ExitRates', 'C', 'id', 'D' ו-'BounceRates'.
יתר על כן, דאגנו לשים את עמודת ה-'purchase' אחרי כל העמודות.

כעת, כדי לוודא שהפיצ'רים תורמים בצורה שווה למודלים שנריץ, עלינו לבצע נרמול לדאטה. כאמור, כמעט כל הפיצ'רים לא מתפלגים בצורה נורמלית. כתוצאה מכך, בחרנו לבצע scale של הדאטה בעזרת MinMaxScaler ולא לבצע נרמול בעזרת פונקציית נרמול הדורשת התפלגות נורמלית. פונקציית ה-MinMaxScaler תשנה את הערכים בכל פיצ'ר כך שהם ינועו בטווח בין 0 ל-1.

פיצלנו את ה-train ל-2 חלקים: train ו-validation. בעזרת פונקציית train_test_split, השארנו ב-validation 20% מהדאטה.

רגע לפני שניגש למודלים, עלינו לתת את הדעת על בעיית המימדיות. בשלב זה, צמחנו מ-21 פיצ'רים ל-63. אין ספק שהמימדיות של הבעיה גדלה מאוד ביחס למצב ההתחלתי. באופן כללי, הפחתת מימדיות הינה תהליך חיוני כדי להילחם ב-overfitting. ה-bias-variance trade-off משפע ממספר הפיצ'רים בדאטה. ככל שיהיו יותר פיצ'רים, כך המורכבות גדלה וגם השונות ביחד איתה, בזמן שה-bias קטן. מימדיות גדולה עלולה להשפיע בצורה שלילית על המודלים ולהוביל ל-overfitting. עלינו להימנע ממצב זה. נרצה למצוא את האיזון – להגיע למצב שהן השונות והן ה-bias לא גבוהים ולא נמוכים מדי.

במהלך החיפוש אחר הדרך העדיפה להורדת המימדים, התחלנו עם PCA אך לא השתמשנו בה. הדרך שבחרנו בסוף להורדת המימדים היא שימוש בפונקציית selectKBest עם פרמטר score_func = chi2. זה הוביל לתוצאות גבוהות משל PCA. selectKBest זו פונקציה מתוך החבילה sklearn.feature_selection. לפונקציה מגדירים מס' מטרה (K) שמהווה את מס' הפיצ'רים הרצוי. הפונקציה מחזירה אך ורק את K הפיצ'רים בדאטה עם הציון הגבוה ביותר. על סמך מה נקבע הציון? למשל, הגדרת הציון לפי 'chi2' יחשב לפי מבחן כי בריבוע תלות בין משתנים סטוכסטיים, ובחירה בו תסיר את הפיצ'רים שהכי צפויים להיות חסרי תלות ב-class מסוים וכתוצאה מכך לא רלוונטיים לסיווג הלייבל. ניסינו מגוון רחב של מס' פיצ'רים רצוי (הפרמטר K). לאחר מספר ניסיונות ידניים, מצאנו ש-K=30 הינו האידיאלי למודלים שהרצנו.

חלק שלישי – הרצת מודלים

המודלים שבחרנו להריץ:

מודלים ראשוניים: KNN ו-Logistic Regression.

מודלים מתקדמים: Multi-Layer Perceptron (ANN) ו-Random Forest.

בכל אחד מהמודלים השתמשנו ב-GridSearch על מנת למצוא את הערכים האופטימליים עבור ההיפר פרמטרים המרכזיים של המודל.

חלק רביעי – הערכת מודלים

- את המודלים הערכנו לפי ה-train לאחר הפיצול ואת ה-validation נשמור להמשך משום שהוא מדמה עבורנו את ה-test. הערכנו את כל המודלים באמצעות K-Fold Cross Validation על ידי שימוש בסט ה-train העדכני לאחר הפיצול. הגדרנו את K=12 (K – מס' הפולדים). הצגנו את ה-roc curve של כל איטרציה ולאחר מכן הצגנו בקו כהה יותר באותו גרף את ה-roc הממוצע.
  ביצענו את התהליך פעמיים (נספח 18), פעם אחת על ה-validation שמתקבל מה- K-Fold Cross Validation ופעם נוספת על ה-train המתקבל מה-K-Fold Cross Validation. עשינו זאת בשביל לקבל את פערי הביצועים בין המודלים השונים על ה-train ועל ה-validation ובנוסף, כדי להבין האם המודל הוא Overfitted.
  את ה-mean roc הגבוה ביתר קיבלנו בהרצת המודל **Random Forest** ואילו את הנמוך ביותר קיבלנו בהרצת המודל KNN. התוצאה השנייה בטיבה הייתה של מודל ה-MLP ולאחריו של מודל Logistic Regression.

- פערי הביצועים הקטנים ביותר בין הרצת המודל על ה-train לבין הרצת המודל על ה-validation היו בסדר הבא (מהפער הקטן ביותר לפער הגדול ביותר): Logistic Regression, MLP, Random Forest ו-KNN.

  הפער במודל ה-KNN היה גדול בצורה משמעותית (כ-24%), דבר אשר עלול להעיד על overfitting. הפערים ב-Logistic Regression וב-MLP עמדו על כ-1%-2% וב-Random Forest הפער הסתכם לכ-6%. במבט ראשוני, נעדיף את מודל ה-Logistic Regression / MLP, אולם מודל ה-Random Forest קיבל את ה-mean roc הגבוה ביותר ופער ביצוע של 6% הינו לגיטימי בעינינו. אנחנו נעדיף את התוצאה ביותר בהינתן פער הביצוע הנ"ל ולכן **בחרנו במודל ה-Random Forest.**

  לבסוף, הרצנו את המודל הנבחר על סט ה-train וה-validation (לאחר הפיצול) והצגנו את ה- roc curve של כל אחד מהם (נספח 19).

  על מנת להגדיל את יכולת ההכללה של המודל ניתחנו כל פיצ'ר – הסתכלנו על הקורלציות שלו, על האופן בו הוא מתפלג. יתרה מכך, ביצענו הורדת מימדים – הורדנו את כמות הפיצ'רים באופן סביר (כחצי ממספר הפיצ'רים) והשארנו את הפיצ'רים להם יכולת חיזוי הלייבל בצורה הטובה ביותר.

- אימנו את המודל הנבחר לפי ההיפר-פרמטרים שמצאנו על סט ה-train ובנינו confusion matrix (נספח 20) על סט ה-validation. פירוט תאי המטריצה וייצוגם:
  TP – הלקוח רכש והמודל חזה זאת.
  TN – הלקוח לא רכש והמודל חזה זאת.
  FP – הלקוח לא רכש אך המודל חזה שכן רכש.
  FN – הלקוח רכש אך המודל חזה שלא רכש.

חלק חמישי – ביצוע פרדיקציה

לבסוף, אימנו את המודל הנבחר באמצעות סט הנתונים המלא (ה-train וה-validation). עשינו ניבוי על סט ה-test. לעת סיום, יצרנו pipeline אשר מקבל סט נתונים של train ו-test, מבצע את כל הליך העיבוד המקדים, הורדת ממדים, נרמול, אימון המודל הנבחר וביצוע תחזיות.

סיכום

לסיכום, עברנו תהליך ארוך, מלמד ומאתגר. הפרויקט הינו מעגלי אשר מתחילים בשלד, חוקרים לעומק, מגלים כיוונים חדשים וחוזרים לשנות, לטייב ולשפר את כל החלקים. דרך הפעולה שלנו הייתה להתייחס לכל פיצ'ר בנפרד ועליו להחיל את העיבוד המקדים הרלוונטי לו. מסקנות מרכזיות:

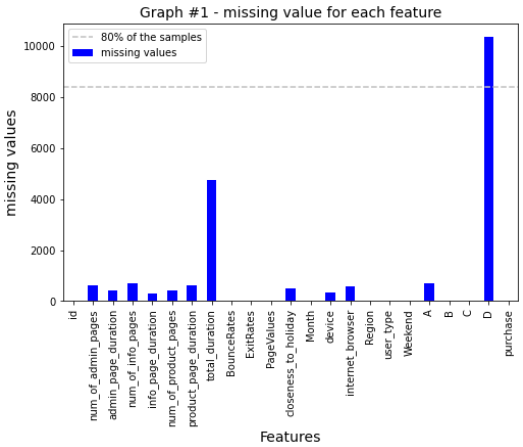
- בשלב יחסית מוקדם של הפרויקט, נוכחנו לדעת שכל פעולה אשר עושים על הדאטה עשויה להשפיע באופן ניכר על חיזוי המודלים.
- ניסינו דרכים שונים להוריד מימדים. בראש מעיינינו היה ה-Bias-Variance Tradeoff. לאחר העיבוד המקדים, הגענו ל-63 פיצ'רים אשר עלולים להוביל ל-overfitting. לכן, ניסינו לבחון אופציות שונות להורדת המימדים. גילינו, שהורדה משמעותי של המימדים (למשל ל-10) פגעה בתוצאות המודלים ומנגד, הורדה זניחה פגעה גם כן. כתוצאה מכך, בחרנו במודל עם 30 פיצ'רים אשר מאזן את ה-Bias וה-Variance ומוביל לתוצאות טובות במודלים.
- באופן כללי, המודלים המורכבים ידעו לחזות בצורה טובה יותר את הלייבל מאשר המודלים הפשוטים ובפרט, מודל ה-Random Forest שבו בחרנו.

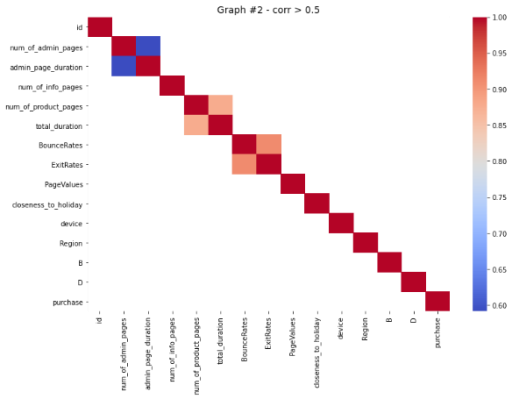
נספח אחראיות כל שותף

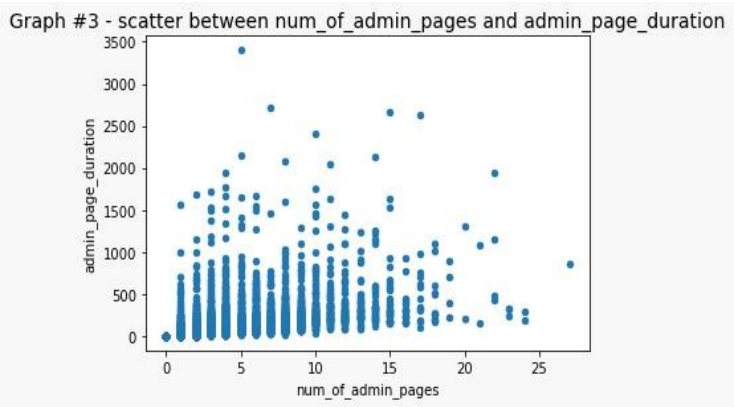
מתחילתו ועד סופו של הפרויקט, ביצענו את התהליך ביחד – פיזית או ע"י פגישות ב-zoom. במפגשים הראשונים ניסינו להבין את הדאטה, מה כל פיצ'ר אומר, מה נדרש מאיתנו לבצע וקיימנו סיעור מוחות מתמשך כדי לבחור את דרך הפעולה האידיאלית עבורנו. לאורך כל התהליך, כל שותף לקח חלק שווה בקידוד ובגיבוש רעיונות והחלטות בפרויקט.

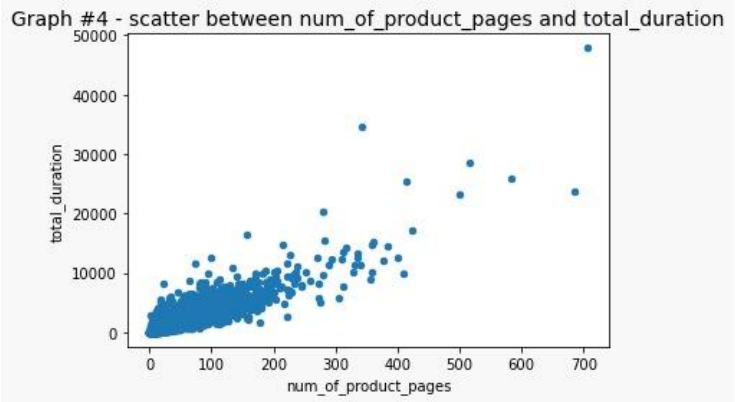
לסיום, הפרויקט היה מעשיר, מאתגר, מלמד ומהנה. אנחנו מרגישים שרכשנו כלים הן תיאורטיים והן פרקטיים בנושא למידת מכונה.

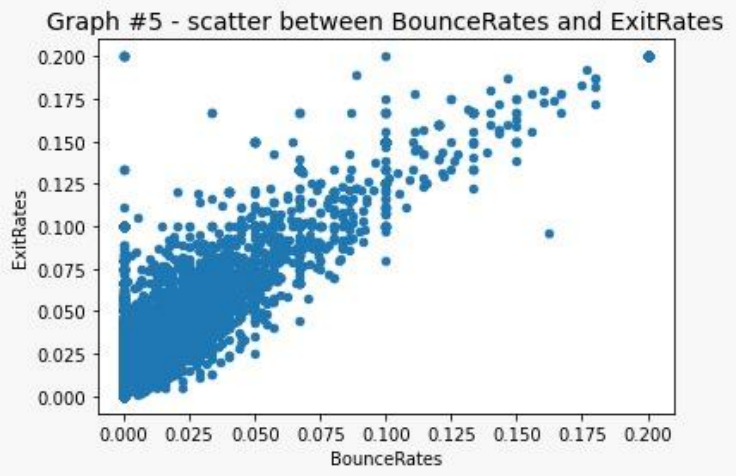
נספחים


Graph #1 - missing value for each feature
80% of the samples
missing values
missing values
Features
id
num_of_admin_pages
admin_page_duration
num_of_info_pages
info_page_duration
num_of_product_pages
product_page_duration
total_duration
BounceRates
ExitRates
PageValues
closeness_to_holiday
Month
device
internet_browser
Region
user_type
Weekend
A
B
C
D
purchase

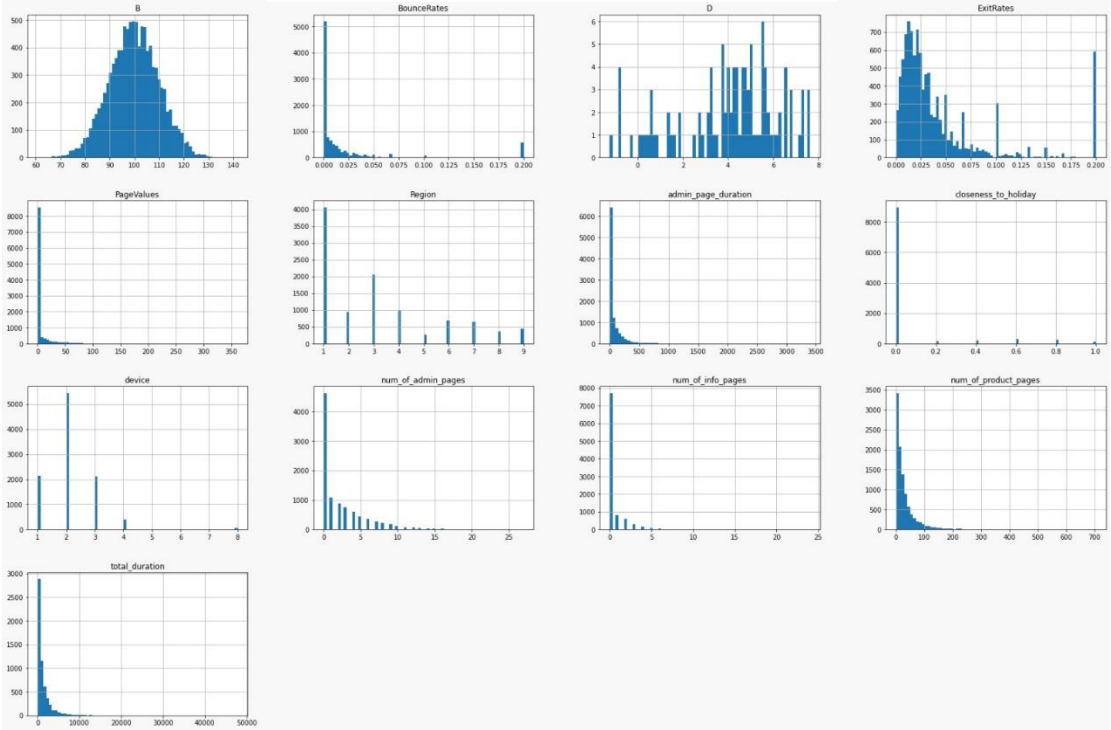

Graph #2 - corr > 0.5

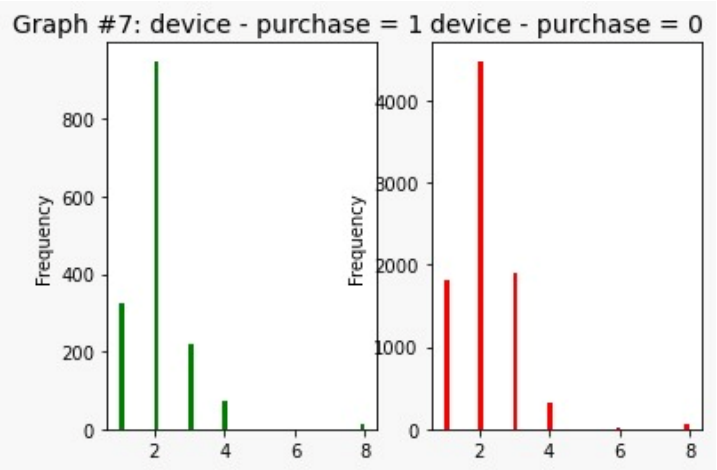

Graph #3 - scatter between num_of_admin_pages and admin_page_duration
admin_page_duration
num_of_admin_pages

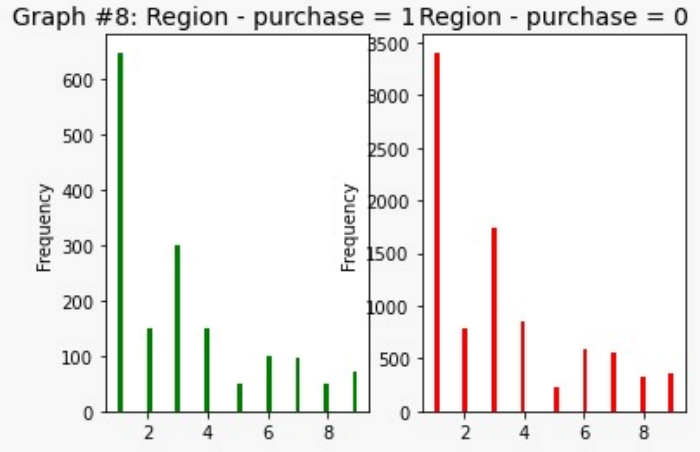

Graph #4 - scatter between num_of_product_pages and total_duration
total_duration
num_of_product_pages

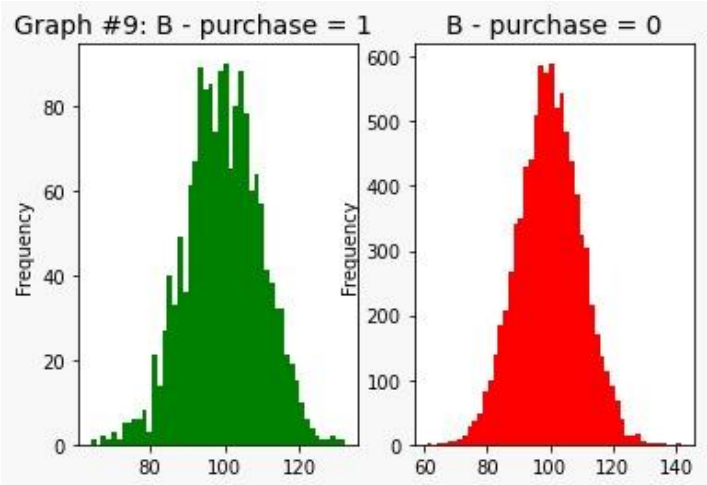

Graph #5 - scatter between BounceRates and ExitRates
ExitRates
BounceRates

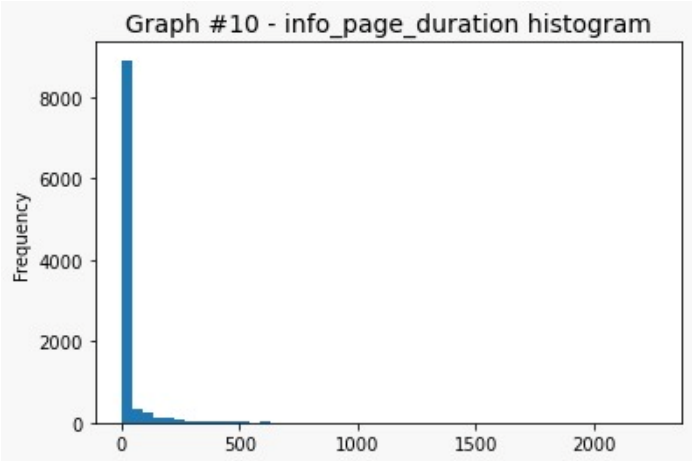
Graph #6 – plotting the numeric features:

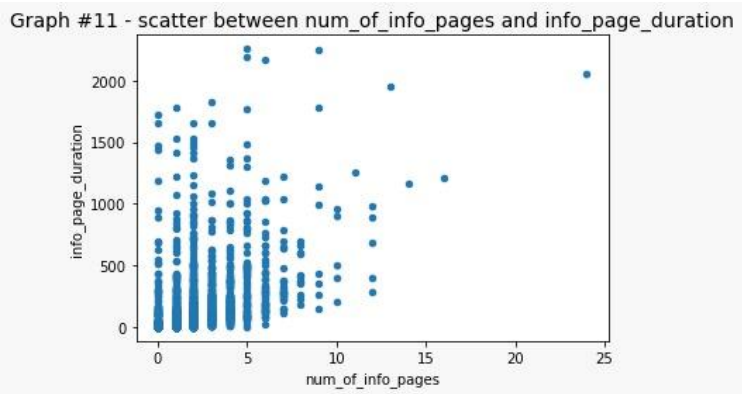


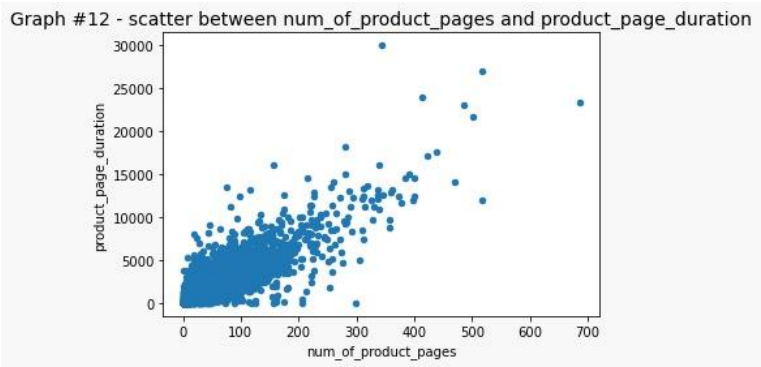


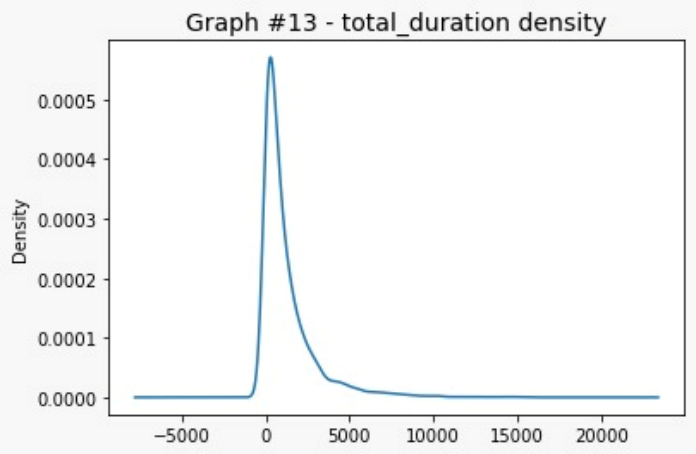




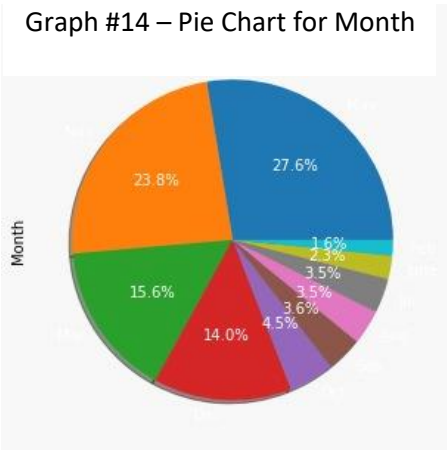




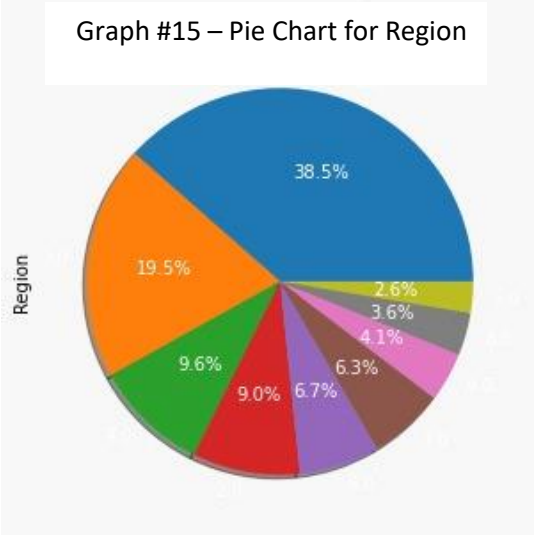


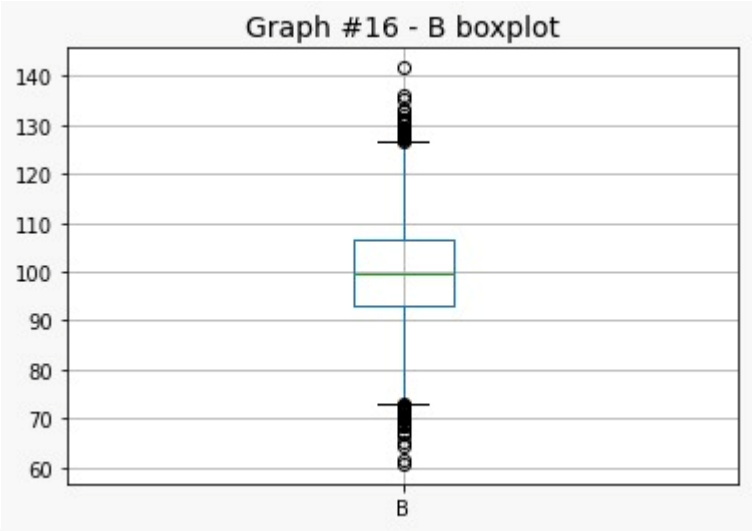


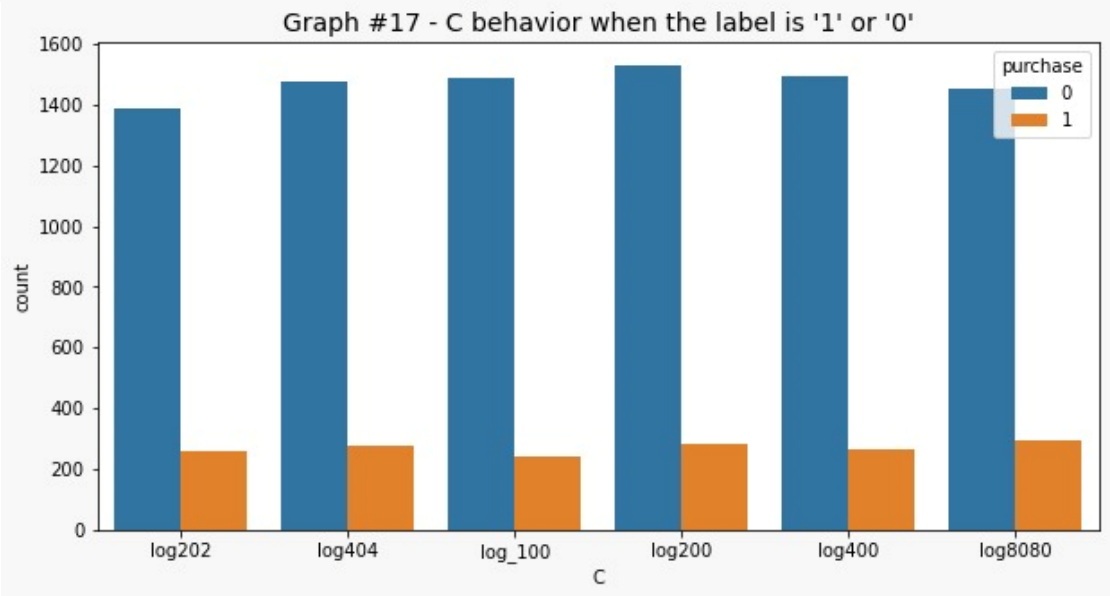
Graph #14 – Pie Chart for Month



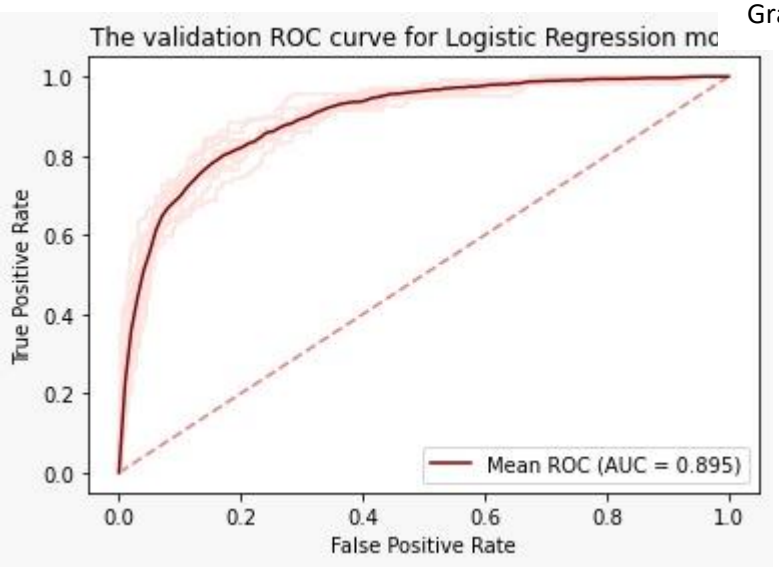
Graph #15 – Pie Chart for Region

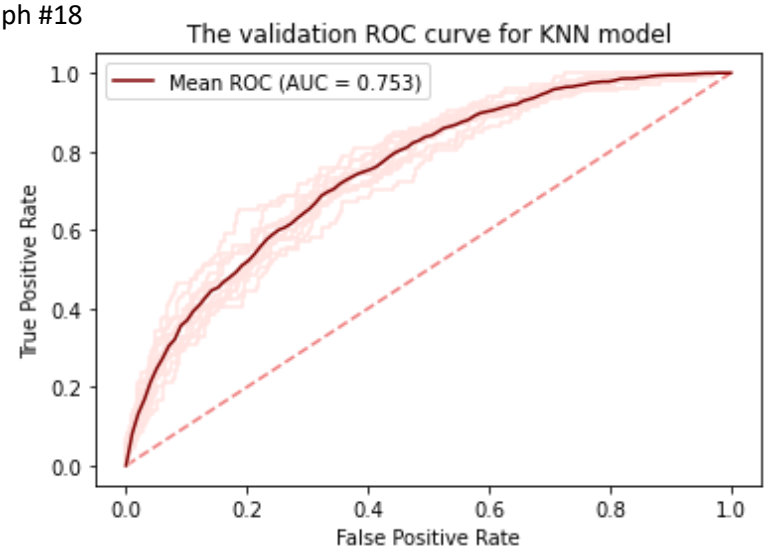


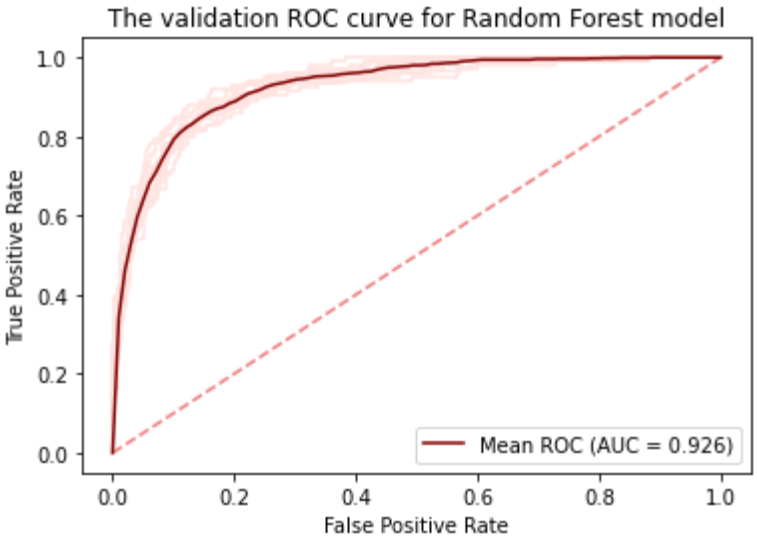


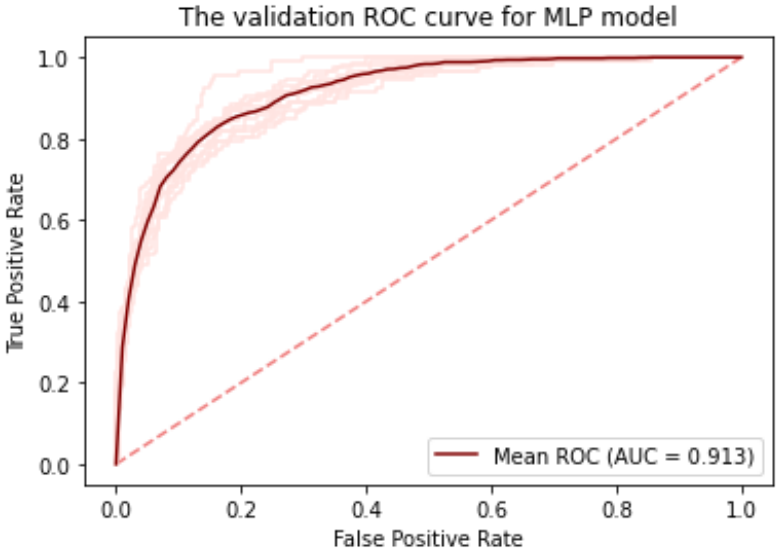


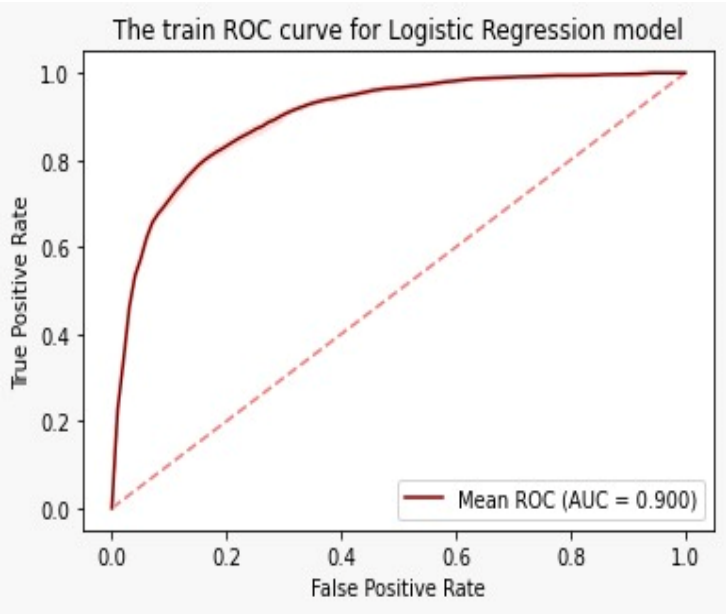
Graph #18

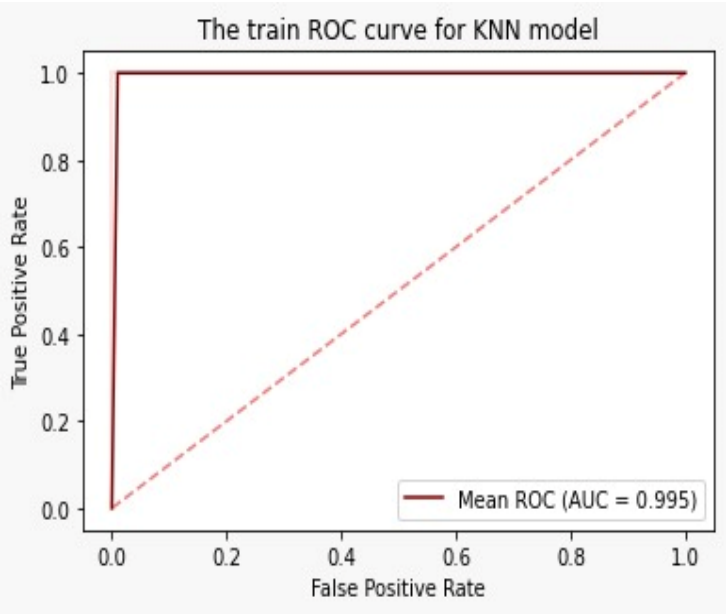

The validation ROC curve for Logistic Regression mo
Mean ROC (AUC = 0.895)
True Positive Rate
False Positive Rate

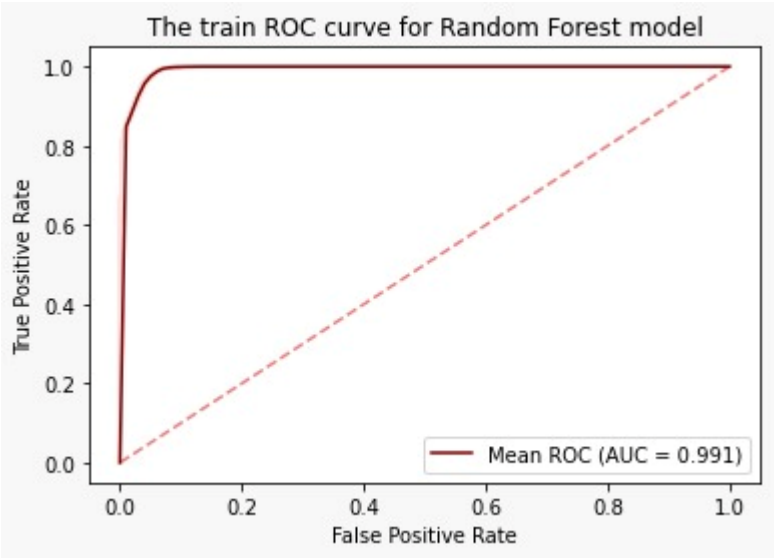

The validation ROC curve for KNN model
Mean ROC (AUC = 0.753)
True Positive Rate
False Positive Rate

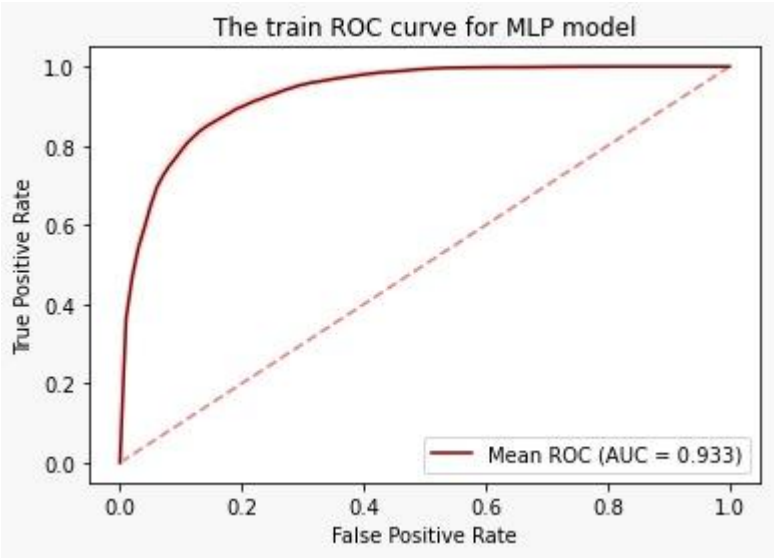

The validation ROC curve for Random Forest model
Mean ROC (AUC = 0.926)
True Positive Rate
False Positive Rate


The validation ROC curve for MLP model
Mean ROC (AUC = 0.913)
True Positive Rate
False Positive Rate

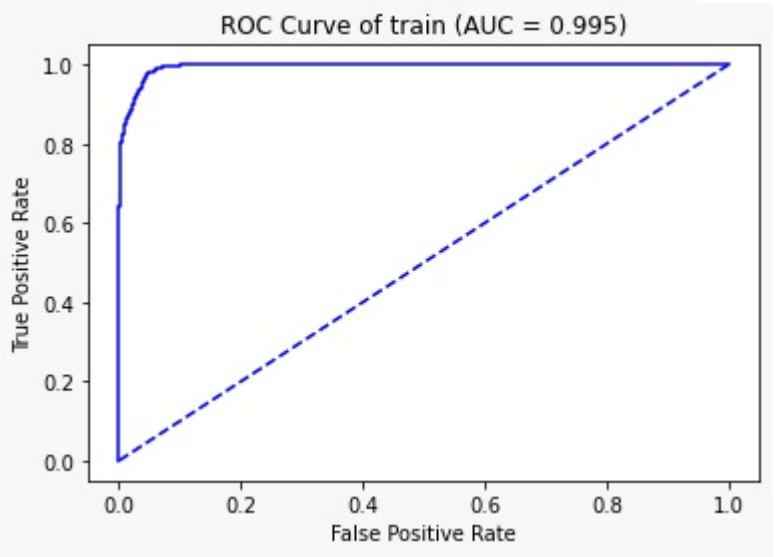

The train ROC curve for Logistic Regression model
Mean ROC (AUC = 0.900)
True Positive Rate
False Positive Rate

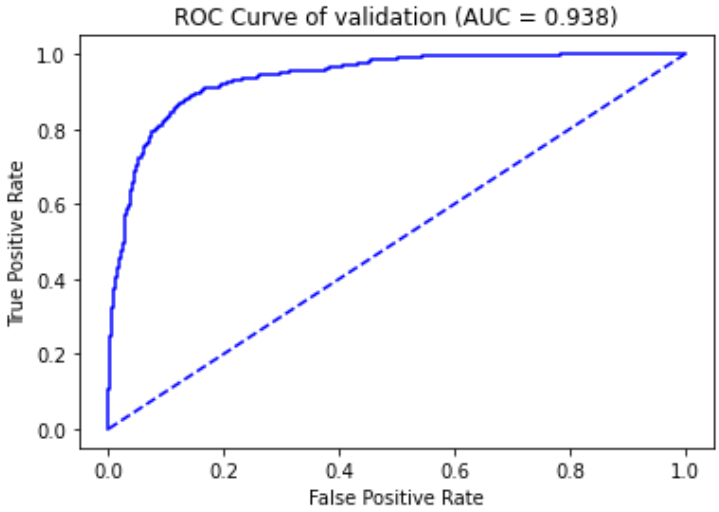

The train ROC curve for KNN model
Mean ROC (AUC = 0.995)
True Positive Rate
False Positive Rate





Graph #19





Graph #20