15.05.2022

אליהו אטין

רון קנדלשיין

כריית מידע - חלק ב'

מטרת העל:

יצירת מודל אשר יסייע לדרג הניהולי במחלקת השיווק להחליט לאילו לקוחות פוטנציאלית לפנות.

הקדמה:

שיווק ע"י קמפיינים ממוקדים הינה אחת הדרכים היותר אפקטיביות להגיע לאנשים בכדי שירכשו מוצרים מסוימים.
כפי שאנחנו חווים בעצמנו בעידן הרשתות החברתיות, הקמפיינים שחברות מבצעות מכוונות עבור לקוחות פוטנציאלים המאופיינים ע"י אותה חברה כרוכשים פוטנציאליים.
היכולת לפתח מודל אשר ידע לאפיין את אותם לקוחות על בסיס מידע קיים יכול למקסם את רווחי החברה למול עלות הקמפיינים השונים.

הערות:

1. מכיוון שהיה לנו מספר בעיות בהגשת חלק א' ערכנו את המידע (train data) וכמובן ביצענו עיבוד מתאים לtest data. על כן, לאחר עיון בהערות ומחשבה נוספת תוך כדי התהליך - שינינו את ייצוגו של חלק מהמידע. לשם הפשטות, והתייחסות למה שרלוונטי לתרגיל הנוכחי – נתייחס לכך שביצענו את חלק א' כפי במוצג להלן. את חובת עיבוד מידע (test and train) צירפנו בחוברת נפרדת לחוברת המתארת את בניית המודל. בנוסף, צירפנו גם את קבצי ה-csv הסופיים של תהליך העיבוד.
2. <u>נציין את הקבצים המצורפים לעבודה:</u>
   - Part2_Train_(Part1-catchup).ipynb – ההשלמה שביצענו לחוסרים מחלק 1.
   - Part2_Test.ipynb – עיבוד test data
   - Classifiers.ipynb – קובץ בניית המודל ואימונו עם **הגרסא הראשונית** של המידע (הסבר בהמשך)
   - Classifiers_2nd_Iteration.ipynb – קובץ בניית המודל ואימונו עם **הגרסא הסופית** של המידע
   - Campain_Predict_v1_ada.csv – קובץ ה-TEST אותו חזינו בעזרת המסווג שנבחר
   - Data_Test_V3.csv – קובץ ה-TEST לאחר עיבוד שביצענו (אותו אנו טוענים)
   - Data_Train_V3.csv – קובץ עם מידע האימון אותו אנו טוענים לעבודה.

1. עיבוד מקדים

בפרק זה נציג את העיבוד המקדים שביצענו על מנת לייעל את המודל שיצרנו. נציג את הצעדים שביצענו ואת ההצדקה לביצועם. לאימון המודל עשינו שימוש במידע אודות 1665 לקוחות.

א. סיווג המידע

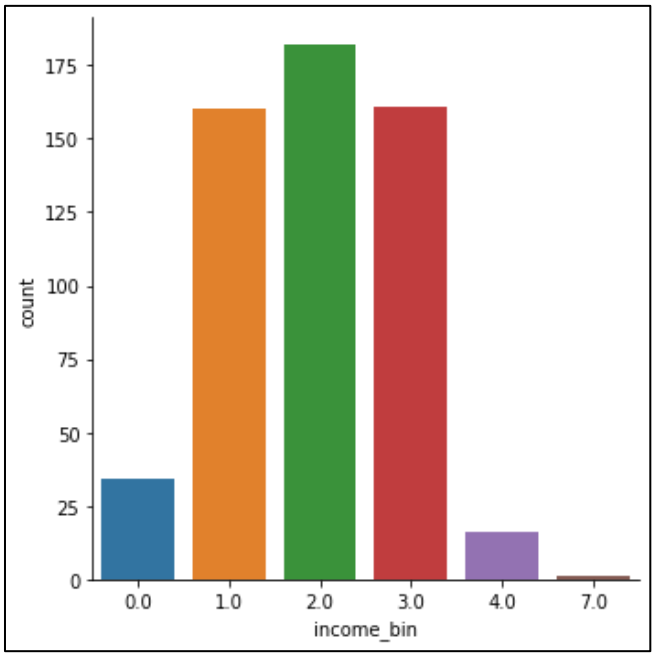
על מנת לייעל את עבודת המודל, במסגרת העיבוד המקדים, ביצענו סינון של מידע לא רלוונטי לפי מסקנות שהגענו אליהן בחלק א׳ ושינינו ייצוג של מידע מסוים.
בכדי ליצור מידע אשר מתאים לתהליך הסיווג העברנו את המידע בסינון ראשוני, אפיינו את המידע ראינו התפלגויות עבור כל הערכים ועל סמך זה את הסיווגים הבאים:

1. **Education:**

מידע זה הינו מידע קטגורי זה אשר הופיע בצורה של מחרוזת. המרנו לערך מספרי.

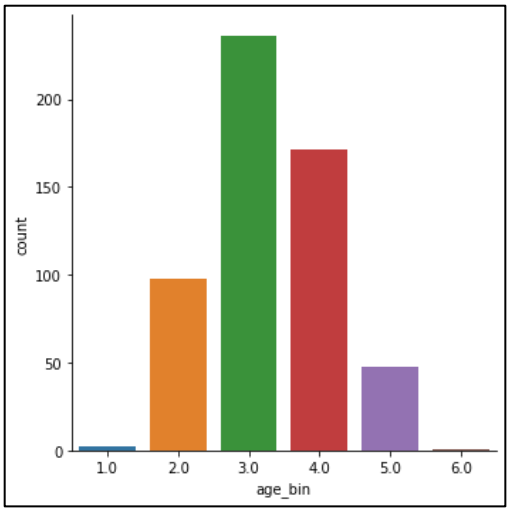
2. **Income:**

את המידע הרציף הנ״ל המרנו למידע קטגורי. הבחנו שמרבית המשכורות הינן בטווח של 0 עד מאה אלף ₪ (באופן דומה למידע האימון). על כן ביצענו את החלוקה באופן שנקבל עד כמה שאפשר חלוקה אחידה בין הקבוצות.



איור 1: חלוקת קטגורית של ההכנסת הלקוחות

3. Age:
את המידע הרציף הנ״ל המרנו למידע קטגורי. באופן זהה להכנסה, ניסינו ע״י ביצוע equal-length partition לחלק לקבוצות כך שהחלוקה תהיה אחידה.



איור 2 : חלוקת קטגורית של גיל הלקוחות

4. **Status** :
המרנו את המידע הקטגורי הנ״ל למידע מספרי לטובת המודל.

5. Mnt_x:
עבור עמודות הנוגעות לכמה לקוח בזבז על סוג מסוים של מוצרים ביצענו min-max normalization.

6. Response_Campain_x:
עבור עמודות קיבלנו החלטה לאחד ע״י האופרטור OR לכדי עמודה אחת המסמלת :
האם לקוח הגיב לאחד מ-5 הקמפיינים האחרונים שבוצעו.
נרחיב עוד על כך בהמשך.

2. חלוקת המידע

א. **חלוקת מידע האימון:**
חילקנו את המידע שלנו למאגר train ומאגר validation כדרוש. היחס שבחרנו להשתמש הינו 70% עבור train ו-30% עבור validation .

ב. **cross validation:**
בהרצת האלגוריתמים נעשה שימוש בשיטת cross validation כנדרש.
על מנת לחלק את training data ל train ו- validation , נחלק מראש את כל הtraining set למספר חלקים, וכל פעם חלק אחר יהיה סט הוולידציה. נעשה זאת על ידי שימוש בפונקציה KFold.

ג. במהלך העבודה שלנו ראינו לנכון לבצע 2 הרצות שונות ונסביר מה הרציונל העומד מאחורי כל הרצה:

- **הרצה 1**
עמודת היעד שלנו, הינה איחוד (OR) של כלל הקמפיינים שנעשו עבור אותו הלקוח.
בעצם הינה מייצגת האם הלקוח הגיב לאחד מהקמפיינים אשר בוצעו (ישנו יעוד ל-5 ישנים ואחד נוכחי). בכך רצינו לבדוק אילו לקוחות פוטנציאליים לתגובה לקמפיין ובעצם בעזרת המודל לדעת האם ישנם מאפיינים משותפים לאותם לקוחות.
- **הרצה 2:**
עמודת היעד שלנו הייתה תגובתו של הלקוח לקמפיין האחרון שנערך – כפי שהנחנו בחלק א'. כלומר, נרצה שמודל יאפיין מהם המאפיינים של לקוח אשר הגיב לקמפיין האחרון.

מטרת הפיצול של 2 הרצות אלו הינה 2 נקודות מבט שונות אודות מטרת העל:
1. הסתכלות על המידע כאשר אין חשיבות לנקודת הזמן הנוכחית – נרצה לנתח את התגובתיות של כל לקוח רשום במהלך כלל תקופת הקנייה שלו. וע"י זאת לבנות מודל אשר ידע להצביע על פרמטרים ספציפיים שיעזרו להצביע על לקוחות פוטנציאלים וקמפיינים אפקטיביים.
2. התייחסות לזמן הנוכחי על מנת למקסם את יעילותו של הקמפיין הבא. היתרון של גישה זאת על הקודמת הינה שלדעתנו צמצום עמודת היעד לכדי תגובה לקמפיין האחרון תשיג תוצאות טובות יותר בטווח הקצר.

3. <u>סיווג המידע</u>

tבכדי למצוא מסווג מתאים לצורך שלנו עשינו שימוש במספר מסווגים שונים.
עבו כל מסווג ביצענו בדיקות בכדי לבדוק את רמת הדיוק בחיזוי שלו. החלוקה של כל מסווג הייתה זהה למצוין.

<u>רקע</u>

ראשית מעט רקע אוות הפרמטרים של המסווגים:

- **Sensitivity**
  מתאר כמה מה- positive חזינו בהצלחה כ-positive.
  נוסחא:

$$\frac{TP}{TP + FN}$$

- **Accuracy**
  פרמטר זה מצביע כמה חזינו נכון – ללא קשר האם TNאו TP.
  כלומר מתאר את הדיוק הכללי של המודל. כפי שניתן לראות בנוסחה:

$$\frac{TP + TN}{N + P}$$

- **Precision**
  מודד בכמה מתוך כל שחזינו כ-positive צדקנו ואכן ערכם positive.

$$\frac{TP}{TP + FP}$$

- המדדים החשובים שבחרנו להתמקד בהם בפרוייקט הינם:

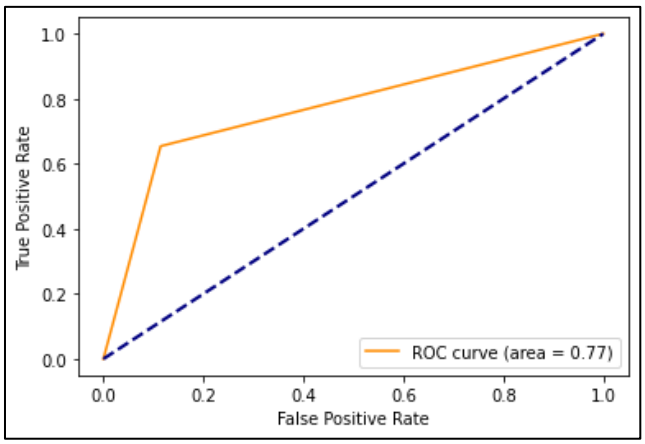
1. **Sensitivity**
   בחרנו להשתמש במדד זה מכיוון שכחברת שיווק, נרצה למקסם את מספר הלקוחות שאנחנו "מגיעים" אליהם. כלומר, לא נרצה למקסם את יחס ה-TP שלנו למול כלל הלקוחות הפוטנציאליים. ע"י מקסום המדד הנ"ל נדע נבצע קמפיין אפקטיבי אשר יניב רווחים גדולים יותר לחברה.
2. **Precision**
   חשוב בעינינו, מכיוון שנרצה לוודא שהמשאבים שלנו מנוצלים כראוי ושלא נתכנן קמפיינים עבור קהל יעד לא נכון.
3. **Accuracy**:
   מדד זה מייצג את הדיוק של הכללי של המודל. דיוק גבוה יוביל לפחות טעויות בין אם הם FP או FN.

א. Decision Tree

המסווג הראשון שבחרנו הוא עץ החלטות .

1. אימון ראשוני:

accuracy= 0.836
sensitivity= 0.6542056074766355
precision= 0.6086956521739131



עקומת ה-ROC לאחר האימון

2. תוצאות החיזוי על הtest:

accuracy= 1.0
sensitivity= 1.0
precision= 1.0

3. הסבר והנחות:

ניתן לראות שקיבלנו חיזוי מאוד גבוה הודות להיווצרות של . overfitting כלומר היצמדות יתר של המסווג לערכים ויצירת מצג לא נכון בעקבות תכנון לקוי של המסווג
בכדי להימנע מכך, וכך גם עשינו במסווגים הבאים בדקנו את ההיפר פרמטרים של המסווג בכדי להגיע לתוצאה הכי יעילה ואמינה . בעץ החלטות ללא הגבלת ההיפר פרמטרים ניתן בקלות להגיע ל- overfitting מכיוון שהוא עובר לצומת הבאה עבור כל מספר קטן של דוגמאות עד שנשאר מספר מאוד קטן של דוגמאות בקצה.

4. <u>היפר פרמטרים</u>:

בעזרת שינוי ההיפר מטרים הגענו לתוצאה מדויקת ואמינה יותר . בעזרנו בפונקציה בהיפר פרמטר טיונינג בכדי להוביל אותנו לערכים שיבטיחו אמינות .בתהליך זה בחרנו טווח ערכים רחב עבור האפשרויות המרכזיות כפי שמובא כאן:

max_depth[3,5,8,10,15,20] =
criterion = ['gini','entropy']
splitter = ['best', 'random']
min_samples_split=[500 ,10,50,75,100,125,200]
min_samples_leaf=[500 ,200 ,1,5,10,20,50,100]

4.1. <u>הסבר על ההיפר פרמטרים שהתמקדנו בהם</u>:

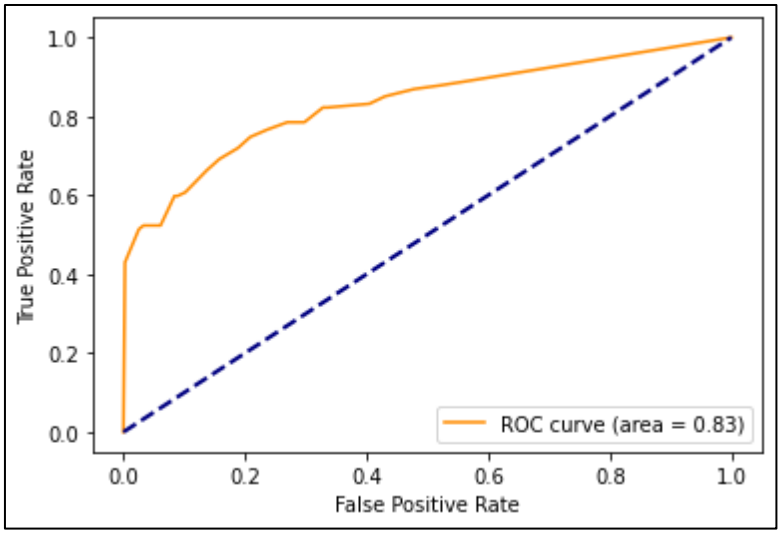
- **Max_depth:** משפיע על כמה העץ יהיה עמוק ככל שהעץ עמוק יותר המסווג הדוק יותר וישפיע על רמת הדיוק שלנו.
- **Min_sample_split:** קובע עבור כמה דוגמאות יווצר פיצול חדש בעץ ובעצם קובע כמה פיצולים יהיו בעץ.
- **Min_sample_leaf**: מספר הדוגמאות שיהיו בעלה הסופי משפיע על הדיוק שלנו, לא נרצה להתייחס לעלים עם מספר קטן מידי של דוגמאות מהמידע.
- **Criterion**: קובע איך נבחר את החלוקה לפי entropy/gini המסווג בודק איזו חלוקה נותנת תוצאות טובות יותר.

4.2. ראשית השתמשנו באלגוריתם רנדומלי למציאת ההיפר פרמטרים האופטימליים וקיבלנו שהם:

{'splitter': 'random',
'min_samples_split': 50,
'min_samples_leaf': 10,
'max_depth': 15,
'criterion': 'gini'}

אימנו מחדש לפי ההיפר פרמטים האופטימליים ואלה תוצאות האימון:

accuracy= 0.85
sensitivity= 0.5233644859813084
precision= 0.7



עקומת ה-ROC לאחר אימון מחדש

5. בדיקת המודל

בדקנו את המודל החדש על הtest כדי לבדוק שאנו לא שוב בoverfitting וקיבלנו את התוצאות הבאות:

accuracy= 0.8935622317596567
sensitivity= 0.6050420168067226
precision= 0.8275862068965517

וניתן לראות שאכן הגענו לתוצאות טובות יחסית ודומות לתוצאות האימון הראשוני וכנראה שלא נוצר overfitting. לאחר מכן הפעלנו אלגוריתם greedy למציאת ההיפר פרמטרים:

{'splitter': [best]
min_samples_split,[500 ,10,50,75,100,125,200]
min_samples_leaf,[500 ,200 ,1,5,10,20,50,100]
max_depth,[3,5,8,10,15,20]
criterion': ['gini', 'entropy']

וקיבלנו שהפרמטרים הטובים ביותר הם:

'criterion': 'entropy'
'max_depth': 3
'min_samples_leaf': 1
'min_samples_split': 50
'splitter': 'best'

אימנו עם הפרמטרים הללו והרצנו חיזוי על הtest ואלה התוצאות:

accuracy= 0.85
sensitivity= 0.5233644859813084
precision= 0.7

נראה שקיבלנו תוצאות קצת פחות טובות מהאלגוריתם הרנדומלי להיפר פרמטרים אבל בכל מקרה המודל לא בoverfitting.

ב. Random Forest

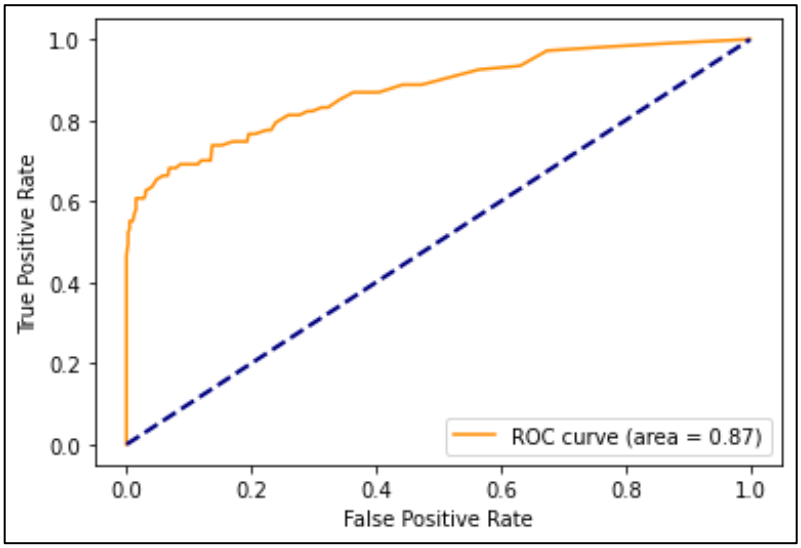
1. הסבר
מודל שיוצר מספר סוגים של עצים משווה ביניהם ולוקח את העץ שנותן את התוצאה הטובה ביותר. הוא לא מבצע תהליך של תיקון אלא יוצר כמה עצי h החלטה מחלקים שונים של הdata. יתרון על מסווג עץ החלטה רגיל שכאן אנו מקבלים מספר עצים אם פיצולים שונים שלא היו נבחרים בעץ ההחלטה הרגיל ואז אנו בודקים מי העץ עם התוצאות הטובות ביותר.

2. ההיפר פרמטרים למסווג זה הם:

- **Max_depth:** משפיע על כמה העץ יהיה עמוק ככל שהעץ עמוק יותר המסווג הדוק יותר וישפיע על רמת הדיוק שלנו.
- **Min_sample_split**: קובע עבור כמה דוגמאות יווצר פיצול חדש בעץ ובעצם קובע כמה פיצולים יהיו בעץ.
- **Min_sample_leaf**: מספר הדוגמאות שיהיו בעלה הסופי משפיע על הדיוק שלנו, לא נרצה להתייחס לעלים עם מספר קטן מידי של דוגמאות מהדאטא.
- **N_estimators:** מספר העצים ביער.

2.1 ראשית אימנו את המודל ללא כיוונון ההיפר פרמטרים ואלו תוצאות האימון:

- accuracy= 0.896
- sensitivity= 0.5514018691588785
- precision= 0.9365079365079365



עקומת ה-ROC לאחר אימון מחדש

2.2 חיזוי המודל:

לאחר מכן הרצנו חיזוי של המודל על הtest ואלה התוצאות שקיבלנו:

accuracy= 1.0
sensitivity= 1.0
precision= 1.0

ניתן לראות בבירור שהמודל בoverfitting ולכן חיפשנו את ההיפר פרמטרים לשינוי:

criterion: ['gini', 'entropy'],
max_depth': [5, 10, 15, 20, 25, 30, 35, 40, 45, 50, None]
max_features': ['auto', 'log2', 2, 5, 8, 16, 19]
max_samples': [0.1, 0.3, 0.5, 0.7, 0.9]
min_samples_leaf': [10, 20, 50, 100]
min_samples_split': [10, 20, 50, 100]
n_estimators': [100-1000 with jumps of 18]

זה הערכים שרצינו לבחון, השתמשנו באלגוריתם הרנדומלי למציאת ההיפר פרמטרים האופטימליים וקיבלנו שהם:

n_estimators: 742
min_samples_split: 50
min_samples_leaf: 10
max_samples: 0.9
max_features: 16
max_depth: 35
criterion: 'gini'

2.3 כיוונון ההיפר פרמטרים

לאחר כיוונון ההיפר פרמטרים אימון מחד של המודל והרצתו על הtest אלה התוצאות שקיבלנו:

random model accuracy = 88.60%
random model sensitivity = 52.34%
random model precision = 90.32%

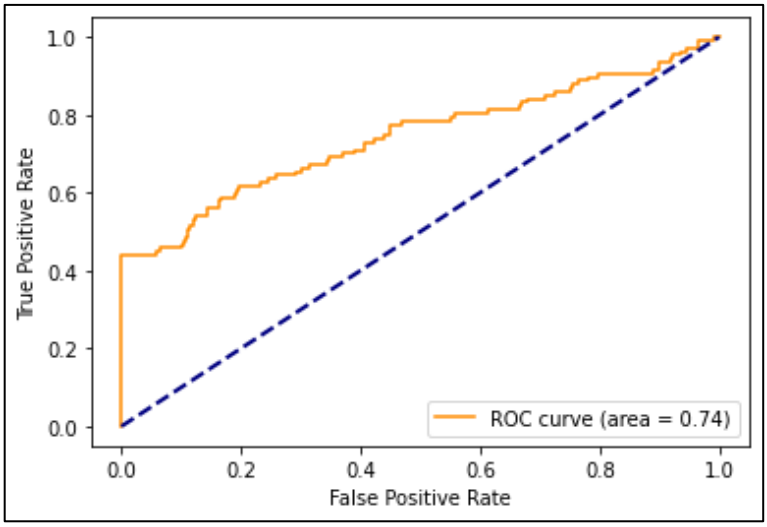
ניתן לראות שהתוצאות אכן קרובות לתוצאות האימון הראשוני וכנראה שהמודל לא בoverfitting.

ג. SVM classifier

1. הסבר:
מודל זה מנסה לחלק את המידע לינארית מכיוון שהמידע אינו בהכרח לינארי המודל מנסה לעלות את מימד המידע למימד שבו יהיה ניתן לבצע את החלוקה.

2. אימון ראשוני:
ראשית אימנו את המודל על הtraining data שלנו ואלה תוצאות האימון:

- base model accuracy = 87.80%
- base model sensitivity = 43.93%
- base model precision = 97.92%



עקומת ה-ROC לאחר אימון ראשוני

הרצנו את המודל על הtest ואלה התוצאות שקיבלנו:

- accuracy= 0.871244635193133
- sensitivity= 0.3739495798319328
- precision= 0.9888888888888889

לא לגמרי ברור עם המודל בoverfit

3. <u>היפר-פרמטרים</u>

לשיפור המודל בחרנו את הפרמטרים הבאים ואת טווח הערכים שלהם:

C: [0.1, 0.3, 0.5, 0.7, 0.9]
decision_function_shape: ['ovo', 'ovr']
kernel: ['linear', 'poly', 'rbf', 'sigmoid']
probability: [True]

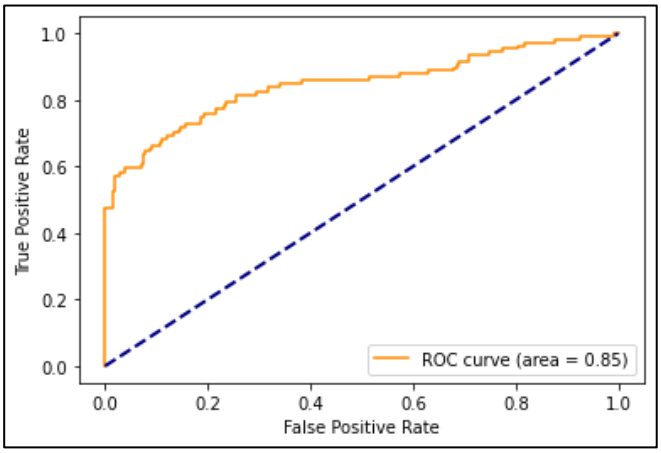
- kernel- פונקציית הגרעין להעלאת המימד.
- C- גורם רגולריזציה שמשפיע על המשקלים בפונקצייה.

לאחר הרצת האלגוריתם הרנדומלי לחיפוש ההיפר פרמטרים האופטימליים קיבלנו שהם:

probability: True
kernel: 'linear'
decision_function_shape: 'ovo'
C: 0.5

לאחר אימון עם ההיפר פרמטרים קיבלנו את תוצאות האימון הבאות:

random model accuracy = 88.60%
random model sensitivity = 43.93%
random model precision = 100.00%



<u>עקומת ה-ROC לאחר אימון עם היפר פרמטרים</u>

הרצנו את המודל על הtest ואלה התוצאות שקיבלנו:

accuracy= 0.8772532188841202
sensitivity= 0.40756302521008403
precision= 0.9797979797979798

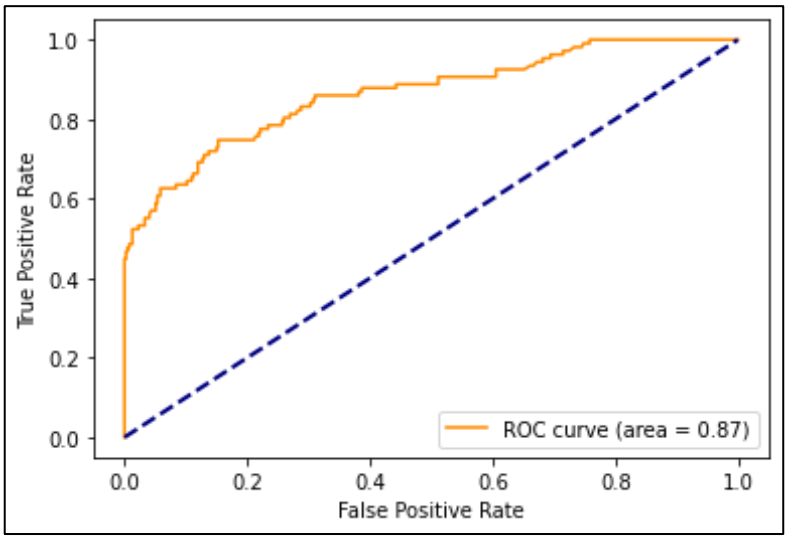
ד. AdaBoost

1. הסבר
אלגוריתם adaptive boost- אלגוריתם למידה המחפש מספר קטן של מסווגים "חזקים" מתוך קבוצה של מסווגים "חלשים" האלגוריתם מעניק משקל גדול לשגיאות בזיהוי כאשר כל משקל מסמל את חשיבות התכונה.

2. אימון ראשוני:

להלן תוצאות אימון המודל:

accuracy= 0.87
sensitivity= 0.6074766355140186
precision= 0.7386363636363636



עקומת ה-ROC לאחר אימון ראשוני

3. תוצאות הרצת המודל על הtest:

accuracy= 0.9201716738197425
sensitivity= 0.6638655462184874
precision= 0.9239766081871345

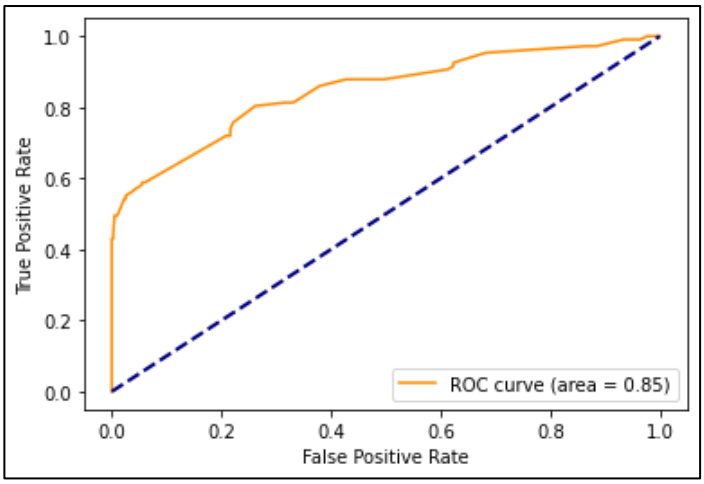
4. <u>היפר פרמטרים:</u>

הפרמטרים שבחרנו לשם כיוונון המודל:

n_estimators: [10, 20, 25, 30, 35, 50, 100, 150, 200, 400]

תוצאות האימון לאחר בחירת ההיפר פרמטר: 'n_estimators': 10:

accuracy= 0.884
sensitivity= 0.5420560747663551
precision= 0.8656716417910447



<u>עקומת ה-ROC לאחר אימון עם היפר-פרמטרים</u>

<u>תוצאות האימון של המודל החדש על הtest:</u>

accuracy= 0.8918454935622318
sensitivity= 0.5672268907563025
precision= 0.8544303797468354

4. ניתוח המודל:

א. **הצגת תוצאות האימון:**
ראשית נציג את מיפוי התוצאות של כלל המסווגים :
נציין, כי עבור SVM נתקלנו בבעיה בקוד ועל כן לא התייחסנו למסווג זה בניתוח המודל.

Classifier	Accuracy	Sensitivity	Precision
Basic decision tree	1.0	1.0	1.0
Decision tree – hyperparameters	0.8935622317596567	0.6050420168067226	0.8275862068965517
Decision tree – grid search	0.85	0.5233644859813084	0.7
Basic random forest	1.0	1.0	1.0
Random forest – hyperparameters	0.886	0.5234	0.9032
Adaboost	0.9201716738197425	0.6638655462184874	0.9239766081871345
Adaboost – hyperparameters	0.8918454935622318	0.5672268907563025	0.8544303797468354

נציין כי הרצה מס׳ אחד של המידע ,כפי שתואר בפרק 3 , הייתה המוצלחת יותר מבחינת תוצאות המודל ועל כן נציג אותה. כלומר, המודל מוצלח יותר כאשר מתייחסים לתגובתיות של לקוח לקמפיין האחרון בלבד ללא יחס לקמפיינים מהעבר.

ב. <u>ניתוח התוצאות</u>

בשלב הראשון הרצנו את שלושת המסווגים הטובים ביותר שלנו:

1. אימנו אותם 10 פעמים כאשר כל פעם חלק אחר של ה data אשר הוגדר ב set validation בצורה של cross validation.
2. חישבנו את הממוצע של המדדים הרלוונטיים של ההרצות עבור כל מסווג .
3. בעזרת המדדים שקיבלנו השוונו בין כל זוג מסווגים ע״י מבחנים סטטיסטים.
4. כפי שציינו קודם לכן, ביצענו 2 הרצות שונות אשר נבעו מעיבוד שונה של המידע. לאור הגדרת אילו פרמטרים חשובים יותר למודל שלנו נוכחנו לדעת כי גרסה 2 של ההרצה המתייחסת לתגובתיות של הלקוח בהווה ולא שמה דגש על תגובה לקמפיינים קודמים השיגה תוצאות טובות יותר. המסווג הטוב ביותר הינו – **<u>ADA BOOST</u>**

Classifier	Mean Accuracy	Mean Sensitivity	Precision Mean
Adaboost	0.9183284034340955	0.6300069495721669	0.7840123928359223
RandomForestClassifier	0.9321441454440516	1.0	0.5440447233925496
DecisionTreeClassifier	0.8943366279489215	0.646885674065801	0.6772596968249143

- מסווג זה לדעתנו מוצא את האיזון הדרוש להצלחת הקמפיין. נציין כי RandomForestClassifier השיג תוצאה כללית טובה יותר אך זאת היות שערך ה-Precision שלו יצא 1.0 דבר המצביע לעינינו שמשהו לא בוצע כשורה על ידינו בתהליך.

ג. <u>הרצת המודל על סט המבחן</u>

1. הרצנו את תהליך העיבוד (הזהה לעיבוד שבוצע על מידע האימון) על המידע.
2. הרצנו את המסווגים, כאשר נציג רק את תוצאות המסווג שנבחר, על סט המידע הנ״ל כאשר יצרנו עמודה של –will_responde .
3. לאחר ההרצה ניתן לראות בקובץ המצורף כי ישנם 54 לקוחות אשר אופיינו כאלו שיגיבו לקמפיין השיווק. מידע זה ניתן לספק למנהלי השיווק על מנת לכוון את הקמפיין לאותם לקוחות.