רוני מינדלין מילר 302242870
עמית מגן 206348005

למידה חישובית – עבודה 2

שם הקבוצה: AmitRonnie

תיעוד הריצות שביצענו:

#version	accuracy
1	0.32
2	0.678
3	0.682
4	0.684
5	0.685
6	0.66
7	0.696
8	0.696
9	0.678
10	0.67
11	0.67
12	0.67
13	0.68
14	0.66
15	0.64
16	0.59
17	0.58
18	0.66
19	0.657
20	0.644
21	0.685
22	0.549
23	0.693
24	0.689
25	0.704
26	0.704
27	0.668
28	0.687
29	0.693

ניתן לראות שלניסיונות 15,16,17 רמת דיוק מאוד נמוכה כיוון שאלו היו גרסאות שניסינו להשתמש ב-SMOTE שעשה לנו overfitting ל-data (מפורט בהמשך).

רוני מינדלין מילר 302242870
עמית מגן 206348005

<u>טיפול בערכים חסרים:</u>

ראשית הסרנו את כל השורות שכל הערכים בהם היו Null. לאחר מכן החלפנו את כל הערכים המייצגים ערכים חסרים ('', ' ', 'None', 'nan', 'NaN') ב-np.nan על מנת ליצור ייצוג יחיד עבור כל הערכים החסרים.

בחרנו להוריד את כל העמודות בהן יותר ממחצית מהערכים הינם ערכים חסרים, מתוך הנחה כי עמודות אלו יכניסו יותר מדי רעש למודל. יש לציין כי ניסינו לתת threshold שונה ולראות איך הדבר משפיע על המודל (0.4, 0.6, 0.7, 0.8, 0.85), אך עם 0.5 התוצאות היו הכי טובות. לאחר ההורדה נשארנו עם 34 עמודות (מתוך 52).

עבור ערכים קטגוריאליים השלמנו את הערכים החסרים עם תווית קבועה המייצגת ערכים חסרים. כלומר הוספנו ערך חדש בשם null_value לייצוג ערכי null. לסיום השתמשנו ב-label encoder כדי להמיר את המשתנים הקטגוריאליים ונומריים.

עבור ערכים נומריים השלמנו את הערכים החסרים באמצעות ממוצע הערכים של אותו הפיצ'ר.

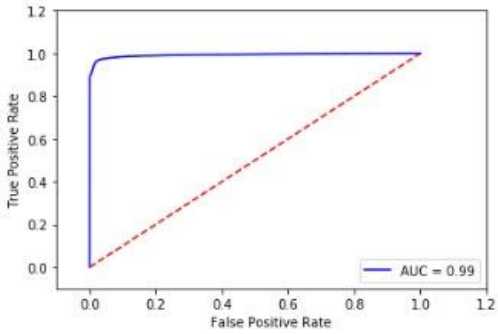
<u>Validation</u>:

שיטת הולידציה בה השתמשנו היא stratified 5-fold-cross validation, כיוון ששיטה זו מבצעת פיצול של הנתונים ל-folds כך שהתפלגות הנתונים ב-folds השונים תהיה זהה להתפלגות הנתונים ב-data המקורי. עובדה זו קריטית עבור הנתונים שלנו מאחר ויש Imbalance משמעותי בין שתי מחלקות הסיווג: עבור סיווג 1 יש 502 רשומות, ועבור סיווג 0 יש 29,949 רשומות. כלומר יש יחס של 1 ל-60 בין כמות הרשומות עבור כל סיווג. השימוש ב-stratified 5-fold-cross validation על פני שיטת הולידציה k-fold-cross validation מאפשר לנו להתחשב בחוסר השוויון של הנתונים בכל אחד מה-folds וגורם לזה שלא יקרה מצב שיהיה מודל שיתאמן רק על רשומות בעלות סיווג 0.

<u>Imbalance data</u>:

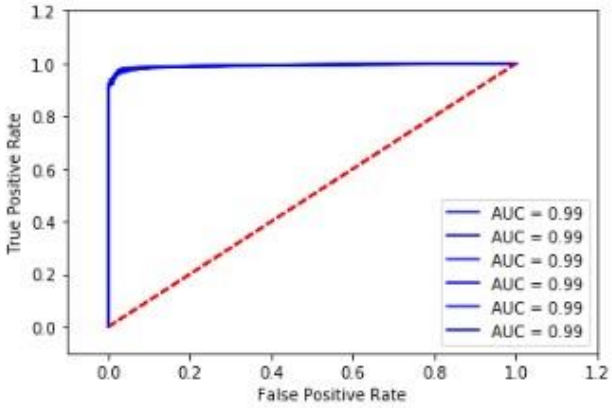
כדי להתמודד עם בעיית ה-imbalance date שהזכרנו לעיל, ניסינו להשתמש ב-smote כדי לייצר רשומות מלאכותיות בעלות הסיווג הפחות שכיח ב-data. לאחר השימוש ב-smote על קובץ ה-train, קיבלנו כ-59898 רשומות (29,949 רשומות עבור כל אחד מהסיווגים). הרצה על כל אחד מהמודלים הביאה אותנו ל- overfitting ולכן בחרנו לא להשתמש בשיטה זו. להלן תוצאות המודלים שניסינו עם שימוש ב-smote:

- LgbRegressor (הפרמטרים מצוינים במחברת):

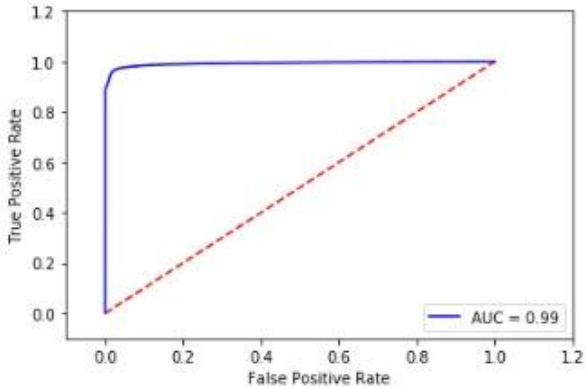


רוני מינדלין מילר 302242870
עמית מגן 206348005

- Ensemble of 3 models: XgbRegressor, XgbClassifier, lgbRegressor



- SVR:



ניתן לראות כי בכל אחד מהמודלים קיבלנו דיוק גבוהה מאוד, ועל כן אנו נמצאים במצב של overfitting.
ניסינו לבצע ניסיונות להתאים את הפרמטרים של המודל להימנע ממצב זה (למשל במודל xgboost שינינו את הפרמטרים subsample, eta, max_depth שאמורים למנוע overfitting) , אך גם אז המודל סבל מ-overfitting חמור ולכן החלטנו לוותר על השימוש ב-smote.

רוני מינדלין מילר 302242870
עמית מגן 206348005

פירוט הניסיונות והשיטות שהשתמשנו:

Random Forest:

N_estimators	Max_feature	Max_depth	Min_sample_split	Mean_sample_leaf	ROC
100	auto	7	20	100	0.5
100	auto	4	20	100	0.5
100	none	4	20	100	0.5
100	none	4	200	100	0.5

לאחר לא מעט ניסיונות לשפר את המודל ללא הצלחה, בחרנו לוותר על מודל זה.

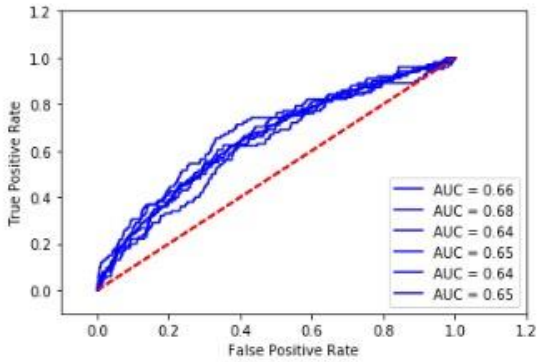
XgbRegressor:

הפרמטרים המסומנים הינם הניסיון הכי טוב שהיה לנו עבור מודל זה, ואלו גם הפרמטרים שהשתמשנו בהגשה.

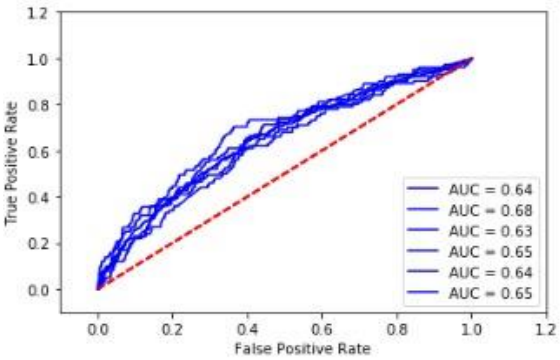
Nnestimators	Learning_rate	gama	subsample	Col sample by tree	Max depth	AUC	AUC image
100	0.01	0	0.8	1	7	0.65	1
100	0.08	0	0.8	1	7	0.605	2
100	0.008	0	0.8	1	7	0.649	3
100	0.01	0.2	0.8	1	7	0.642	4
100	0.01	0	0.6	1	7	0.645	5
100	0.01	0	1	1	7	0.648	6
100	0.01	0	0.8	0.8	7	0.646	7
100	0.01	0	0.8	1	5	0.643	8
100	0.01	0	0.8	1	8	0.647	9
150	0.01	0	0.8	1	8	0.647	10
50	0.01	0	0.8	1	8	0.641	11

רוני מינדלין מילר 302242870
עמית מגן 206348005

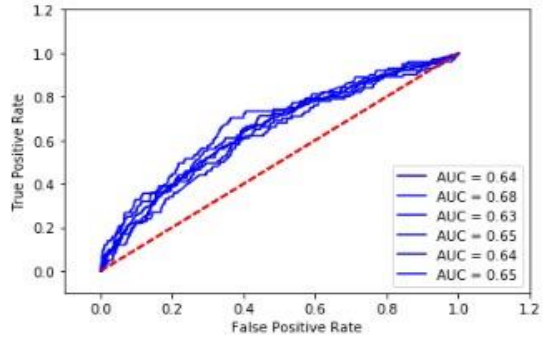
מודל ראשון:



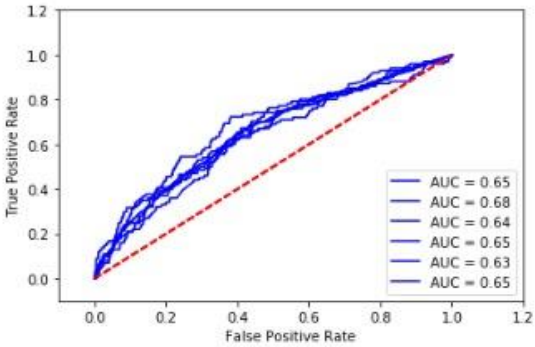
מודל שני:



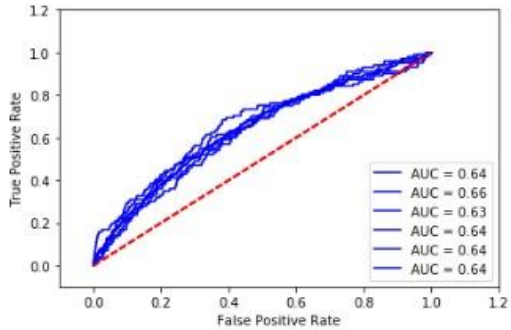
מודל שלישי:



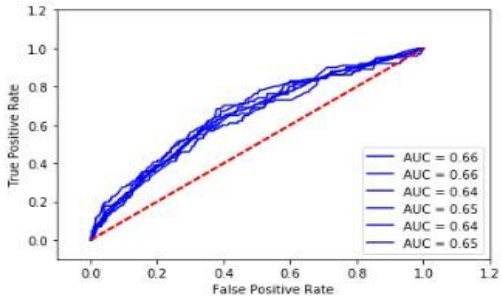
מודל רביעי:



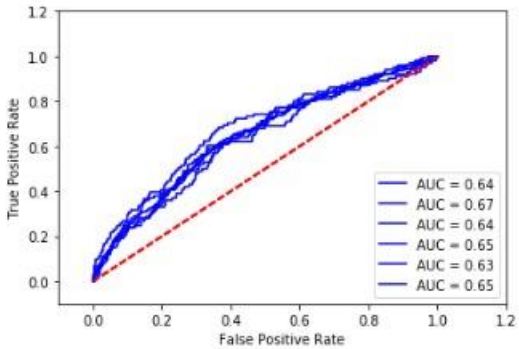
מודל חמישי:



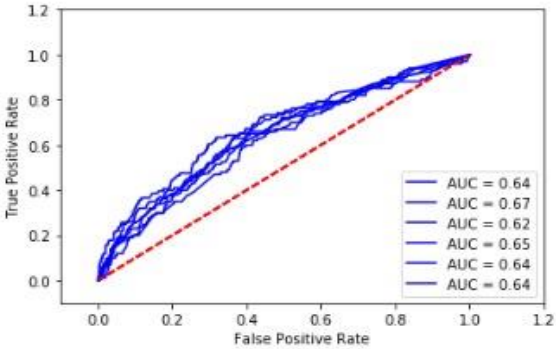
מודל שישי:



מודל שביעי:



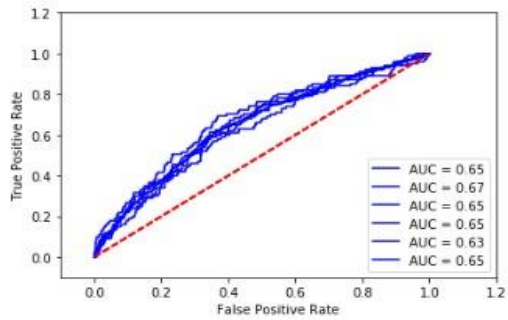
מודל שמיני:

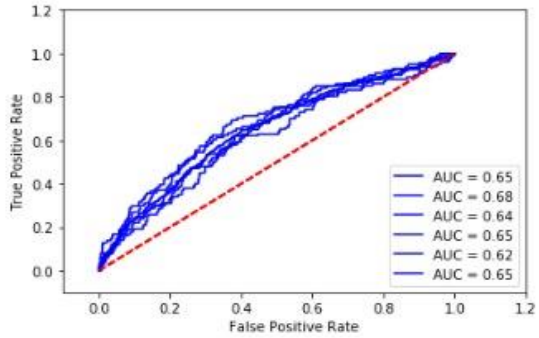


מודל תשיעי:

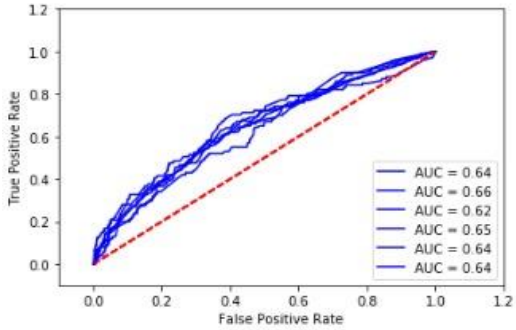
מודל עשירי:

רוני מינדלין מילר 302242870
עמית מגן 206348005





מודל אחד עשר:



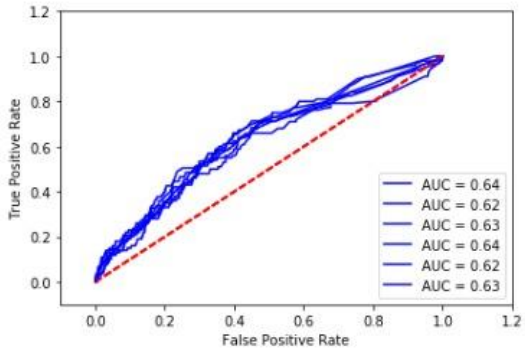
:lgbmRegressor

זהו מודל מבוסס עצי החלטה שמשתמש ב-gradient boosting. להלן הניסיונות שביצענו לשיפור המודל:

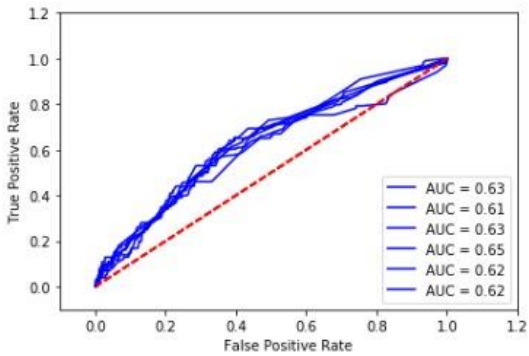
	Is_unbalance	N_estimators	Num_ leaves	Learning_rate	subsample	Max_depth	AUC
1	true	100	20	0.01	0.8	7	0.62
2	true	50	20	0.01	0.8	7	0.621
3	true	150	20	0.01	0.8	7	0.627
4	true	200	20	0.01	0.8	7	0.634
5	true	300	20	0.01	0.8	7	0.632
6	true	200	10	0.01	0.8	7	0.64
7	true	200	5	0.01	0.8	7	0.643
8	true	200	3	0.01	0.8	7	0.637
9	true	200	5	0.08	0.8	7	0.636
10	true	200	5	0.008	0.8	7	0.641
11	true	200	5	0.01	0.6	7	0.643
12	true	200	5	0.01	0.6	5	0.642
13	true	200	5	0.01	0.6	3	0.642
14	true	200	5	0.01	0.6	9	0.647

רוני מינדלין מילר 302242870
עמית מגן 206348005

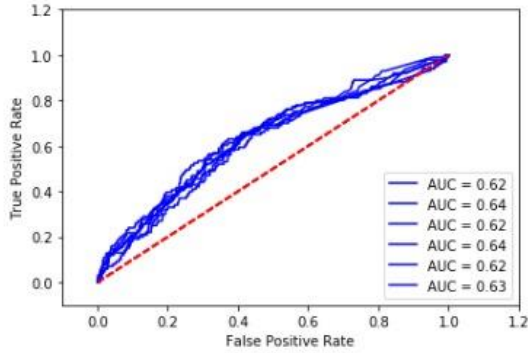
מודל ראשון:



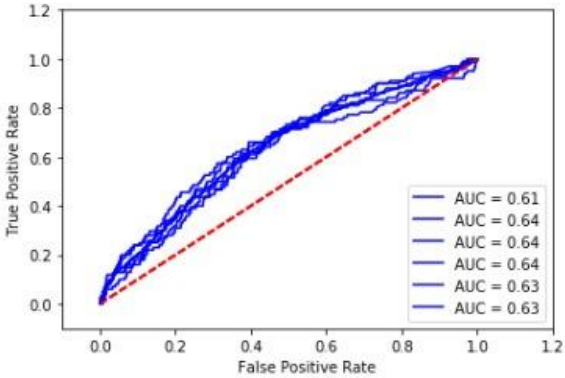
מודל שני:



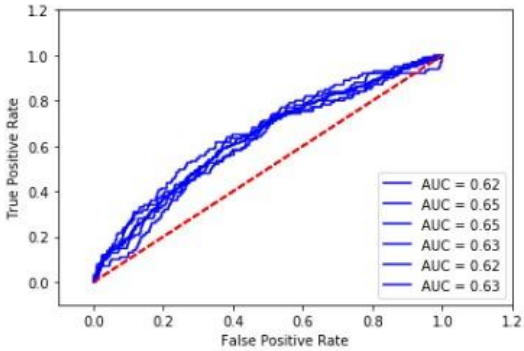
מודל שלישי:



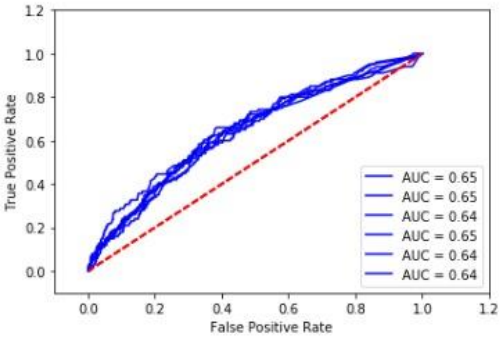
מודל רביעי:



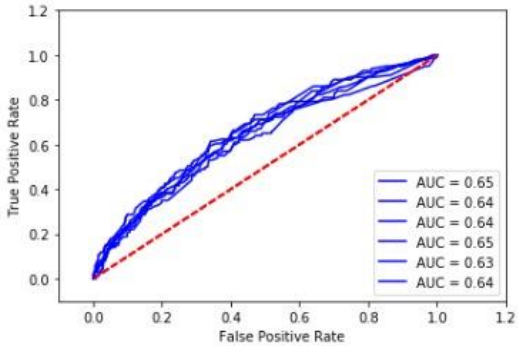
מודל חמישי:



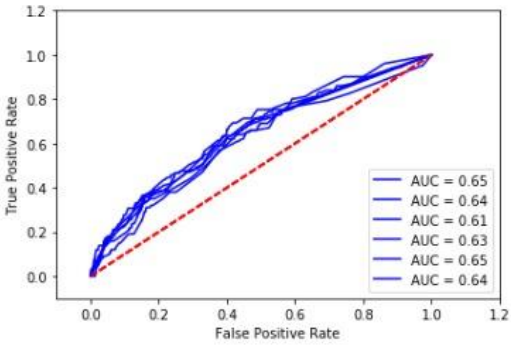
מודל שישי:



מודל שביעי:

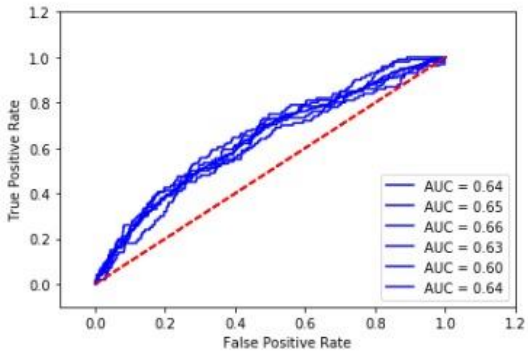


מודל שמיני:

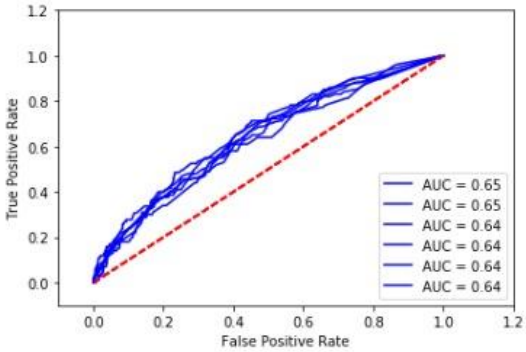


רוני מינדלין מילר 302242870
עמית מגן 206348005

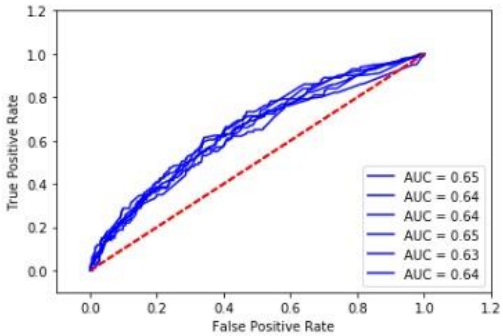
מודל תשיעי:



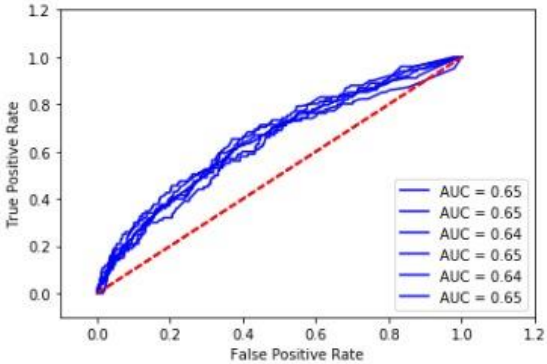
מודל עשירי:



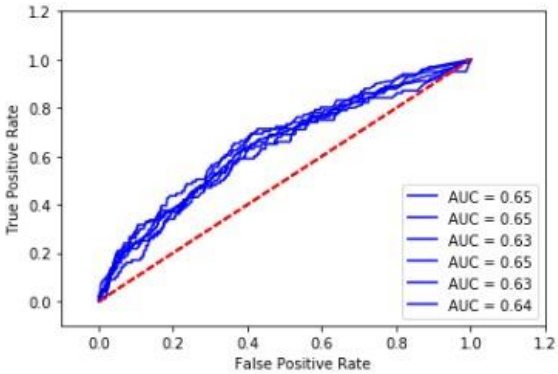
מודל אחד עשרה:



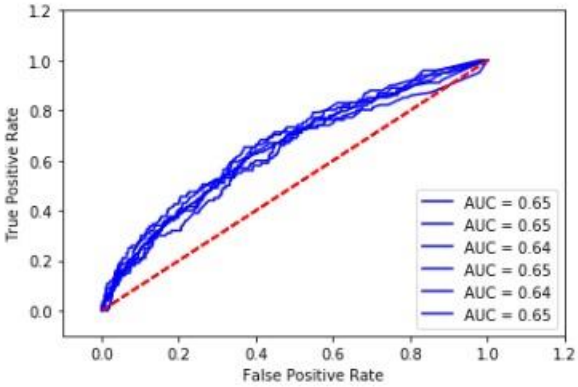
מודל שניים עשר:



מודל שלוש עשר:



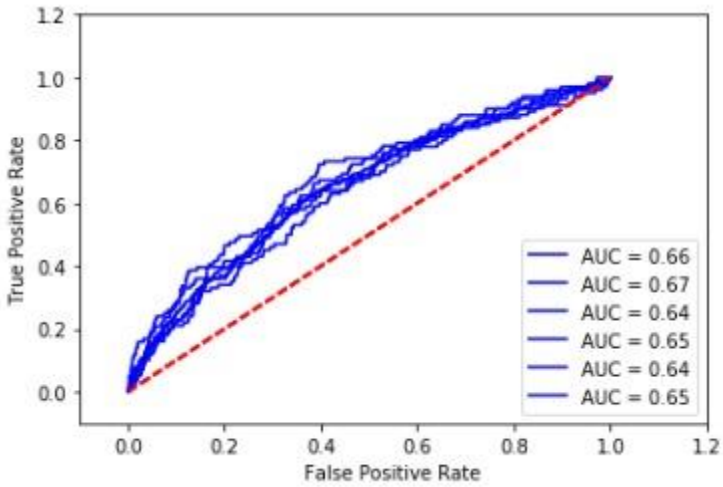
מודל ארבע עשר:



רוני מינדלין מילר 302242870
עמית מגן 206348005

לאחר מכן ניסינו לבצע ensemble של שני המודלים הנ"ל (XGBRegressor ו-lgbmRegressor).
השתמשנו ב-stretifiedKCross validation וביצענו stacking לתוצאות שני המודלים עבור כל אחד מה-folds (ממוצע על תוצאות החיזוי של כל אחד מהמודלים). קיבלנו תוצאה של AUC=0.651.

להלן הגרף שהתקבל:



מלבד המודלים שהצגנו ניסינו גם שימוש ב-SVM, naïve bays ו-fully connected neural network אך התוצאות לא היו מספיק טובות.

בנוסף ניסינו גם לבצע feature selection ע"י בחירת הפרמטרים המשמעותיים ביותר לאחר סיום האימון והורדת הפיצ'רים שרמת ההשפעה שלהם הייתה המינימלית ביותר – אך גם זה לא שיפר את המודל.

בסופו של דבר המודל שנתן לנו את התוצאה הטובה ביותר הינו XGBRegressor עם הפרמטרים שבחרנו. על אף שהרצת המודל עם K-cross validation נתן תוצאה פחות טובה, בחרנו להגיש אותו כיוון שה-validation בו יותר טוב. בסוף ביצוע ה-k-cross אנו מריצים פעם אחרונה את המודל על כל הנתונים (ללא חלוקה לtrain ו-test), ואת תוצאות החיזוי של מודל הזה הוספנו לממוצע של החיזוי של ה- test set.