

דאטא סיינס

בתעשייה

עבודה 1

מגישים:

בן איידלין

שחר קרמר

## תיאור כללי של ה-pipeline:

### הכנת נתונים

- (1) החלפת ערכים חסרים- עבור משתנים רציפים החלפנו בממוצע, ועבור משתנים בדידים החלפנו בערך הנפוץ ביותר.
- (2) המרת משתנים מסוג STRING למשתנים נומריים.
- (3) הורדת פיצ'רים שאינם רלוונטיים (עמודות המהוות ids)
- (4) ניקוי המידע-
  - a. תחילה ביצענו חילוף של טווח ההפרש בין הרבעון הראשון Q1 לרבעון שלישי Q3 עבור עמודת גיל.
  - b. לאחר מכן סיננו את כל הרשומות אשר ערך הגיל שלהן גדול מהרבעון השלישי ועוד הטווח שחילצנו כפול משקל מסוים.
  - c. ביצענו שינויים בנוסחה ובתנאים עד שקיבלנו את הנוסחה שהיתה לנו הכי הגיונית מבחינת התוצאות שלה, והיא-  $age > Q3 + 0.4 * IQR$ .
  - d. קיבלנו כי:
    - i. סה"כ נזרקו 7471.
    - ii. טווחי הגילאים שנזרקו נע בין 84-89.
    - iii. מתוך 7471 רשומות שנזרקו, ישנם 1042 מקרי ממוות (שהינם מהווים בערך 13.165% מתוך סך כלל הנפטרים ב-dataset).
  - e. בסופו של דבר, קיבלנו החלטה שלא לבצע ניקוי של המידע ולא להשתמש בסינון הזה מאחר ואנחנו מאבדים אחוז יחסית גבוה של נפטרים, וגיל הוא בעינינו אחד הפיצ'רים המשמעותיים שיכול להוות סיבה הגיונית לסיווג או פיצול וקביעת מותו של חולה שהגיע למיון (הגיוני שכלל שגיל המטופל עולה כך הסיכוי למותו גדל).
- (5) הוספת פיצ'רים-
  - a. Hospital\_visit\_counts- כמות המבקרים סה"כ באותו בית חולים.
  - b. Hospital\_death\_count- כמות המתים סה"כ באותו בית חולים.
  - c. Precent\_of\_death\_in\_hospital- אחוז המתים באותו הבית חולים, חושב באמצעות שני הפיצ'רים מעלה (לאחר חישוב זה הם נמחקו).
  - d. K-mean column- הוספנו פיצ'ר ע"י שימוש באלגוריתם זה, ככה שכמות הקלסטרים היא לוג בבסיס 4 של כמות הרשומות הקיימות (יצא 8 קלסטרים). להלן התפלגות הפיצ'ר (משמאל-מספר הקלסטר, מימין- כמות הרשומות שמופא לאותו הקלסטר).

0	45048
5	15804
2	9727
3	8249
1	6321
7	2983
6	2782
4	799
- (6) בחירת פיצ'רים- השתמשנו במבחן ANOVA F value ובפונקציה קיימת SelectKBest של sklearn המאפשרת בחירת פיצ'רים. סיננו את כל הפיצ'רים שה-p\_value שלהם גדול מ-0.01. סה"כ נשארו עם 167 פיצ'רים.

## מידול אבלואציה

5 שיטות הקלספיקציה השונות שבחרנו הן:

- Neural Network (1)
- Logistic Regression (2)
- Naive Bayes (3)
- Random Forest (4)
- KNN (5)

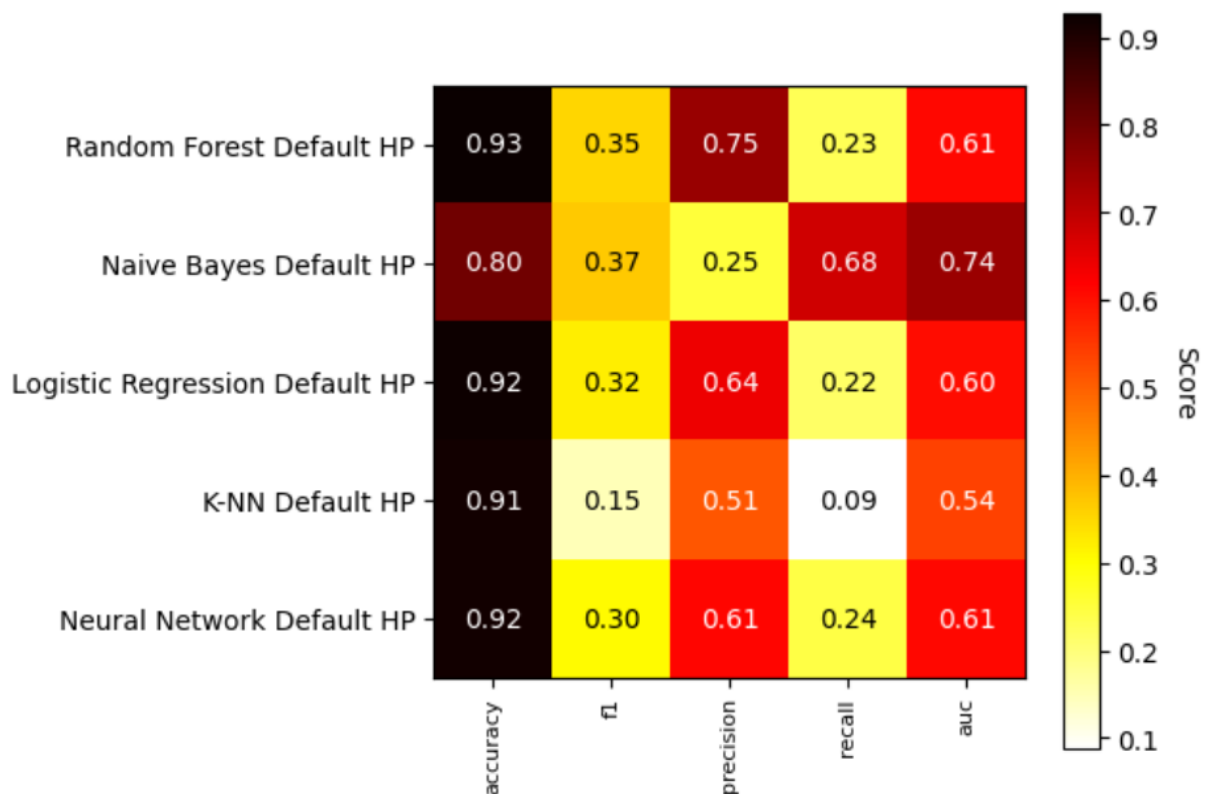
### **ביצוע של 10-fold-cross-validation:**

הגדרנו מחלקה הנקראת FOLD. המחלקה מייצגת fold ספציפי מתוך העשרה ומופע שלה מכיל את סט האימון והמבחן וטווחי ה-binnings. לבסוף יצרנו קובץ Pickle שישמור את אותם folds כדי שבמידה ונרצה להרי את הקוד מהתחלתו, כל החלק של הדסקרטיזציה (שהיה ארוך מאוד) יחסך ונקרא פשוט את הקובץ פיקל עם האובייקטים המייצגים את ה-folds.

### **לכל fold בוצעו השלבים הבאים:**

- (1) דסקרטיזציה- לכל פיצ'ר נומרי בחרנו את כמות ה-bins (המינימום בין 10 לכמות הערכים הייחודיים) ואת שיטת החלוקה עומק שווה.
- (2) אימון על גבי סט האימון (תחת כל fold בנפרד) של כל אחד מחמשת סוגי המודלים.
- (3) בחינת המודל בעזרת סט המבחן.
- (4) הערכת התוצאות ע"י חמשת המדדים הבאים:
  - a. Accuracy
  - b. F1
  - c. Precision
  - d. Recall
  - e. Auc

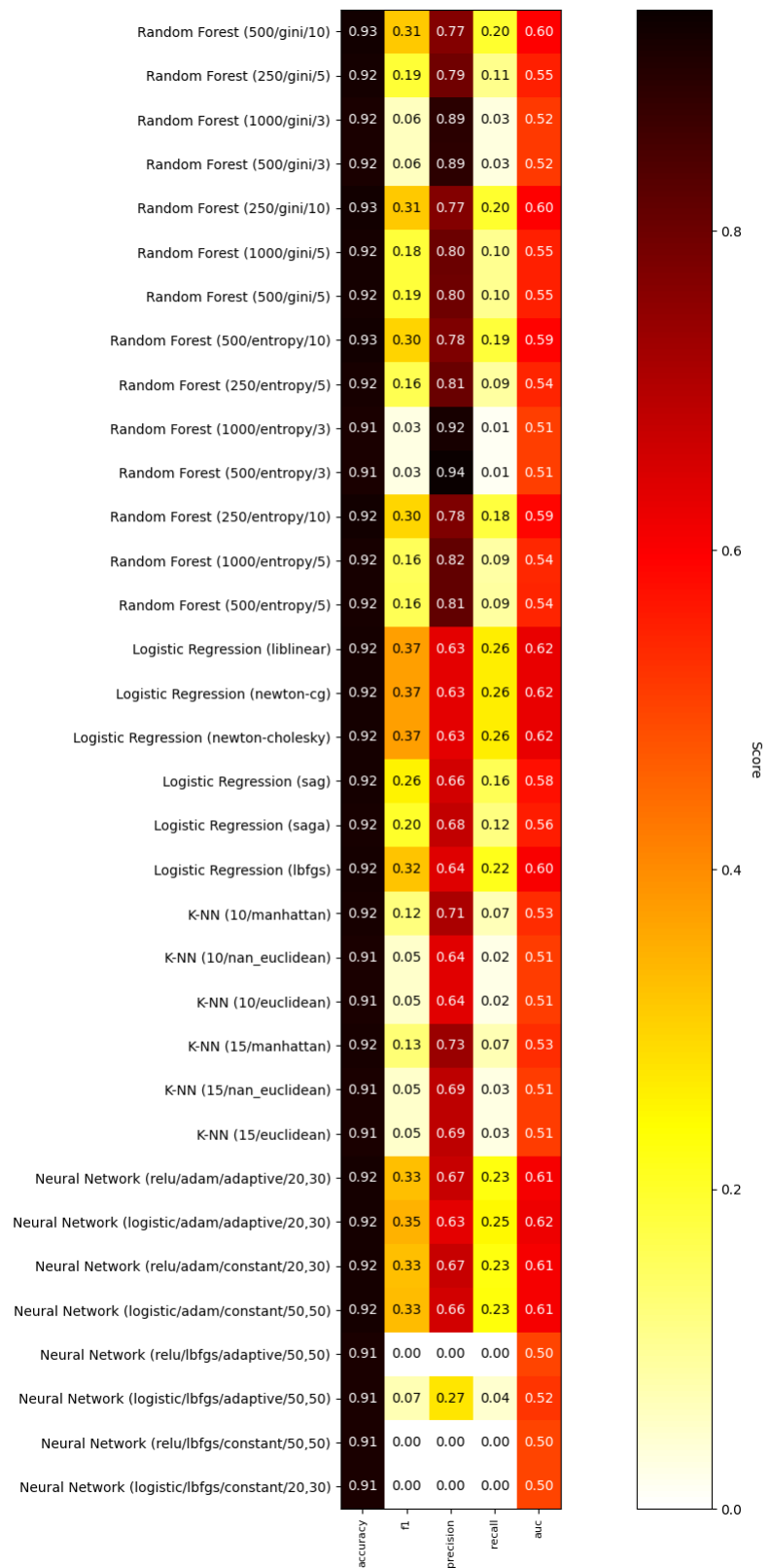
תחילה ביצענו את השלבים המתוארים מעלה עבור היפר-פרמטרים דיפולטיביים עבור כל אחד מהמודלים שבחרנו. סה"כ לאחר חישוב משוקלל של מדדי הערכה על גבי כל ה-folds תחת כל אלגוריתם הניבו את התוצאות הבאות:



```
{'Random Forest Default HP': {'accuracy': (0.926902368351137,
0.0032630942075040394),
'f1': (0.3505515962742344, 0.031081914602455334),
'precision': (0.7495184550691754, 0.03371180483033407),
'recall': (0.22905585709687135, 0.02411851030014563)},
'Naive Bayes Default HP': {'accuracy': (0.7989598878753886,
0.004282836064663516),
'f1': (0.36840524710346784, 0.01078759379617323),
'precision': (0.25271088643696604, 0.008036309436739143),
'recall': (0.679620796923019, 0.017618483946838785)},
'Logistic Regression Default HP': {'accuracy': (0.9219085604840112,
0.0030108039172654024),
'f1': (0.3224752660000427, 0.030983359868721073),
'precision': (0.6402207621024156, 0.026693223676031434),
'recall': (0.21598096083582544, 0.02537717008307037)},
'K-NN Default HP': {'accuracy': (0.9140143602416138, 0.0030866304345777006),
'f1': (0.14893984845657665, 0.01323111306725455),
'precision': (0.5113276337004479, 0.04171921222854151),
'recall': (0.08722363471605231, 0.00823371843549271)},
'Neural Network Default HP': {'accuracy': (0.9159878514551952,
0.006412394159747676),
'f1': (0.30417937743877416, 0.1437869329683791),
'precision': (0.6128235490740601, 0.1285773579554448),
'recall': (0.24425310955041432, 0.15840471022678976)}}
```

אפשר לראות שמדד הדיוק הוא מאוד גבוה בכל המדדים בזמן שמדדים אחרים די חלשים יחסית.  
 נראה כי Naïve base יוצא דופן- הוא היחיד שמדד הrecall שלו גבוה יחס לשאר, ומדד הprecision שלו נמוך יחס לשאר.

לאחר הרצה ראשונית, התחלנו "לשחק" עם ההיפר-פרמטרים של כל מודל, ובחנו את תוצאות המדדים בכל הרצה. להלן התוצאות:



## בחירת האלגוריתם:

נדגיש כי התוצאות המוצגות מעלה אינן מספקות לרמת ה deployment production אבל עושות את העבודה עבור שלב בחירת המודל.

נראה כי מבחינת דיוק, לכלל המודלים יש ביצוע דומה וכי מדד הדיוק גבוה בכולם ובטווח זהה.

בהיבט של recall אל מול precision, לדעתנו מדד ה precision הינו חשוב יותר למודל שנבחר מאחר וחשוב לנו יותר לזהות את אלו שישארו בחיים לא פחות מאשר לזהות את אלו שימותו. נסביר- נסתכל על מקרה הקיצון בו על כל חולה נחזה שימות. כלומר משמעות מקרה זה הוא recall גבוה. בהנחה והמודל שלנו יעזור לרופאים במיון להחליט איך לחלק את תשומת הלב שלהם, עדיף להפנות את המשאבים למקרים בהם הסיכויים לחיות גדולים יותר. לכן מקרה הקיצון המתואר פחות עדיף במקרה זה, ונרצה שהמודל פחות יחמיר ויוכל לזהות גם במקרים בהם החולה לא ימות.

בנוסף, אנחנו גם יותר נתחשב במדד AUC כמדד חשוב מאחר והוא מעריך בצורה טובה עד כמה המודל מצליח להבדיל בין רשומות שהם positive ל-negative.

זאת ועוד, מדד f1 פחות יהווה גורם משמעותי בהחלטת המודל מאחר והינו שיקלול של מדדי recall וה precision ונותן לכל אחד מהם משקל דומה, וכפי שהזכרנו למעלה אנחנו רוצים לתת חשיבות גבוהה יותר למדד ה precision.

תחילה נבחר עבור כל מודל את ההיפר-פרמטרים האופטימליים:

### 1) Neural Network:

- a. הפרמטר solver עם הערך lbfgs הניב את התוצאות הכי נמוכות (0 בכל המדדים).
- b. שאר תוצאות ההרצה די דומות, ולכן ניקח את קומבינציית הפרמטרים שממקסמת את ה precision (על חשבון ה recall).
- c. קומבינציית הפרמטרים הסופית- 20, 30, adaptive, adam, relu.

### 2) Logistic Regression:

- a. קומבינציית הפרמטרים הסופית- liblinear.

### 3) Naive Bayes:

- a. אין למודל זה היפר-פרמטרים שאפשר לשחק איתם, ולכן ניקח את תוצאותיו מההרצה הראשונית.

### 4) Random Forest:

- a. קומבינציית פרמטרים הסופית- 10, gini, 250.

### 5) KNN:

- a. קומבינציית פרמטרים הסופית- 15, manhattan.

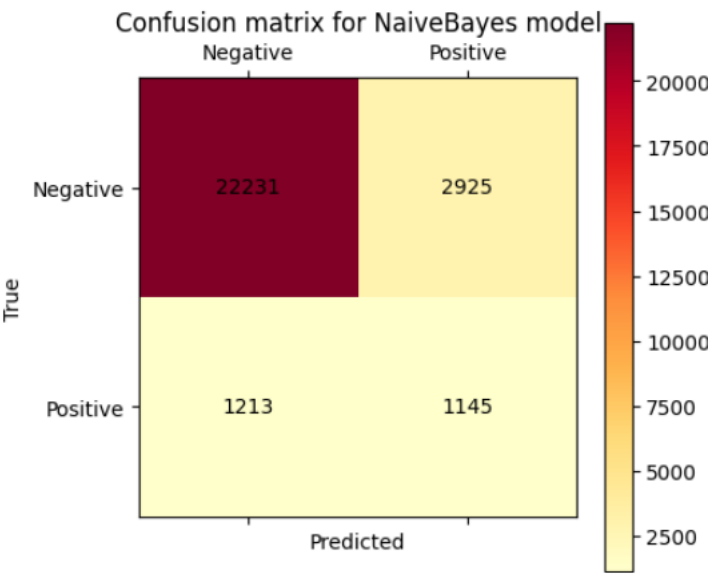
	ACC	F1	PRECISION	RECALL	AUC	Run time in sec	Run time std
ANN	0.92	0.33	0.67	0.23	0.61	104.47	80.32
LR	0.92	0.37	0.63	0.26	0.62	31.912	48.89
NB	0.798	0.36	0.25	0.67	0.74	0.124	0.023
RF	0.93	0.31	0.77	0.2	0.6	45.296	22.456
KNN	0.92	0.13	0.73	0.07	0.53	107.842	65.701

המודלים שבחרנו להמשיך איתם הם Naive Bayes, Random Forest ו- Neural Network.

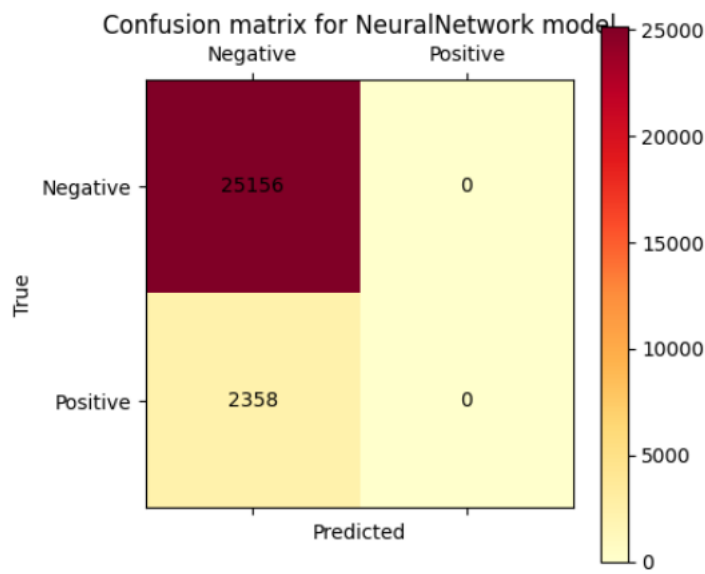
עבור כל אחד מהמודלים ביצענו הרצה נוספת ללא חלוקה ל-folds, והגדלנו משמעותית את כמות bins בשלב binning כיוון שראינו שדבר זה משפר את תוצאות המודלים. בהרצה זו החלוקה לסט אימון ומבחן על גבי על הדאטא סט נעשתה רנדומלית כך ש-70% מהרשומות היו תחת סט האימון והשאר תחת סא המבחן.

:confusion matrix

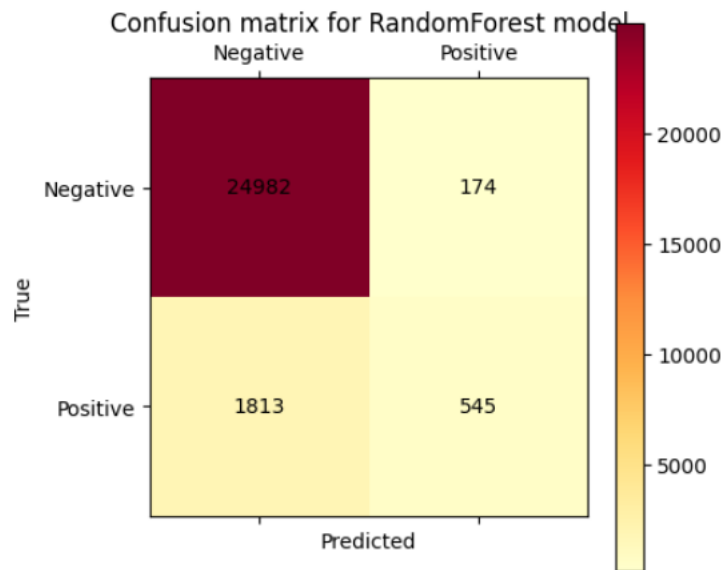
Naive Bayes (1)



Neural Network (2)

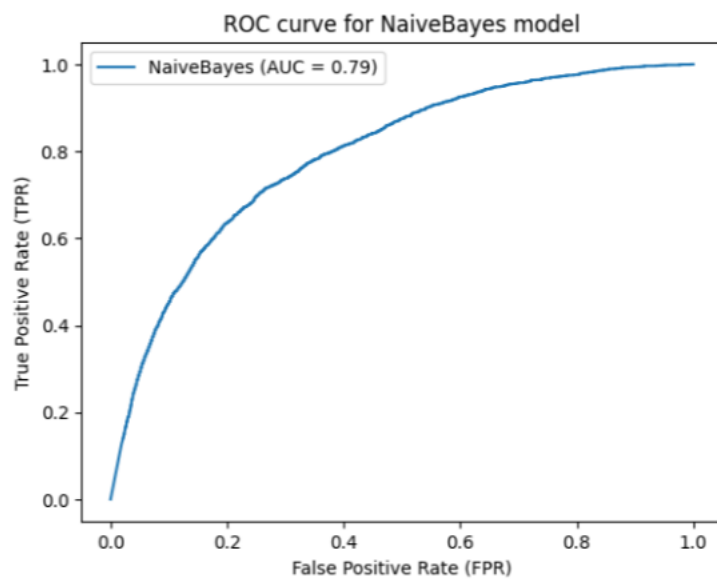


Random Forest (3)



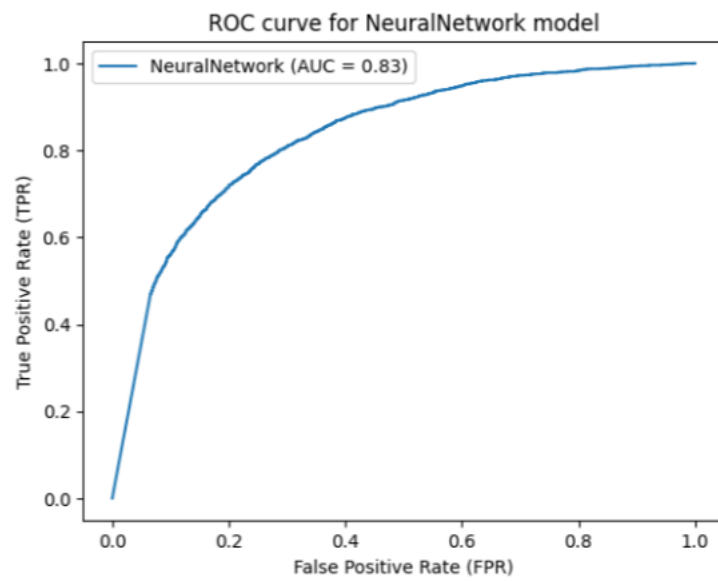
:ROC curve

Naive Bayes (1

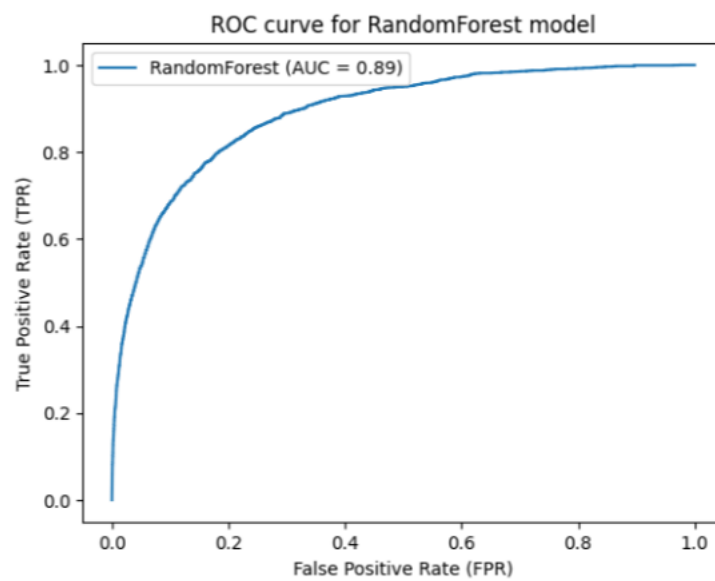




## Neural Network (2)



## Random Forest (3)



## אימון המודלים הנבחרים ותוצאות הערכה החיצונית

הרצנו את שלב pre-processing על הנתונים הנמצאים בקובץ unlabeled (בצורה שבה הפעלנו את אותה חוקיות שעשינו על dataframes המקורי, גם על הקובץ הלא מתוייג).

ביצענו אימון ראשוני כך שקובץ האימון הוא כל הדאטא סט שעבר pre-processing ולאחר אימון כל אחד מהמודלים ביענו פרדיקציה עבור הקובץ unlabeled.

את התוצאות העלנו ל-kaggle ולהלן ציוני ה-auc שהעניק:

 <b>NeuralNetworkPred.csv</b> Complete (after deadline) - 16m ago	0.84363	0.84072	<input type="checkbox"/>
 <b>NaiveBayesPred.csv</b> Complete (after deadline) - 17m ago	0.77545	0.78346	<input type="checkbox"/>
 <b>RandomForestPred.csv</b> Complete (after deadline) - 17m ago	0.88365	0.88819	<input type="checkbox"/>


זמני הריצה:

	Run time in sec
<b>ANN</b>	91.62
<b>NB</b>	1.41
<b>RF</b>	72.86

לאחר מכן הגדלנו את קובץ האימון פי 3 ובחנו את המודלים. להלן התוצאות:

 <b>NaiveBayesPredMoreTrain.csv</b> Complete (after deadline) - now	0.77545	0.78346	
 <b>NeuralNetworkPredMoreTrain.csv</b> Complete (after deadline) - 23s ago	0.8525	0.85113	
 <b>RandomForestPredMoreTrain.csv</b> Complete (after deadline) - 1m ago	0.88503	0.88873	

ראינו כי יש שיפור בכל המודלים ולכן בחרנו להגדיל את קובץ האימון פי 10 ולאמנם מחדש. להלן התוצאות:

 <b>NeuralNetworkPred10TimesTrained.csv</b> Complete (after deadline) - now	0.85573	0.85456	<input type="checkbox"/>
 <b>NaiveBayesPred10TimesTrained.csv</b> Complete (after deadline) - 20s ago	0.77545	0.78346	<input type="checkbox"/>
 <b>RandomForestPred10TimesTrained.csv</b> Complete (after deadline) - 1m ago	0.88405	0.88815	<input type="checkbox"/>

זמני ריצה

	Run time in sec
<b>ANN</b>	512
<b>NB</b>	7.52
<b>RF</b>	1577.1

ניתן לראות כי עבור מודל NB הגדלת קובץ האימון לא שינתה בהרבה (לא להגדלה של פי 3 או 10). לעומתו, מודל RF בעל ביצועים גבוהים יותר בהגדלת סט האימון פי 3, אך בעל ביצועים פחותים יותר בהגדלה פי 10.

עבור המודל ANN כל הגדלה של קובץ האימון שיפרה את הביצועים.

ניתן לראות כי המגמה עקבית למידי עם ההערכה הפנימית חוץ מכך ש-NB סיפק את תוצאת ה-AUC הטובה ביותר בהערכה הפנימית ואת הגרועה ביותר בהערכה החיצונית (הציון שניתן מקגל).

### נעשה ניסיון לבצע אנסמבל המשלב את תוצאות שלושת המודלים יחדיו.

נעשו שלושה ניסיונות כאלו:

(1) באופן נאיבי נתן משקול שווה לכל אחת מהתוצאות של המודלים



aggregated\_prediction.csv

Complete (after deadline) · 16h ago

0.85192

0.86222

(2) משקול תוך התחשבות בתוצאות הערכה חיצונית- כלומר מודל בעל הערכה חיצונית גבוהה יותר קיבל משקול גבוהה יותר ולהפך (לדוג' RF קיבל את הציון הטוב ביותר)



aggregated\_prediction.csv

Complete (after deadline) · 16h ago

0.85293

0.86331

(3) האנסמבל השלישי והמורכב יותר- מודל NB בעל ציון מדדי auci recall הגבוהים ביותר, ובעל ציון precision הנמוך ביותר. משמעות הדבר הזו בעינינו היא שמודל זה מרבה לחזות מוות של חולה ולכן FP שלו גבוה, אך מנגד כאשר המודל חוזה כי החולה לא ימות, חיזוי זה הוא חזק יותר מאשר החיזוי של שאר המודלים, כלומר ה-TN שלו גבוה. לכן ניסינו לבצע משקול לפי התיאור הבא- עבור דגימת חולה, כארש NB נתן חיזוי שבו החולה לא ימות, נתנו משקול גבוה יותר עבור NB לעומת שאר המודלים, ולהפך- כאשר NB חזה כי החולה ימות נתנו לו את המשקול הנמוך ביותר.



aggregated\_prediction.csv

Complete (after deadline) · 16h ago

0.79151

0.79956

מאחר ותוצאות האנסמבל של הניסיון הראשון והאחרון לא יוצאות דופן באף אחד מהניסיונות ואף נמוכות יותר, זנחנו את הרעיון להשתמש בו.

הניסיון השני שמשקלל את כל אחד מהמודלים במשקל שונה לפי ההערכה החיצונית, נתן ביצועים דומים מאוד למודל RF שהוא בעל הביצועים הגבוהים ביותר, ולכן השערנו כי שיכול להיות כי בעולם האמיתי הוא יכול להיות אפילו יותר טוב.

## סיכום ותובנות

בפריקט הזה ניסינו להשתמש בגישות שונות כלפי הדאטא מבחנת מורכבות הניתוח שעשינו. כבר מלכתחילה בשלב האבלואציה (10 fold cross) בחרנו מודלים מרמות מורכבות שונות- מרמה פשוטה יותר (כמו naïve bayse) ועד לרמה מורכבת (כמו רשת נוירונים).

גם כאשר המשכנו לשלב הבא של הניתוח (מבחנים מול הדאטא הלא מתויג והערכה חיצונית ע"י Kaggle) השתמשנו בשלושה מודלים ברמות שונות, שלכל אחד יתרונות וחסרונות בהיבטים שונים- לדוגמא המודל הפשוט ביותר שבחרנו NB בעל מדדי הערכה הכי נמוכים מלבד מדד הערכה AUC שקיבל את הציון הגובה ביותר משאר המודלים. בנוסף, מודל זה בעל הביצועים המהירים ביותר (זמן אימון נמוך משמעותית). זאת ועוד, במודל RF קיבלנו את התוצאות הטובות ביותר מבחינת המדדים אך מצד שני הוא בעל זמן ריצה הארוך ביותר. בסופו של דבר בחירת שלושת המודלים נעשתה משיקולים שונים שיתחשבו ברמות מורכבות שונות תוך בחירה פנימית של הביצועים הטובים ביותר פר מודל על גבי הרצות שונות עם קומבינציות של היפר-פרמטרים.

בשלב האחרון רצינו לחשוב על שיטות שונות לאמן את המודלים שבחרנו עם ההיפר-פרמטרים האופטימליים עבור כל המודל, כדי לקבל את ההערכה החיצונית הכי מוצלחת. תחילה, הרצנו הרצת אימון ראשונית ונאיבית לכל אחד מהמודלים עם הסט מבחן הלא מתויג, ואת התוצאות העלנו ישירות להערכה חיצונית ע"י Kaggle. ראינו כי התוצאות אכן לא מרשימות במיוחד ומכאן פתחנו בשלב אופטימיזציה של תהליך האימון ע"י שיפור הן כמות ה-binnings והן גודל סט האימון, ואף ניסיון לביצוע אנסמבל המשלב את שלושת המודלים הנבחרים.

הגדלת כמות ה-Bins שיפר בצורה דרמטית את ביצועי המודלים. הגדלת סט האימון שיפר באופן חלקי (תלוי במודל ותלוי בכמות ההגדלה). ביצוע אנסמבל היה דומה בתוצאותיו לתוצאות של המודל הטוב ביותר שהוא random forest (בניסיון אנסמבל ספציפי).

לסיכום, למדנו כי המודל המורכב הוא לא תמיד הכי טוב והמורכבות צריכה להיות תואמת לדאטא עצמו ולדומיין עצמו. בהקשר שלנו, הינו בוחרים להמשיך עם מודל random forest ולהמשיך לבחון את האופציה לאנסמבל.