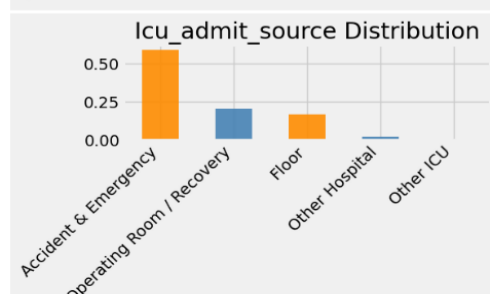
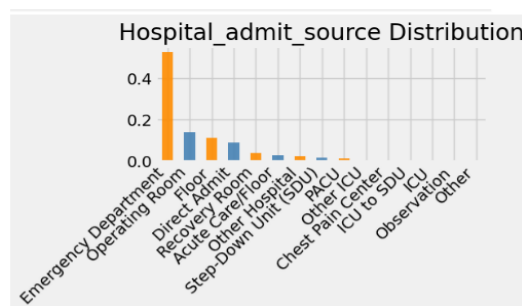
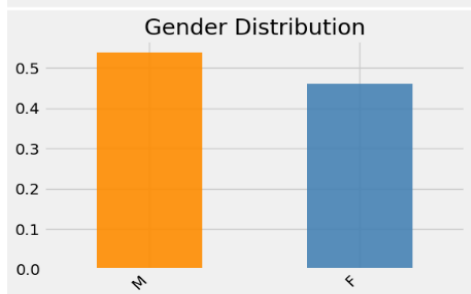
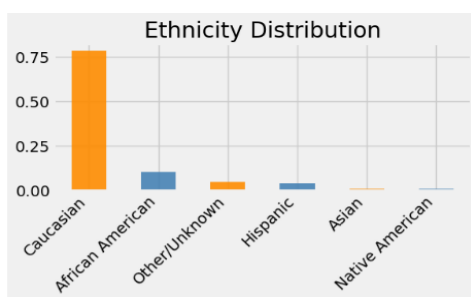


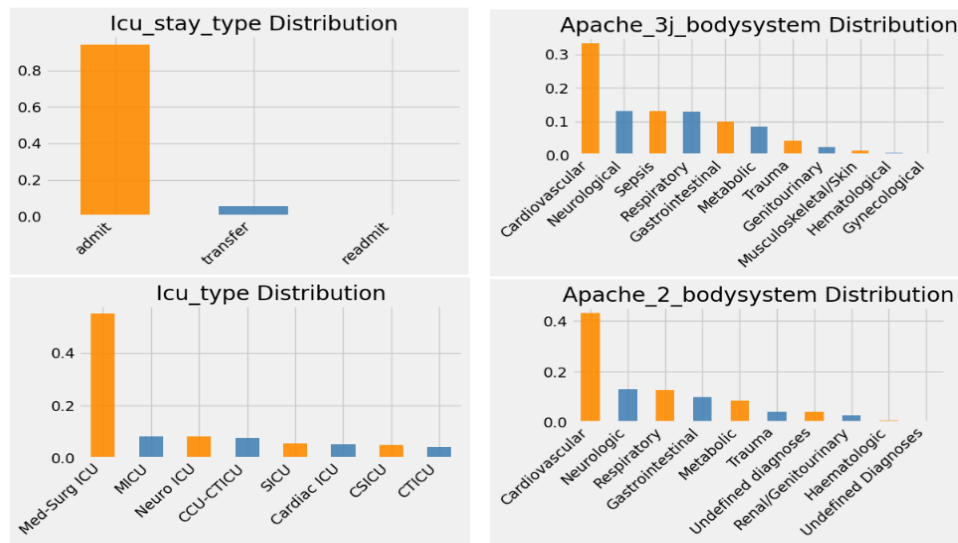
Data Science in the Industry - תרגיל 1

עיבוד מקדים

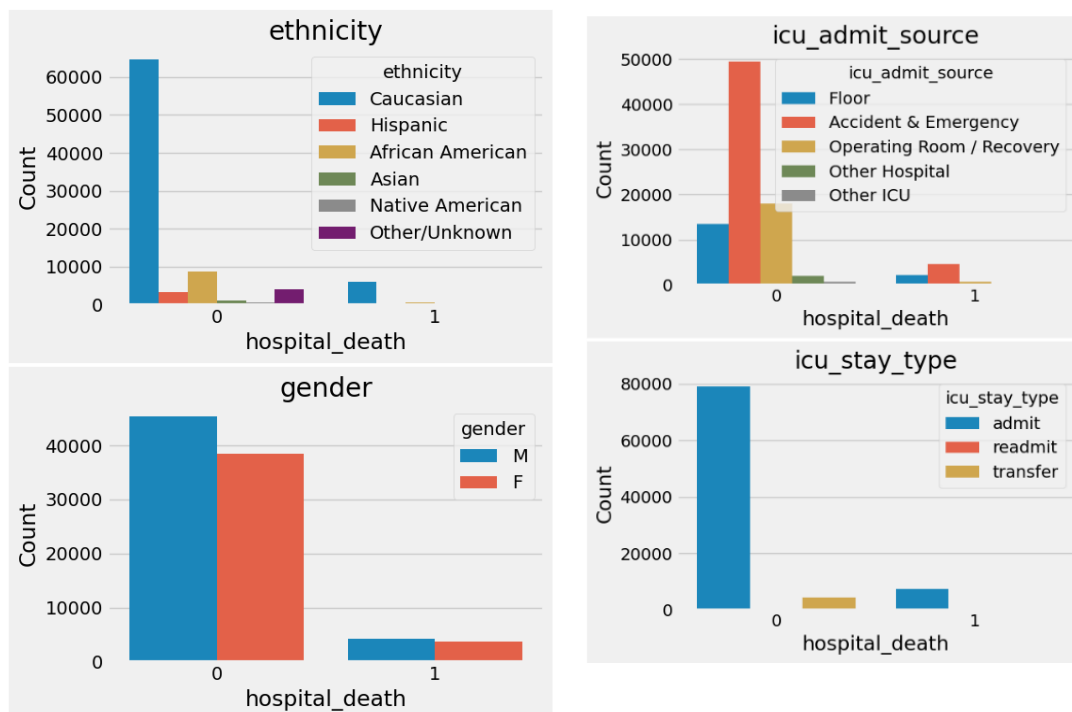
בקוד שסופק, נבצע מספר שלבים עבור ניתוח נתונים ראשוני בכדי לזהות את המאפיינים החשובים מהקובץ האימון. נתאר את עיקרי השלבים:

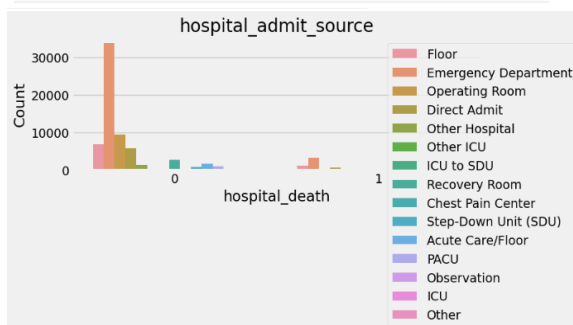
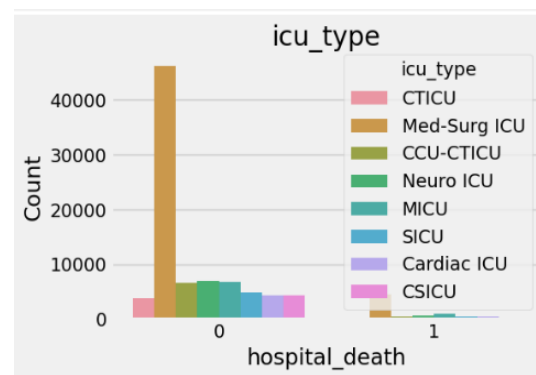
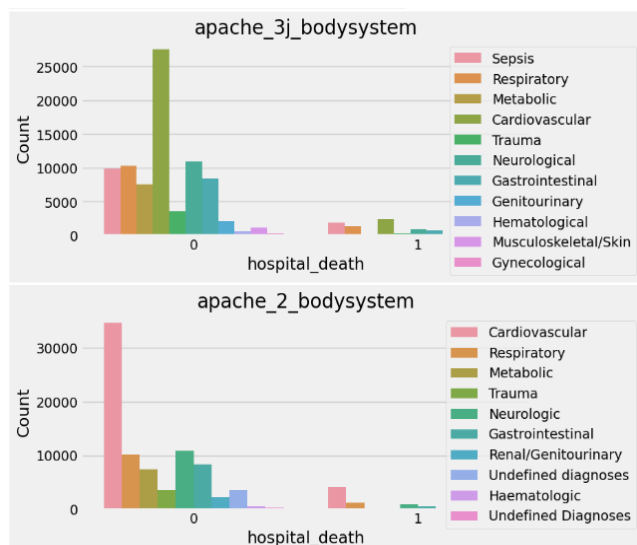
1. ייבוא הספריות הדרושות: מתבצעת ייבוא של הספריות הנדרשות לניתוח נתונים, ביניהן pandas, numpy, seaborn, matplotlib, scikit-learn ואחרות.
2. טעינת המערכת: טעינה של קבצי הנתונים האימון, הבדיקה, המילון ותבנית הפתרון.
3. בדיקת עמודות עם ערכים ייחודיים: נערך בדיקה על עמודות בסט האימון והמבחן כדי לזהות עמודות שמכילות ערכים ייחודיים או את אותו מספר ערכים ייחודיים כמספר השורות הכולל. מהבדיקה שביצענו קיבלנו את העמודות הבאות:
['encounter_id', 'patient_id', 'readmission_status']
עמודות אלו למעשה לא מועילות לנו מפני שכל רשומותיהן אחידות. ולכן הן מועברות לרשימת cols_to_drop ומוסרות מסט הבדיקה המבחן.
4. ניתוח מאפיינים קטגוריים: נבחין אילו מאפיינים הינם קטגוריאליים ונקבל את:
['ethnicity', 'gender', 'hospital_admit_source', 'icu_admit_source', 'icu_stay_type', 'icu_type', 'apache_3j_bodysystem', 'apache_2_bodysystem']
עבור כל מאפיין קטגורי במערכת הנתונים, ייצרנו גרף עמודות כדי להציג את התפלגות הקטגוריות. פעולה זו מספקת תובנות בנוגע לתדירות ואחוזי הקטגוריות השונות בתוך כל מאפיין ותאפשר לנו לשים לב אם יש מאפיין שכדאי לנו לתת לו התייחסות מיוחדת בהמשך.
גרפי התפלגות ערכים בעמודות קטגורליות:





5. ניתוח מאפיינים קטגוריים עם מוות בבית החולים: עבור מאפיינים קטגוריים שיש להם יותר מ-10,000 מופעים, ניצור גרף עמודות כדי להציג את התפלגות מקרי המוות באופן חוזר על הקטגוריות השונות בתוך כל מאפיין. הגרף מציג את התפלגות הקטגוריות ואחוזי ההופעה של כל קטגוריה בתוך כל מאפיין קטגורי. ניתוח זה מאפשר קבלת תובנות חשובות לגבי המאפיינים הקטגוריים והם עשויים לשפר תהליכי ניתוח בהמשך:





- ethnicity - מראה שמעל 78% מהפציינטים הם לבנים, ושיעור ההופעה של פציינטים עם מוצא אזורי אחר/לא ידוע יחסית נמוך. מידע זה יכול לספק רקע חשוב על האוכלוסיות המרכזיות ולעזור בהבנה של התפלגות ההופעה של פציינטים ביחד לתוצאה הסופית.
- gender – מראה שזהות המינים מחולקת ביחס די זהה, עם 53.9% גברים ו-46.1% נשים. מידע זה מספק רקע על ההתפלגות המינית של הפציינטים.
- hospital_admit_source - מציג התפלגות נפוצה על פני הכניסה לבית החולים, כגון מחלקת מיון במרפאה חירום וחדר הפעולות. מידע זה מאפשר להבין את האופן שבו הפציינטים מתמקדים במחלקות מסוימות ויכול לשפר את תהליכי המיון הראשוני של הפציינטים.

שלבים אלו של ניתוח נתונים ראשוני מסייעים בהבנת המערך הנתונים, בזיהוי המאפיינים החשובים ובהבנה של מגמות פוטנציאליות או יחסים עם משתנה המטרה (hospital_death). הוויזואליזציות והסטטיסטיקות שנמצאות בגרפים האלו מספקים ניתוח נוסף של הנתונים ולפיתוח המודל. מבחינתנו, הסתכלות על הנתונים בצורה זו עזרה להכיר טוב יותר את ה-DATA אך בהמשך המודל, לא נתנו להם התייחסות משמעותית. נגיד אבל שכן אפשר להסיק מהם מסקנות נרחבות במידת הצורך פשוט במקרה שלנו, ראינו כי יש להם השפעה זניחה.

התמודדות עם ערכים חסרים

הסרת מאפיינים-

אנו משתמשים בפונקציה שנקראת `missing_values_pracent` כדי לחשב ולהציג מידע על הנתונים החסרים. הפונקציה מחשבת את כמות הנתונים החסרים בכל עמודה ואת אחוז הנתונים החסרים בכל עמודה מתוך כל הנתונים החסרים בסך הכל.

החלטנו לטפל בנתונים החסרים על ידי הסרת עמודות שמכילות יותר מ-80% נתונים חסרים. ההחלטה להגדיר את הסף ל-80% התקבלה לאחר התבוננות על המידע שלנו. לקחנו בחשבון את הצורך לשמור על יכולת המודל ללמוד מידע וכמובן, לא רצינו "להיפרד" מכמות גדולה מידי של מאפיינים מפני שגם מצב זה יכול להסית מידי את תוצאות המודל. לצד זה, עמודות עם כמות גבוהה מאוד של נתונים חסרים עשויות להטיל אי ודאות על התוצאות המשוערות של המודל.

לכן, קביעת הסף ל-80% נראתה כהחלטה הגיונית. עוד עמד לגד עיננו המחשבה על מה הוא אחוז העמודות הללו (עם מעל 80% ערכים חסרים) מכלל העמודות:

	Total	Percent
h1_bilirubin_min	84619	92.265001
h1_bilirubin_max	84619	92.265001
h1_lactate_min	84369	91.992411
h1_lactate_max	84369	91.992411
h1_albumin_max	83824	91.398166
h1_albumin_min	83824	91.398166
h1_pao2fio2ratio_max	80195	87.441257
h1_pao2fio2ratio_min	80195	87.441257
h1_artierial_ph_max	76424	83.329517
h1_artierial_ph_min	76424	83.329517
h1_hco3_min	76094	82.969699
h1_hco3_max	76094	82.969699
h1_artierial_pco2_min	75959	82.822501
h1_artierial_pco2_max	75959	82.822501
h1_wbc_max	75953	82.815958
h1_wbc_min	75953	82.815958
h1_artierial_po2_min	75945	82.807236
h1_artierial_po2_max	75945	82.807236
h1_calcium_max	75863	82.717826
h1_calcium_min	75863	82.717826
h1_platelets_min	75673	82.510658
h1_platelets_max	75673	82.510658
h1_bun_max	75091	81.876070
h1_bun_min	75091	81.876070
h1_creatinine_min	74957	81.729962
h1_creatinine_max	74957	81.729962
h1_diasbp_invasive_max	74928	81.698342
h1_diasbp_invasive_min	74928	81.698342
h1_sysbp_invasive_min	74915	81.684167
h1_sysbp_invasive_max	74915	81.684167
h1_mbp_invasive_max	74844	81.606751
h1_mbp_invasive_min	74844	81.606751
h1_hematocrit_max	73420	80.054082
h1_hematocrit_min	73420	80.054082

ראינו שמדובר ב-34 מאפיינים שהם 18% מכלל המאפיינים. זה נראה לנו אחוז הגיוני "לנקות" מכלל העמודות כך שישמר לנו מגוון מספיק עשיר של מאפיינים. כמו כן, אנו עדיין לא יודעים מה הן עמודות יותר חשובות ולכן נרצה לא להיות פזיזים בבחירת הסרת העמודות בשלב זה. בנוסף, עצם הגדרת ערך הסף ל-80% תשליך בכל הנראה על כך שלעמודות אלו לא תהיה יותר מידי מפני שהן חסרות נתונים ואין להן משמעות סטטיסטית גבוהה. הסרת העמודות מתבצעת על נתוני האימון (train) והבדיקה (test).

השלמת ערכים חסרים-

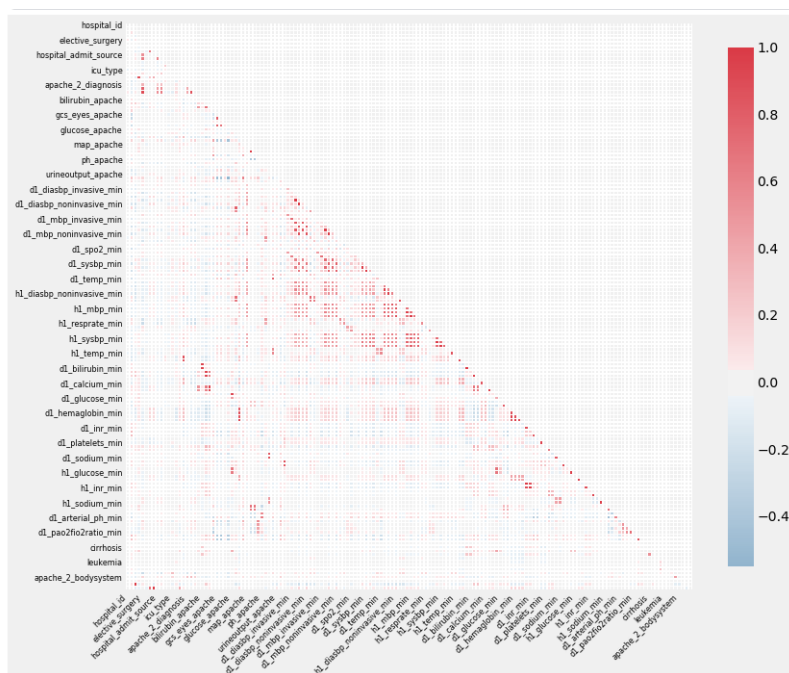
כדי להתמודד עם הנתונים החסרים בצורה המתאימה, ביצענו החלטות שונות בהתאם לסוג המאפיין. למאפיינים מספריים הנמצאים בהתפלגות נורמלית, החלטנו למלא את הנתונים החסרים בחציון של העמודה. בכך אנו משתמשים בערכים מרכזיים וקרובים להתפלגות המקורית כדי לשמור על איזון הנתונים ולא להשפיע על התפלגות המאפיין. למאפיינים מספריים שאינם בהתפלגות נורמלית, החלטנו למלא את הנתונים החסרים בערך הממוצע של העמודה. זה מאפשר לשמור על הסטטיסטיקה הכללית של המאפיין ולא להשפיע על התפלגות המאפיין. למאפיינים קטגוריים, החלטנו למלא את הנתונים החסרים בערך הנפוץ ביותר בעמודה. זה מסייע לשמור על התפלגות הקטגוריות המקוריות ולא להשפיע על התפלגות המאפיין. עם השלמת הנתונים החסרים, ביצענו כמובן את ההתאמות המתאימות על מערכות הנתונים האימון והבדיקה, בהתאם לשימור העמודות המשמעותיות והמתאימות למודל.

תיאור פעולות נרמול\חלוקה ל-bin\אנקודינג ויצירת מאפיינים-

1. יצירת בינים למשתנה BMI: אנו מגדירים Bin-ים עבור ערכי BMI, ומתייגים את הערכים המתאימים לקטגוריות בהתאם. התחומים מוגדרים כ-[0, 18.5, 25, 30, 100] והתוויות המתאימות הם 'תת משקל', 'משקל תקין', 'עודף משקל' ו'שמן'. הסיבה שבחרנו במשתנה זה היא בגלל הרלוונטיות הקלינית של משתנה זה. BMI משמש לעתים קרובות כאינדיקטור למצב משקל וסיכון בריאותי. סיווג BMI לקטגוריות רלוונטיות מבחינה קלינית עולה בקנה אחד עם הנחיות והמלצות שנקבעו לסיווג משקל. זה יכול לעזור לנו להעריך את השכיחות של קטגוריות משקל שונות בתוך מערך הנתונים ולנתח את השפעתם על תוצאות או משתנים אחרים בעלי עניין.
2. אנו משתמשים ב-MinMaxScaler כדי להפוך את ערכי הגיל והגובה לטווח 0-1 באופן סטנדרטי. אנו מעדכנים את הנתונים המתוקנים ומוסיפים עמודות חדשות בשם "age_normalized" ו-"height_normalized" למטריצות האימון והבדיקה.
3. קידוד לפי קטגוריות עבור המאפיינים הקטגוריים המעודכנים. אנו משתמשים ב-LabelEncoder לצורך קידוד הערכים הקטגוריים אשר יאפשר למודל לעבוד עם משתנים קטגוריים מפני שנייצגם כמשתנים מספריים.
4. שימוש בשיטת Shapiro-Wilk כדי לבדוק את ההנחה שהנתונים המספריים מתפלגים בצורה גאוסיאנית. אנו מדפיסים את הסטטיסטיקה הסטטיסטית וערך p של הבדיקה. על פי ערך p, אנו יכולים להסיק אם הנתונים נראים כמו תפוצה גאוסית או לא. תוצאת הבדיקה בשיטת Shapiro-Wilk מסקנת שהנתונים המספריים אינם מתפלגים בצורה גאוסית, זה עוזר לנו בהחלטה של שלא נבצע standardization.

הסרת מאפיינים עם קורלציה גבוהה-

השיטה שבה פעלנו כאן היא זיהוי תכונות עם קורלציה גבוהה על מנת להפחית את הממדיות של הנתונים ולשפר את ביצועי המודל. בכדי לזהות תכונות עם קורלציה גבוהה. תחילה רצינו לקבל מעט מושג לגבי המצב הנתון ב-dataset, ישנם הרבה מאפיינים במודל הנוכחי ולכן תחילה נעזרנו בגרף חום כדי לראות היכן אנחנו עומדים:



ממבט חטוף ניתן לראות כי אכן קיימת קורלציה גבוהה (מעל 0.8) בין מספר לא מבוטל של מאפיינים, ואם נראה תמונה זו יותר בירור נקבל:

	hospital_id	hospital_death	age	bmi	elective_surgery	ethnicity	gender	height	hospital_admit_source	icu_admit_source	...	hepatic_failure	immunosuppression	leukemia	lymphoma	solid_tumor_with_metastasis	apache_3j_bodysystem	apache_2_
hospital_id	NaN	0.001255	0.008472	0.012735	0.052123	0.009538	0.013650	0.027722	0.010713	0.010648	...	0.001356	0.000145	0.002980	0.002378	0.004711	0.043766	
hospital_death	NaN	NaN	0.106603	0.030535	0.093574	0.005274	0.006811	0.019299	0.031352	0.022554	...	0.038660	0.043743	0.029632	0.018624	0.050837	0.054570	
age	NaN	NaN	NaN	0.083680	0.066359	0.024442	0.024679	0.105900	0.039518	0.067800	...	0.019972	0.024855	0.029739	0.022959	0.025703	0.098180	
bmi	NaN	NaN	NaN	NaN	0.015645	0.025439	0.041314	0.055653	0.019364	0.026632	...	0.001817	0.030365	0.013116	0.009892	0.042335	0.042777	
elective_surgery	NaN	NaN	NaN	NaN	NaN	0.020595	0.029948	0.023507	0.527983	0.620941	...	0.034670	0.014682	0.017571	0.008208	0.015355	0.246577	
...	
apache_3j_bodysystem	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
apache_2_bodysystem	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
bmi_category	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
age_normalized	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
height_normalized	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	

לכן, בהצגת התכונות עם קורלציה גבוהה, החלטנו להגדיר קורלציה גבוהה לערך סף של 0.99 זאת מפני שניתן לראות כי יש הרבה מאפיינים בעלי אחוז קורלציה גבוהה, דבר שמאפשר לנו לדייק את החתך למאפיינים הגבוהים ביותר.

לבסוף התכונות שהתקבלו, בעלות קורלציה שגבוהה מערך הסף שהגדרנו הן:

```
paco2_for_ph_apache .1
d1_diasbp_noninvasive_max .2
d1_diasbp_noninvasive_min .3
d1_mbp_noninvasive_min .4
d1_sysbp_noninvasive_max .5
d1_sysbp_noninvasive_min .6
h1_inr_max .7
h1_inr_min .8
age_normalized .9
height_normalized .10
```

כל אלו, נמחקו מהנתונים ולא הועברו למודל. בסיום פעולה זו, עדכנו את רשימת התכונות המספריות בנתונים והסרנו את התכונות הגבוהות ביותר מהנתונים.

הפעלת שיטה לבחירת מאפיינים רלוונטים

בחרנו להשתמש ב-chi square כדי לבחור את המאפיינים המשמעותיים ביותר למודל החזוי שלנו. בדיקת chi square היא כלי סטטיסטי שמאפשר לנו לבחון את התלות בין מאפיינים למשתנה המטרה שלנו.

ביצענו שני מימושים שונים של בחירת מאפיינים באמצעות chi square (בינהן השיטה שהשתמנו בהרצאה 3), ראינו כי כל פונקציה כזו מעט שונה ולכן נותנת לנו תוצאות אחרות. החלטנו לבדוק שני מימושים שונים של chi square, להשוות את ההבדלים בין התוצאות ולבסוף להריץ את המודל עבור כל אחת מהם כך שנדע על בסיס התוצאות הסופיות להחליט מי מבין השיטות נתנה לנו ערכים שתרמו יותר למודל.

השיטה הראשונה:

```
X_train = X_train.applymap(abs)
selector = SelectKBest(chi2, k=10)
selector.fit(X_train, Y_train)
X_new = selector.transform(X_train)
```

הפלט שהתקבל עבור המאפיינים המשמעותיים:

```
apache_2_diagnosis .1
apache_3j_diagnosis .2
bun_apache .3
glucose_apache .4
urineoutput_apache .5
d1_sysbp_min .6
d1_bun_max .7
d1_bun_min .8
d1_glucose_max .9
d1_arterial_po2_max .10
```

השיטה השנייה (זהה להרצאה):

```
X = modified_data.loc[:,modified_data.columns!='hospital_death']
y = modified_data[['hospital_death']]
selector = SelectKBest(chi2, k=10)
selector.fit(X, y)
X_new = selector.transform(X)
```

הפלט שהתקבל עבור המאפיינים המשמעותיים:

- 1. elective_surgery
- 2. apache_post_operative
- 3. gcs_eyes_apache
- 4. gcs_motor_apache
- 5. gcs_verbal_apache
- 6. intubated_apache
- 7. ventilated_apache
- 8. d1_bun_max
- 9. d1_lactate_max
- 10. d1_lactate_min

ההבדל העיקרי בין שני השיטות מתרחש בשלב העיבוד המוקדם של הנתונים. בשיטה הראשונה, השתמשנו בפונקציה `abs()` לשינוי כל הערכים השליליים לחיוביים בסט האימון. זה מתבצע משום שבדיקת חי בריבוע מחייבת ערכים שאינם שליליים, מאחר וזהו חישוב מתמטי שאין לו פרשנות משמעותית עם ערכים שליליים.

בשיטה השנייה, לעומת זאת, לא ביצענו שינוי של הנתונים לפני שהפעלנו את בחירת המאפיינים. השיטה השנייה הוחלה על נתונים ששמרו על הפורמט המקורי שלהם, מה שאולי מאפשר ניתוח מעודכן יותר של המאפיינים המשמעותיים.

בסופו של דבר, **בחרנו בשיטה השנייה** (רשימת המאפיינים השניה שמוצגת), מאחר שהיא הציגה תוצאות מעניינות ומשמעותיות יותר בהמשך הפרויקט. דינמיקה זו מציינת את חשיבותו של שלב העיבוד המוקדם של הנתונים, ואת האפשרות שלו להשפיע על הממצאים של בדיקת חי בריבוע. בהמשך נציג את הפער וההשלכות ברמת התוצאה הסופית של בחירה זו.

בחירת שיטות קלסיפיקציה שונות על מנת למדל את בעיית הסיווג

בחרנו לבדוק 6 מודלים שעניינו אותנו במיוחד.

תחילה, השתמשנו בפונקציה `train_test_split` מספריית `sklearn` כדי לחלק את הנתונים לסטים של אימון ובדיקה.

לאחר מכן, הגדרנו שישה מודלים שונים של למידת מכונה:

1. Gradient Boosting
2. Logistic Regression
3. Decision Tree
4. Random Forrest (שיטת אנסמבל)
5. Naive Bayes
6. Neural Network

בחירה זו של מודלים מאפשרת מגוון של אלגוריתמים, כולל שיטות ליניאריות ולא ליניאריות, כדי לתפוס יחסים שונים אפשריים בין המאפיינים למשתנה המטרה. הרצנו את המודלים במספר תצורות שונות. נסביר כעת על כולן ונשווה בין התוצאות. במבחן התוצאה אנחנו מתמקדים בפרמטרים של: Time, Accuracy, Recall ו-Precision. כמו כן, נסתכל על קובץ ה-טסט שכן, מטרתנו לבסוף לקבל על בסיסו את החיזוי הטוב ביותר.

תחילה הרצנו את המודלים עם ההיפר פרמטרים הבאים:

LightGBM:

- `model__learning_rate`: מהירות בה המודל למד והתאים את הפרמטרים שלו. מהירות למידה גבוהה גורמת למודל ללמוד מהר, אך יכולה לגרום לו להתעלם מהערכים האופטימליים.
הוגדר ל- [0.001, 0.01, 0.05] כדי לנסות שלושה קצבי למידה שונים: גבוה, בינוני ונמוך.
- `model__n_estimators`: זהו מספר העצים שבונה המודל. מספר גדול של עצים יכול להוביל לגבורה-למידה (overfitting), בעוד מספר קטן מדי של עצים יכול לא לתפקד טוב מספיק.
הוגדר ל- [100, 50, 20] מנימוק זהה לזה שבפרמטר הקודם.

Logistic Regression:

- `model__C`: הפרמטר המגדיר את חומרת הגרדיאנט. הוגדר ל- [10.0, 1.0, 0.1]

Decision Tree:

- `model__max_depth`: עומק המרבי של העץ. הוגדר ל- [5, 10]
- `model__min_samples_split`: זהו מספר הדוגמאות הנדרשות לחלוקת צומת בעץ. הוגדר ל- [2, 5, 10]

Random Forest:

- model__n_estimators: כמו ב-LightGBM, זהו מספר העצים ביער.

- model__max_depth: כמו ב-Decision Tree, זו העומק המרבי של העץ.

באשר ל- Naïve Bayes ו- Neural Network: לא הוגדרו הייפר-פרמטרים מיוחדים למודלים אלו.

תוצאות הרצת המודלים

נסיון ראשון:

ביצענו הרצה עם המודלים כפי שתארנו, ו-cross validation 10 עבורם. כמו כן השתמשנו ב- 10 הפיטצ'רים הראשונים שקיבלנו מהרצת פונקציית חי בריבוע (הפיטצ'רים שהתקבלו מהרצת הפונקציה הראשונה).

התוצאות שהתקבלו הינן:

Model	Training Accuracy	Training Precision	Training Recall	\
LightGBM	0.921126	0.689825	0.155341	
Logistic Regression	0.920608	0.660089	0.163875	
Decision Tree	0.920458	0.694929	0.138590	
Random Forest	0.921998	0.684789	0.177149	
Naive Bayes	0.870547	0.328315	0.478982	
Neural Network	0.920499	0.644991	0.173989	

Model	Test Accuracy	Test Precision	Test Recall	\
LightGBM	0.922205	0.712766	0.168872	
Logistic Regression	0.920842	0.658080	0.177064	
Decision Tree	0.921550	0.722892	0.151229	
Random Forest	0.923077	0.698198	0.195337	
Naive Bayes	0.869596	0.326284	0.476371	
Neural Network	0.923240	0.654045	0.239445	

Model	Training Time	Classification Time	ROC AUC	\
LightGBM	237.509397	90.640439	0.581213	
Logistic Regression	5.885445	3.909564	0.584175	
Decision Tree	15.395049	2.242977	0.572869	
Random Forest	244.897298	92.128948	0.593670	
Naive Bayes	0.474934	0.880380	0.691605	
Neural Network	240.501501	445.459820	0.613725	

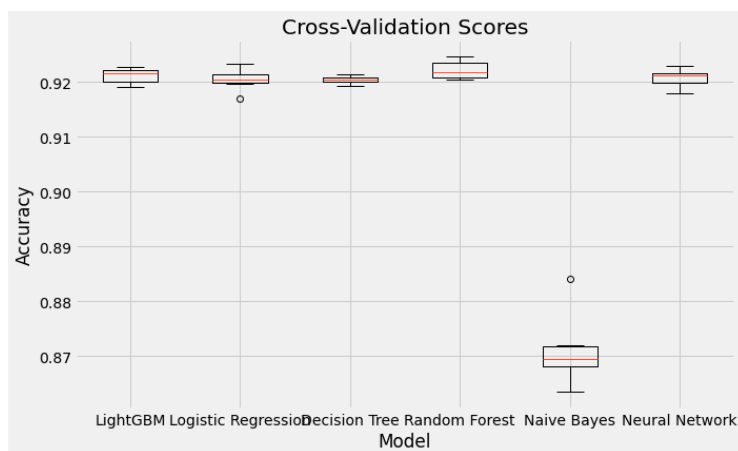
Model	FPR	\
LightGBM	[0.0, 0.00644545237526856, 1.0]	
Logistic Regression	[0.0, 0.00871329672952972, 1.0]	
Decision Tree	[0.0, 0.00549057054189544, 1.0]	
Random Forest	[0.0, 0.00799713535449988, 1.0]	
Naive Bayes	[0.0, 0.09316065886846503, 1.0]	
Neural Network	[0.0, 0.011995703031749821, 1.0]	

Model	TPR	\
LightGBM	[0.0, 0.1688720856962823, 1.0]	
Logistic Regression	[0.0, 0.1770636420919975, 1.0]	
Decision Tree	[0.0, 0.15122873345935728, 1.0]	
Random Forest	[0.0, 0.1953371140516698, 1.0]	
Naive Bayes	[0.0, 0.4763705103969754, 1.0]	
Neural Network	[0.0, 0.23944549464398235, 1.0]	

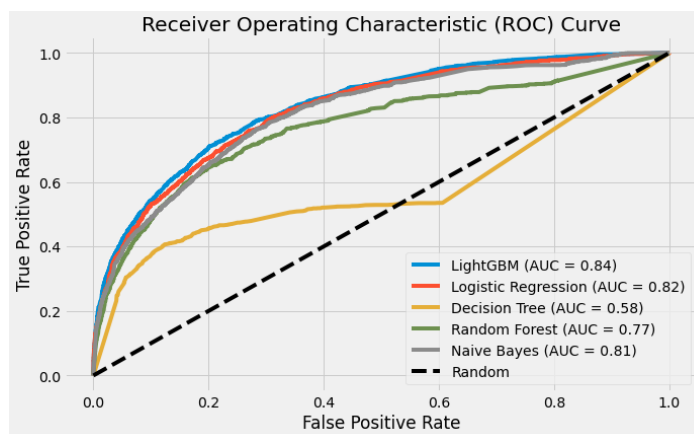
תוצאות מרכזיות ב-Test:

	Accuracy	Precision	Recall	AUC
	Natural Network 0.923	Decision Tree 0.722	Naïve Base 0.476	Naïve Base 0.691
Time for each model	240	15	0.5	0.5

: 10 Cross Validation

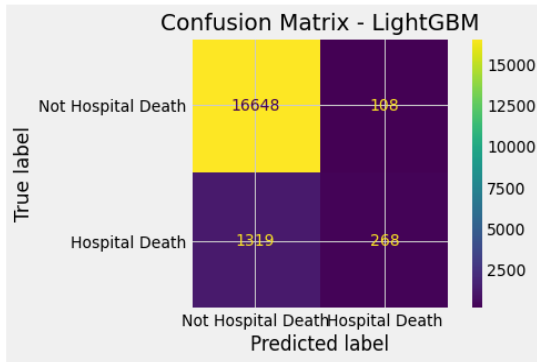


עקומת ROC:

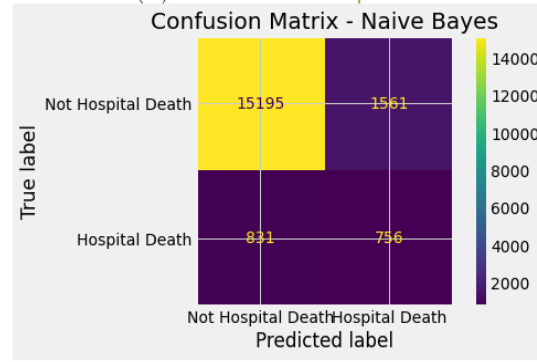


במקרה זה, מודל Naïve Bayes בעל ה-TPR הגבוה ביותר (0.476), אך גם ה-FPR שלו הרבה גבוה יותר (0.093) לעומת השאר.

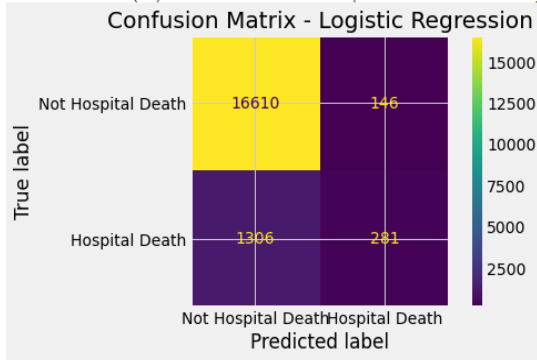
עם זאת, חשוב לציין כי אף על פי שלמסווג Naïve Bayes יש את ה-AUC ה-ROC הגבוה ביותר ו-TPR גבוה, הדיוק שלו בסטי האימון והבדיקה הוא הנמוך ביותר בין כל המודלים. זה מצוין שאף על פי שהוא מזהה נכונה חלק גדול מההופעות החיוביות, הוא גם מסווג באופן שגוי מספר גדול של הופעות שליליות כחיוביות. לכן, בהיבט של דיוק ייתכן שנבחר במודל שונה.



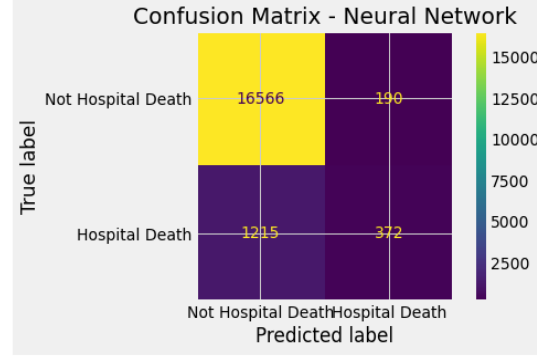
Explanation - LightGBM:
 True Negatives (TN): 16648 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 108 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1319 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 268 - Predicted as Hospital Death and Actually Hospital Death



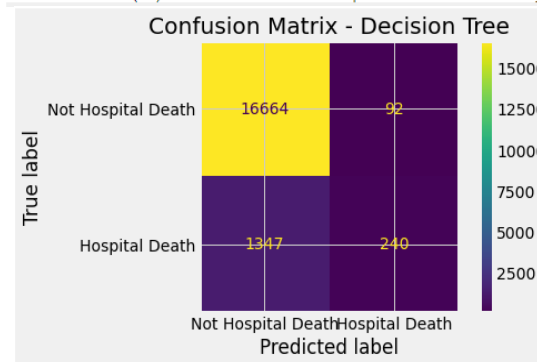
Explanation - Naive Bayes:
 True Negatives (TN): 15195 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 1561 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 831 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 756 - Predicted as Hospital Death and Actually Hospital Death



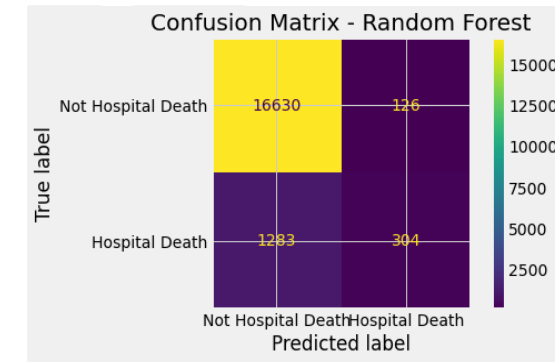
Explanation - Logistic Regression:
 True Negatives (TN): 16610 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 146 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1306 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 281 - Predicted as Hospital Death and Actually Hospital Death



Explanation - Neural Network:
 True Negatives (TN): 16566 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 190 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1215 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 372 - Predicted as Hospital Death and Actually Hospital Death









Explanation - Decision Tree:
 True Negatives (TN): 16664 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 92 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1347 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 240 - Predicted as Hospital Death and Actually Hospital Death



Explanation - Random Forest:
 True Negatives (TN): 16630 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 126 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1283 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 304 - Predicted as Hospital Death and Actually Hospital Death

תוצאות בקאגל:

Submission and Description	Private Score	Public Score	Selected
 Decision Tree_predictions1.csv Complete (after deadline) · now	0.55337	0.58497	<input type="checkbox"/>
 Random Forest_predictions1.csv Complete (after deadline) · 18s ago	0.5436	0.57684	<input type="checkbox"/>
 Naive Bayes_predictions1.csv Complete (after deadline) · 30s ago	0.51368	0.50964	<input type="checkbox"/>
 LightGBM_predictions1.csv Complete (after deadline) · 42s ago	0.54426	0.5725	<input type="checkbox"/>
 Neural Network_predictions1.csv Complete (after deadline) · 1m ago	0.55284	0.56521	<input type="checkbox"/>
 Logistic Regression_predictions1.csv Complete (after deadline) · 1m ago	0.60245	0.62335	<input type="checkbox"/>

את התוצאה הטובה ביותר קיבלנו עבור מודל ה-Logistic Regression עם ערך של 0.62

נסיון שני:

עשינו הכל באופן זהה לזה שתואר בנסיון הראשון, רק שבעת בחרנו את 10 הפיטצ'רים השניים שקיבלנו מוקדם יותר מהרצת פונקציית חי בריבוע. התוצאות שהתקבלו הן:

	Training Accuracy	Training Precision	Training Recall	\
Model				
LightGBM	0.921126	0.689825	0.155341	
Logistic Regression	0.920608	0.660089	0.163875	
Decision Tree	0.920458	0.694929	0.138590	
Random Forest	0.922039	0.682473	0.179678	
Naive Bayes	0.870547	0.328315	0.478982	
Neural Network	0.921221	0.683400	0.161346	

	Test Accuracy	Test Precision	Test Recall	\
Model				
LightGBM	0.922205	0.712766	0.168872	
Logistic Regression	0.920842	0.658080	0.177064	
Decision Tree	0.921550	0.722892	0.151229	
Random Forest	0.923459	0.708428	0.195967	
Naive Bayes	0.869596	0.326284	0.476371	
Neural Network	0.920842	0.696793	0.150599	

	Training Time	Classification Time	ROC AUC	\
Model				
LightGBM	231.926968	89.238540	0.581213	
Logistic Regression	5.819801	3.868256	0.584175	
Decision Tree	14.575157	2.162014	0.572869	
Random Forest	233.027854	86.976661	0.594164	
Naive Bayes	0.460418	0.853430	0.691605	
Neural Network	235.828972	441.445266	0.572196	

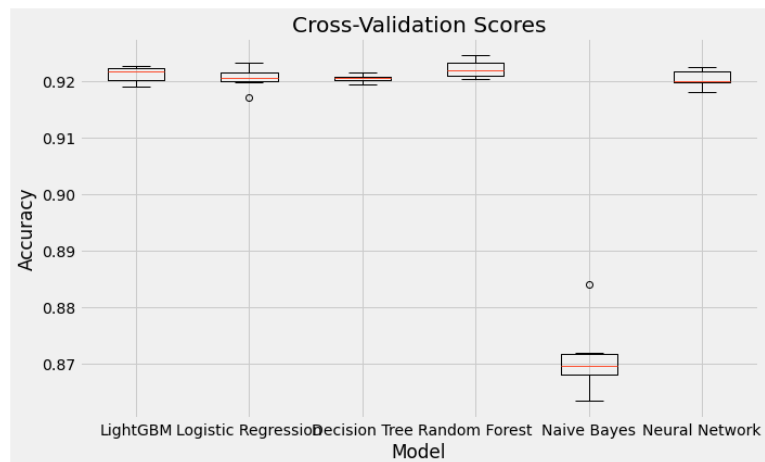
	FPR	\
Model		
LightGBM	[0.0, 0.00644545237526856, 1.0]	
Logistic Regression	[0.0, 0.00871329672952972, 1.0]	
Decision Tree	[0.0, 0.00549057054189544, 1.0]	
Random Forest	[0.0, 0.0076390546669849605, 1.0]	
Naive Bayes	[0.0, 0.09316065886846503, 1.0]	
Neural Network	[0.0, 0.0062067319169252805, 1.0]	

	TPR	\
Model		
LightGBM	[0.0, 0.1688720856962823, 1.0]	
Logistic Regression	[0.0, 0.1770636420919975, 1.0]	
Decision Tree	[0.0, 0.15122873345935728, 1.0]	
Random Forest	[0.0, 0.19596723377441713, 1.0]	
Naive Bayes	[0.0, 0.4763705103969754, 1.0]	
Neural Network	[0.0, 0.15059861373660996, 1.0]	

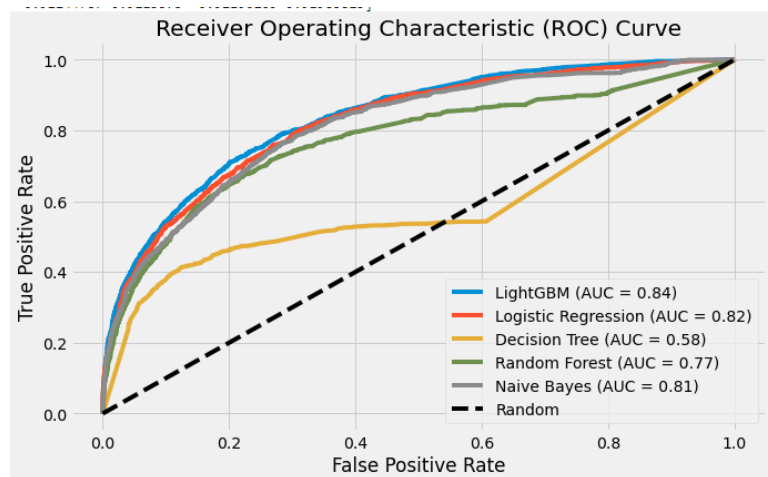
תוצאות מרכזיות ב-Test:

	Accuracy	Precision	Recall	AUC
	Random Forest 0.923	Decision Tree 0.722	Naïve Base 0.476	Naïve Base 0.69
Time for each model	233	14	0.46	0.46

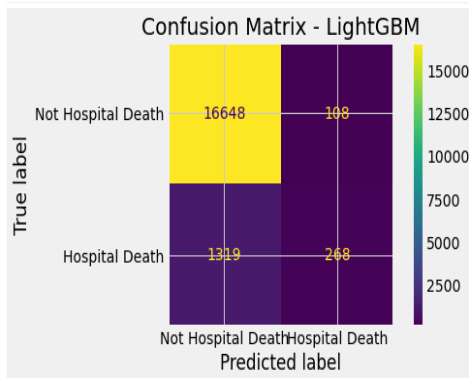
:10 Cross Validation



:ROC Curve

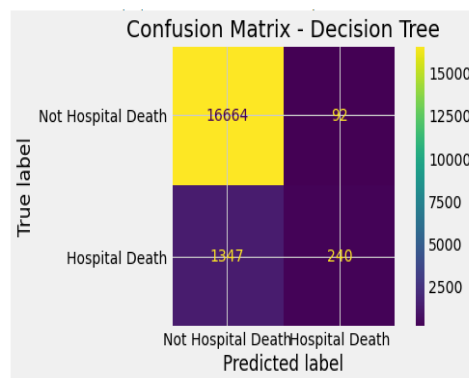


גם כאן יצאו תוצאות דומות להרצה הקודמת מבחינת ההצלחה של Naïve Base . אך מבחינת דיוק היינו בוחרים ב-Random Forest .



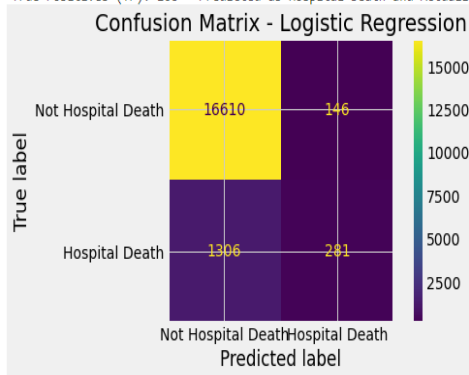
Explanation - LightGBM:

True Negatives (TN): 16648 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 108 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1319 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 268 - Predicted as Hospital Death and Actually Hospital Death



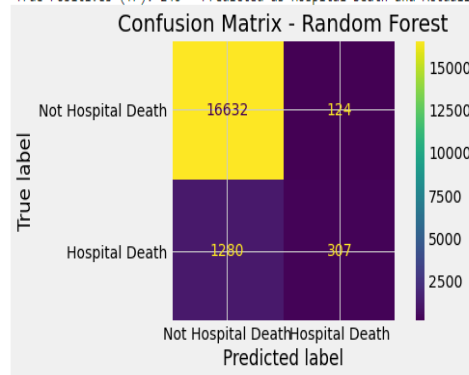
Explanation - Decision Tree:

True Negatives (TN): 16664 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 92 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1347 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 240 - Predicted as Hospital Death and Actually Hospital Death



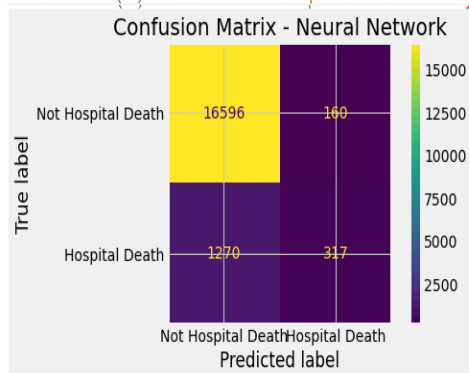
Explanation - Logistic Regression:

True Negatives (TN): 16610 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 146 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1306 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 281 - Predicted as Hospital Death and Actually Hospital Death



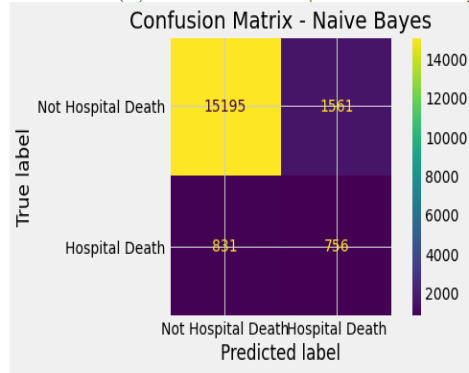
Explanation - Random Forest:

True Negatives (TN): 16632 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 124 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1280 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 307 - Predicted as Hospital Death and Actually Hospital Death



Explanation - Neural Network:







True Negatives (TN): 16596 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 160 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1270 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 317 - Predicted as Hospital Death and Actually Hospital Death



Explanation - Naive Bayes:

True Negatives (TN): 15195 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 1561 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 831 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 756 - Predicted as Hospital Death and Actually Hospital Death

תוצאות בקאגל:

Submission and Description		Private Score ⓘ	Public Score ⓘ	Selected
	Neural Network_predictions.csv Complete (after deadline) · now	0.64544	0.66409	<input type="checkbox"/>
	LightGBM_predictions.csv Complete (after deadline) · 16s ago	0.6112	0.6308	<input type="checkbox"/>
	Decision Tree_predictions.csv Complete (after deadline) · 37s ago	0.60667	0.62403	<input type="checkbox"/>
	Naive Bayes_predictions.csv Complete (after deadline) · 1m ago	0.67446	0.64996	<input type="checkbox"/>
	Random Forest_predictions.csv Complete (after deadline) · 1m ago	0.60716	0.62869	<input type="checkbox"/>
	Logistic Regression_predictions.csv Complete (after deadline) · 2m ago	0.71377	0.72554	<input type="checkbox"/>

את התוצאה הטובה ביותר קיבלנו עבור מודל ה-Logistic Regression עם ערך של 0.72, מדובר בשיפור משמעותי!

מסקנות והמשך מידול:

משתי ריצות אלו הגענו ל-2 מסקנות:

1. השפעת הפיטצ'רים עליהם נריץ את המודל היא בעלת השפעה מכרעת על התוצאות שנקבל בסוף. ראינו איך שינוי של 10 פיטצ'רים הביא לעלייה של 10% בהצלחת המודל. מסקנה זו הביאה אותנו להחלטה, שאנו רוצים להתבסס על יותר מ-10 פיטצ'רים משמעותיים גם על חשבון זמן ריצת המודל (כל עוד זה לא שינוי חריג מדי). לכן, החלטנו להריץ את פונקציית חי בריבוע ולקחת את 40 הפיטצ'רים המשמעותיים ביותר. כמו כן, כפי שכבר ציינו, הפונ' השנייה של חי בריבוע נתנה לנו תוצאות טובות יותר (שיפור של 10%) ולכן בחירת 40 המאפיינים תהיה באמצעותה.

הרצה שלישית עם 40 פיטצ'רים:

40 הפיטצ'רים שהתקבלו הינם:

```
'elective_surgery', 'apache_2_diagnosis', 'apache_3j_diagnosis',
'apache_post_operative', 'bun_apache', 'creatinine_apache',
'fio2_apache', 'gcs_eyes_apache', 'gcs_motor_apache',
'gcs_unable_apache', 'gcs_verbal_apache', 'intubated_apache',
'resprate_apache', 'ventilated_apache', 'wbc_apache', 'd1_diasbp_min',
'd1_heartrate_max', 'd1_mbp_min', 'd1_sysbp_min', 'h1_heartrate_max',
'h1_mbp_min', 'h1_mbp_noninvasive_min', 'h1_resprate_max',
'h1_sysbp_min', 'h1_sysbp_noninvasive_min', 'd1_bun_max', 'd1_bun_min',
'd1_creatinine_max', 'd1_creatinine_min', 'd1_inr_max', 'd1_inr_min',
```


'd1_lactate_max', 'd1_lactate_min', 'd1_wbc_max',
 'apache_4a_hospital_death_prob', 'apache_4a_icu_death_prob',
 'cirrhosis', 'hepatic_failure', 'immunosuppression',
 'solid_tumor_with_metastasis'

התוצאות שהתקבלו:

	Training Accuracy	Training Precision	Training Recall	\
Model				
LightGBM	0.928867	0.708224	0.298040	
Logistic Regression	0.924192	0.662840	0.246365	
Decision Tree	0.925733	0.648731	0.302939	
Random Forest	0.999905	1.000000	0.998894	
Naive Bayes	0.842156	0.297197	0.608249	
Neural Network	0.941284	0.805320	0.420986	

	Test Accuracy	Test Precision	Test Recall	ROC AUC	\
Model					
LightGBM	0.927057	0.679137	0.297417	0.642054	
Logistic Regression	0.923840	0.662116	0.244486	0.616335	
Decision Tree	0.923350	0.617380	0.299937	0.641166	
Random Forest	0.928038	0.698366	0.296156	0.642021	
Naive Bayes	0.837268	0.284126	0.579710	0.720686	
Neural Network	0.917625	0.547739	0.274732	0.626624	

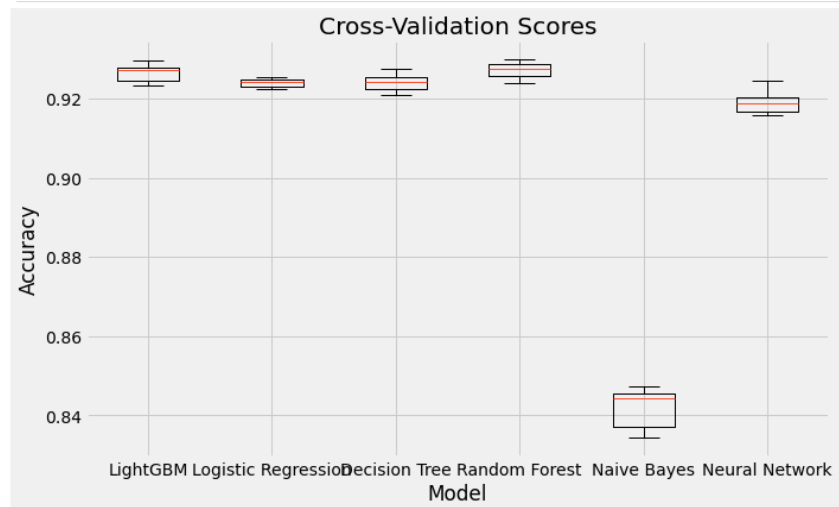
	FPR	\
Model		
LightGBM	[0.0, 0.013308665552637862, 1.0]	
Logistic Regression	[0.0, 0.011816662687992361, 1.0]	
Decision Tree	[0.0, 0.017605633802816902, 1.0]	
Random Forest	[0.0, 0.01211506326092146, 1.0]	
Naive Bayes	[0.0, 0.13833850560993077, 1.0]	
Neural Network	[0.0, 0.0214848412508952, 1.0]	

	TPR
Model	
LightGBM	[0.0, 0.297416509136736, 1.0]
Logistic Regression	[0.0, 0.24448645242596093, 1.0]
Decision Tree	[0.0, 0.2999369880277253, 1.0]
Random Forest	[0.0, 0.2961562696912413, 1.0]
Naive Bayes	[0.0, 0.5797101449275363, 1.0]
Neural Network	[0.0, 0.27473219911783237, 1.0]

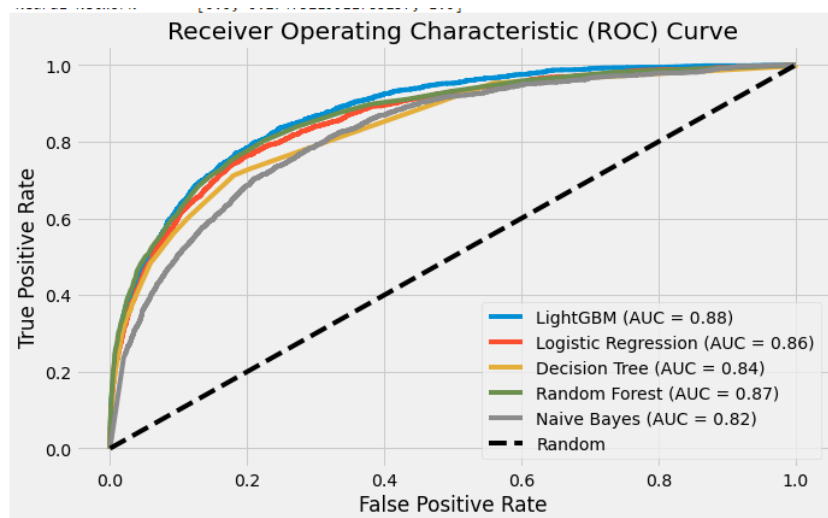
תוצאות מרכזיות ב-Test:

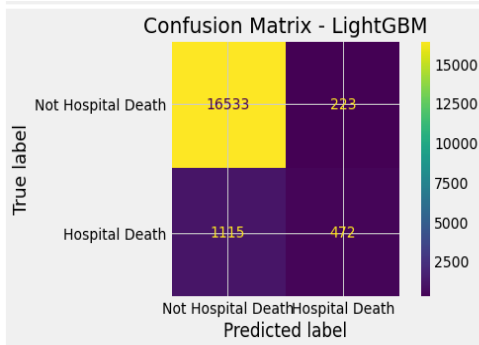
	Accuracy	Precision	Recall	AUC
	Random Forest 0.928	Random Forest 0.6983	Naïve Base 0.579	Naïve Base 0.72

:10 Cross Validation

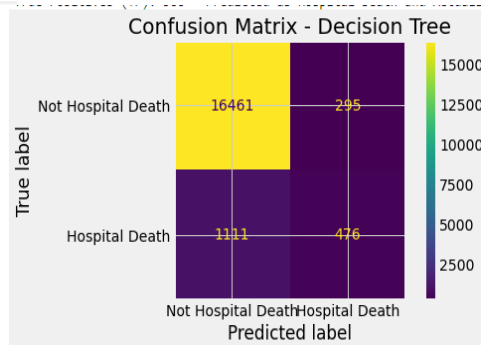


:ROC Curve

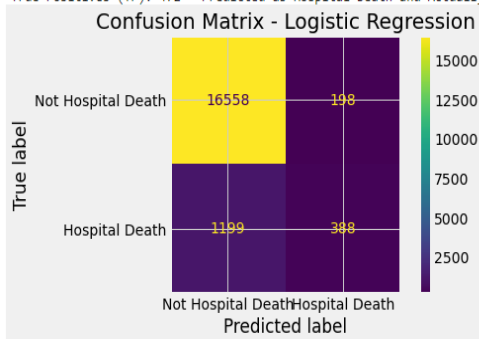




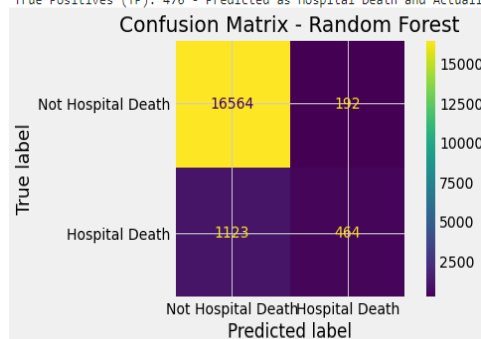
Explanation - LightGBM:
 True Negatives (TN): 16533 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 223 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1115 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 472 - Predicted as Hospital Death and Actually Hospital Death



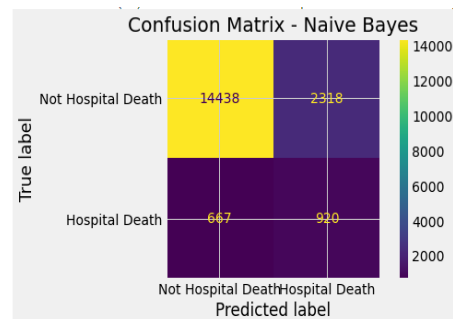
Explanation - Decision Tree:
 True Negatives (TN): 16461 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 295 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1111 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 476 - Predicted as Hospital Death and Actually Hospital Death



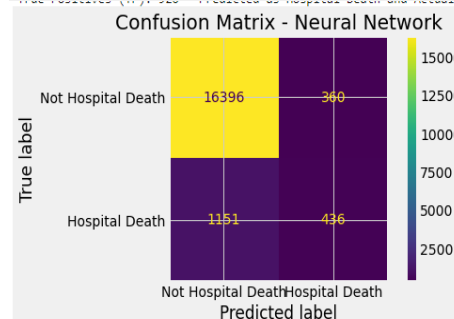
Explanation - Logistic Regression:
 True Negatives (TN): 16558 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 198 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1199 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 388 - Predicted as Hospital Death and Actually Hospital Death



Explanation - Random Forest:
 True Negatives (TN): 16564 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 192 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1123 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 464 - Predicted as Hospital Death and Actually Hospital Death



Explanation - Naive Bayes:
 True Negatives (TN): 14438 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 2318 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 667 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 920 - Predicted as Hospital Death and Actually Hospital Death



Explanation - Neural Network:
 True Negatives (TN): 16396 - Predicted as Not Hospital Death and Actually Not Hospital Death
 False Positives (FP): 360 - Predicted as Hospital Death but Actually Not Hospital Death
 False Negatives (FN): 1151 - Predicted as Not Hospital Death but Actually Hospital Death
 True Positives (TP): 436 - Predicted as Hospital Death and Actually Hospital Death

תוצאות בקאגל:

 Decision Tree_predictions_chi40.csv Complete (after deadline) · 4m ago	0.61905	0.65921	<input type="checkbox"/>
 Logistic Regression_predictions_chi40.csv Complete (after deadline) · 5m ago	0.76373	0.77939	<input type="checkbox"/>
 LightGBM_predictions_chi40.csv Complete (after deadline) · 5m ago	0.69742	0.7293	<input type="checkbox"/>
 Random Forest_predictions_chi40.csv Complete (after deadline) · 6m ago	0.68321	0.70936	<input type="checkbox"/>
 Neural Network_predictions_chi40.csv Complete (after deadline) · 6m ago	0.65517	0.66325	<input type="checkbox"/>
 Naive Bayes_predictions_chi40.csv Complete (after deadline) · 6m ago	0.51279	0.50797	<input type="checkbox"/>

את התוצאה הטובה ביותר קיבלנו עבור מודל ה-Logistic Regression עם ערך של 0.77, מדובר בשיפור יפה של 5%, אך ציפינו לשיפור יותר משמעותי.

תוצאות אלו הציגו בירור שהמודל הטוב ביותר כרגע הוא Logistic Regression, בנוסף בריצה זו גם מודל - LightGBM התקרב הראה שיפור משמעותי של 11% והגיע לתוצאה יפה. אבחנה זו, של המודלים שמיטבים באופן עיקרי עם הנתונים שלנו, יותר מהשאר, הביאה אותנו למסקנה השנייה- שימוש ב- Ensemble Predictions.

2. Ensemble Predictions - אנחנו רוצים לקחת בחשבון את כלל המודלים אך לתת לכל אחד מהם משקל אחר-ראינו כי יש מודלים שנתנו לנו באופן עקבי תוצאות יותר טובות כמו הרגרסיה הלוגיסטית. יחד עם זאת, ראינו כי יש גם מודלים שמובילים במדדים אחרים באופן עקבי לדוגמא, בשתי הריצות Naïve Base סיפק את ה-precision recall הגבוהה ביותר. לאחר שלקחנו בחשבון את התוצאות שקיבלנו ומעט ניסוי וטעייה, עם משקולים שונים, הגענו למשקול האופטימלי הבא:

```
weights = {
    'LightGBM': 0.25,
    'Logistic Regression': 0.35,
    'Decision Tree': 0.1,
    'Random Forest': 0.1,
    'Naive Bayes': 0.1,
    'Neural Network': 0.1
}
```

הרצנו את המודל בפעם השלישית עבור 40 פיצורים והוספת משקולות בהתאמה לתרומה של כל מודל. וקיבלנו את התוצאות הבאות

תוצאות קאגל:



ensemble_predictions (2).csv
Complete (after deadline) · 3m ago

0.81281

0.83331



קיבלנו דיוק של 0.83 !

סיכום ותובנות עיקריות

תוצאה זו של 83% מספקת אותנו בשלב זה ושמחנו לראות כיצד גם לאחר העיבוד המקדים יש עוד מקום רב לשיפור התוצאות (שיפור של 20% בין התוצאה הראשונה הטובה ביותר לאחרונה) אם בוחרים את המודל שנכון ביותר לסוג הנתונים שברשותך. ככל הנראה יש מודלים נוספים שלא בדקנו שאולי היו מספקים נו תוצאות טובות יותר אבל ברמת הלמידה הרגשנו שבחנו מודלים שעניינו אותנו והרגשנו שמתאימים לסוג הנתונים שלנו. כמו כן הופתענו לגלות תוצאות שלפעמים לא עמדו בקנה אחד עם הדברים שלהם ציפינו, למשל, בחלוקת המשקלים שביצענו תחילה ב-ensemble prediction, ראינו לנכון לתת ל-Naïve Base משקל גבוהה יחסית לאור התוצאות שהניב בריצות שעברו. אך למרות מה שחשבנו, לאחר מספר מדגמים, הופתענו לגלות שדווקא משקל גדול יותר עבור מודל ה-LightGBM הניב תוצאה טובה יותר.

הסתייגות- זמן הריצה של ensemble prediction לקח לנו כ-3 שעות. שזה יותר מפי 2 מהריצות הקודמות. הייתה לנו מחשבה לייעל את זמן הריצה על ידי כך שנריץ את המודל עבור פחות פיצ'רים, או להשתמש בפחות מודלים ב-ensemble predictions.

אך לבסוף, הסתכלנו על "התמונה הגדולה" והיא שמדובר במודל שמתעסק בחיזוי לגבי חיי אדם, אנחנו מאמינים שבמקרה זה לדיוק יש חשיבות ניכרת בהרבה על פני זמן הריצה. כמובן שיש מקרים בהם מדובר בחיי אדם וההחלטות צריכות להיות הרבה יותר מידידות, אבל מהסתכלות שלנו על משמעות החיזוי, 3 שעות נראה לנו זמן לגיטימי על מול התוצאה הסופית ולכן החלטנו להישאר איתו.