

ממ"ן 21 – תומר ריפין – 322230608

שאלה 1 - הגדרת הבעיה והכנת הנתונים

מטרת כריית המידע

מטרת כריית המידע בפרויקט הנוכחי היא לחזות לאיזה אדם יש סיכוי גבוה לחלות בשבץ באמצעות פרמטרים כגון: מין, גיל, מחלות עבר, עישון ופרמטרים נוספים.

הנחות: הנתונים נאספו באופן מהימן על ידי החוקרים.

הגדרת הנתונים

נעזר במידע אשר ניתן ב-kaggle על כל אחת מהעמודות ובניתוח מידע בסיסי על הנתונים עם הספריה Pandas בפייטון. בנוסף, נעזר בחיפושי אינטרנט ובמאמרים המצורפים כדי ללמוד על יחידות המידה של חלק מהעמודות.

נוכל לראות בטבלה למטה סיכום על כל עמודה שהשתמשנו בה בניתוח זה, העובדה החשובה ביותר שביתן לראות מניתוח זה היא שההתפלגות בעמודת המטרה שלנו stroke אינה אחידה. אחוז האנשים שקיבלו שבץ קטן משמעותית מאחוז האנשים שלא (~5% בערך).
סה"ב רשומות לפני הניקוי - 5110

שם עמודה	תיאור התכונה	יחידות מידה	סוג הנתונים	תחומי ערכים	נתונים חסרים
Gender	מין	Male Female Other	קטגורי		אין
Age	גיל	שנים	נומרי	[0.08-82]	אין
Hypertension	האם יש להם יתר לחץ דם		בינארי	0,1	אין
heart_disease	האם חוו מחלת לב		בינארי	0,1	אין
ever_married	האם היה נשוי בעבר		בינארי	Yes, No	אין
work_type	סוג העבודה בה האדם עובד	Private Self-employd Govt_job Children Never_worked	קטגורי		אין
Residence_type	סוג שכונת המגורים שבה התגורר האדם	Urban Rural	קטגורי		אין

avg_glucose_level	רמת הגלוקוז הממוצעת	$\frac{mg}{dL}$	נומרי	[55.12, 271.74]	אין
Bmi	מדד שמטרתו לתאר את האם משקל אדם תקין	$BMI = \frac{weight}{height^2} = \frac{kg}{m^2}$	נומרי	[10.3, 97.6]	יש – חסרים ערכים אצל 201 אנשים
smoking_status	האם האדם עישן או מעשן בעברו	formerly smoked, never smoked, smokes, Unknown	קטגורי		יש – 1544 עם סטטוס עישון לא ידוע
Stroke	האם קיבל שבץ		בינארי	0, 1	אין

שלבי ה-KDD

מטרת כריית המידע: מטרת כריית המידע בפרויקט הנוכחי היא לחזות לאיזה אדם יש סיכוי לחלות בשבץ באמצעות פרמטרים כגון: מין, גיל, מחלות עבר, עישון ופרמטרים נוספים.

איסוף ושמירת הנתונים:

הנתונים הנדרשים לתהליך כריית מידע התקבלו מ-Kaggle במלואם ולכן אין צורך באינטגרציה. הנתונים עצמם מקורם במחקר של חברת McKinsey & Company. במחקר המקורי של החברה היו במאגר 29,072 רשומות אבל רק בערך 30% התפרסם באתר ונגיש לנו. הנתונים שכן נגישים לנו הם נתונים על 5110 אינדיבידואלים שגרים בבנגלדש. בנוסף, לפני הפרסום לאתר החוקרים ערכו מעט את הנתונים מהמקור אצל McKinsey & Company.

ניקוי הנתונים ועיבוד מוקדם:

בשלב הזה נחפש נתונים רועשים: נוודא שאין נתונים חסרים או ריקים. נוכל לטפל ברשומות כאלו על ידי השמה של הממוצע או של הערך הנפוץ ביותר במקומם. בנוסף, נוודא שאין חריגות קיצוניות בנתונים או נתונים לא הגיוניים, מכיוון שיש לנו כמות גדולה יחסית של נתונים, נוכל למחוק רשומות עם נתונים חריגים בצורה קיצונית אך נצטרך להיות זהירים ולא למחוק יותר מדי רשומות של אנשים שחלו בשבץ כי יש יחסית מעט כאלו. לכן, נוכל להעזר בשיטה נוספת – binning שמטרתה לבצע דיסקרטיזציה של ערכים רציפים יחסית לקבוצות וכך "להחליק" רשומות חריגות מדי. כדי לזהות האם יש לנו רשומות עם נתונים חריגים מדי, נעזר במידע נוסף מהאינטרנט. עם רשומות כפולות נתמודד בכך שנזהה אותן ונמחק אותן.

טרנספורמציות הנתונים ורדוקציה:

בשלב זה, נבחן האם אנחנו צריכים לבצע טרנספורמציה למידע שלנו. נדרש לעיתים לבצע פעולה

כזו כדי להתאים את הנתונים לכרייה. פעולות טרנספורמציה יכולות להיות נורמליזציה או דיסקרטיזציה.

בחירת שיטות וכלים לביצוע כריית מידע:

אשתמש ב-python בספריות Pandas, matplotlib לייצוג הנתונים בצורה גרפית ולסידור המידע. בנוסף, אעזר בספריה sklearn אשר מממשת את האלגוריתמים הפוטנציאליים (עצי החלטה, יערות אקראיים וכו') לסיווג לטובת תהליך כריית המידע.

כריית המידע:

יישום האלגוריתמים של כריית הנתונים שנבחרו בשלבים קודמים על הנתונים כדי לחלץ דפוסים או מגמות.

סקירת התוצאות וניתוחן:

בשלב הזה נבחן את התוצאות של המודל שיצרנו. נבדוק דיוק, פשטות, רלוונטיות וכו'. במקרה שתוצאות המודל לא יספקו, נחזור על השלבים שרשמנו למעלה שוב, על מנת לשפר את המודל ולהגיע לתוצאה מספקת.

הסקת מסקנות:

נייצג את המודל שהתקבל מהתהליך, ונוכל לבחון כל רשומה חדשה שנקבל ונוכל לחזות את הסיכוי של המטופל החדש לחטוף שבץ.

סקירת חלופות לכריית מידע

בהמשך לסעיפים א' וב', נבצע סקירה של 4 חלופות שונות לכריית מידע. נתייחס בכל אחת מהן ליתרונות והחסרונות שלה בהקשר הבעיה הנתונה.

1. Random Forest -

Random Forest משלב עצי החלטה מרובים כדי לשפר את הדיוק ולהפחית התאמת יתר. במהלך תהליך יצירת המודל, מאומנים עצי החלטה רבים על תת קבוצה אקראית של הנתונים והתכונות. כאשר כדי לקבל את ההחלטה בסוף מחפשים את התוצאה שהתקבלה על ידי רוב העצים האקראיים.

יתרונות:

- a. **דיוק:** משפר את הדיוק על פני עץ החלטה יחיד
- b. **רובוסטיות:** המודל פחות סביר שיבצע Overfit לעומת עץ החלטה יחיד מפאת שהוא מסתמך על ממוצע של עצי החלטה רבים

חסרונות:

- c. **יקר מבחינה חישובית:** ייצור כל העצים וחישוב התוצאה על כל אחד מהם מכביד מאוד על החישוב
- d. **לא טוב עם נתונים לא מאוזנים:** יכול להיות מוטה כלפי קבוצת הרוב בנתונים לא מאוזנים.

2. עץ החלטה ID3 מבוסס Information Gain – ID3 הינו אלגוריתם רקורסיבי אשר מייצר

עצי החלטה ויוצר פיצולים בעץ על בסיס קריטריון הנבחר. הקריטריון הנבחר אצלנו הוא Information Gain (רווח אינפורמטיבי) - תחילה מחושבת האנטרופיה של עמודת היעד המבוקשת לסיווג/וחיזוי, ובכל רמה בעץ נבחרת התכונה בעלת האנטרופיה המותנית הנמוכה ביותר, כלומר זו שתביא לרווח האינפורמטיבי המירבי לסיווג/חיזוי משתנה המטרה בהתאם לידיעת הערך של תכונת הפיצול. אלגוריתם זה רץ בצורה רקורסיבית עד לתנאי העצירה שבו לכל הרשומות בעלה יש סיווג זהה או שאין יותר תכונות לפצל על פיהן. לאלגוריתם זה קיים האילוץ שהוא עובד רק עם ערכים קטגוריאליים, כלומר דורש דיסקרטיזציה של הנתונים.

יתרונות:

- a. **פשטות:** אלגוריתם פשוט ליישום ופרשנות.
- b. **מהירות:** חישובים מהירים.
- c. **לא דורש הנחת לינאריות:** בניגוד לגרסיה לינארית, באלגוריתם זה, אין צורך בהנחה של קשר לינארי בין המשתנים.

חסרונות:

- a. **נטייה ל-overfitting:** קל להגיע באמצעות אלגוריתם זה למצב של overfitting עקב מעט רשומות ונתונים או עמודה עם אופציות רבות שלא ניתנות לדיסקרטיזציה אשר מקבלות תעדוף על ידי אלגוריתם זה.
- b. **לא מתמודד היטב עם נתונים חסרים**

3. עץ החלטה C4.5 עם קריטריון פיצול gain-ratio – עץ החלטה זה מנסה להתגבר על

הבעיה באלגוריתם ID3 עם information gain – הנטייה לבחירה של תכונות בעלות ערכים רבים על ידי בחירת קריטריון אחר לפיצול. Gain-ratio הוא קריטריון שמנסה "להעניש" תכונות מרובות ערכים על ידי חלוקה של ה-information שלהן ב-information של הפיצול.

יתרונות:

- a. **טוב לתכונות מרובות ערכים:** עשוי להביא תוצאות טובות יותר מאלגוריתמים אחרים למאגרי מידע עם תכונות מרובות ערכים שונים.
- b. **גמישות:** מתאים לעבודה עם תכונות קטגוריאליות ורציפות כאחד. באמצעות גזימת העץ יודע להפחית התאמת יתר ויודע להתמודד עם ערכים חסרים.

חסרונות:

- a. **ענפים לא מאוזנים:** המדד המשמש לפיצול מוטה לבחירת תכונות שמביאות לחלוקה לא מאוזנת של ענפים. בעץ הסופי לעיתים רבות יש ענפים שמשמעותית קצרים מכל השאר.

4. **עץ החלטה CART עם קריטריון פיצול gini-index** – תוצאת האלגוריתם הינה עץ החלטה בינארי בלבד. לאחר סיום הריצה מבצעים גיזום על העץ לטובת דיוק שלו.
Gini-index הינו קריטריון הפיצול בשיטה זו, gini-index מניח שכל התכונות רציפות וכי יש כמה וכמה נקודות פיצול לכל תכונה. האלגוריתם בוחר את הפיצול שיביא לרמת אי דיוק הנמוכה ביותר. אלגוריתם זה מעדיף פיצול לשתי קבוצות שיהיו יחסית שוות בגודלן.

יתרונות:

- a. מתאים במיוחד לתוצאות בוליאניות
- b. **גיזום:** לאחר בניית העץ המלא, האלגוריתם יוצר קבוצה של תתי-עצים גזומים ובוחר בעץ עם פונקציית העלות-סיבוכיות המינימלית על מנת לצמצם overfit. ולכן שיטה זו לרוב נותנת עצי החלטה קומפקטים ומדויקים יותר.

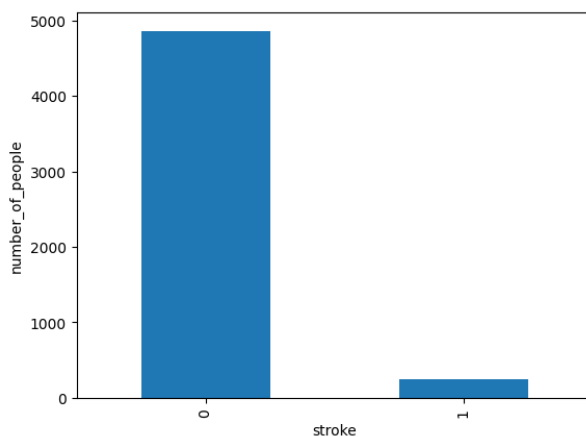
חסרונות:

- a. **יעילות:** זמן ריצת האלגוריתם גדול משמעותית משאר האלגוריתמים המוצעים.

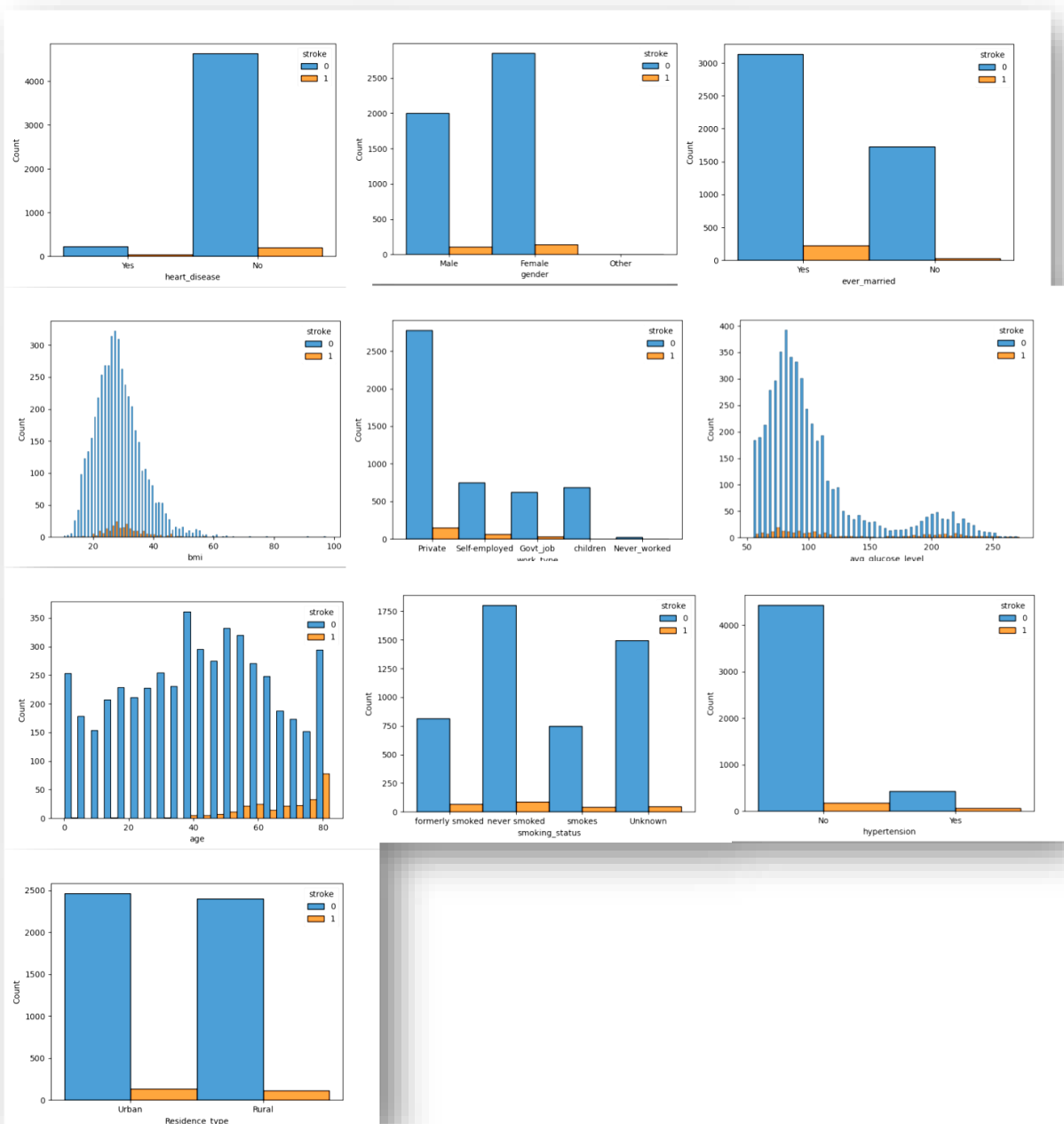
הכנת הנתונים וטיובם

תצוגה גרפית, סידור וניקוי הנתונים:

נשים לב שיש לנו מעט מאוד אנשים שקיבלו שבץ – עובדה זו תנחה אותי בהמשך ניקוי הנתונים בכך שאנסה להמנע כמה שאפשר ממחיקת שורות של אנשים שקיבלו שבץ כדי להמנע מאיבוד מידע יקר. להלן היסטוגרמה של העמודה שבץ במאגר המידע:



כדי לקבל תחושה טובה יותר של הנתונים, נייצר היסטוגרמות של העמודות השונות ביחד עם עמודת המטרה:



נוכל לשים לב לכמה תופעות מעניינות שינחו אותנו בהמשך –

1. אפשר לראות שיש קורלציה חיובית חזקה בין הגיל לבין שבץ.
 2. התפלגות התכונה avg_glucose_lvl (מעטה AGL) נראית לא נורמלית
- נמשיך בסידור של הנתונים בצורה שתהיה נוחה יותר לעבוד איתם בהמשך:

1. את העמודה `ever_married`, נחליף לבינארית כך ש-Yes יהיה 0 ו-No יהיה 1.
2. את העמודה `gender` נחליף גם לבינארית כך ש-Male יהיה 0 ו-Female יהיה 1. את השורה היחידה שיש בה `Other` נמחק כי היא יחידה ואותו יחיד לא קיבל שבץ ולכן אנחנו לא מאבדים נתונים קריטיים.
3. את העמודה `Residence_type` נחליף גם לבינארית כך ש-Rural יהיה 0 ו-Urban יהיה 1.
4. נמחק את העמודה `id` כי הנתונים בה אינם נותנים מידע על הסיכוי של האדם לקבל שבץ.

טיפול בערכים חסרים -

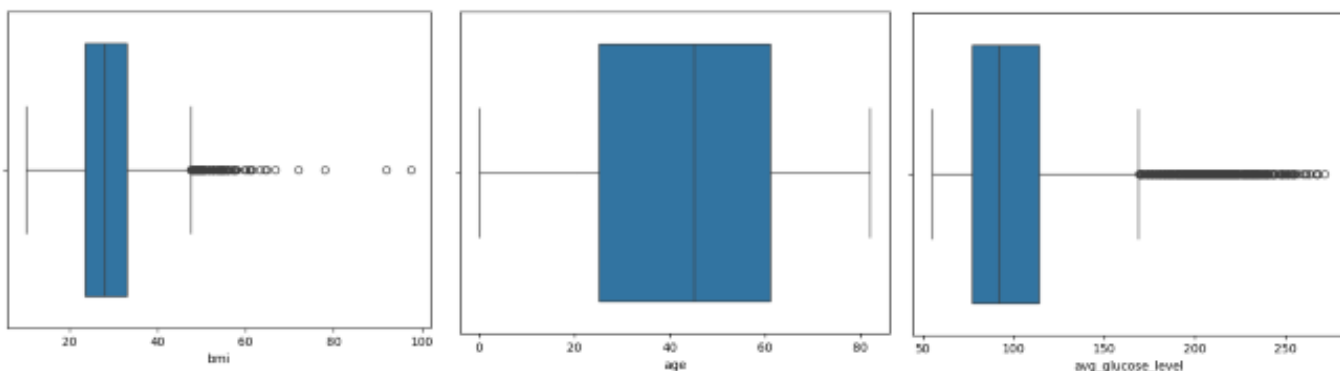
נטפל בערכים החסרים – כמעט בכל התכונות אין לנו ערכים חסרים, יש לנו בעמודת ה-BMI ובעמודת הסטטוס עישון (Unknown).

1. בעמודת ה-BMI אמלא את כל הערכים החסרים בערכי חציון ה-BMI.
2. בעמודת `smoking-status` נחליף את כל הערכים שהם גם Unknown וגם $\text{age} < 15$ או `work_type == children` להיות `never smokes`. אני יוצא מהנחה שרוב מוחלט של הילדים הקטנים בימינו לא מעשנים או מעשנים בכמות זניחה. לאחר פעולה זו צמצמנו את כמות ה-Unknown מ-1544 ל-909!

טיפול בערכים קיצוניים -

נחפש ערכי קיצון שיכולים להיות לא הגיוניים. נשתמש ב-BoxPlot כדי לקבל תחושה ויזואלית של האם בעמודות הנומריות יש ערכים חריגים.

נוכל לראות שבעמודת הגיל, אין לנו ערכים שנראים חריגים מדי אבל עמודות ה-BMI וה-AGL יש לנו זנב ימני יחסית ארוך של ערכים שחשודים כחריגים (שמחושבים על ידי ערכים שהם מעל 1.5IQR מערך הרביעון השלישי כאשר $\text{IQR} = Q_3 - Q_1$). לפי הויקיפדיה ומאמרים נוספים, אנשים עם רמת סוכר ממוצעת גבוהה הם ברמת סיכון גבוהה לסכרת ויש גם סכנות בריאותיות ב-BMI גבוה אך נתונים אלו אינם לא הגיוניים לחלוטין. לכן, אני מעדיף כמה שפחות למחוק שורות שיכולות לאבד מידע חשוב ואמחק רק ערכים קיצוניים מאוד ($\text{BMI} > 80$ ו- $\text{AGL} > 225$).



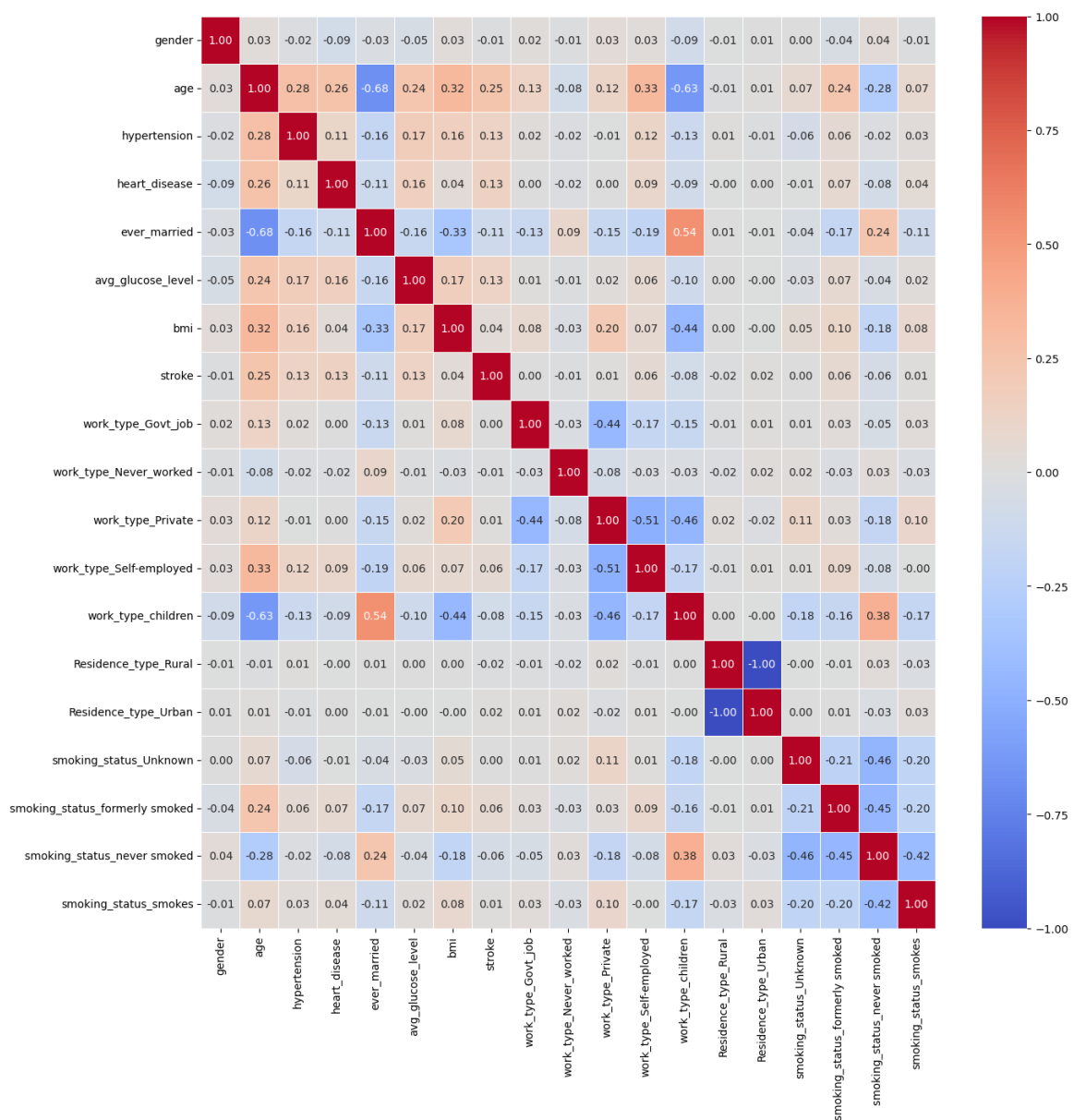
לאחר מחיקת הערכים הקיצוניים, נשארתי עם 4892 שורות במאגר.

– pd.get_dummies()

מכיוון שהמימוש הפייתוני של DecisionTree ו-RandomForest (אותן אבחר בהמשך לסיווג הנתונים) לא יודע להתמודד עם ערכים מילוליים. השתמשתי בפונקציה של חבילת הפייתון Pandas אשר לוקחת עמודות קטגוריאליות שמכילות יותר משתי אופציות וממירה אותן לעמודות בינאריות של האם העמודה בקטגוריה המסוימת או לא.

מציאת קורלציה בין תכונות -

כדי להבין טוב יותר את הנתונים, ואת היחסים בין העמודות, יצרתי heatmap אשר מראה את הקורלציה בין כל אחת מהעמודות:



ניתוח הגרף:

1. יש קורלציה שלילית בין age ו- ever_married וגם עם ו-work_type_children. זה כמובן הגיוני בתור נתונים שקשורים לגיל הנבדק (רוב הנבדקים שלא התחתנו היו צעירים).

2. עבור עמודת היעד – stroke ניתן לראות כמה דברים:

a. כמו שראינו בהיסטוגרמות הקשר הכי חזק ל-stroke זה כמובן הגיל.

b. ישנן עוד כמה עמודות עם קשר לא זניח (מעל 0.1):

i. Hypertension

ii. Heart_disease

iii. AGL

אני בכוונה מתעלם מ-ever_married כי כמו שראינו יש קורלציה חזקה בין age לעמודה זו בצורה שכנראה משפיעה על הקשר שלה ל-stroke.

תצוגה זו יכולה לעזור לנו להבין טוב יותר את הקשרים בין העמודות השונות ולנסות לחזות על אילו תכונות יתבסס המודל שנבנה. נוכל להסיק שכנראה בעץ החלטה, השורש כמעט תמיד יהיה age כי תכונה עם קורלציה חזקה מאוד. אפשר גם לחשוב האם אנחנו רוצים להוריד מהמודל תכונות עם קורלציה חזקה ל-age כמו ever_married כדי למנוע מהמודל שיבחר להתבסס אך ורק על הגיל של הנבדקים וכך להתעלם ממידע נוסף. אציין שלאחר שניסיתי למחוק את העמודות האלו לפני הלמידה של המודל, לא קיבלתי שינויים משמעותיים בתוצאות ולכן החלטתי להשאיר אותם.

דיסקרטיזציה לתכונות נומריות -

אני עומד להשתמש בשיטות סיווג מבוססות עצי החלטה שיודעות לעבוד טוב עם נתונים קטגוריאליים ולכן אבצע דיסקרטיזציה לנתונים הנומרים, הדיסקרטיזציה אף תוכל לעזור להקטין אף יותר את ההשפעה של הערכים הקיצוניים שעוד נותרו. ניסיתי 3 שיטות שונות לדיסקרטיזציה–

1. דיסקרטיזציה מבוססת מידע פומבי – לקחתי שיטות חלוקה נפוצות על פי מחקרי עבר וכך נוצרו הקטגוריות הבאות:

```
def categorize_AGL(x):
    if 0 <= x < 72:
        return "Low"
    if 72 <= x < 108:
        return "Normal"
    if 108 <= x < 126:
        return "Prediabetes"
    if 126 <= x:
        return "Diabetes"

def categorize_age(x):
    if 0 <= x < 10:
        return "Children"
    if 10 <= x < 18:
        return "Teens"
    if 18 <= x < 60:
        return "Adults"
    if 60 <= x:
        return "Elderly"

def categorize_BMI(x):
    if x < 16.5:
        return "Underweight (Severe)"
    if 16.5 <= x < 16.9:
        return "Underweight (moderate)"
    if 16.9 <= x < 18.4:
        return "Underweight (mild)"
    if 18.4 <= x < 25:
        return "Normal"
    if 25 <= x < 30:
        return "Overweight"
    if 30 <= x < 35:
        return "Obese (1)"
    if 35 <= x < 40:
        return "Obese (2)"
    if 40 <= x:
        return "Obese (3)"
```

2. גישת רוחב שווה – חילקתי את התכונות השונות לפי סלים ברוחב שווה כאשר יצרתי 5

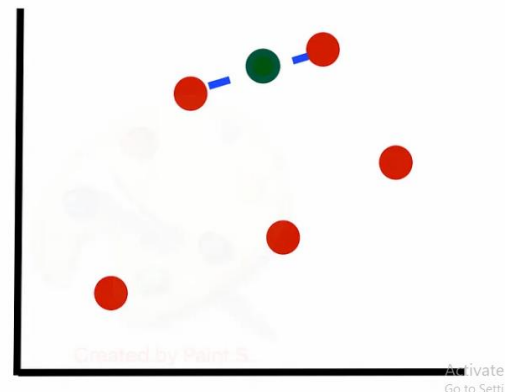
סלים שונים, זה נתן לי את התוצאות הטובות ביותר.

3. גישת עומק שווה – גם פה, חלוקה ל-5 תאים נתנה לי את התוצאות הטובות ביותר.

לבסוף, שיטת עומק שווה (Equal Frequency) נתנה לי את התוצאות הטובות ביותר ולכן הלכתי איתה.

Oversampling לטובת איזון עמודת המטרה על ידי Synthetic Minority Over-sampling Technique (SMOTE)

כבר בתחילת העבודה על הנתונים, ניתן היה לראות שההתפלגות של עמודת המטרה אינה מאוזנת. אנחנו רוצים שהמודל שלנו ידע לחזות חולי שבץ וכרגע יש לו דגימה קטנה מאוד ללמוד ממנה. נרצה לייצר עוד שורות באופן שלא יוצר over-fitting ולכן נשתמש באלגוריתם SMOTE. אלגוריתם זה מייצר באופן סינטי שורות נוספות כך - שורה סינטטית מיוצרת על ידי בחירה רנדומלית של שורה מה-minority class ומציאת השכן הקרוב ביותר שלה, לאחר מכן מגרילים מספר בין 0 ל-1 ובאמצעותו מייצרים דגימה סינטטית שהיא יושבת על הווקטור בין השורה שמצאנו לשורה הקרובה ביותר שלה.



בתמונה זו¹ – לקחנו את הדגימה השמאלית העליונה ביותר, חיפשנו את השכן הקרוב שלה וייצרנו דגימה סינטטית חדשה ביניהן.

השתמשתי באלגוריתם SMOTE עד שמספר הדגימות של אנשים שקיבלו שבץ יהיה זהה למספר הדגימות לאלו שלא.

חשוב מאוד להשתמש ב-SMOTE רק על נתוני האימון, אחרת יכול להיווצר מצב של data-leakage והמודל שלנו יהיה לא מדויק עבור נתוני אמת. בנוסף, חשוב לבצע את ה-SMOTE לפני שמבצעים החלקות לנתונים כמו דיסקרטיזציה, כדי שהדגימות הסינטטיות שנוצרות לאחר ריצת האלגוריתם כמה שיותר מגוונות וקרובות למקור.

¹ <https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88c>

טבלה מסכמת של הנתונים לאחר סידור וניקוי -

שם עמודה	תיאור התכונה	סוג הנתונים	תחומי ערכים	נתונים חסרים	איך טופלה
Gender	מין	בינארי	0,1	אין	
Age	גיל	נומרי	0,1,2,3,4	אין	בוצעה דיקרטיזציה שוות עומק
Hypertension	האם יש להם יתר לחץ דם	בינארי	0,1	אין	
heart_disease	האם חוו מחלת לב	בינארי	0,1	אין	
ever_married	האם היה נשוי בעבר	בינארי	0,1	אין	הומר מ-Yes ו- No ל-0 ו-1.
Work_type_self-employd	האם האדם הוא מועסק עצמאי	בינארי	0,1	אין	
Work_type_Govt_job	האם האדם הוא עובד ממשלתי	בינארי	0,1	אין	
Work_type_Children	האם האדם הוא ילד	בינארי	0,1	אין	
Work_type_never_worked	האם האדם מעולם לא עבד	בינארי	0,1	אין	
Residence_type	סוג שכונת המגורים שבה התגורר האדם	בינארי	0, 1	אין	
avg_glucose_level	רמת הגלוקוז הממוצעת	נומרי	0,1,2,3,4	אין	נמחקו ערכים קיצוניים, בוצעה דיסקרטיזציה שוות עומק
Bmi	מדד שמטרתו לתאר את האם משקל אדם תקין	נומרי	0,1,2,3,4	אין	נמחקו ערכים קיצוניים, ערכים חסרים מולאו בחציון, בוצעה דיסקרטיזציה שווה עומק
smoking_status_formaly_smoked	האם האדם עישן עישן בעברו	בינארי	0,1	אין	

	אין	0,1	בינארי	האם האדם מעולם לא עישן	smoking_status_never_smoked
	אין	0,1	בינארי	האם האדם מעשן	smoking_status_smokes
נשתמש ב-Smote כדי לייצר איזון בנתונים לאחר הפיצול לנתוני אימון ולמידה	אין	0, 1	בינארי	האם קיבל שבץ	Stroke

לאחר ניקוי וסידור הנתונים נשארו לי רק 909 שורות עם סטטוס עישון לא ידוע, החלטתי להוריד עמודה זו כי היא לא מוסיפה לנו מידע מתאים.

לאחר ביצוע ה-SMOTE תוך כדי ה-10-Fold cross validation היו במאגר הלמידה 8750 שורות, מתוכן חצי עם 1 == stroke וחצי עם 0 == stroke כאשר רוב השורות לאנשים שיש להם שבץ יוצרו בצורה סינטטית על ידי האלגוריתם.

שאלה 2 – סיווג

בחירת השיטות לסיווג הנתונים

בחרתי בשתי שיטות לסיווג הנתונים:

1. עץ החלטה מבוסס Gini-index

2. Random Forest

ניסיתי את כלל השיטות שתיארתי ב-4 חלופות אפשריות לסיווג לביצוע כריית המידע, בחרתי מתוכן בשתי שיטות אלו כי הן החזירו לי את התוצאות הטובות ביותר והן הכי מתאימות לביצוע פעולת הסיווג. Gini-index הוא מדד פיצול המתאים בעיקר לחיזוי תכונות בינאריות וראיתי שפיצול על Information-gain נטה לחוסר דיוק שנבע מהעדפה בעץ לתכונות רבות משתנים כמו smoking-work-type-i status. בנוסף, שיטת הסיווג RandomForest הכי עמידה בפני Overfitting מבין שאר האלגוריתמים שמתבססים על עץ החלטה אחד ויחיד.

פסאודו קוד לכל אחת מהשיטות שנבחרו

פסאודו קוד ל-Random Forest

```
# Initialize the RandomForest with parameters
```

```
Initialize RandomForest(num_trees, max_depth, min_samples_split)
```

```
# Fit the RandomForest model
```

```
Function fit(X, y):
```

```
    For each tree in num_trees:
```

```
        X_sample, y_sample = random_sample(X, y)
```

```
        # Train a decision tree
```

```
        tree = train_decision_tree(X_sample, y_sample, max_depth, min_samples_split)
```

```
        # Add the trained tree to the list of trees
```

```
    Add tree to trees
```

```
# Predict using the RandomForest model
```

Function predict(X):

Collect predictions from all trees

tree_predictions = collect_predictions(trees, X)

Majority vote to determine final prediction

final_predictions = majority_vote(tree_predictions)

Return final_predictions

פסאודו קוד לעץ החלטה מבוסס Gini-index

Initialize the DecisionTree with parameters

Initialize DecisionTree(max_depth, min_samples_split)

Fit the DecisionTree model

Function fit(X, y):

tree = build_tree(X, y, depth=0)

Return tree

Build the tree recursively

Function build_tree(X, y, depth):

If stopping_criteria_met(X, y, depth):

Return create_leaf(y)

best_split = find_best_split(X, y)

left_X, left_y, right_X, right_y = split_data(X, y, best_split)

left_subtree = build_tree(left_X, left_y, depth + 1)

right_subtree = build_tree(right_X, right_y, depth + 1)

Return create_node(best_split, left_subtree, right_subtree)

Calculate Gini index

Function calculate_gini_index(X, y, feature, value):

left_y, right_y = split_labels(y, feature, value)

gini_left = gini_impurity(left_y)

gini_right = gini_impurity(right_y)

gini = (gini_left * len(left_y) + gini_right * len(right_y)) / len(y)

Return gini

Calculate Gini impurity

Function gini_impurity(y):

proportions = calculate_proportions(y)

gini = 1 - sum(proportions^2)

Return gini

Check stopping criteria

Function stopping_criteria_met(X, y, depth):

If depth >= max_depth or len(y) < min_samples_split or all_labels_same(y):

Return True

Return False

Cart Pruning Algorithm (from Mark Last presentation)

1. Initialize the list of optimal trees with the maximal tree
2. Initialize $a=0$
3. Increase a until the tree ceases to be optimal

4. Find a new sub-tree, which is optimal with a new value of a
5. Add the new sub-tree to the list of optimal trees
6. If the new sub-tree has more than one terminal node, go to Step3. Otherwise,
Stop

תוצאות הניתוחים ומידת הדיוק של כל שיטה

מכיוון שהיו לנו יחסית מעט שורות במאגר עם אנשים שיש להם שבץ, ומכיוון שלחזות בדיוק גבוה מחלות זה קשה, עוד לפני שראיתי את התוצאות הראשוניות הנחתי שהתוצאות בחיזוי אנשים שיש להם שבץ לא יהיו טובות.

ואכן, בתוצאות הראשוניות שהמודלים החזירו לי, כשזה הגיע לחזי ראיתי שהמודלים בקושי הצליחו להבדיל בין חולים שקיבלו שבץ לאלו שלא ופספסו הרבה חולים.

ניסיתי לחשוב מה עולה יותר – FP (זיהוי שגוי של אדם כפוטנציאל גבוה לשבץ) או FN (פספוס ואי התרעה בפני אדם שעלול לחטוף שבץ). קראתי על מקרי שבץ ועל טיפול מניעתי וראיתי שבמקרים שבהם יש חשש לשבץ, אין צורך בסדרת בדיקות מחמירה ויקרה אלא לרוב ממליצים להתחיל לחיות אורח חיים בריא יותר, בין אם לאכול בריא או לעשות פעילות ספורטיבית וטיפול זה יחסית אפקטיבי ובמחיר מועט גם לחולה וגם לבית החולים. לעומת זאת, הטיפול לאדם שקיבל שבץ הוא יקר, החיים יכולים להרס, נדרש ליווי קפדני בבית חולים, תרופות וטיפולים יקרים ולרוב החולים אינם מתואששים בצורה מלאה מהתקרית. מפאת הסיבות הללו, היה לי קריטי שיהיו לי כמה שפחות FN (אנשים שעומדים לחלות בשבץ ולא הזהרתי אותם) ופחות היה אכפת לי מאחוזי FP גבוהים יותר.

האינסטינקט שלי אמר לי שכדאי לי להגדיר למודלים threshold נמוך יותר מ-0.5 כדי לחזות האם יש לאדם שבץ או לא. בעזרת הפונקציה המובנית predict_proba בדקתי מה ההסתברות של ה-sample להיות 0 או 1 והגדרתי שזה 0 רק עם ההסתברות קטנה מה-threshold אחרת זה 1. ובחברתי threshold-ים קטנים מ-0.5 כדי להעלות את הסיכוי ל-1 לדוגמה:

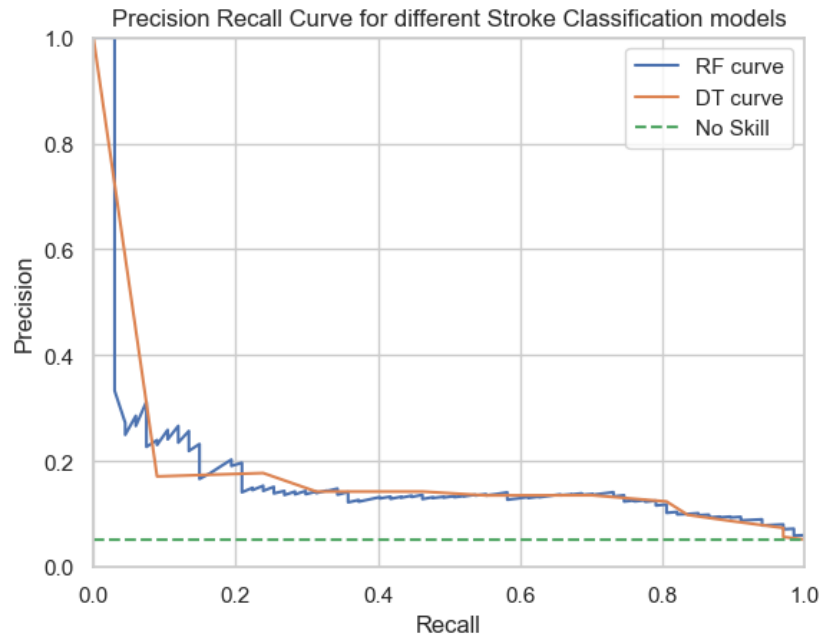
```
threshold = 0.45
y_proba = clf.predict_proba(X_test)[:,-1]
y_pred = (y_proba >= threshold).astype("int")
```

כדי לראות מה ה-threshold הטוב ביותר, השתמשתי ב-precision-recall-curve, גרף אשר משמש כדי לראות את tradeoff בין recall ל-precision על פני threshold-ים שונים.

אזכיר –

$$recall = \frac{TP}{TP+FN}, precision = \frac{TP}{TP+FP}$$

לכן אשתדל למצוא threshold שיתן לי recall כמה שיותר גבוה (סימן לכמות FN מועטה) ועדיין ישמור לי על Precision סביר.

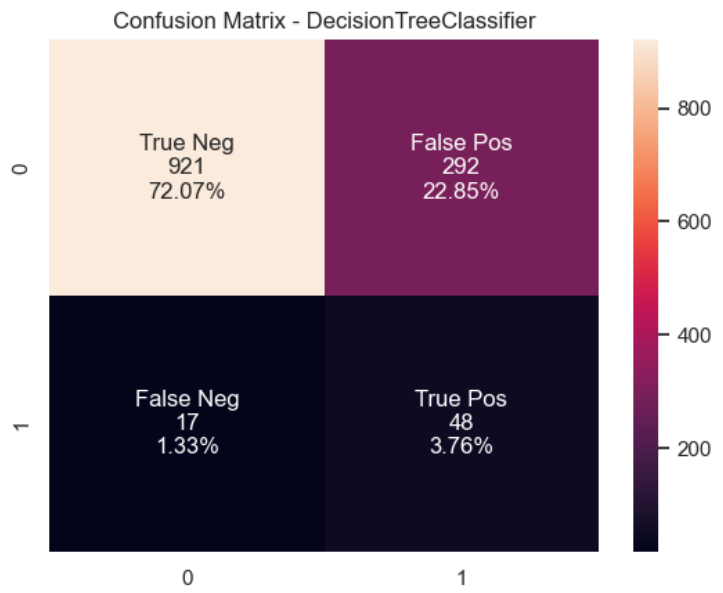
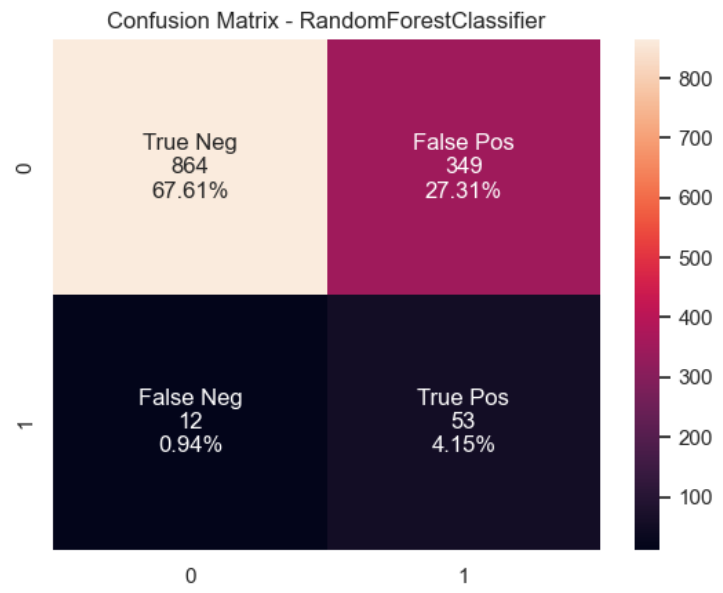


נוכל לראות בנקל מהגרף שכמעט לא משנה מה ה-threshold שאשים, ה-precision שלי יהיה נמוך מאוד. ולכן כיוונתי ל-threshold נמוך מאוד (0.15) שיתן לי recall גבוה ו-precision סביר יחסית אבל אז כשהסתכלתי על ה-confusion matrix ראיתי שהכמות של FP שלי יותר מדי גבוהה. מכיוון שהנתונים מאוד לא מאוזנים, על כל שיפור של חיזוי מדויק על כמה נבדקים בודדים שחולים בשבץ אני צריך לדווח על מאות נבדקים נוספים שיש להם פוטנציאל לשבץ למרות שזה לא נכון.

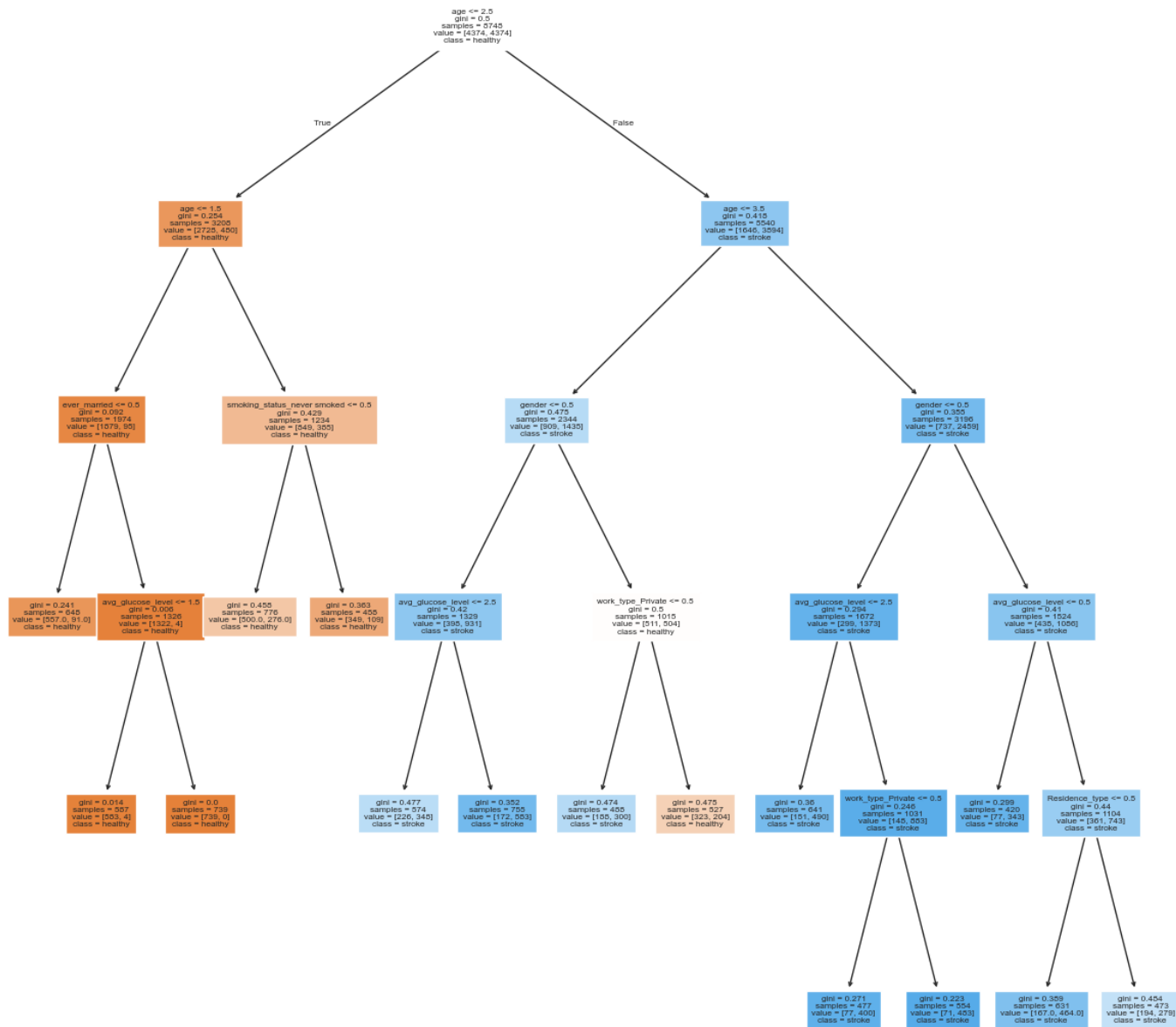
לכן, שיחקתי עוד עם ה-threshold עד שהגעתי לאחד שהייתי מרוצה ממנו ונתן לי את היחס הטוב ביותר שנראה לי סביר. ה-threshold היה 0.45, זה אכן קטן מ-0.5 כמו שצפיתי אבל לא אגרסיבי כמו שאולי ה-precision-recall curve יציע.

כידוע, יש עוד Settings חוץ מ-threshold שאפשר להוסיף לעצי החלטה ול-Random Forest. כדי למנוע Overfitting אני הגדרתי לעץ ההחלטה $\text{min_samples_leaf} = 400$ ו- $\text{max_depth} = 6$, ועבור ה-Random Forest הגדרתי רק $\text{Min_samples_leaf} = 200$.

להלן ה-confusion matrix שעשיתי עבור ה-threshold המתקבל בחלוקה ל-train ו-test של שני שלישי – שליש לשני המסווגים השונים (כאשר לפני הלמידה הורץ SMOTE לאיזון הנתונים על מאגר הלמידה).



אוסף את הויזואליזציה של ה-DecisionTree מבוסס Gini-index שיצרת.

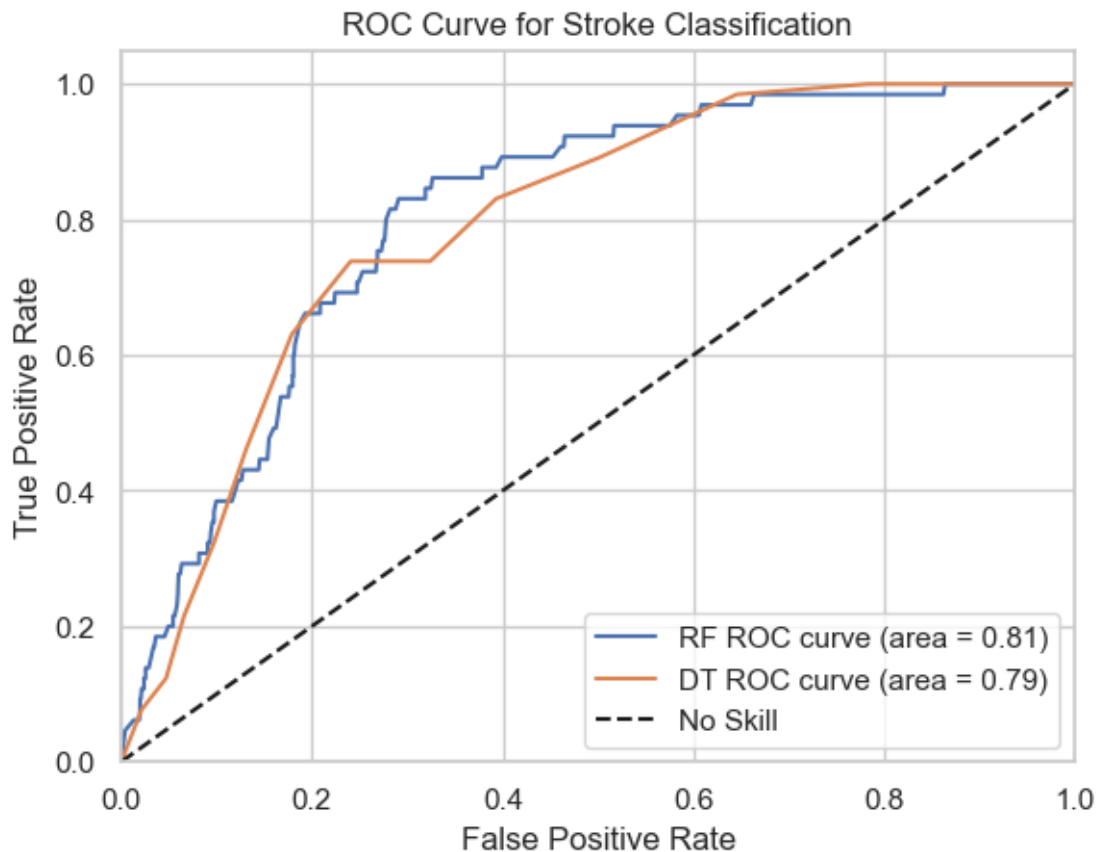


כמו שצפינו בהתחלה, ל-Age יש השפעה גדולה מאוד על הסיווג, היא מופיעה כאן בשלושה פיצולים שונים של העץ! בנוסף, AGI שראינו שגם לה יש הרבה קורלציה באופן יחסי עם stroke מקבלת פה נוכחות מרשימה בעץ.

ל-RandomForest היה קשה יותר לעשות ויזואליזציה ולכן לא אצרף אותו כאן.

ניתוח השוואתי של התוצאות

:ROC Curve



שני המסווגים נותנים Area Under Curve קרוב ל-0.8 שזו נחשבת תוצאה טובה אך לא מצוינת.

מכיוון שמדובר ב-Imbalanced Data, וכמות האנשים עם stroke ב-test נמוכה מאוד ויכולה להיות עם שונות גבוהה שתשפיע משמעותית על התוצאות בחלוקה של שני שלישי בצעתי 10 fold cross validation כדי להתגבר על הרעש שיכול להווצר מכך ולדייק את ההערכה וההשוואה בין המודלים.

ב-10 fold cross validation אי אפשר לייצר confusion matrix ולצייר ROC curve אחד ויחיד, אלא צריך לייצר 10 כאלו או לעשות גרפים שיציגו את הממוצע שלהם. חשוב מאוד לציין שגם כאן, כדי לאזן את הנתונים ולתת למודל יותר דגימות להתאמן עליהן, לפני כל הרצה של הסיווג יוצרו דגימות סינתטיות על גבי נתוני הלמידה (לאחר החלוקה ל-train ו-test) באמצעות אלגוריתם SMOTE שהוזכר לעיל.

בחרתי במקום להציג השוואה רק בין ממוצע התוצאה שלהם:

Metric	RandomForest	CART gini-index
Accuracy	0.757	0.691
Recall	0.767	0.692
ROC AUC	0.824	0.815

Precision	0.134	0.129
-----------	-------	-------

סיכום ומסקנות

אפשר לראות בטבלת ההשוואה ש-RandomForest טוב יותר מעץ החלטה כמעט בכל פרמטר. אני מאמין שהסיבה לכך היא בעיקר ביכולת של RandomForest להתבסס על תוצאות מרובות מהרבה עצים אקראיים וכך להמנע מתופעת ה-Overfitting שעץ החלטה יחיד יכול לסבול ממנה בקלות. בנוסף, כל עץ ב-RandomForest לומד על feature-ים שונים ולכן הרבה פעמים המודל מצליח להשתמש בנתונים רבים יותר כדי לקבל החלטה ולתפוס את המשמעות מאחורי כל הנתונים בניגוד לעץ החלטה יחיד שבסופו של דבר מתבסס על סט התכונות המצומצם שאצלו בעץ.

למרות ש-RandomForest טוב יותר מ-CART כדי לחזות שבץ בניסוי זה עם נתונים אלו, עדיין הוא מודל שלא משיג תוצאות טובות במיוחד. יש לו הרבה מאוד FP ו-Precision נמוך מאוד. Recall ו-Accuracy של מתחת ל-0.8 היא גם תוצאה נמוכה מאוד. חיזוי שבץ זה קשה מאוד, גוף האדם הוא מכונה מורכבת מאוד שתלויה בהמון פרמטרים שונים, פנימיים וחיצוניים ולכן גם מאוד בלתי צפויה. כמו כן, כמות הנבדקים שהיה להם שבץ שמהם יכולנו ללמוד על המשותף לחולי שבץ היה נמוך מאוד ונאלצנו להשתמש בטכניקות של ייצור דגימות סינתטיות ו-Oversampling כדי להתגבר על כך, פתרון לא אידיאלי שפגע בהצלחת המודל שלנו לכל הדעות.

אני חושב שניתן היה לשפר משמעותית את יכולות המודל שלנו בכמה אופנים –

1. העשרת מאגר הנתונים: אני מאמין שאם היינו מקבלים את כל מאגר הנתונים של McKinsey & Company שכולל כמעט 30 אלף רשומות של נבדקים היינו יכולים לקבל עוד שורות רבות ונתונים על אנשים שהיה להם שבץ כדי לחזק את יכולת הלמידה שלנו על הגורמים לשבץ. בנוסף, אם היה אפשרי להוסיף עוד עמודות עם נתונים על הנבדקים כמו תוצאות בחינה גופנית, קוגניטיבית וגנטית אולי היו לנו עוד פרמטרים לפיהם היינו יכולים לנסות לאפיין את המשותף לסיכון גבוה לשבץ.
2. שימוש טוב יותר בפרמטרים של CART ושל RandomForest ב-sklearn: אני חושב שיכולתי עם משחק נוסף והבנה מעמיקה יותר של משמעויות הפרמטרים של המסווגים בפייתון להשיג תוצאות מעט טובות יותר.
3. שימוש באלגוריתמי סיווג נוספים שלא נלמדו בקבוצה שלי (אך כן נמצאים בהרצאות של מארק): אלגוריתמים כמו XGBoost, AdaBoost ואלגוריתמי חיזוק גרדיאנט נוספים שלא נלמדו בקורס אמורים להיות טובים יותר מ-RandomForest ופוטנציאלית יצליחו לחזות בצורה טובה ממנו לנתונים שלנו.

לסיכום, מאוד נהניתי מהפרויקט, אני מאמין מאוד בלמידה דרך הידיים. עד כה באו"פ זה היה הפרויקט שאני מרגיש שדחף אותי הכי הרבה ללמידה עצמית, קריאה באינטרנט וניסוי וטעייה.

תודה רבה ומקווה שנהנתם (: