מיני פרויקט- קלסיפיקציה של חולי סכרת ויצירת API

**מגישות:**

**נירית טרבלסי 207781220**

**עדי רגב (ששר)208884536**

קישור לתיקיית הפרויקט:

<https://drive.google.com/drive/folders/1KRvmC-E0zX9Xmzo4Qu8OcPByg6NxQMXA>

מטרת הפרויקט שלנו הייתה ליצור כתובת API, אליה יוכל

המשתמש להזין נתונים מסוימים ולקבל סיווג לפי מודל למידת מכונה

האם הוא חולה סכרת או לא.

במהלך הפרויקט עשינו שימוש במגוון אלגוריתמים כגון רשתות נוירונים, עצי החלטה,

וחבילות כגון flask\_ngrokעל מנת ליצור כתובת API ובחבילת Sklearn בשביל לבנות את מודל החיזוי.

קיבלנו נתוני חולים בסכרת ואנשים בריאים של אנשים מאפריקה, מדובר במאגר נתונים מוכר.

ניתן לעיין בו כאן:

**url** = 'https://github.com/rosenfa/nn/blob/master/pima-indians-diabetes.csv?raw=true'

מאגר זה מאגד בתוכו מגוון נתונים על מאפיינים שונים כגון: לחץ דם, אינסולין, מספר הריונות, גיל ועוד עבור כל אחד.

במהלך העבודה עם מאגר זה, נתקלנו בערכים חסרי הגיון. לדוגמא,

אפס בנתון של מדידת לחץ דם.

הנחנו שבאותם נתונים חסרי הגיון, ערכי האפס החליפו ערכים חסרים None.

לכן, עיבוד מקדים שעשינו על מאגר הנתונים היה להחליף ערכים אלו בערכי חציון ובכך מזערנו איבוד מידע.

את התהליך הנ"ל ביצענו במאפיינים הבאים:

גלוקוז, לחץ דם, עובי עור, אינסולין ו-BMI.

לאחר מכן ביצענו נרמול של הנתונים בעזרת **StandardScaler**.

לאחר הנרמול, סידרנו את הנתונים כך שיייצגו התפלגות גאוסייאנית

בעזרת **PowerTransformer**(method='yeo-johnson').

בעזרת אלגוריתם **SelectKBest**(f\_classif, k=5) איתרנו את המאפיינים

החשובים ביותר עבור הסיווג ומצאנו שהם:

*מספר הריונות, גלוקוז, ,diabetes pedigree function גיל ו-BMI.*

שלבי עיבוד הנתונים השונים כגון בחירת K המאפיינים הרלוונטיים ביותר,

עם נרמול או בלי נרמול, עם התפלגות גאוסייאנית או בלי ועוד,

כללו בדיקות מרובות והרצות שונות של מודלים על ווריאציות שונות של הנתונים, על מנת להגיע לתוצאה האופטימאלית.

במקביל, עבדנו על יצירת כתובת API שתאפשר להזין נתונים לשורת הכתובת

ולקבל תוצאה מיידית של הסיווג.

מטרת API לחבר בין שרתים למשתמש קצה. המשתמש לא צריך לגשת ישירות

לשרת, אלא העברת הנתונים מתבצעת דרך ה-API והמשתמש מקבל את הפלט הרצוי גם כן על ידו. (אפשר לדמות את זה למסעדה, בה המתארח אינו ניגש ישירות למטבח, אלא מעביר ומקבל את מבוקשו דרך המלצר)

נעזרנו בחבילת flask כדי שתהיה אפשרות לקבל מהמשתמש את הנתונים השונים עבור המאפיינים השונים, שיעברו לאחר מכן למודל שאותו ייבאנו

לקוד. לאחר הרצת המודל, ה-flask מחזיר למשתמש את הסיווג באותה כתובת API. כך שבסופו של דבר, הסיווג מתבצע על ענן בלי צורך בהתקנות או

גישה למחשב אישי.

תהליך בחירת המודל:

ניסינו מודלים שונים כגון עץ החלטה, רגרסיה לוגיסטית, רשתות נוירונים,

Random forest ועוד.

לבסוף התמקדנו במודל של רשתות נוירונים וחיפשנו את הפרמטרים

שיביאו לתוצאות הטובות ביותר במודל זה כגון: מספר שכבות, מספר קודקודים בכל שכבה.

בחירות אלו ביצענו על ידי לולאות for ושמירת תוצאות המודלים השונים באופן

אוטומטי עם סף של ערכי recall ו-accuracy.

Recall>0.79

Accuracy>0.68

למעשה הרצנו עשרות מודלים עם מספר שכבות משתנה ומספר קודקודים משתנה באופן אוטומטי על ידי האלגוריתם.

משלל התוצאות שהאלגוריתם שמר, בחרנו את הטובים ביותר.

ביצענו בדיקות נוספות עבור המודלים הטובים ביותר כגון בחירת k-best

של המאפיינים, עם המודלים השונים.

לאחר מכן, קיבלנו את המודל הטוב ביותר שמבנהו: (משמאל לימין)

(1, 4, 28, 1)

K-best=5

בנוסף לכך בדקנו את הנרמול המועדף ומצאנו שנרמול בשילוב התפלגות גאוסייאנית נתן את התוצאה הטובה ביותר.

סוג הדטה שנבחר: ygN5 ( yg-התפלגות גאוס

N-נרמול

5- חמישה מאפיינים טובים ביותר בעזרת t-test)

**טבלת סיכום ביצועים:**

(1, 4, 28, 1) רשת נוירונים

['**ygN5**'] סוג הדטה

|  |  |
| --- | --- |
| **'accuracy\_score'** | 0.75 |
| **'recall\_score'** | 0.89 |
| **'precision\_score'** | 0.6 |
| **'F1\_score'** | 0.72 |
|  |  |

**confusion\_matrix:**

|  |  |  |
| --- | --- | --- |
|  | **0** | **1** |
| **True** | 102 | 48 |
| **False** | 9 | 72 |

חשוב לציין, בנוסף לכך שחיפשנו accuracy גבוה, היה לנו חשוב שיהיה גם recall גבוה (לאתר כמה שיותר חולים), לכן במקרים מסוימים נאלצנו לוותר על accuracy גבוה יותר

עבור recall גבוה.

כפי שהזכרנו, המאפיינים הכי חשובים עבור המסווג:

*מספר הריונות, גלוקוז, ,diabetes pedigree function גיל ו-BMI.*

קישור לתיקיית הפרויקט:

<https://drive.google.com/drive/folders/1KRvmC-E0zX9Xmzo4Qu8OcPByg6NxQMXA>