



עבודה מסכמת בקורס ניהול התפעול והייצור

הפקולטה למדעי החברה
המחלקה לניהול – מסלול ניהול טכנולוגיה

קורס מספר 55030
מרצה: ד"ר אמיר אלאוף
מתרגל: מר דויד פרויד

מגיש: סהר עבודי
קבוצה מספר 12 – סהר עבודי
מייל: saharabudi@yandex.com
תעודת זהות: 207388604

הצגת הבעיה

שבץ מוחי הוא מצב רפואי שבו זרימת דם לקויה למוח גורמת למוות של תאי מוח. גורם הסיכון העיקרי לשבץ הוא יתר לחץ דם. גורמי סיכון נוספים כוללים רמות כולסטרול גבוהות בדם, עישון טבק, השמנה, סוכרת, אירוע שבץ חולף קודם, מחלת כליות סופנית, ופרפור פרוזדורים.

באמצעות סט נתונים מאתר 'Kaggle' הנקרא 'Brain_Stroke.csv' ננסה לבצע זיהוי מוקדם של הפוטנציאל לקבלת שבץ מוחי. קובץ ה-CSV שוקל 280 קילו-בייטים, מכיל 11 עמודות (gender, age, hypertension, heart disease, ever-married, work-type, residence-type, avg-glucose-level, bmi, smoking status, stroke) ומבוסס על 54791 תאים, חלקם מספריים וחלקם איכותיים.

סטטיסטיקות תיאוריות

גיל

הגיל הממוצע של האנשים בנתונים הוא 43.42 שנים, עם סטיית תקן של 22.66 שנים, מה שמראה על פיזור רחב של הגילאים בקרב האוכלוסייה הנבדקת. הגיל המינימלי בנתונים הוא כ-0.08 שנים (בערך חודש), בעוד שהגיל המקסימלי מגיע ל-82 שנים. הטווח של הגילאים הוא 81.92 שנים, מה שמצביע על קשת רחבה של גילאים בנתונים.

יתר לחץ דם

רק כ-9.6% מהאנשים בנתונים סובלים מיתר לחץ דם, כפי שמראה הממוצע של משתנה זה (0.096). רוב האנשים בנתונים אינם סובלים מיתר לחץ דם, מה שמודגם גם על ידי העובדה שהשכיח של משתנה זה הוא 0.0. נתון זה מצביע על כך שרוב האוכלוסייה שנבדקה נמצאת ברמות לחץ דם תקינות.

מחלות לב

כ-5.5% מהאנשים סובלים ממחלות לב, כפי שעולה מהממוצע של משתנה זה (0.055). גם כאן, רוב האנשים בנתונים אינם סובלים ממחלות לב, כפי שמראה השכיח של משתנה זה (0). נתון זה מדגיש כי רוב האוכלוסייה אינה סובלת מבעיות לב.

רמת גלוקוז ממוצעת

רמת הגלוקוז הממוצעת בדם של האנשים בנתונים היא 105.94 מ"ג/ד"ל, עם סטיית תקן של 45.08 מ"ג/ד"ל, מה שמראה על פיזור רחב של רמות הגלוקוז. הטווח של רמות הגלוקוז נע בין 55.12 מ"ג/ד"ל ל-271.74 מ"ג/ד"ל, טווח רחב זה מצביע על שוני משמעותי ברמות הגלוקוז בקרב האוכלוסייה הנבדקת.

BMI - מדד מסת גוף

ה-BMI הממוצע בנתונים הוא 28.5, מה שמצביע על כך שרוב האנשים נמצאים בקטגוריית "עודף משקל". ה-BMI בנתונים נע בין 14.0 ל-48.9, עם טווח של 34.9, המראה על וריאציה משמעותית במסת הגוף בין האנשים שנבדקו. נתון זה מדגיש את השונות הרבה במצבי המשקל בקרב האוכלוסייה.

שבץ

כ-5% מהאנשים בנתונים עברו שבץ, כפי שמראה הממוצע של משתנה זה (0.0498). נתון זה מראה ששבץ הוא מצב רפואי נדיר יחסית בקרב האוכלוסייה שנבדקה. בנוסף, בטבלת החיתוך בין מגדר לשבץ נמצא כי השכיחות של שבץ מעט גבוהה יותר אצל גברים (5.2%) בהשוואה לנשים (4.8%).

חיתוך גיל ממוצע לפי יתר לחץ דם ושבץ

עבור אנשים ללא יתר לחץ דם וללא שבץ, הגיל הממוצע הוא כ-40.29 שנים. לעומת זאת, אנשים ללא יתר לחץ דם אך עם שבץ הם בגיל ממוצע של 66.95 שנים, מה שמרמז על כך ששבץ נוטה להתרחש בגיל מבוגר יותר גם בקרב אלו שאין להם יתר לחץ דם. עבור אנשים עם יתר לחץ דם וללא שבץ, הגיל הממוצע הוא כ-61.55 שנים, ואילו אנשים עם יתר לחץ דם ושבץ הם בגיל ממוצע של 70.21 שנים. נתונים אלו מצביעים על כך שהשילוב של יתר לחץ דם וגיל מבוגר מגביר את הסיכון לשבץ.

Work Type	Ever Married	Gender	Stroke	BMI	Avg Glucose Level	Heart Disease	Hypertension	Age	Parameter
4981	4981	4981	4981	4981	4981	4981	4981	4981	Count
-	-	-	0.050	28.50	105.94	0.055	0.096	43.42	Mean
-	-	-	0.218	6.79	45.08	0.228	0.295	22.66	Std (Std Dev)
-	-	-	0	14.0	55.12	0	0	0.08	Min
-	-	-	0	23.7	77.23	0	0	25.0	25% (Q1)
-	-	-	0	28.1	91.85	0	0	45.0	50% (Median)
-	-	-	0	32.6	113.86	0	0	61.0	75% (Q3)
-	-	-	1	48.9	271.74	1	1	82.0	Max
4	2	2	-	-	-	-	-	-	Unique
Private	Yes	Female	-	-	-	-	-	-	Top
2860	3280	2907	-	-	-	-	-	-	Frequency

ניתוח הסטוגרמות לערכים מספריים בדאטה:

- גיל: הגיל החציוני הוא 43, מה שמעיד על אוכלוסייה צעירה יחסית.
- יתר לחץ דם: רק 13% מהנבדקים סובלים מיתר לחץ דם.
- מחלות לב: 10% מהנבדקים סובלים ממחלות לב.
- רמת גלוקוז ממוצעת: רמת הגלוקוז הממוצעת החציונית היא 9.105.
- מדד מסת גוף (BMI): מדד מסת הגוף החציוני הוא 3.28.
- שבץ מוחי: רק 5% מהנבדקים חוו שבץ מוחי.

ניתוח חוזק המתאם על פי מטריצה:

stroke	bmi	avg_glucose_level	heart_disease	hypertension	age	
0.25	0.37	0.24	0.26	0.28	1	age
0.13	0.16	0.17	0.11	1	0.28	hypertension
0.13	0.061	0.17	1	0.11	0.26	heart_disease
0.13	0.19	1	0.17	0.17	0.24	avg_glucose_level
0.057	1	0.19	0.061	0.16	0.37	bmi
1	0.057	0.13	0.13	0.13	0.25	stroke

מתאמים חזקים (יחסית):

- **גיל ו-BMI:** קיים מתאם חיובי בינוני עד חזק בין הגיל למדד מסת הגוף. ככל הנראה, זהו אחד הקשרים החזקים ביותר במטריצה.

מתאמים בינוניים:

- **יתר לחץ דם, מחלות לב ושבץ מוחי:** בין המשתנים הללו קיימים מתאמים חיוביים חלשים עד בינוניים, מה שמצביע על קשר בין מחלות לב וכלי דם.
- **רמת גלוקוז ממוצעת ו-BMI:** קיים מתאם חיובי חלש בין רמת הגלוקוז למדד מסת הגוף, מה שמצביע על קשר אפשרי בין השמנה לרמות סוכר גבוהות.

מתאמים חלשים:

- **יתר לחץ דם ורמת גלוקוז ממוצעת:** למרות שקיים קשר בין מחלות לב וכלי דם, הקשר בין יתר לחץ דם לרמת הגלוקוז הוא חלש יותר.
- **יתר לחץ דם ושבץ מוחי ורמת גלוקוז ממוצעת:** גם הקשרים בין יתר לחץ דם ושבץ מוחי לבין רמת הגלוקוז ממוצעת הם חלשים יחסית.

Main Analysis

1. **עץ החלטות – פתחתי מודל ואלידי של עץ החלטות,** לניבוי משתנה 'stroke' באמצעות המשתנים smoking_status ו avg_glucoase_level.
 1. תחילה טענתי את הנתונים לכדי דאטה-פריים(df) כדי שנוכל לעבוד עליו.
 2. בזיהוי סוגי הערכים הבחנתי כי המשנה 'smoking status' הינו איכותי, לכן תרגמתי אותו למספרים נומריים כאשר:

```
d = {'formerly smoked': 0, 'never smoked': 1, 'smokes': 2, 'Unknown' : 3}
df['smoking_status'] = df['smoking_status'].map(d)
```

3. לאחר מכן חלקתי את המשנים ל X1 וY1, על מנת שנוכל לחלק אותם ל train וtest.

```
X1 = df[['smoking_status', 'avg_glucose_level']]
y1 = df['stroke']
```

4. אופי החלוקה ל train/test – כאשר החלוקה היא (25-75)

```
X1_train, X1_test, y1_train, y1_test = train_test_split(X1, y1, test_size=0.25, random_state=42)
```

5. הגדרת העץ החלטות לפי המשתנים:

```
dtree3 = DecisionTreeClassifier(max_depth=3)
dtree3.fit(X1_train, y1_train)
```

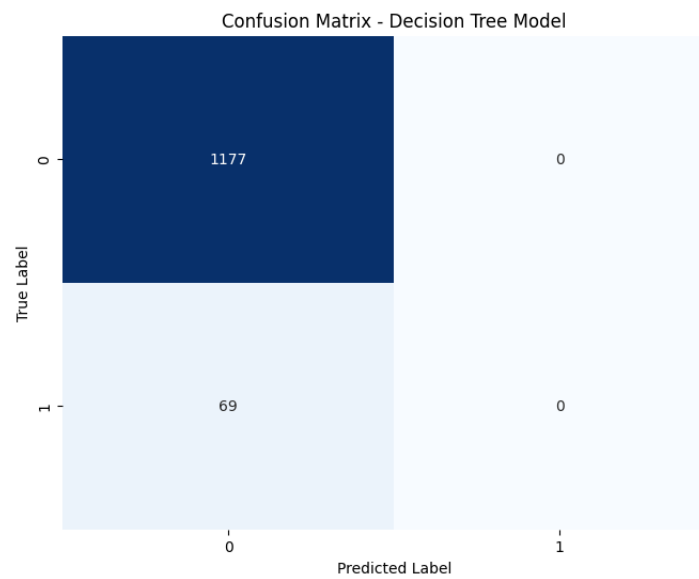
6. הגדרת משתנה חיזוי לטובת מטריצת בלבול:

```
y1_pred = dtree3.predict(X1_test)

cm1 = confusion_matrix(y1_test, y1_pred)

plt.figure(figsize=(8, 6))
sns.heatmap(cm1, annot=True, fmt='d', cmap='Blues', cbar=False, xticklabels=[0, 1], yticklabels=[0, 1])
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix - Decision Tree Model')
plt.show()
```

תוצאות המטריצה:



7. הגדרת מידת הדיוק:

```
accuracy1 = accuracy_score(y1_test, y1_pred)
print("Accuracy:", accuracy1)

Accuracy: 0.9446227929373997
```

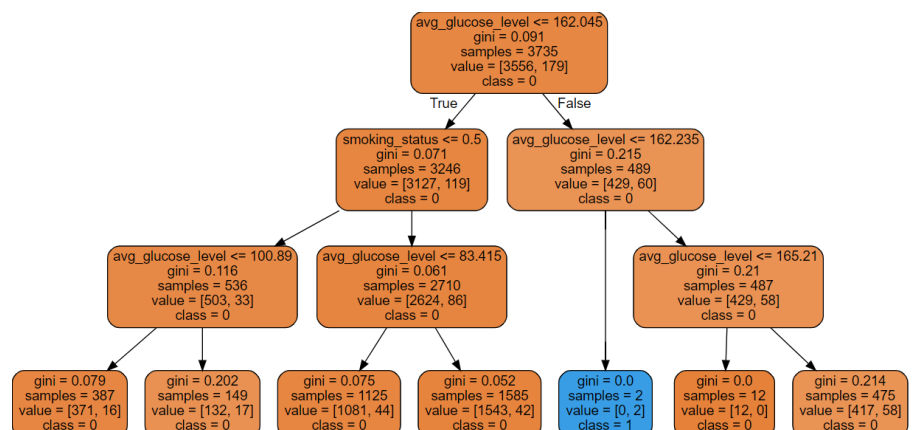
8. קלאסיפיקציה:

```
print(classification_report(y1_test, y1_pred))
```

	precision	recall	f1-score	support
0	0.94	1.00	0.97	1177
1	0.00	0.00	0.00	69
accuracy			0.94	1246
macro avg	0.47	0.50	0.49	1246
weighted avg	0.89	0.94	0.92	1246

9. ויזואליזציה של עץ ההחלטות

```
export_graphviz(decision_tree=dtree3,
out_file='C:\Residence_type.dot',
feature_names=X1_test.columns,
class_names=dtree3.classes_.astype(str),
leaves_parallel=True,
filled=True,
rotate=False,
rounded=True)
from graphviz import Source
Source.from_file('C:\Residence_type.dot')
```



2. הרחבת עץ ההחלטות ושיפור accuracy

במודל השני של עץ ההחלטות אנו נדרשים לשיפור המודל הראשון על מנת להגיע למדד דיוק גבוהה יותר, באמצעות שינוי עומק העץ, וכן הוספת שדה אחד לבחירתו. להלן השלבים:

1. בחרתי להוסיף המשתנה "Ever_Married" כמשתנה מנבא נוסף, כמו כן המרתי את הערכים שלו לנומריים על מנת שאוכל לעבוד איתם:

```
d = {'Yes' : 0, 'No' : 1}
df['ever_married'] = df['ever_married'].map(d)
```

2. לאחר מכן חלקתי את המשתנים ל X_1 ו Y_1 , על מנת שנוכל לחלק אותם ל $train$ ו $test$.

```
X2 = df[['smoking_status', 'avg_glucose_level', 'ever_married']]
y2 = df['stroke']
```

3. חלוקת המשתנים ל $train$ ו $test$, תחת ההוראה לחלוקה לפי 25-75. כמו כן, על מנת להגיע לתוצאות משופרות יותר, $randomstate$ ירד ל 0 במקום 42 וכן נוספה רמה אחת לעץ max

$depth = 4$

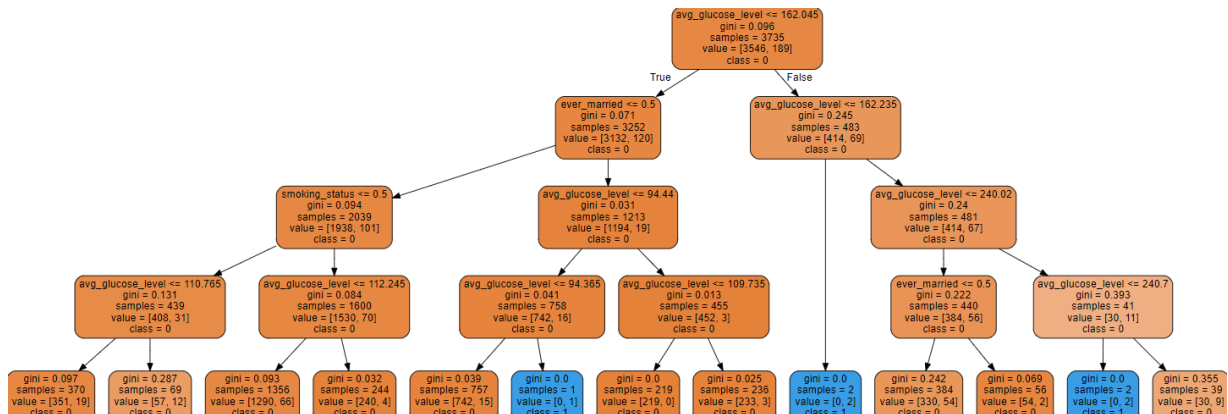
```
X2_train, X2_test, y2_train, y2_test = train_test_split(X2, y2, test_size=0.25, random_state=0)
dtree3 = DecisionTreeClassifier(max_depth=4)
dtree3.fit(X2_train, y2_train)
y2_pred = dtree3.predict(X2_test)
```

4. תוצאות מידת הדיוק:

```
accuracy2 = accuracy_score(y2_test, y2_pred)
print("Accuracy:", accuracy2)

Accuracy: 0.9510433386837881
```

5. ויזואליזצית העץ החדש



3. רגרסיה לוגיסטית

במודל הרגרסיה הלוגיסטית התבקשתי לנבא את המשתנה ever_married באמצעות המשתנים BMI ו avg_glocuse_level.

```
df4 = df.copy()
df4.head()
```

1. ייצרתי עותק של data frame:

2. חלוקת המשתנים לקבוצות:

```
X4 = df4[['avg_glucose_level', 'bmi']]
y4 = df4['ever_married']
X4.head()
```

	avg_glucose_level	bmi
0	228.69	36.6
1	105.92	32.5
2	171.23	34.4
3	174.12	24.0
4	186.21	29.0

3. סטנדרטיזציה של הנתונים, קוד הזה נועד לתקן את הנתונים כך שכל התכונות במערך הנתונים X4 יהיו בעלי ממוצע 0 וסטיית תקן 1:

```
from sklearn.preprocessing import StandardScaler
# Standardizing the data
scaler = StandardScaler()
scaler.fit(X4)
X4_scaled = scaler.transform(X4)
df_scaled = pd.DataFrame(X4_scaled, columns=X4.columns)
df_scaled.head()
```

	avg_glucose_level	bmi
0	2.723411	1.193238
1	-0.000523	0.589390
2	1.448529	0.869222
3	1.512650	-0.662492
4	1.780895	0.073909

4. חלוקת המשתנים ל train ו test, תחת ההוראה לחלוקה לפי 70-30

```
X4_train, X4_test, y4_train, y4_test = train_test_split(X4, y4, test_size=0.30, random_state=42)
```

5. בניית מודל הרגרסיה הלוגיסטית על פי החלוקה למשתני הניבוי:

```
from sklearn.linear_model import LogisticRegression
# Building the logistic regression model
logreg = LogisticRegression()
logreg.fit(X4_train, y4_train)
```

6. ניבוי מידת הדיוק וכן קלסיפיקציה:

```
# Predicting the target variable
y4_pred = logreg.predict(X4_test)

# Finding the accuracy score
accuracy4 = accuracy_score(y4_test, y4_pred)
print("Accuracy:", accuracy4)

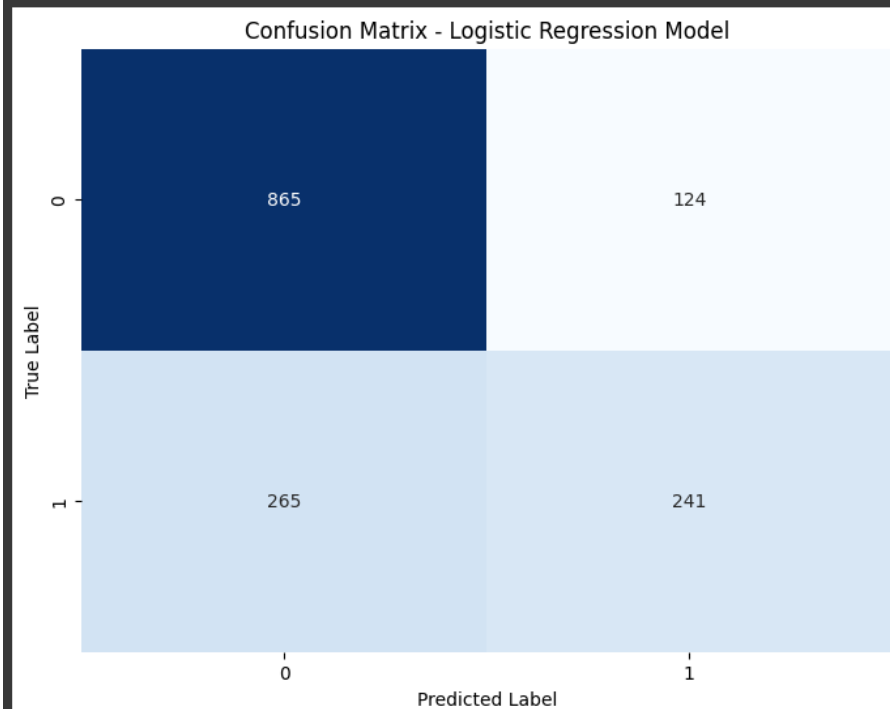
Accuracy: 0.7397993311036789

print(classification_report(y4_test, y4_pred))
```

	precision	recall	f1-score	support
0	0.77	0.87	0.82	989
1	0.66	0.48	0.55	506
accuracy			0.74	1495
macro avg	0.71	0.68	0.68	1495
weighted avg	0.73	0.74	0.73	1495

7. מטריצת בלבול:

```
# Calculate the confusion matrix
cm3 = confusion_matrix(y4_test, y4_pred)
# Plot the confusion matrix using seaborn's heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(cm3, annot=True, fmt='d', cmap='Blues', cbar=False, xticklabels=[0, 1], yticklabels=[0, 1])
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix - Logistic Regression Model')
plt.show()
```



4. מודל k-nearest neighbor

התבקשתי לפתח מודל ואלידי לKNN, לניבוי המשתנה 'hypertension' באמצעות משתני הקובץ bmi, age, על פי חלוקה ל train ו test של 80-20.

1. נחלק את המשתנים לקבוצות:

```
X5 = df[['age', 'bmi']]
y5 = df['hypertension']
```

2. סטנדריזציה של הנתונים:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X5)
X5_scaled = scaler.transform(X5)

df_scaled = pd.DataFrame(X5_scaled, columns=X5.columns)
df_scaled.head()
```

	age	bmi
0	1.040584	1.193238
1	1.614270	0.589390
2	0.246250	0.869222
3	1.570141	-0.662492
4	1.658400	0.073909

3. חלוקת הקבוצות ל train ו test לפי 80-20:

```
from sklearn.model_selection import train_test_split
# Splitting the data into train and test sets:
X5_train, X5_test, y5_train, y5_test = train_test_split(df_scaled, y5, test_size=0.20, random_state=42)

# Importing the KNeighborsClassifier class from the sklearn.neighbors library:
from sklearn.neighbors import KNeighborsClassifier

# Building the model
knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X5_train, y5_train)
```

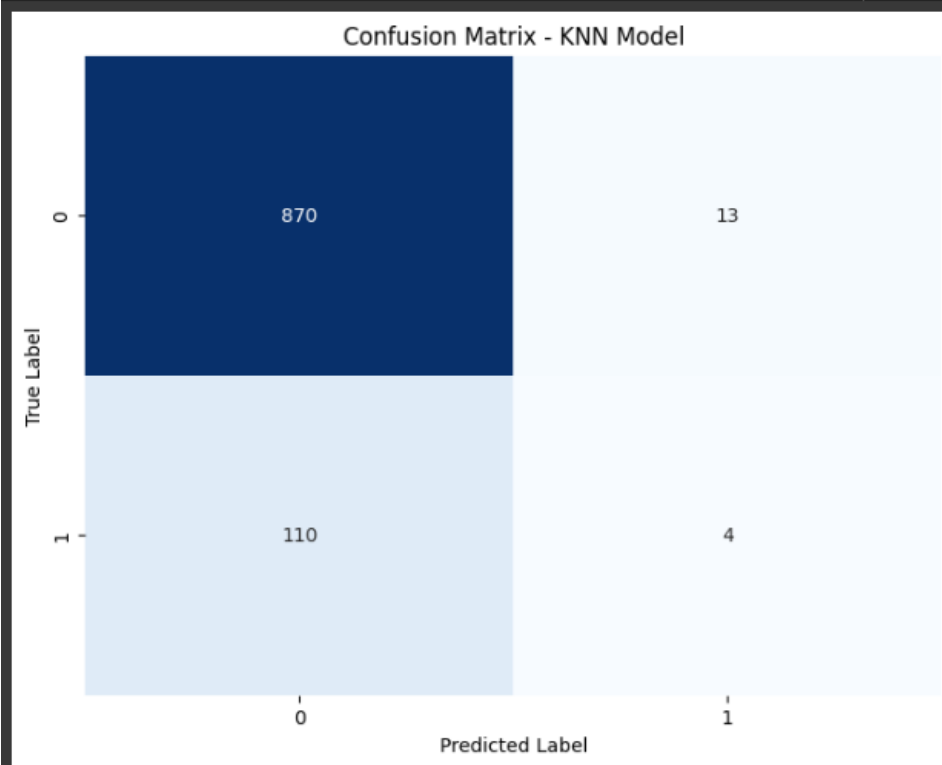
▼ KNeighborsClassifier
KNeighborsClassifier(n_neighbors=7)

4. מטריצת בלבול למודל KNN:

```
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score

cm5 = confusion_matrix(y5_test, y5_pred)

plt.figure(figsize=(8, 6))
sns.heatmap(cm5, annot=True, fmt='d', cmap='Blues', cbar=False, xticklabels=[0, 1], yticklabels=[0, 1])
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix - KNN Model')
plt.show()
```



5. מידת הדיוק והקלסיפיקציה:

```
accuracy5 = accuracy_score(y5_test, y5_pred)
print("Accuracy:", accuracy5)

print(classification_report(y5_test, y5_pred))
```

	precision	recall	f1-score	support
0	0.89	0.99	0.93	883
1	0.24	0.04	0.06	114
accuracy			0.88	997
macro avg	0.56	0.51	0.50	997
weighted avg	0.81	0.88	0.83	997

ניתוח התוצאות:

עץ החלטה (Decision Tree)

מודל עץ ההחלטה השיג מידת דיוק גבוהה של 94.46% עד 95.10%, והצליח לסווג במדויק כמעט את כל המקרים שבהם לא התרחש שבץ. עם זאת, המודל כשל לחלוטין בזיהוי מקרים של שבץ, כפי שמשתקף במטריצת הבלבול, שבה ניכר כי המודל לא זיהה אף מקרה של שבץ נכון. מבנה עץ ההחלטות הציג תלות גבוהה ברמת הגלוקוז הממוצעת בדם ובסטטוס העישון של המטופלים, כאשר פיצולים אלו הובילו במרבית המקרים לסיווג של אי-שבץ. תוצאות אלו מצביעות על כך שהמודל מוטה באופן מובהק לזיהוי מחלקה שלילית (ללא שבץ) בלבד.

רגרסיה לוגיסטית (Logistic Regression)

מודל הרגרסיה הלוגיסטית הציג ביצועים מאוזנים יותר, עם מידת דיוק של 73.98%. המודל הצליח בצורה סבירה לזהות מקרים של אי-שבץ, אך התקשה בזיהוי מקרים חיוביים של שבץ. דוח הסיווג מצביע על כך שהמודל מצליח יותר בזיהוי מחלקה 0 (ללא שבץ) מאשר מחלקה 1 (עם שבץ), כאשר ערכי ה-Precision וה-Recall עבור מחלקה 1 הם נמוכים יותר. תוצאות אלו מעידות על יכולת חלקית בלבד של המודל לזהות מקרים של שבץ.

מודל שכן קרוב ביותר (K-Nearest Neighbors)

מודל ה-KNN השיג מידת דיוק של 87.66%, והראה ביצועים טובים בזיהוי מקרים של אי-שבץ, אך כמו המודלים האחרים, התקשה בזיהוי מקרים של שבץ. ממטריצת הבלבול עולה כי המודל זיהה נכון רק 5 מתוך 114 מקרים של שבץ. דוח הסיווג מראה כי למודל יש הטיה ברורה לטובת המחלקה השלילית (ללא שבץ), עם ערכי Precision ו-Recall נמוכים מאוד למחלקה החיובית (עם שבץ).

מסקנות

מסקנות הניתוח מצביעות על בעיה משותפת לכל המודלים שנבחנו: קושי ניכר בזיהוי מקרים של שבץ מוחי (מחלקה חיובית). כל המודלים הציגו ביצועים טובים בסיווג מקרים של אי-שבץ, אך כשלו בזיהוי המקרים שבהם התרחש שבץ, מה שמעיד על חוסר איזון במודל לטובת המחלקה השלילית. תוצאות אלו מצביעות על הצורך בשיפור האיזון בין המחלקות באמצעות טכניקות כגון איזון מחלקות, או בבחינת מודלים מתקדמים יותר שיכולים להתמודד טוב יותר עם מחלקות לא מאוזנות.