

דוח יישום k-Nearest Neighbors (k-NN) אוחנה

מטרה

דוח זה מTARGET את יישום והערכת אלגוריתם k-Nearest Neighbors (k-NN) ל-SIUG. נעשה שימוש במערכת הנתונים students_data.csv תוך חקירת הנתונים, עיבוד מקדים, יישום המודל והערכת ביצועיו. המשימה כוללת גם זיהוי וטיפול בחירויות אפשריות במערכת הנתונים, כדי להבטיח דיוק מרבי של המודל.

Data Exploration and Preprocessing .1

Load the Dataset .1.1

מערכת הנתונים נתען באמצעות ספרייה pandas והשורות הראשונות הוצגו כדי להבין את מבנה הנתונים.

	feature1	feature2	feature3	feature4	label
0	3.984735	15.767828	0.136371	3.043915	1
1	11.142359	24.628361	0.196689	3.083318	1
2	10.487314	18.346126	0.055332	0.128279	0
3	7.819583	17.449196	0.198131	3.007801	1
4	8.028921	20.747280	0.009201	-0.130778	0

Basic Statistics .1.2

חשבו סטטיסטיות מרכזיות באמצעות פונקציית *:describe*

	count	mean	std	min	25%	50%	75%	max
feature1	1000.0	9.926965	1.983449	3.646592	8.636113	9.995765	11.295162	16.225820
feature2	1000.0	19.801200	5.055980	5.502431	16.311453	19.772042	23.224596	35.491497
feature3	1000.0	0.481443	0.284950	0.000243	0.233310	0.464065	0.719020	0.996013
feature4	1000.0	1.549059	1.507720	-0.368837	-0.002409	2.802161	3.008193	3.337738
label	1000.0	0.516000	0.499994	0.000000	0.000000	1.000000	1.000000	1.000000

השונות הגבוהה בטווח הערכים וה ממוצעים בין המאפיינים בטבלה מצביעה על צורך בNORMALISATION, על מנת למנוע השפעה לא מאוזנת של תכונות מסוימות על המודל.

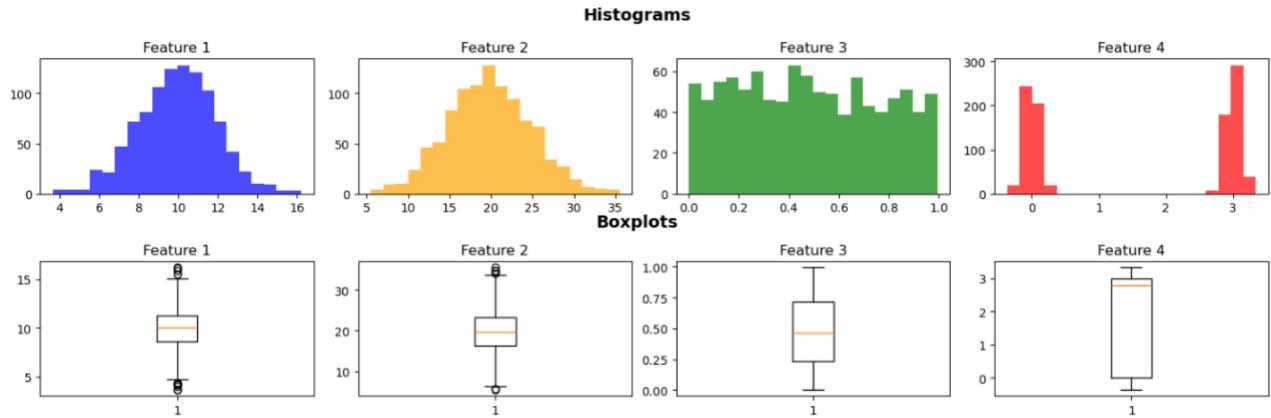
Data Cleaning .1.3

לא זהה בדата ערכים חסרים - בוצע שימוש בפונקציית *info* :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 5 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   feature1  1000 non-null    float64
 1   feature2  1000 non-null    float64
 2   feature3  1000 non-null    float64
 3   feature4  1000 non-null    float64
 4   label      1000 non-null    int64  
dtypes: float64(4), int64(1)
memory usage: 39.2 KB
```

Visualizations .1.4

A. Histograms & Boxplots



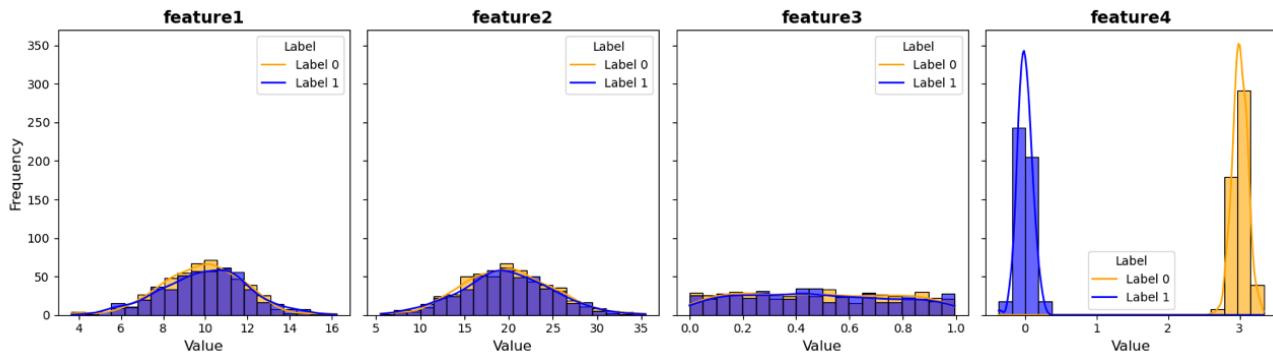
.I. Feature 1 | Feature 2 - מציגים התפלגות נורמלית, אך קיימים ערכים חריגים בקצוות.

.II. Feature 3 מפוזר באופן אחד, ללא חריגות.

.III. Feature 4 יש מבנה ייחודי עם הפרדה ברורה לשתי קבוצות.

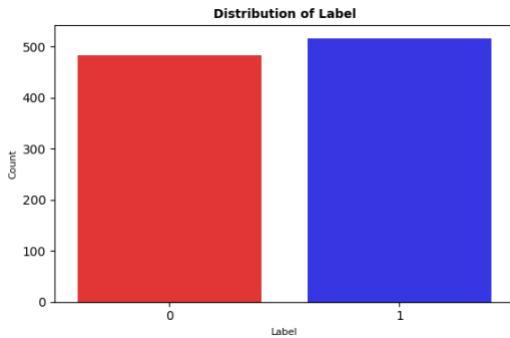
ב. Histograms Separated by Labels

Histograms of Features with Target Variable Separation



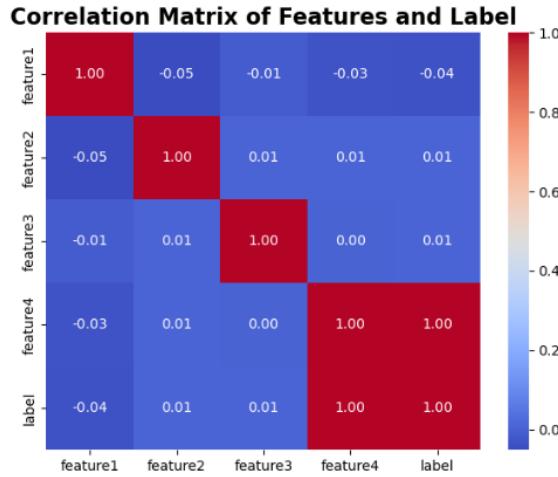
- I. Feature 4 מספק את ההפרדה הברורה ביותר בין הקבוצות.
- II. Feature 2 ו-Feature 1 לא מראים מראים הפרדה ברורה.
- III. Feature 3 אינו מספק הבדל משמעותי בין הקבוצות ולכן יהיה פחות משמעותי להפרדה במודל.

ג. Countplot Label



ניתן לראות מהגרף שהدادה מאוזנת.

Heatmap. 7



- הקורלציה בין המאפיינים עצם נמוכה מאוד, הדבר מעיד על כך שאין חפיפה ממשמעותית ביניהם. לכן, נראה שכל אחת מהמאפיינים יסייע ייחודי למודל.
- ישנה קורלציה מושלמת בין Feature 4 לעמודת ה-Label.

Feature Engineering . 2

2.1 Identify and Handle Anomalies

- כיוון שבუיף הקודם זיהתה קורלציה מושלמת בין Feature 4 לבין עמודת ה-Label- הוחלט להסיר את Feature 4 .
קורלציה מושלמת מצביעה על כך שהמאפיין מסביר באופן מלא את משתנה המטרה. מצב זה עלול להטעות את המודל בצורה משמעותית, לגרום לאובדן כליליות ולפגוע בביצועיו על נתונים חדשים.
בעזרה גרפי Boxplot- זיהו ערכים חריגים ב 1-Feature ו- 2-Feature- לאחר בדיקה, נמצא כי ישנים בסך הכל 17 ערכים חריגים בשני המאפיינים יחד, מהווים אחוז קטן מכלל מערכת הנתונים. אך, הוחלט להסיר את הערכים החיריגים הללו כדי למנוע הטיה של המודל ולשפר את ביצועיו. החלטה זו נועדה לשמר על איזון במודל ולמנוע השפעה לא פרופורציונלית של הערכים החיריגים.

Feature 1: 11 outliers (1.10% of the data)
Feature 2: 6 outliers (0.60% of the data)

2.2 Feature Scaling

- כדי למנוע זילוג נתוניים הוחלט לבצע קודם כל את שלב חלוקת הנתונים ורק לאחר מכן לבצע את הנרמול.
לכן שלב הנרמול יתבצע בשלב 3.

Feature Scaling+data Splitting .3

- 3.1. הנתונים חולקו לשולשה חלקים- Train 70%, Validation 15%, Test 15% random_state=42 כדי שהחלוקת תהיה זהה בכל הרצה ו shuffle=True כדי שאמם הנתונים ממויינים הדאטה תתערבב לפני החלוקה כך זה לא ישפיע על אימון המודל ועל הביצועים.

Train Features:

X_train: (688, 3)
y_train: (688,)

Validation Features:

X_val: (147, 3)
y_val: (147,)

Test Features:

X_test: (147, 3)
y_test: (148,)

- .3.2. כתוב בוצע נרמול הנתונים, ובחרה באופן אקראי שיטת הנרמול הסטנדרטית(Standard Scaling) הנרמול יכול לעמוד בסיס נתונים X_train בלבד, בעוד השינוי בוצע גם על X_val ו-X-test וזו זאת כדי למנוע דיליפה של מידע בין קבוצות הנתונים.

Model Implementation .4

Model Selection .4.1

בثور התחלת, נבחרו באופן שরירותי הפרמטרים הבאים עבור מודל KNN :

- **k=5** (מספר השכנים הקרובים)
- **p=2** (מרחק אוקלידי)

המודול אומן על נתונים האימון ובוצע חיזוי על סט הולידציה. הערכת הביצועים נעשו במדדים

הבאים:

- **Accuracy** (דיוק)
- **Precision** (דיוק סיווג)
- **Recall** (רגישות)
- **F1 Score** (מדד משולב)

תוצאות הביצועים:

Performance Metrics for KNN Model (k=5, p=2):

accuracy score : 0.5374149659863946

precision score: 0.5232558139534884

recall score: 0.625

F1 score: 0.569620253164557

5. Hyperparameter Tuning

בוצע כיוון הiper-פרמטרים באמצעות `GridSearchCV` לצד בדיקה ידנית של ערכי k ומדד המרחק (distance metric) כדי ליזהות את השילוב האופטימלי עבור המודל.

1. `GridSearchCV`

כיוון המודל בוצע באמצעות טווח ערכים עבור הפרמטרים הבאים:

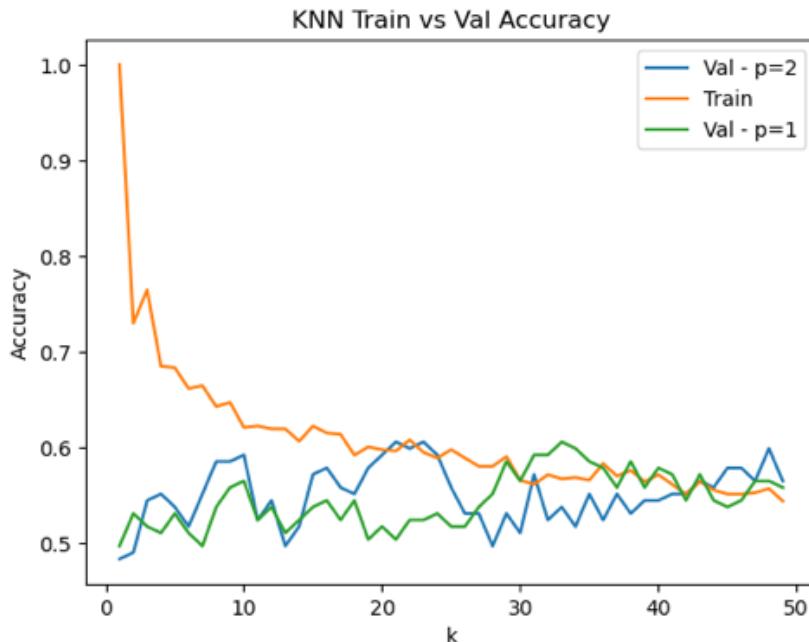
- **מספר השכנים:** בין 1 ל-50.
- **מדד המרחק:** אוקלידי (Euclidean) ומנהטן (Manhattan).
- **מדד ביצועים:** Accuracy: מכיוון שהנתונים מאוזנים ניתן להסתמך על מדד זה.

תהליך הכוון הzbוצע באמצעות חלוקת הנתונים ל-5 תת-קבוצות (5-Fold Cross-Validation)

לאחר כיוון הiper-פרמטרים, המודל אומן על סט האימון, והפרמטרים הטובים ביותר שנמצאו הם:

```
{'n_neighbors': 39, 'p': 1}
```

2. בשלב זה, לאחר ביצוע כיוון הiper-פרמטרים באמצעות `GridSearchCV` נ被执行 בדיקה נוספת בשיטת **Manual Iteration** למציאת הערכים האופטימליים עבור k ומטריקת המרחק. ננסה לראות אם ניתן לשפר את ביצועי המודול ולהשיג דיק גובה יותר על ידי התאמת ידנית של הפרמטרים, מעבר למה שנמצא בכיוון האוטומטי.



הגרף מציג את השינוי בדיק (Accuracy) של מודל KNN על סט האימון והולידציה עבור ערכים
שוניים של k ושתי מטריקות מרחק.

1. **מගמות האימון והולידציה:**

- עבור ערכי k נמוכים, דיק האימון גבוה מאוד בעוד שדיק הולידציה נמוך
(Overfitting)
- ככל ש- k -גדל, הדיק על סט הולידציה מת>'צבר סביר 0.55–0.6, עם שיפור מסוים
בとともに ביןוניים של k .

2. **השוואת מטריקות מרחק:**

- מרחק אוקלידי ($k=2$) מראה ביצועים מעט טובים יותר בהשוואה למנהטן ($k=1$)
עבור מרבית ערכי k

בחירה בפרמטרים:

נבחרו הפרמטרים $10 \geq k \geq 2$ מכיוון שהם משיגים איזון טוב בין ביצועי האימון להולידציה, עם
שיפור ב-Accuracy-על סט הולידציה ומניעת Overfitting .

Model Evaluation .6

- 6.1. בשלב זה, המודל עבר הערכת ביצועים בשתי שיטות עיקריות:
I. **GridSearchCV** - לחיפוש אוטומטי של הפרמטרים האופטימליים.
II. **Manual Iteration** - לבחינה ידנית של ערכי k ומטריקת המרחק, במטרה לשפר את
ביצועי המודל.
אלו תוצאות המודל בכל שיטה:
I. **GridSearchCV**

Performance Metrics for KNN Model (GridSearchCV) ($k=39, p=1$):

```
accuracy score : 0.5578231292517006
precision score: 0.5327102803738317
recall score: 0.7916666666666666
F1 score: 0.6368715083798882
```

Manual Iteration .II

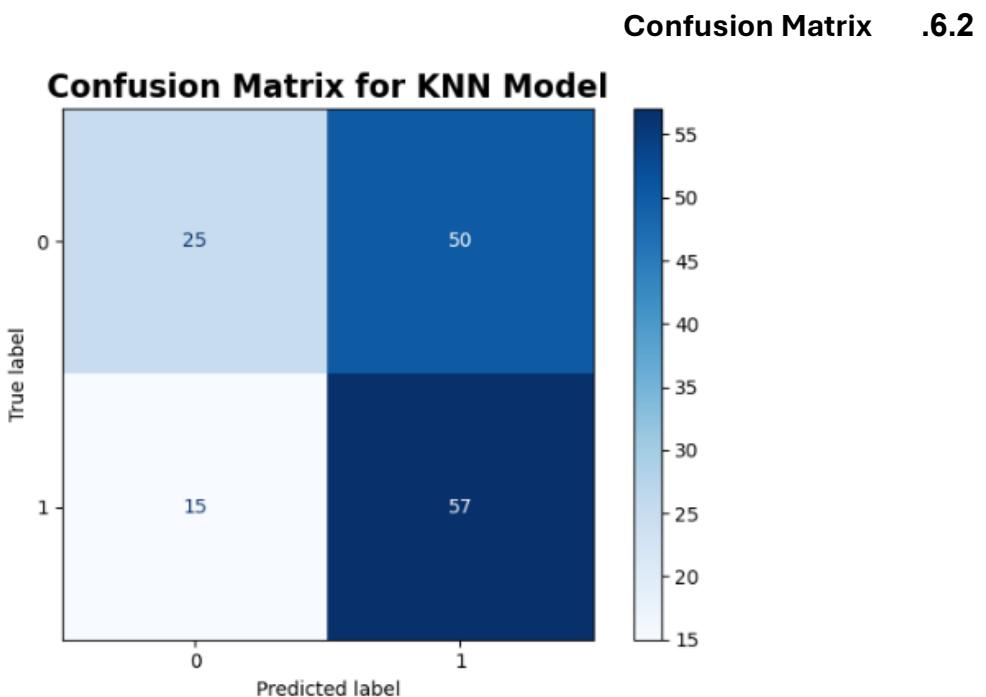
Performance Metrics for KNN Model (Manual Iteration) ($k=10, p=2$):

```
accuracy score : 0.5918367346938775
precision score: 0.5882352941176471
recall score: 0.5555555555555556
F1 score: 0.5714285714285715
```

בاهיעדר מידע על הקשר היישומי של הנתונים בפרויקט זה, נבחר להתמקד במדד ה-**F1 Score** מכיוון שהוא משלב באופן מאוזן בין **Precision** (דיוק תחזיות חיוביות) ו- **Recall** (זיהוי מקרים חיוביים).

מדד זה מתאים במיוחד כאשר אין העדפה ברורה לאחד משני ההיבטים הללו, ומטרתו להבטיח ביצועים מאוזנים.

בהתאם לכך, נבחר במודול **GridSearchCV**-עם הפרמטרים, $p=1$, $k=39$ אשר הציג את ה-**F1 Score**-הגבוה ביותר מבין האפשרויות, לצד **Recall** משופר. הבחירה במודול זה מאפשרת גישה יציבה ומאוזנת להמשך העבודה.



מתוך הגרף ניתן לראות:

- **True Positives (TP)** - המודל זיהה 57 מקרים חיוביים באופן נכון
- **True Negatives (TN)** - המודל זיהה 25 מקרים שליליים באופן נכון
- **False Positives (FP)** - המודל סיוג בעטעות 50 מקרים כחיוביים (בפועל שליליים)
- **False Negatives (FN)** - המודל פספס 15 מקרים חיוביים

.7 bonus challenge

לאחר יישום **PCA (Principal Component Analysis)** לצורך הקטנת מדדיות הנתונים, ביצועי המודל נמדד מחדש באמצעות מדד ה-**F1 Score**

f1_score with PCA: 0.6444444444444446

f1_score without PCA: 0.6368715083798882

car_shopping PCA הביא לשיפור קל (0.0075+) במדד ה-**F1 Score**

Model Implementation -Test set .8

התוצאות בוצעה על סט המבחן תוך שימוש בפרמטרים שנבחרו במהלך האימון. ($k=39$) התוצאות שהתקבלו הן:

accuracy score : 0.4594594594594595
precision score: 0.44660194174757284
recall score: 0.6666666666666666
F1 score: 0.5348837209302325

ניתוח התוצאות:

- **Accuracy** נמור יחסית: ממד הדיוק מצבי על כך שהמודל הצליח לזהות תוצאות נכונות רק בכ- 45.95% מהמקרים.
- גבוח משמעותית מ: **Recall** - המודל הצליח לזהות מקרים חיוביים באופן טוב יותר (66.67%) בהשוואה לדיקן התוצאות החיוביות (44.66%).
- **F1 Score** מואزن יחסית: עם ערך של 0.5349, המודל מציג איזון מסוים בין דיוק התוצאות (Recall) לבין מקרים חיוביים (Precision).

מסקנה:

למרות ש **Recall** - גבוח (66.67%), ביצועי המודל בסט המבחן אינם מספקים בכל הממדים, במיוחד במדד **Precision** ו-**Accuracy** - תוצאות אלו מעידות על מגבלותם במייד הקיימים במודל או אי ההתאמת של שיטת KNN לסט המבחן.

להמשך שיפור הביצועים, מומלץ:

1. **הוספת נתונים נוספים:** הרחבת סט הנתונים עשויה לשפר את המודל על ידי הגדלת כמות הדוגמאות הזרזינות למידה.
2. **שילוב עמודות נוספות:** הוספת מאפיינים רלוונטיים (features) שיוכולים לשפר את הפרדת הקבוצות ולהעלות את איותות התוצאות.