

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

מטרה

דוח זה מתאר את יישום והערכת אלגוריתם ה k-Nearest Neighbors (k-NN) לסיווג. נעשה שימוש במערך הנתונים students_data.csv תוך חקירת הנתונים, עיבוד מקדים, יישום המודל והערכת ביצועיו. המשימה כללה גם זיהוי וטיפול בחריגות אפשריות במערך הנתונים, כדי להבטיח דיוק מרבי של המודל.

1. Data Exploration and Preprocessing

1.1. Load the Dataset

מערך הנתונים נטען באמצעות ספריית 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 |

1.2. Basic Statistics

חושבו סטטיסטיקות 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 בטווחי הערכים והממוצעים בין המאפיינים בטבלה מצביעה על צורך בנרמול הנתונים, על מנת למנוע השפעה לא מאוזנת של תכונות מסוימות על המודל.

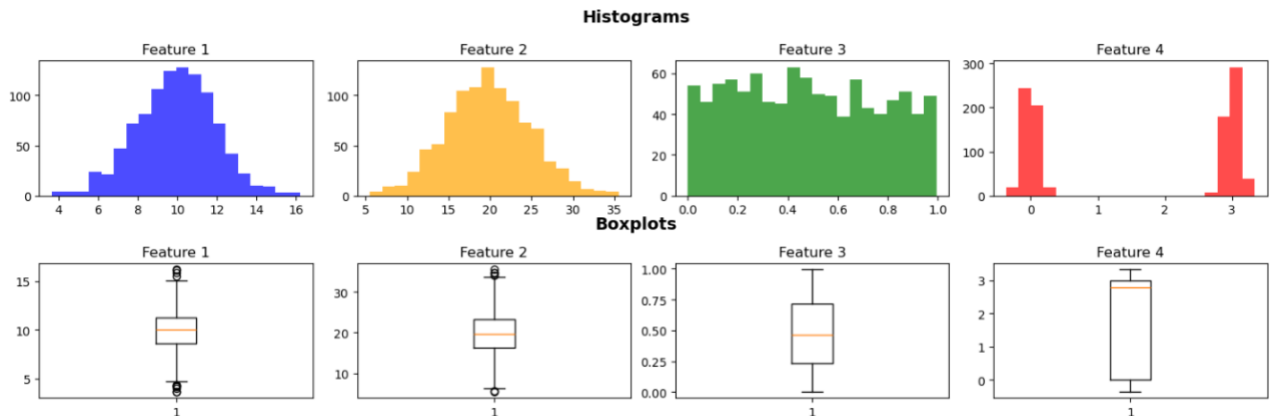
1.3 Data Cleaning

לא זוהו בדאטה 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
```

1.4 Visualizations

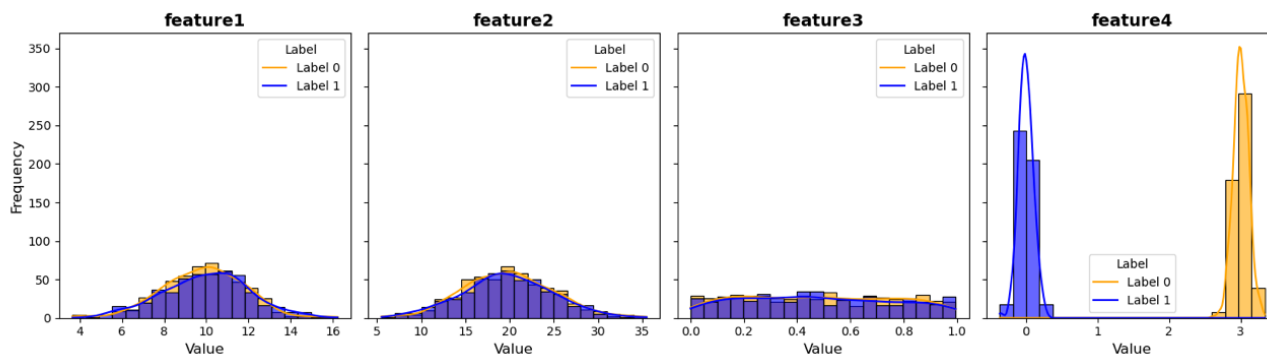
א. Histograms & Boxplots



- I. **Feature 1** ו **Feature 2**-מציגים התפלגות נורמלית, אך קיימים ערכים חורגים בקצוות.
- II. **Feature 3** מפוזר באופן אחיד, ללא חריגות.
- III. ל- **Feature 4** יש מבנה ייחודי עם הפרדה ברורה לשתי קבוצות.

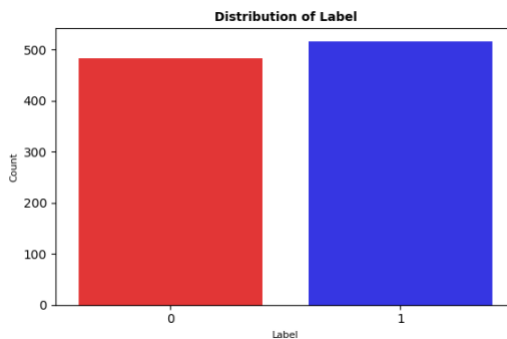
ב. Histograms Separated by Labels

Histograms of Features with Target Variable Separation



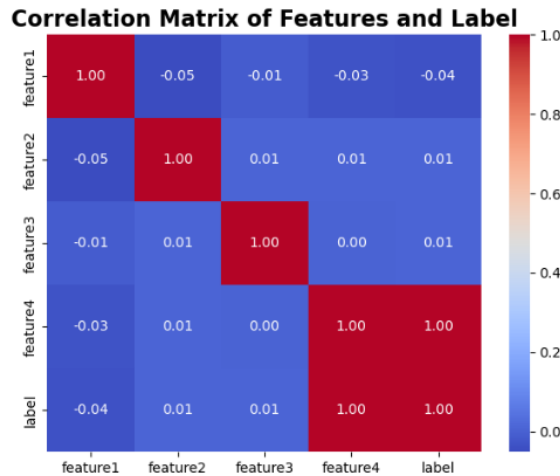
- י. **Feature 4** מספק את ההפרדה הברורה ביותר בין הקבוצות.
- י. **Feature 1** ו**Feature 2**-עשויים לתרום מעט בניתוחים נוספים, אך אינם מראים הפרדה ברורה.
- י. **Feature 3** אינו מספק הבדל משמעותי בין הקבוצות ולכן ככל הנראה יהיה פחות משמעותי להפרדה במודל.

ג. Countplot Label



ניתן לראות מהגרף שהדאטה מאוזנת.

Heatmap.7



- הקורלציה בין המאפיינים עצמם נמוכה מאוד, הדבר מעיד על כך שאין חפיפה משמעותית ביניהם. לכן, כנראה שכל אחת מהמאפיינים יוסיפו מידע ייחודי למודל.
- ישנה קורלציה מושלמת בין Feature 4 לעמודת הlabel.

2. Feature Engineering

2.1 Identify and Handle Anomalies

- כיוון שבסעיף הקודם זוהתה 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 להטות את המודל בצורה משמעותית, לגרום לאובדן כלליות ולפגוע בביצועיו על נתונים חדשים. בעזרת גרפי ה Boxplot-זוהו ערכים חריגים ב Feature 1 וב Feature 2-לאחר בדיקה, נמצא כי ישנם בסך הכול 17 ערכים חריגים בשני המאפיינים יחד, המהווים אחוז קטן מכלל מערך הנתונים. לכן, הוחלט להסיר את הערכים החריגים הללו כדי למנוע הטיה של המודל ולשפר את ביצועיו. החלטה זו נועדה לשמור על איזון במודל ולמנוע השפעה לא פרופורציונלית של הערכים החריגים.

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

2.2 Feature Scaling

כדי למנוע זליגת נתונים הוחלט לבצע קודם כל את שלב חלוקת הנתונים ורק לאחר מכן לבצע את הנרמול.

לכן שלב הנרמול יתועד בשלב 3.

3. Feature Scaling+data Splitting

3.1. הנתונים חולקו לשלושה חלקים- Train 70%, Validation 15%, Test 15% עם random_state=42 כדי שהחלוקה תהיה זהה בכל הרצה ו-shuffle=True כדי שאם 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. כעת בוצע נרמול הנתונים, ונבחרה באופן אקראי שיטת הנרמול הסטנדרטית (Standard Scaling). הנרמול כויל על בסיס נתוני X_train בלבד, בעוד השינוי בוצע גם על X_val ו-X_test וזאת כדי למנוע דליפה של מידע בין קבוצות הנתונים.

4. Model Implementation

4.1 Model Selection

בתור התחלה, נבחרו באופן שרירותי הפרמטרים הבאים עבור מודל KNN :

- k=5 (מספר השכנים הקרובים)
 - p=2 (מרחק אוקלידי)
 - המודל אומן על נתוני האימון ובוצע חיזוי על סט הולידציה. הערכת הביצועים נעשה שימוש 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 כיווןון היפר-פרמטרים באמצעות *GridSearchCV* לצד בדיקה ידנית של ערכי k ומדד המרחק (*distance metric*) כדי לזהות את השילוב האופטימלי עבור המודל.

1. *GridSearchCV*

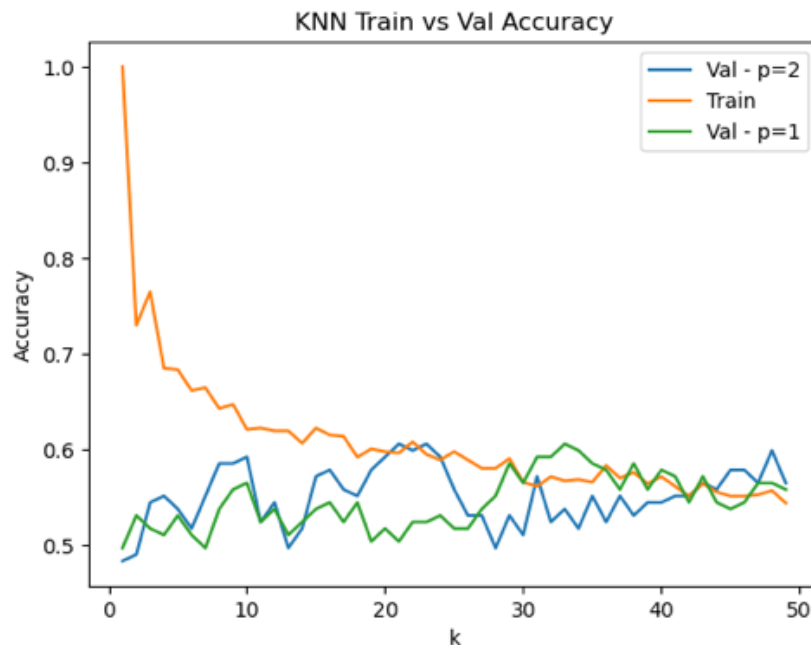
כיווןון המודל בוצע באמצעות טווח ערכים עבור הפרמטרים הבאים:

- **מספר השכנים**: בין 1 ל-50.
 - **מדד המרחק**: אוקלידי (Euclidean) ומנהטן (Manhattan).
 - **מדד ביצועים**, Accuracy, מכיוון שהנתונים מאוזנים ניתן להסתמך על מדד זה.
- תהליך הכיווןון התבצע באמצעות חלוקת הנתונים ל-5 תתי-קבוצות (5-Fold Cross-Validation)

לאחר כיווןון ההיפר-פרמטרים, המודל אומן על סט האימון, והפרמטרים הטובים ביותר שנמצאו הם:

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

1. בשלב זה, לאחר ביצוע כיווןון היפר-פרמטרים באמצעות *GridSearchCV* נבצע בדיקה נוספת בשיטת **Manual Iteration** למציאת הערכים האופטימליים עבור k ומטריקת המרחק. ננסה לראות אם ניתן לשפר את ביצועי המודל ולהשיג דיוק גבוה יותר על ידי התאמה ידנית של הפרמטרים, מעבר למה שנמצא בכיווןון האוטומטי.



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

1. מגמות האימון והולידציה:

- עבור ערכי k נמוכים, דיוק האימון גבוה מאוד בעוד שדיוק הולידציה נמוך (Overfitting)
 - ככל ש- k גדל, הדיוק על סט הולידציה מתייצב סביב 0.55–0.6, עם שיפור מסוים בטווחים בינוניים של k .
- #### 2. השוואת מטריקות מרחק:
- מרחק אוקלידי ($p=2$) מראה ביצועים מעט טובים יותר בהשוואה למנהטן ($p=1$) עבור מרבית ערכי k

בחירה בפרמטרים:

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

6. Model Evaluation

6.1. בשלב זה, המודל עבר הערכת ביצועים בשתי שיטות עיקריות:

- GridSearchCV** - לחיפוש אוטומטי של הפרמטרים האופטימליים.
 - Manual Iteration** - לבחינה ידנית של ערכי k ומטריקת המרחק, במטרה לשפר את ביצועי המודל.
- אלו תוצאות המודל בכל שיטה:
- 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) ($k=10$, $p=2$):

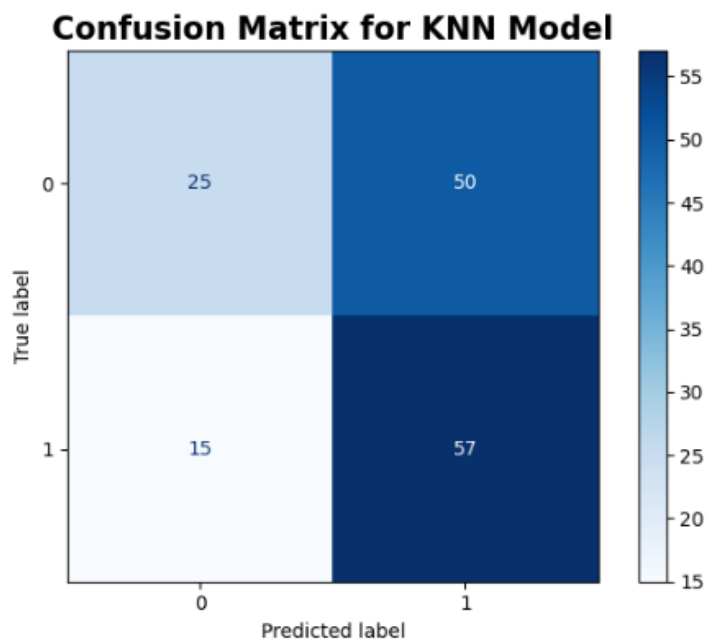
accuracy score : 0.5918367346938775
precision score: 0.5882352941176471
recall score: 0.5555555555555556
F1 score: 0.5714285714285715

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

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

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

6.2 Confusion Matrix



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 סיווג בטעות 50 מקרים כחיוביים (בפועל שליליים)
- **False Negatives (FN)** - המודל פספס 15 מקרים חיוביים

7. bonus challenge

לאחר יישום **PCA (Principal Component Analysis)** לצורך הקטנת ממדיות הנתונים, ביצע המודל נמדדו מחדש באמצעות מדד ה-**F1 Score**

f1_score with PCA: 0.6444444444444446

f1_score without PCA: 0.6368715083798882

כר שיישום PCA הביא לשיפור קל (+0.0075) במדד ה-**F1 Score**.

8. Model Implementation -Test set

התחזית בוצעה על סט המבחן תוך שימוש בפרמטרים שנבחרו במהלך האימון. ($k=39$, $p=1$) התוצאות שהתקבלו הן:

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

ניתוח התוצאות:

- **Accuracy** נמוך יחסית: מדד הדיוק מצביע על כך שהמודל הצליח לזהות תחזיות נכונות רק בכ-45.95% מהמקרים.
- **Recall** גבוה משמעותית מ: Precision- המודל הצליח לזהות מקרים חיוביים באופן טוב יותר (66.67%) בהשוואה לדיוק התחזיות החיוביות (44.66%)
- **F1 Score** מאוזן יחסית: עם ערך של 0.5349, המודל מציג איזון מסוים בין דיוק התחזיות (Precision) ליכולת לזהות מקרים חיוביים. (Recall)

מסקנה:

למרות ש **Recall**-גבוה (66.67%), ביצועי המודל בסט המבחן אינם מספקים בכל המדדים, במיוחד במדדי **Accuracy** ו **Precision**-תוצאות אלו מעידות על מגבלות במידע הקיים במודל או אי התאמה 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 הזמינות ללמידה.
2. **שילוב עמודות נוספות**: הוספת מאפיינים רלוונטיים (features) שיכולים לשפר את הפרדת הקבוצות ולהעלות את איכות התחזיות.