**מבוא למערכות לומדות- תרגיל בית 1- דו״ח עבודה**

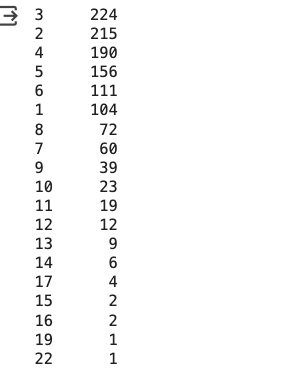
**מגישים:   
  
גל קסטן 316353176**

**חן פרי – 313283657**

(Q1) במסד הנתונים יש 1250 שורות ו26 עמודות.

(Q2)

Value counts of conversations\_per\_day:

****פלט:

בעולם האמיתי, המשתנה ״ מספר שיחות ליום״ מתייחס כנראה לממוצע מספר השיחות ביום שאדם פנים מול פנים. משתנה זה משמש כמעיין מדד לפעילותו החברתית של האדם ולחשיפה שלו למחלות מדבקות.

סוג המשתנה הוא אורדינאלי כיוון שמצד אחד המשתנה קטגורי (הוא דיסקרטי, ויש מספר סופי של ערכים שהוא יכול לקבל) אבל מצד שני קיים סדר טבעי בין הערכים שהמשתנה יכול לקבל (מדובר במספרים טבעיים עם יחס סדר) ויש משמעות כמותית לערכים גדולים/קטנים של כמות שיחות ביום.

(Q3)

|  |  |  |
| --- | --- | --- |
| Type | Description | Feature name |
| Other | Incremental id of the patient | **patient\_id** |
| Ordinal | Age of the paitent | **age** |
| Categorical | Sex of the patient (Male/Female) | **sex** |
| Continuous | Weight of the patient in kilograms | **weight** |
| Categorical | The blood type of the patient(O/A/AB/B and +/-) | **blood\_type** |
| Other | Current location of the patient (latitude and longitude). | **current\_location** |
| Ordinal | The number of siblings the patient has | **num\_of\_siblings** |
| Ordinal | A Score describing the patient’s level of happiness | **happiness\_score** |
| Ordinal | A categorical representation of the patient’s household income | **household\_income** |
| Ordinal | Average number of conversations the patient has everyday | **conversations\_per\_day** |
| Ordinal | The sugar level measurements for the patient | **sugar\_levels** |
| Ordinal | The patient’s level of sport activity on scale from 0-5 | **sport\_activity** |
| Other | Textual description of any symptoms the patient may have reported | **symptoms** |
| Other | The date in which a PCR test was conducted | **pcr\_date** |
| Continuous | Measurement #1 from a PCR test | **PCR\_01** |
| Continuous | Measurement #2 from a PCR test | **PCR\_02** |
| Continuous | Measurement #3 from a PCR test | **PCR\_03** |
| Continuous | Measurement #4 from a PCR test | **PCR\_04** |
| Continuous | Measurement #5 from a PCR test | **PCR\_05** |
| Continuous | Measurement #6 from a PCR test | **PCR\_06** |
| Continuous | Measurement #7 from a PCR test | **PCR\_07** |
| Continuous | Measurement #8 from a PCR test | **PCR\_08** |
| Continuous | Measurement #9 from a PCR test | **PCR\_09** |
| Continuous | Measurement #10 from a PCR test | **PCR\_10** |

**(Q4)**

WE need to write this in Hebrew   
  
**(from the comments on reference)**   
It is important that we use the exact same split for our analysis so that we can reproduce the experiment and obtain the same models. Moreover, when we compare machine learning algorithms we would like them to be evaluated on the same subsets of the dataset.

**(FROM CHAT)**

Using the exact same split for all analyses is crucial because:

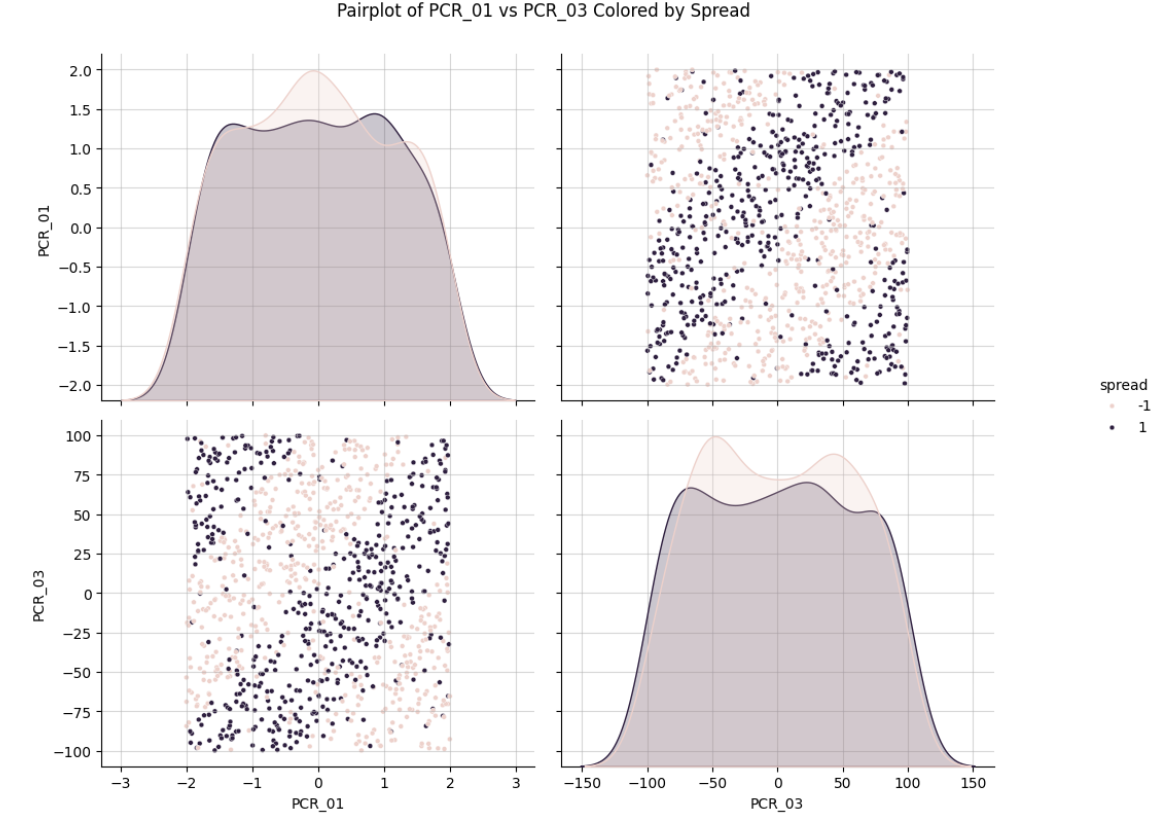
Reproducibility: It ensures that the results of your analysis can be reproduced by anyone else or in future experiments, which is a fundamental aspect of scientific research.

Fair Comparison: When comparing different models or configurations, using the same data split ensures that all models are evaluated under the same conditions, making the comparisons fair.

Avoid Data Leakage: Keeping a consistent split between training and testing data helps prevent data leakage. Data leakage occurs when information from the test set is inadvertently used to train the model, leading to overly optimistic performance estimates. By maintaining a strict separation and not using the test set for decision-making during model selection or tuning, you ensure that the model's performance evaluation is realistic and reflects its ability to generalize to unseen data.

Validation Consistency: When you eventually introduce a validation set for tuning hyperparameters or making decisions about the model, having a consistent training/test split ensures that the validation process is stable and reliable, further avoiding data leakage and ensuring that performance metrics are meaningful.

**(Q5)**



על בסיס התמונה, הפיצ׳רים pcr\_01, pcr\_02 עשויים להיות שימושיים לחיזוי spread ביחד מאחר וניתן לראות כי הדאטה כמעט פריד ל4 אזורים שונים (יש outliers). עם זאת, הדאטה לא פריד לינארית ולכן נצטרך מודל שיודע ללכוד קשרים מורכבים יותר.

כל פיצ׳ר בעצמו לא מספיק כדי לחזות את הדאטה- ההתפלגויות השוליות של שני המשתנים מראות שיש חפיפה בערכים של הקטגוריות של spread ואין כמעט אזורי הפרדה ולכן למשל לא למצוא ערך מפריד שיתן הפרדה לינארית בין תחום ערכי pcr (1 או 3) לבין סוג הspread.

**(Q6)**

**correlation between spread and PCR\_01**: 0.005909176667077393

**correlation between spread and PCR\_03:** -0.004310898790492347

מן הממצאים עולה כי הקורלציה בין pcr\_01 לpcr\_03 היא כמעט 0 כלומר הקשר הלינארי בין המשתנים חלש. ממצאים אלו תומכים במה שמצאנו קודם לכן, ראינו קודם כי כאשר מסתכלים על ההתפלגות השולית של כל אחד מן המשתנים לבד בהפרדה לפי קטגוריות spread, הייתה חפיפה בין הערכים, כלומר לא היתה הפרדה ברורה לעין בין טווח ערכים שpcr\_01 מקבל עבור spread=1 וטווח ערכים עבור spread = -1. באופן דומה גם עבור pcr\_03.

\*\*בבדיקת הקורלציה בודקים רק קשר בין כל משתנה pcr לspread אך לא את האפקט המשולב שלהם.

"Based on the pairplot, the features **PCR\_01** and **PCR\_03** might jointly offer predictive value for 'spread' since the data points are visually grouped into four distinct regions, albeit with some outliers present. This suggests that there may be a non-linear relationship where certain combinations of **PCR\_01** and **PCR\_03** values are associated with specific 'spread' categories. However, since the data does not exhibit linear separability, a model capable of capturing complex relationships, such as a kernel SVM or a neural network, would be more appropriate for this task.

Individually, each feature does not demonstrate a significant ability to predict 'spread'—the marginal distributions for both **PCR\_01** and **PCR\_03** show considerable overlap across the 'spread' categories, indicating no clear threshold or value range that separates 'spread' = 1 from 'spread' = -1.

The calculated correlation coefficients between 'spread' and each PCR feature are very close to zero (**PCR\_01**: 0.005909176667077393, **PCR\_03**: -0.004310898790492347), reinforcing the earlier observation that there is no strong linear relationship between these features and 'spread'. The lack of correlation supports the hypothesis that the relationship between the PCR features and 'spread' is non-linear, as linear correlation would not capture complex patterns such as clusters or non-linear arrangements seen in the pairplot.

Furthermore, the correlation between **PCR\_01** and **PCR\_03** being close to zero suggests that these features are largely independent of each other, which is beneficial for modeling as they may provide unique information to the predictive model.

In conclusion, while the individual linear correlations of **PCR\_01** and **PCR\_03** with 'spread' are negligible, their combined pattern as observed visually suggests the potential for predictive modeling when using non-linear methods. This analysis underscores the importance of considering multivariate effects and non-linear relationships in predictive modeling, and it points towards the need for sophisticated machine learning techniques that can leverage the complex structure in the data."

(Q7)

זמן ריצה של פונקציית החיזוי על נקודה אחת P

* עבור נקודה אחת הפונקציה cdist תחשב את המרחק בין הנקודה p לכל אחת מm הנקודות בdata set. עלות חישוב המרחק תלויה במימד של הנקודה p, אם נתחשב רק במכפלות שצריך כדי לחשב את המרחק האוקלידי- d מכפלות

לכן הסיבוכיות של שלב זה היא O(dm)

* לאחר מכן- מחפשים את k השכנים הכי קרובים לנק p, באמצעות הפונקציה np.argpartition. ניתן להעריך שהפונקציה משתמשת מאחורי הקלעים באלגוריתם partition שסיבוכיות הזמן שלו לינארית בגודל האיברים במערך. לכן עלות שלב זה- O(m).
* בחירת k האינדקסים הראשונים במערך שחוזר מ argpartition- O(1) -
* בחירת הלייבלים של k השכנים הכי קרובים מתוך מערך הלייבלים – O(k)
* סכום k הלייבלים - O(k)
* בחירת הסימן של הסכום – O(1)

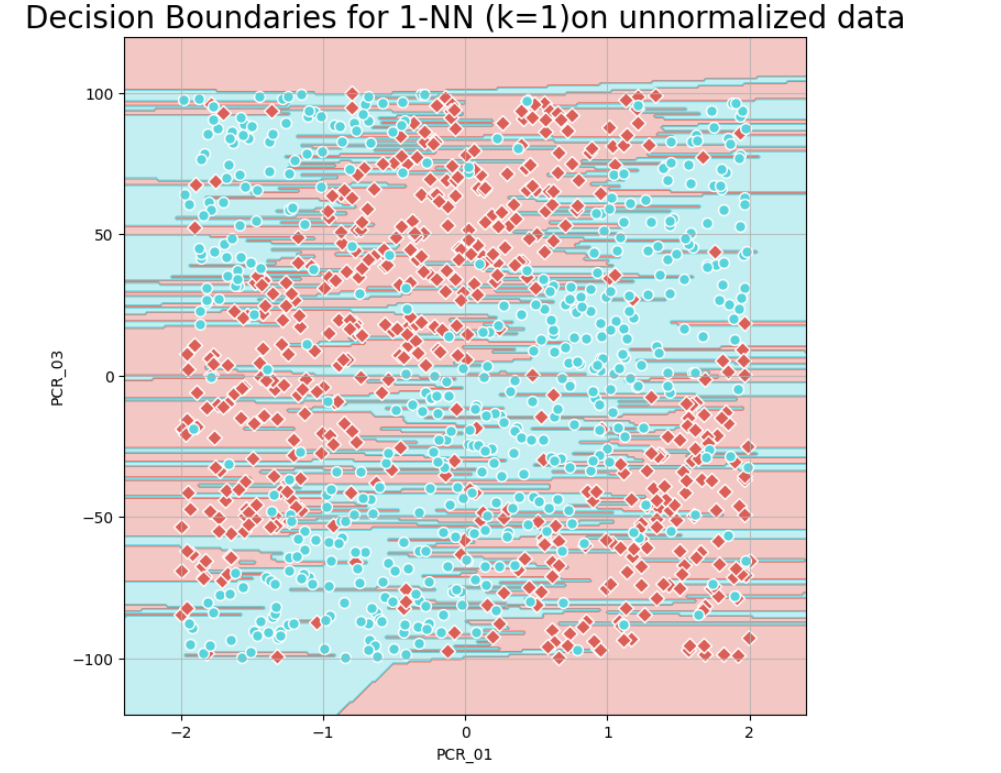
סהכ סיבוכיות – O(md+d+k)

To analyze the time complexity of the prediction function in the k-NN implementation, let's consider the steps involved in making a prediction for a single test data point:

1. **Distance Calculation**: The first step is to compute the distance from the test point to each of the **m** training data points across **d** dimensions. Using the **cdist** function from SciPy, this involves calculating **m** distances, where each distance computation across **d** dimensions is **O(d)**. Thus, the complexity for this step is **O(m\*d)**.
2. **Sorting or Partitioning for Nearest Neighbors**: After calculating the distances, we need to find the **k** nearest neighbors. The implementation uses **np.argpartition** to partition the distances array, which is more efficient than full sorting for this purpose. The average time complexity of partitioning is linear with respect to the number of elements being partitioned, so this step has a complexity of **O(m)** since it operates on the array of distances to each training point.
3. **Majority Voting**: Once the **k** nearest neighbors are identified, the algorithm determines the majority class among these neighbors. This involves looking up the **k** labels and then summing them or counting occurrences, which is **O(k)** in the worst case (if we assume a simple sum or count operation without optimization).

Combining these, the overall time complexity for predicting the label of a single test data point is dominated by the distance calculation and partitioning steps. Thus, the time complexity can be approximated as **O(m\*d + m + k)**. However, since **m** and **d** are typically much larger than **k**, especially in large datasets or high-dimensional data, the **k** term does not significantly affect the overall complexity, and the dominant term is **O(m\*d)** due to the distance computation step.

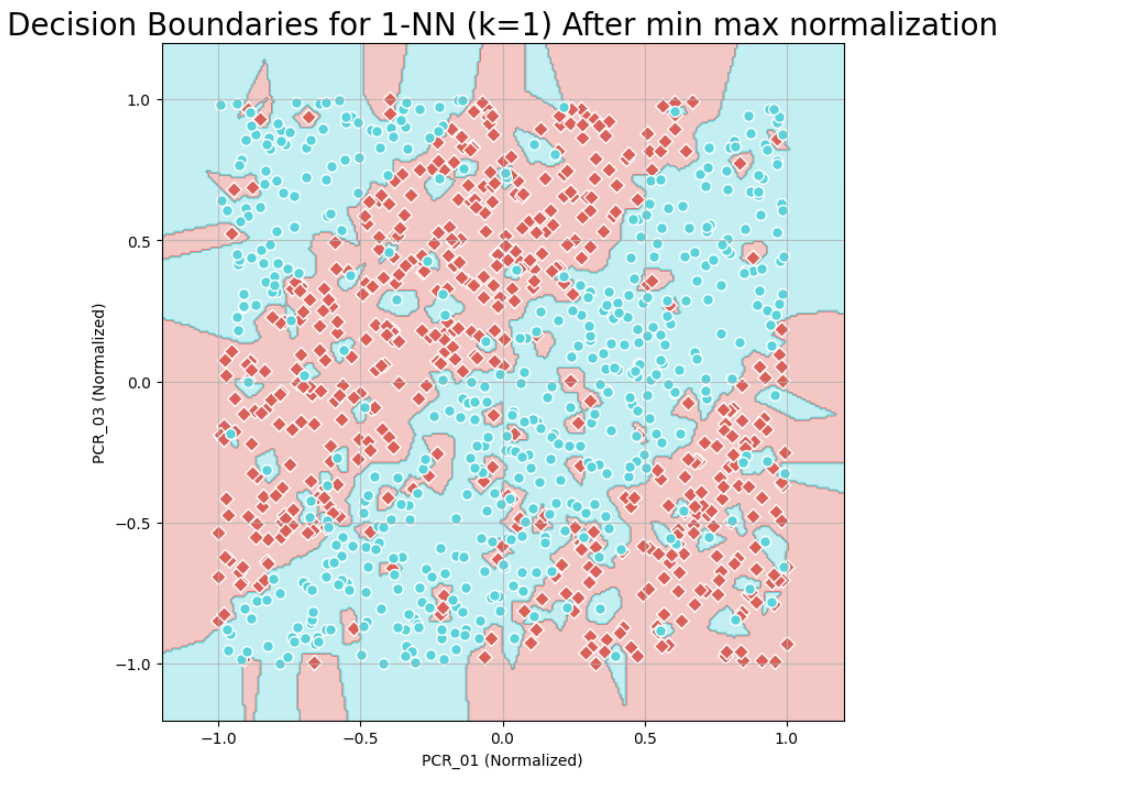
In summary, the prediction function's time complexity for a single test data point is primarily **O(m\*d)**, driven by the need to compute distances across all training points and dimensions.

(Q8)

מידת הדיוק של המודל על קבוצת האימון- 1 (כצפוי, מאחר והנקודה הכי קרובה לנקודה ב1-NN היא הנקודה עצמה).

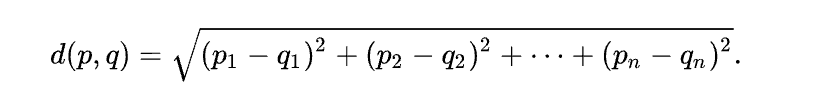
מידת הדיוק של המודל על קבוצת המבחן – 0.664

(Q9)



מידת הדיוק של המודל על קבוצת האימון- 1 (כצפוי, מאחר והנקודה הכי קרובה לנקודה ב1-NN היא הנקודה עצמה).

מידת הדיוק של המודל על קבוצת המבחן – 0.756

נורמליזציה של הנתונים חשובה לאלגוריתם הknn מאחר ואלגוריתם הknn מחפש את השכנים הקרובים ע״י פונקציית מרחק , ספציפית במקרה של התרגיל מרחק אוקלידי, וזוהי פונקציה שרגישה להבדלים בקנה המדידה בו מודדים כל פיצ׳ר. ניזכר כי מרחק אוקלידי בין 2 נקודות נמדד באופן הבא: כאשר .

מהנוסחה ניתן לראות שפי׳צרים הנעים על שטווח הערכים שלהם רחב (קנה מידה גדול יותר) יתרמו יותר לגודל המרחק האוקלידי מאחר וההפרשים בפיצ׳רים האלו יהיו גדולים יותר, על אף שאינם בהכרח חשובים יותר מפיצ׳רים אחרים.

כתוצאה מכך, פיצ׳רים אלו הופכים לדומיננטיים ומאפילים על פיצ׳רים עם קנה מידה נמוך.

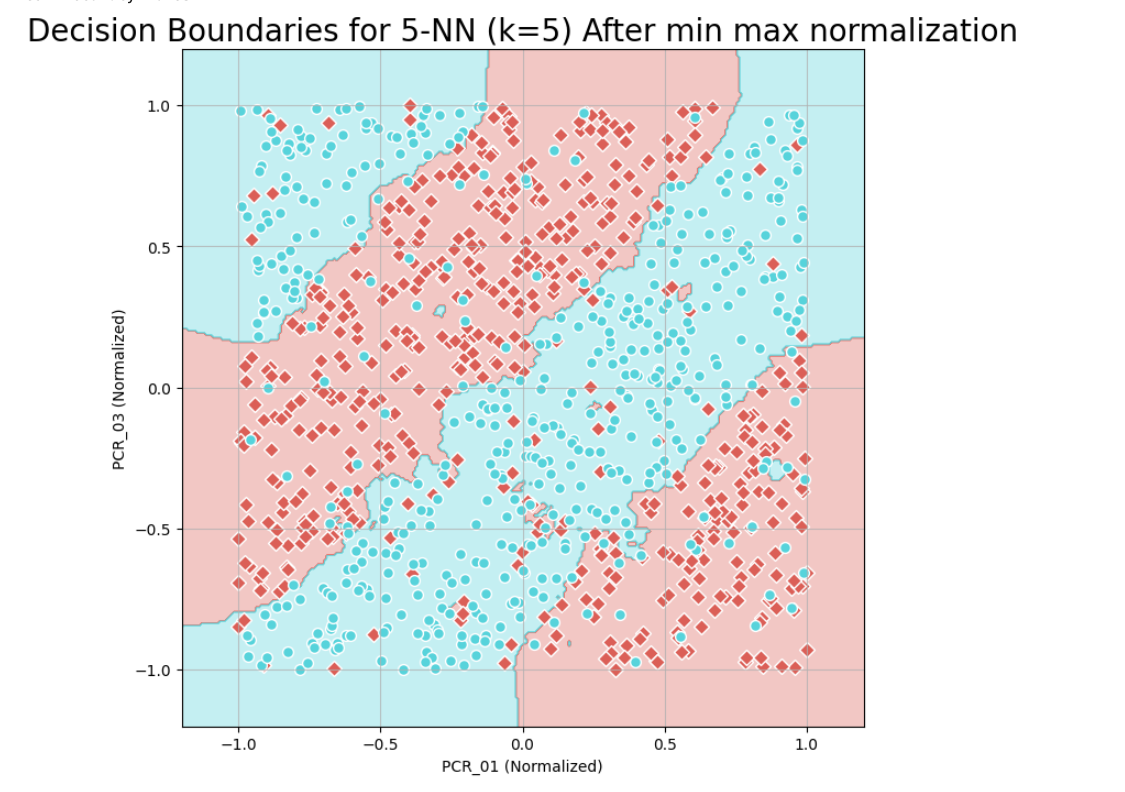
בשאלה 8, לפני שביצענו נורמליזציה על הנתונים, ניתן לראות שגבולות ההחלטה של knn נראים מוזר(פסים אופקיים כחולים על כל אזורי ההחלטה).

הפסים הכחולים האלה קשורים לעובדה שטווח הערכים של pcr\_03 [-100,100] וטווח הערכים של pcr\_01 [-2,2] כלומר פיצ׳ר pcr\_03 הרבה יותר דומיננטי בחישוב המרחק האוקלידי. זה גורם לתופעת ה״פסים האופקיים הכחולים״ גם באזורים בהם ריבוי נקודות אדומות.

מאחר ולפיצ׳ר pcr\_03 קנה מידה גדול יותר, מקבלים שעבור כל נקודה – הנקודות הקרובות יותר אליה הן נקודות בציר האופקי (שם הערך של pcr\_03 קבוע בין כל השכנים ולכן המרחק האוקלידי קטן יותר) ואילו הנקודות בציר האנכי רחוקות יותר (אפילו הפרשים קטנים בין ערכי pcr\_03 הן בקנה מידה גדול יותר). כתוצאה מכך , יש פסים כחולים אופקיים (נקודה כחולה רחוקה בציר האופקי היא דווקא יותר קרובה אוקלידית מאשר נקודות בציר האנכי).

לאחר שעשינו נרמול, קנה המידה של שני הפיצ׳רים היה באותה קנה מידה – [1,-1] ולכן לא היה פיצ׳ר יותר דומיננטי מהשני. ונראה אכן שקיבלנו שאזורי ההחלטה של מסווג הknn הרבה יותר קרובים למה שהיינו מצפים (מאוד קרובים למה שקיבלנו בpair plot).

(Q10)



מידת הדיוק של המודל על קבוצת האימון- 0.879

מידת הדיוק של המודל על קבוצת המבחן –0.852

**ההשפעה של k על אזורי ההחלטה של אלגוריתם knn הם :**

\*k נמוך גורם לoverfitting- כאשר מספר השכנים קטן, כפי שראינו עבור k=1, יש התאמה של המודל לכל הנקודות בקבוצת האימון, אפילו לנקודות הנחשבות רעש, outliert. כך ניתן לראות בתמונה של אזורי ההחלטה עבור k=1, אזורים קטנים כחולים סביב נקודות כחולות , גם כאשר כל הסביבה אדומה ולהפך.

\*כאשר k גדל, אזורי ההחלטה נהיים ״חלקים יותר״, פחות רגישים לרעש ולנקודות בודדות. K גדול יותר משפר את יכולת ההכללה של המודל. ניתן לראות שמידת הדיוק של knn עבור k=5 גדולה יותר (עם זאת מידת הדיוק על קבוצת האימון ירדה).

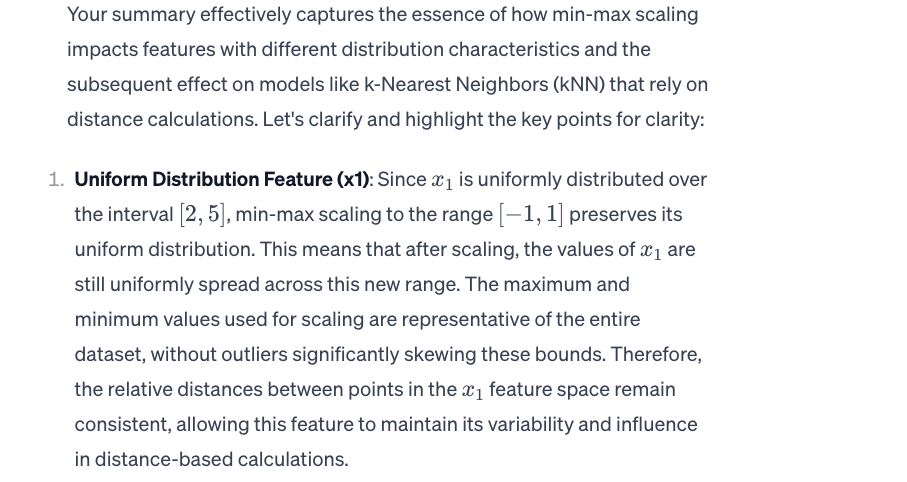
עם זאת, עבור k גדול מדי נקבל שאזורי ההחלטה כבר מתחילים להטשטש (הפשטה גדולה מדי של המודל וחוסר יכולת שלו לתפוס את התבניות בdata , underfitting מאחר וכאשר המסווג יחליט לסווג נקודה הוא יתחשב בתיוג של נקודות רחוקות מדי מהנקודה.

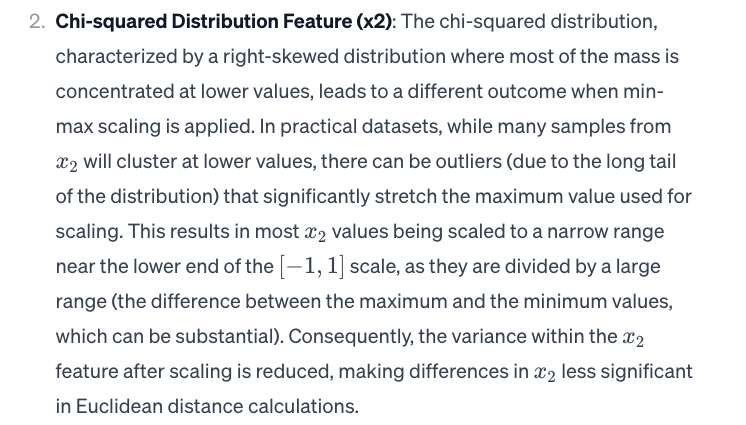
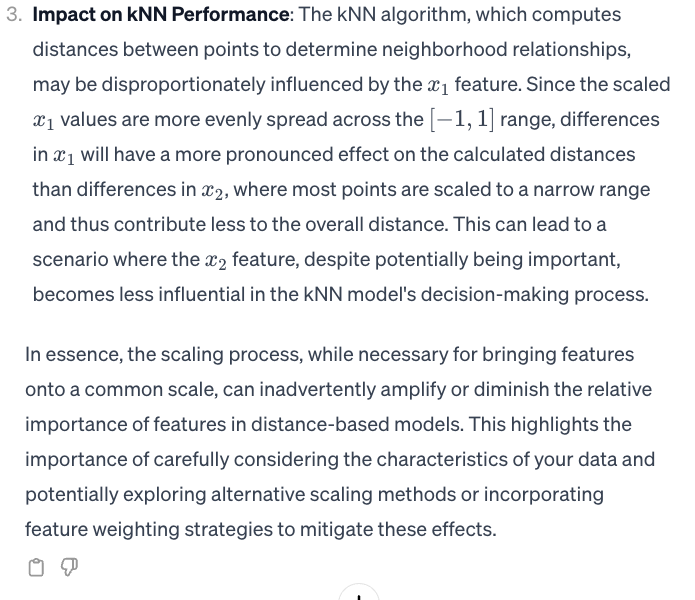
(Q11)

בקצרה-

מאחר והפיצ׳ר x1 מתפלג באופן אחיד על קטע [2,5] , גם לאחר min-max scale , הפיצ׳ר x1 החדש יתפלג גם באופן אחיד בין [1,-1] , כלומר גם אם נדגום n דגימות, הmax ,min לא יהיו outliers ביחד לשאר הנקודות.

לעומת זאת , רוב המסה של ההתפלגות כי בריבוע מרוכזת בטווחים נמוכים [ההסתברות לקבל ערך קטן שווה ל3 היא כמעט 0.8) מה שיכול לגרום לכך שעבור סדרה של הרבה דגימות – רוב הדגימות יהיו מרוכזות בערכים נמוכים אך מעט דגימות יהיו גדולות (outliers) ויביאו לכך שהmaximum גדול. כתוצה מכך בנירמול min-max רוב הנקודות יהיו מרוכזות בטווח קטן , קרוב ל0 (מאחר ומחלקים בהפרש בין המקסימום למינימום שככל הנראה יהיה גדול). דבר זה יגרום לכך שהפיצ׳ר x2 יהיה פחות חשוב במונחים אוקלידים, (בציר של x2 המרחק בין הפיצ׳רים יהיה קטן מאד וכמעט חסר חשיבות בחישוב המרחק האוקלידי, ואילו הפיצ׳ר x1 שדגימותיו מפוזרות יחסית אחיד , יהיה יותר דומיננטי החישוב המרחק האוקלידי. ושוב תקרה אותה תופעה שהסברנו עליה בשאלה 9.





(Q12)

מאחר ויש 8 סוגים שונים של סוגי דם (A+,A-,AB+,AB-,B+,B-,O+,O-) נצטרך 8 קטגוריות בוליאניות. (כעקרון מספיק גם להשתמש ב7 קטגוריות , כך שאם הערך עבור כל הפיצ׳רים האלה false זה אומר שהפיצ׳ר שייך לקטגוריה השמינית)

(Q13)

כן , אפשר להפיק מידע מהפיצ׳ר סימפטומים כדי לסייע בניבוי.   
כאשר חקרנו את הדאטה, שמנו לב שניתן לחלץ מהטקסט החופשי של כל חולה קבוצת של סימפטומים שונים ולייצר עבור כל סימפטום כזה פיצ׳ר בוליאני שהוא חיובי במידה ולחולה יש את הסימפטום הזה.