Part 1

(A1) There are 1250 rows and 25 columns in the dataset

(A2)

conversations_per_day	
3	218
2	204
5	179
4	168
1	108
6	107
7	94
8	54
9	42
10	29
11	16
13	8
12	7
14	6
16	5
15	3
17	1
29	1
Name: count, dtype: int64	

We assume that the feature refers to how many face-to-face conversations each person has every day.
This feature is ordinal because its categories are ordered on a hierarchical scale (high to low). This means that the categories have a meaningful order.

(A3)

Feature	Description	Type
patient_id	Index of each row	Other
age	Age of the person	Continuous
sex	Sex of the person	Categorical
weight	Weight of the person	Continuous
blood_type	Blood type of the person	Categorical
current_location	Location of quarantine (latitude and longitude)	Other
num_of_siblings	Number of siblings the person has	Categorical

happiness_score	Ordinal feature that indicates how happy the person is	Ordinal
household_income	How much money the person makes	Continuous
converstations_per_day	How many face to face conversation the person has on average every day	Ordinal
sugar_levels	Test to check sugar level in blood	Continuous
sport_activity	How many sport activities the person does every week	Categorical
pcr_date	Date the first pcr test was taken	Other
PCR_01	The result of the PCR test is interpreted such that a more negative number indicates a higher certainty of a negative result, while a more positive number suggests a higher likelihood that the person still has COVID	Continuous
PCR_02		
PCR_03		
PCR_04		
PCR_05		
PCR_06		
PCR_07		
PCR_08		
PCR_09		
PCR_10		

(A4) Using the same data split keeps the results consistent, comparisons fair, and conclusions reliable. It ensures everything is repeatable and valid, which is crucial in data analysis.

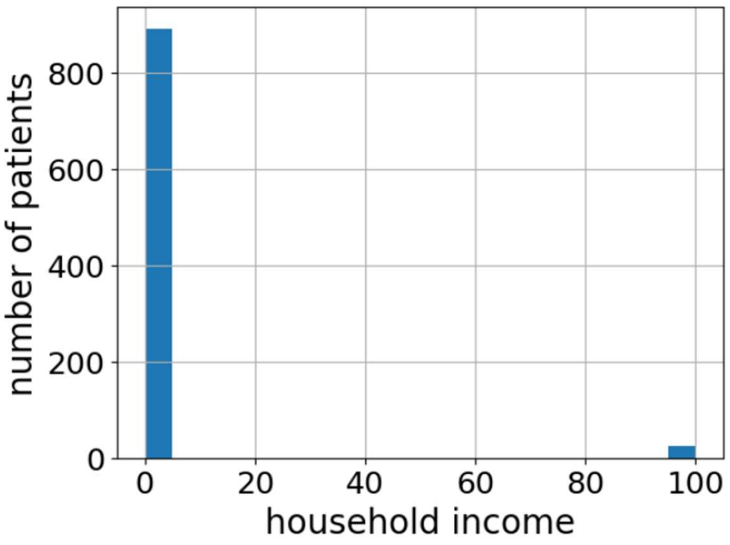
Part 2

(A5)

For both, training set and test set, the only field with missing values was "household_income".

Training set has a total of 83 missing values, the test set has a total of 26 missing values.

(A6)



We can see outliers with a value of about 100 in the histogram.

Other entries are below 5.

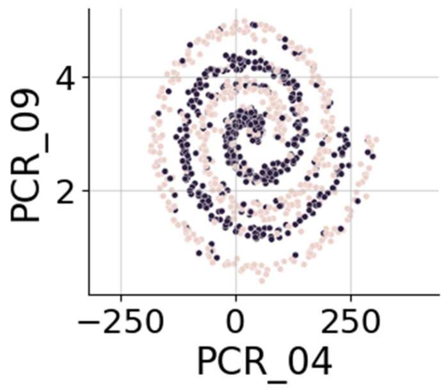
(A7)

Median household income: 0.7
Mean household income: 3.433

There is a significant difference between the two values, because more than half of the values are less than one. So is the median.
Since we have few values that are much greater than 1, as we can deduct from the histogram, the mean value moves toward those values.
Therefore we end up with a mean value much greater than the median value.
The method we think suits our situation best is filling the missing data with the median value.
Since we believe the data influenced the mean value to deviate from the median was an outlier in the first place, using the median which was not influenced is a better idea.

Part 3

(A8)



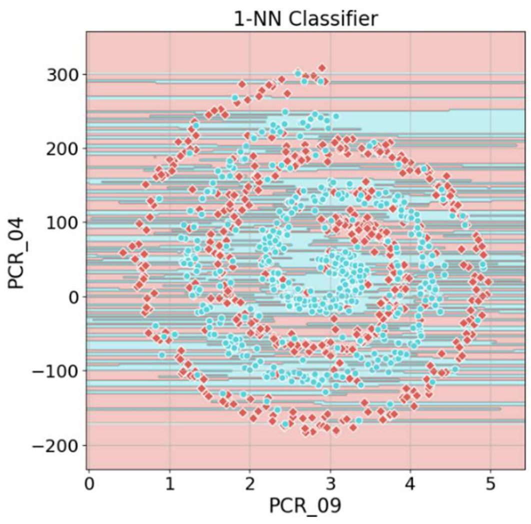
Based on the pairplot, the most visually distinct separation between the spread feature seems to be present in the scatter plot involving PCR_04 and PCR_09. These features exhibit a spiral pattern that distinguishes the two spread categories most clearly.

(A9)

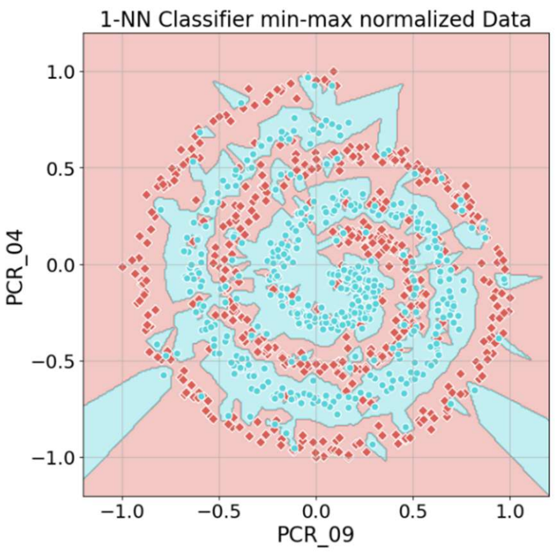
1) Distances: cdist functions computes the Euclidean distance between each pair of the two collections of inputs, X is a single test point, m is the number of data points and the data dimension is d so the complexity is O(m*d)
2) nearest_neighbors_indices: the np.argpartition function sorts the array to find the k nearest neighbors (smallest distances from our test point X) The complexity of this function is O(m) (using partition) for each row of the distances array
3) nearest_neighbors_labels: O(k) because we are retrieving the labels of the k nearest neighbors.
4) Majority vote: O(k) because we are summing the k labels and taking the sign
5) combining these steps, the overall time complexity is O(m*d) + O(m) + O(k) + O(k) = O(m*d) + O(m)+2*O(k) = O(md) because k>m will be equaly treated like k=m and d is the data dimension.

(A10)

Training Accuracy: 1.0
Test Accuracy: 0.58



(A11)



Training Accuracy: 1.0

Test Accuracy: 0.692

Before Normalization

- The decision boundary appears more irregular, and the spread of the data points is influenced by the scale of the original features.

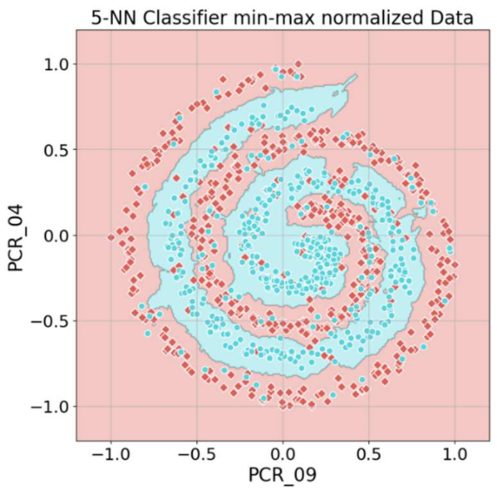
After Normalization

- The decision boundary appears smoother and more regular, indicating a more consistent influence of both features.
- The test accuracy improved to 0.692, showing a better performance of the model on the normalized data.

Normalization is important for kNN models because:

- By normalizing the features to a common scale, each feature contributes equally to the distance calculation, allowing for a fair comparison and improving the model's performance.

(A12)



Training Accuracy: 0.857

Test Accuracy: 0.796

For k=1, the model overfits the training data, resulting in high training accuracy but lower test accuracy.
With k=5, the model generalizes better, reducing overfitting and improving test accuracy.

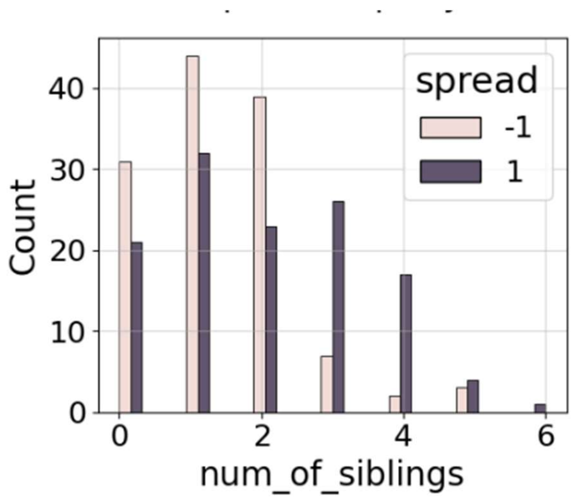
(A13)

Normalizing both features using min-max scaling to
the range [-1, 1] is prablomatic because of their different distributions.
The first feature, which is uniformly distributed between [2, 5], will spread evenly across the new range.
But the chi-squared feature, which is highly skewed with most values near 0, will get compressed with most of its values near -1. This makes it harder for algorithms like k-NN to measure distances properly, as the uniform feature will dominate, leading to a less effective model.

Part 4

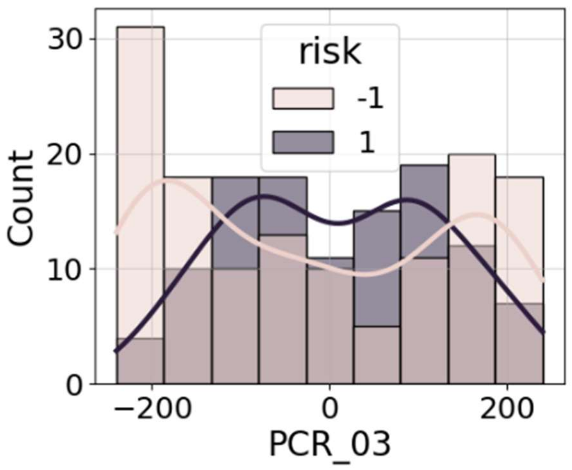
(A14)

According to the univariate analysis, a feature that seems informative for predicting the spread target variable is 'num_of_siblings'. The plot suggests it is an informative feature because of the following: For each value, there is a clear majority of subjects that are positive or negative to the spread variable. Therefore, it might be a good feature to consider as we try to classify which new subjects might be positive to the spread variable.



(A15)

According to the univariate analysis, the 'PCR_3' feature seems informative for predicting the risk target variable. Similary to the reason we chose the 'num_of_siblings' feature as an informative feature to predict the spread target variable, the 'PCR_3' feature shows that for most accepted values, there is a majority of samples that are positive or negative to the risk variable. Furthermore, the distribution estimated by the histogram suggests that negative and positive samples for the risk variable have clearly distinguished distribution with respect to the 'PCR_3' feature.



(A16)

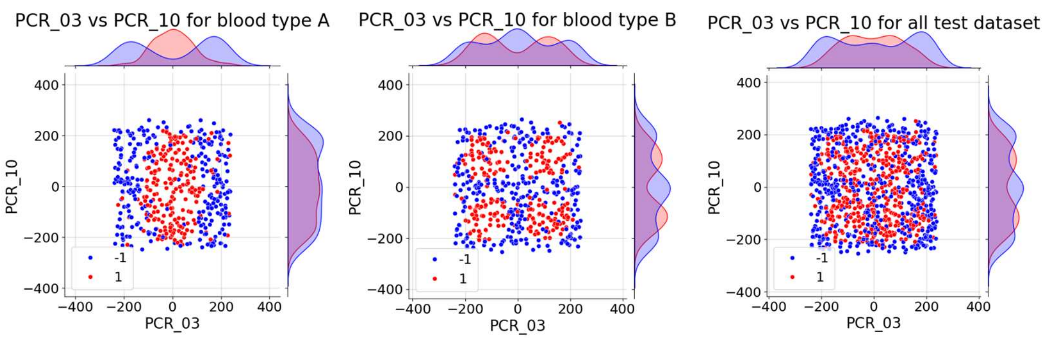
We chose the pair ('PCR_03', 'PCR_10').

Firstly, we noticed that there are 3 candidate pairs for good separation for the positive to special property group.

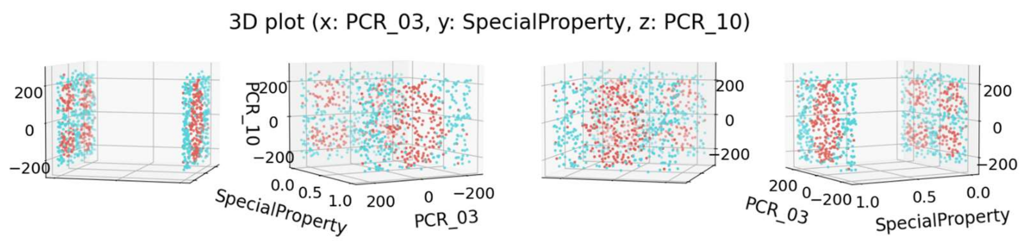
Crossing it with the respective plots of the negative for special property group, it seemed like all of it didn't follow the exact same patterns, but the data was fairly separable for all of it.

Finally, we decided to proceed with that pair, as it seemed like it offers better separation for the negative group than the others.

(A17)



(A18)



(A19)

We don't think a decision tree of max depth 3 would be able to fit the training data well.

According to the plot, such a decision tree would not be able to follow the separability patterns of the data with great resolution, therefore it wouldn't fit the training data well.

If we were to decide manually on how to split the data, we could possibly have a better fit for the positive to SpecialProperty group, but not for the negative to SpecialProperty group.

(A20)

A decision tree of max_depth up to 30 will fit the training data very well (perhaps perfectly). That is since the data is separable with respect to the features we are looking at. Therefore, a decision tree of depth 30 will allow us to take advantage of that separability to classify the training data with great resolution.

(A21)

We don't think a 1-NN model will be able to fit the training data well. Considering the scale of the special property feature (0 or 1) against the scale of the PCR features(range ~-200 to ~200), it is likely that the nearest sample point would be related to the other blood type group.

That will effectively deny the partitioning of the training set according to the special property. Since we had already established that the negative for SpecialProperty and positive for SpecialProperty groups do not follow perfect match in its patterns, related to the target feature, this could possibly lead to lots of misclassifications.

Part 5

(A22)

Q19: Normalization won't impact the performance significantly because the decision tree has a depth of only 3, which is limited for this data. The tree's ability to fit the training data is restricted by its shallow depth rather than the scale of the features.

Q20: Normalization could lead to a more balanced decision tree and potentially improve fitting. By ensuring all features contribute equally, normalization can help the decision tree make more balanced splits, improving the model's overall performance.

Q21: Normalization will significantly improve the performance of the 1-NN model. Before normalization, the varying scales of features can cause issues with distance calculations, leading to incorrect nearest neighbor identification. By bringing all features to the same scale, normalization ensures each feature contributes equally to the distance calculations, leading to more accurate nearest neighbor identification and better fitting of the training data.

Part 6

(A23)

Feature Name	**keep**	**new**	**normalization method**
age	x	x	
sex	x	x	
weight	x	x	
blood_type	x	x	
current_location	x	x	
num_of_siblings	v	x	
happiness_score	x	x	
household_income	x	x	
converstations_per_day	x	x	
sugar_levels	x	x	
sport_activity	x	x	
pcr_date	x	x	
PCR_01	v	x	StandardScaler
PCR_02	v	x	StandardScaler
PCR_03	v	x	MinMaxScaler
PCR_04	v	x	StandardScaler
PCR_05	v	x	MinMaxScaler
PCR_06	v	x	MinMaxScaler
PCR_07	v	x	StandardScaler
PCR_08	v	x	MinMaxScaler
PCR_09	v	x	StandardScaler
PCR_10	v	x	MinMaxScaler
SpecialProperty	v	v	