Gal Kaptsenel, Eitan Gronich 30/11/2022

Major HW1 – Final Report

(Q1)

The dataframe has 1250 rows and 26 columns.

(Q2)

Value counts of *num_of_siblings:*

1	399
2	317
0	271
3	161
4	62
5	31
6	6
7	2
8	1

The meaning of the feature is the number of siblings of the patient (every row in the database represents a patient).

The feature's type is "ordinal", because on the one hand it is categorical (it's discrete, and there is a finite set of reasonable values it can have), but on the other hand it's naturally ordered, and there is a quantitative meaning to a larger or smaller number of siblings.

(Q3)

Feature Name	Description	Type
patient_id	ID of the patient	Other
age	Age of the patient	Ordinal
sex	Sex of the patient (Male/Female)	Categorical
weight	Weight of the patient (probably in KG)	Continuous
blood_type	Blood type of the patient (O/A/B/AB and +/-)	Categorical
current_location	Location of the patient (latitude, longitude)	Other
num_of_siblings	Number of siblings the patient has	Ordinal
happiness_score	A score describing the patient's level of happiness (1-10)	Ordinal
household_income	Income of the household the patient belongs to	Continuous
conversations_per_day	Number of conversations the patient has every day	Ordinal
sugar_levels	The sugar level of the patient (milligrams per deciliter)	Ordinal

sport_activity	Physical Activeness of the patient on scale from 0 to 4	Ordinal
symptoms	Symptoms the patient is suffering from (may be none)	Other
pcr_date	The date the patient had his/her PCR test	Other
PCR_01	Numerical property #1 of the PCR test	Continuous
PCR_02	Numerical property #2 of the PCR test	Continuous
PCR_03	Numerical property #3 of the PCR test	Continuous
PCR_04	Numerical property #4 of the PCR test	Continuous
PCR_05	Numerical property #5 of the PCR test	Continuous
PCR_06	Numerical property #6 of the PCR test	Continuous
PCR_07	Numerical property #7 of the PCR test	Continuous
PCR_08	Numerical property #8 of the PCR test	Continuous
PCR_09	Numerical property #9 of the PCR test	Continuous
PCR_10	Numerical property #10 of the PCR test	Continuous

(Q4)

The calculations, conclusions and learning models we make will be based on the data from the training set. Therefore, if we change the training set in the middle of the analysis, it may result in inconsistency: we may reach different conclusions about the best features to work with, the correct way to normalize them, the best models to use, etc. Also, we want to maintain a strict separation between the train set and the test set, in order to best check our generalization capabilities.

(Q5)

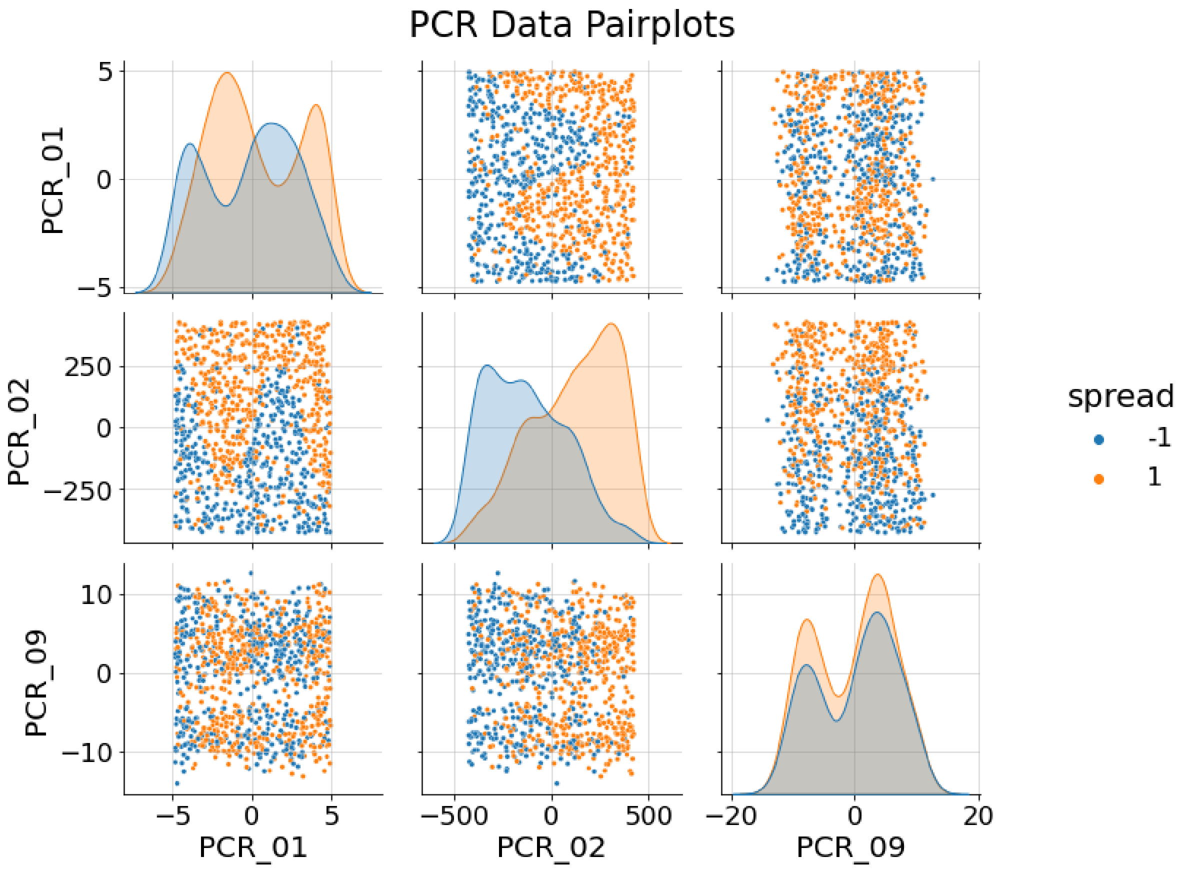
Correlation between PCR_01 and spread: 0.09242800896568669

Correlation between PCR_02 and spread: 0.5076856628032786

Correlation between PCR_09 and spread: -0.035069568953787

(Q6)

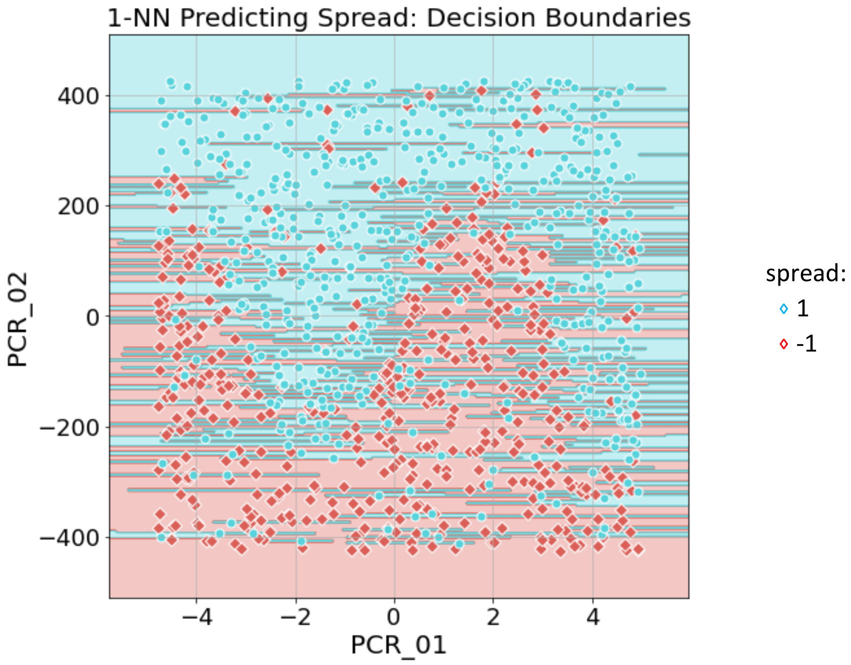
Let us observe pairplots between these variables, categorized according to *spread*:



Clearly, the pair *PCR_01* and *PCR_02* predict *spread* fairly accurately: *spread* is mostly separable on their 2D feature plane, along a wave-like contour line. In contrast, other pairs of features don't seem to separate *spread* very well.

(Q7)

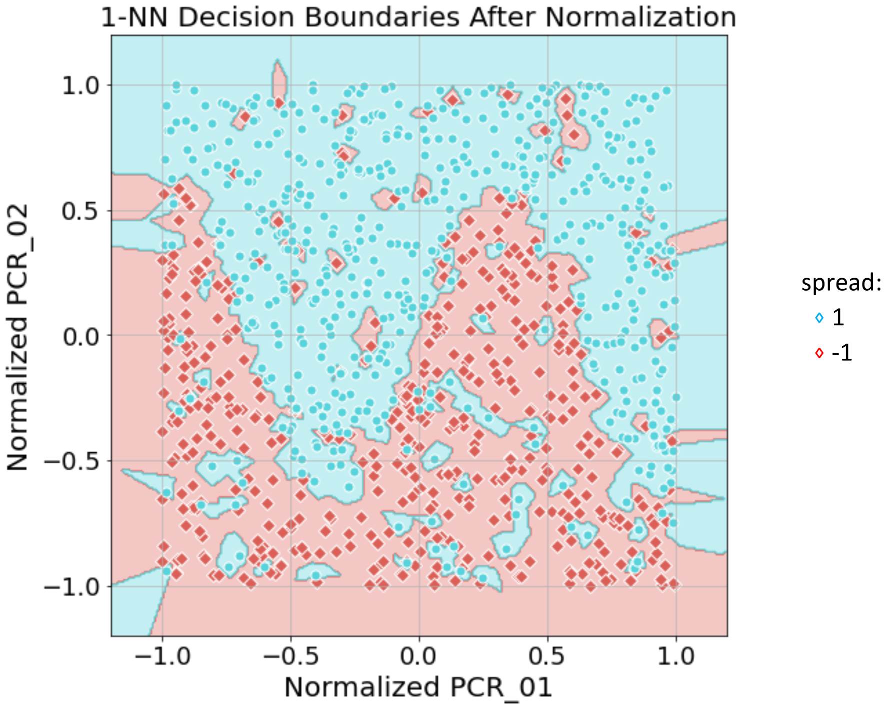
The following figure shows the decision boundaries of a 1NN model fit on the training set:



The training accuracy of the model is 1 (as expected, since the model is 1NN and each point is the closest to itself), and the test accuracy is 0.752.

(Q8)

The following figure shows the decision boundaries for the same model after normalization of both features:



The training accuracy is again 1 as expected, but the test accuracy improved significantly and is now 0.828.

Also, a qualitative difference can be observed between the two figures: while in the first figure we can see narrow horizontal decision boundaries, the boundaries in the second figure are evenly shaped.

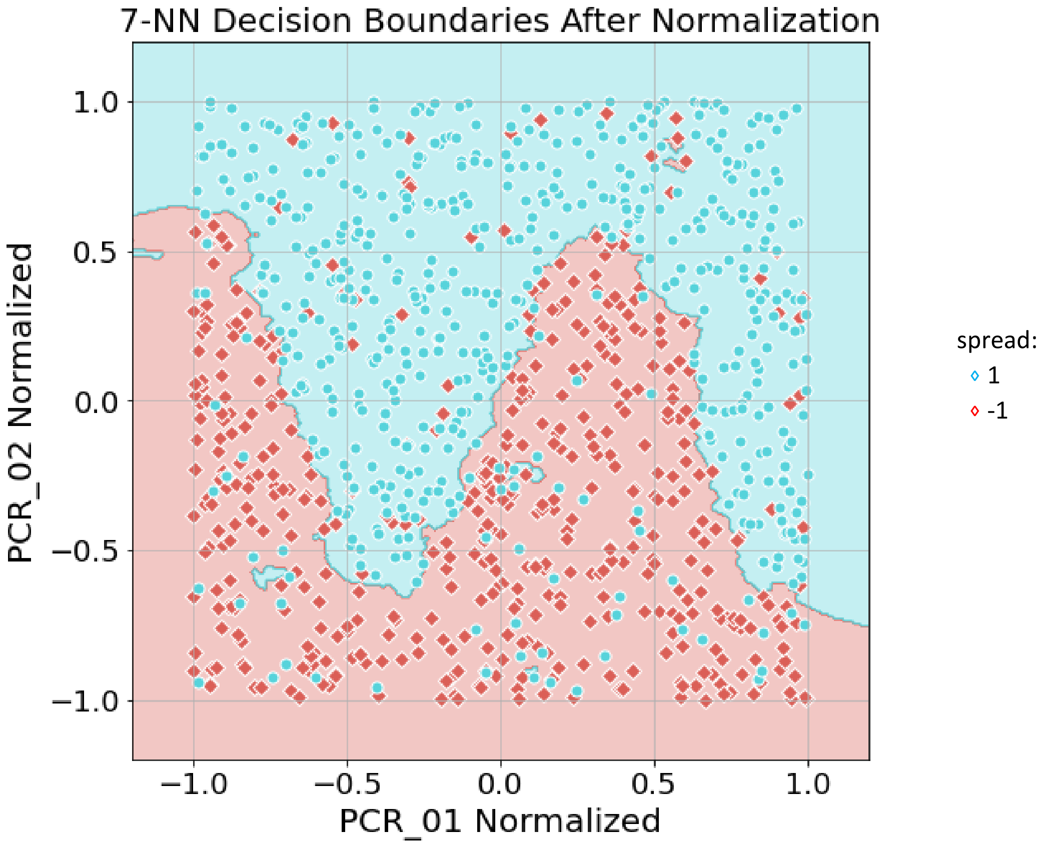
This is due to the **difference in scale** between the two original features: PCR_01 is roughly in $[-4,4]$, while PCR_02 is roughly in $[-400,400]$ – 100 times bigger! If we don't normalize the features, the distances on the PCR_01 coordinate will be much shorter than on the PCR_02 coordinate, and so when considering a point x_0, the points differing from x_0 mostly on the PCR_01 coordinate will be nearer neighbors than those differing on the PCR_02 coordinate, even if the normalized difference (each feature normalized in relation to its own scale) is the same on both axes.

(This is the reason behind the horizontal stripes on the first grid: the PCR_01 coordinate has a stronger "pull" on points nearby, when considering both features on the same visual scale.)

In other words, not normalizing the features gives priority to proximity on some coordinates over others in Nearest Neighbor models (unintentionally and not due to any domain knowledge), and so it is important to normalize them.

(Q9)

The following figure visualizes a KNN model on the same dataset with k=7:



Training accuracy: 0.882

Test accuracy: 0.888

Compared to the previous model, the training accuracy suffered a loss from 1 to 0.882 ($k > 1$ and so when classifying a point in the training set, we take other points into account and therefore may classify wrongly). However, the test accuracy improved significantly from 0.828 to 0.888. In general, the effect of increasing k is that more points are considered and so outliers are neglected (for example, in the previous figure we can see that blue dots in the "red zone" created a small blue decision boundary around them, which mostly didn't happen in the new figure). However, as we discussed in class, it is important to note that increasing k too much would have the result of taking into account points that are too far away (in the extreme – if k is the size of the whole training set then we would classify every points according to the majority of labels in the dataset.)

(Q10)

Assume f is a feature sampled (i.i.d) from $N(0,1)$. Since the maximal and minimal value of f are not bounded, if the number of samples is very large, f could have few samples with large absolute values, while most of the samples are close to 0. Applying a min-max scaling to f to a segment (for example $[-1,1]$) would have the effect of "compressing" f, and since f's minimal and maximal values would be much bigger (in abs. value) than most of f's values, the vast majority of samples would end up concentrated in a small area around the mean. For example: in a $N(0,1)$ distribution, roughly 95% of samples would be in $[-2,2]$. The probability of a sample having an absolute value of over 4 is $6 \cdot 10^{-5}$; if the number of samples is very big, this could occur, and then after applying min-max scaling the aforementioned 95% of samples would only take up at most half of the feature space (assuming $max > 4, min < -4$).

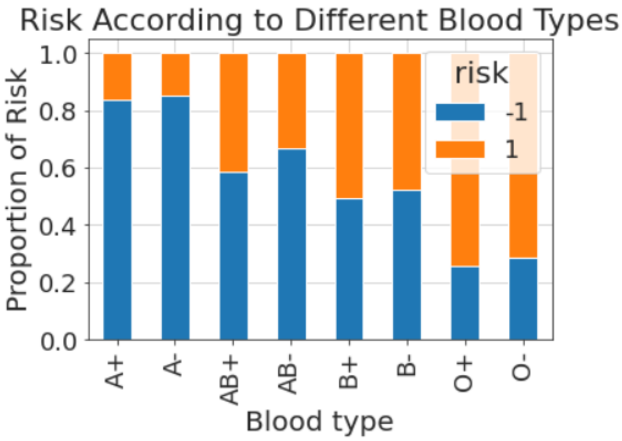
This is detrimental since, in effect, the scale of f would be much decreased, so f would have a much higher effect on proximity than other features, which as we explained in Q8 is harmful to kNN.

(Q11)

There are 8 different types of blood type in the data and so 8 boolean features are needed (with one-hot encoding).

(Q12)

The following is a crosstab of risk according to different blood types:



Based on the plot, we suggest the following three groups: {A+,A-}, {AB+, AB-, B+, B-}, {O+, O-}. The reason is that each group has blood types with similar proportions of *risk*=1, so it is a relatively safe assumption that they have similar effects on *risk* (the underlying assumption is that there are no specific effects of a certain blood type in combination with other specific features, e.g. A+ and A- have similar proportions but A+ only predicts risk for male patients and A- only for female patients)

(Q13)

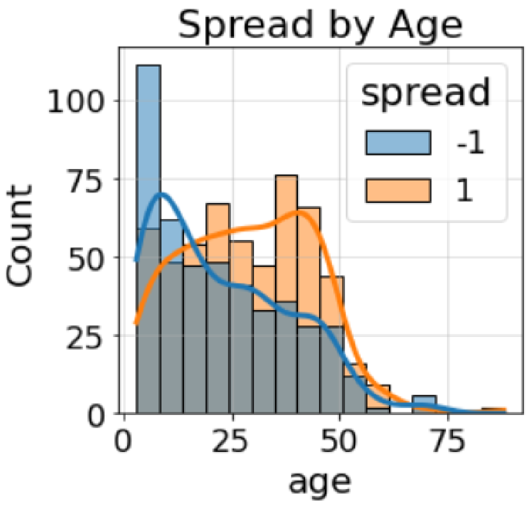
The features we extract from *symptoms* will be boolean features, one for each distinct symptom (the feature equals 1 iff the patient has this symptom). In this way we can take into account any given patient and the symptoms he/she has, and give it a numerical value.

We will explain shortly about the additional features we extracted and turned into numeric features:

Original feature	**Extraction method**
patient_id	Dropped – unmeaningful for prediction
blood_type	Split into 3 groups and turned into OHE, as explained above
symptoms	Turned into Boolean features for each symptom, as explained above
sex	Turned M and F into 1 and -1 (respectively)
current_location	Split the feature into Latitude and Longitude, both continuous variables
pcr_date	Turned the feature into days since 01/01/0001

(Q14)

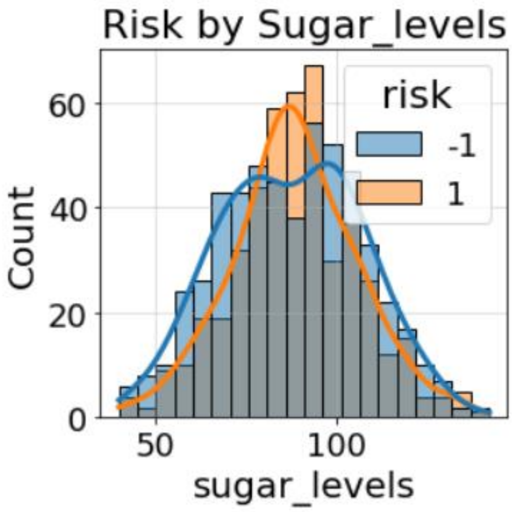
One feature which is informative for predicting *spread* is *age:*



The figure shows that between ages 20-50 the frequency of *spread=1* rises sharply above *spread=-1,* and for ages 0-10, the opposite occurs. Therefore, knowing the age of the patient is informative for predicting their spread.

(Q15)

One feature which could be informative for predicting *risk* is *sugar_levels:*



As the plot shows, a spike of high risk occurs around sugar levels 80-95, so this could be a useful factor in predicting risk for patients. For other sugar levels, it seems that low risk is more probable.

(Q16)

Top 10 most correlated features to *risk* (beside *risk* itself, of course) are:

#	Feature	Correlation
1	blood_type_A	0.496662
2	blood_type_O	0.483319
3	cough	0.088258
4	shortness_of_breath	0.084851
5	PCR_09	0.081810
6	latitude	0.070012
7	household_income	0.056910
8	PCR_06	0.052516
9	PCR_01	0.046551
10	PCR_07	0.038636

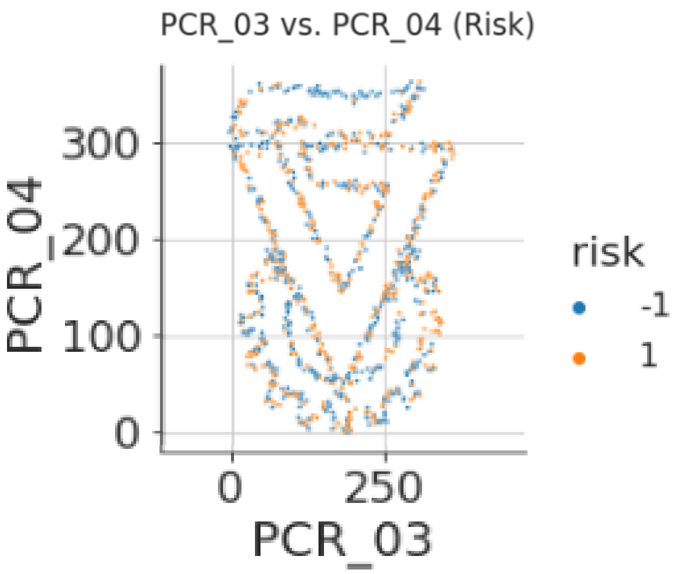
(Q17)

Let us observe two pairs of features from the pairplots, in regard to the *risk* feature.

Pair #1 (PCR_03 X PCR_04)

The features form an interesting structure of the symbol of the Technion Institute.

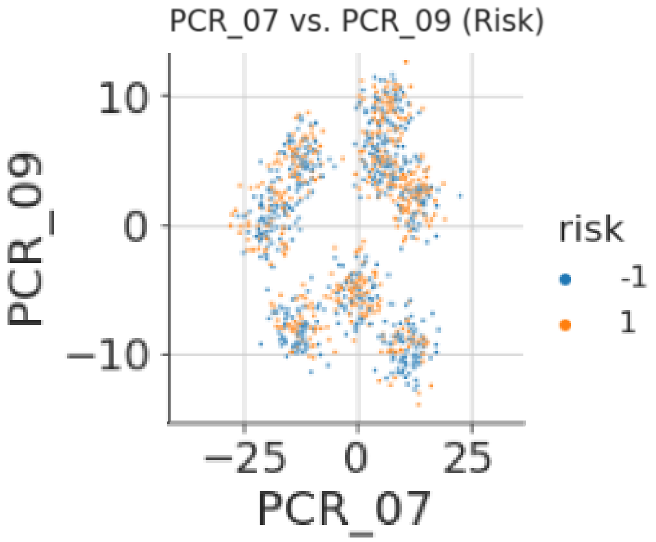
It does not seem that they explain the *risk* feature since different *risk* values are seemingly almost randomly distributed along the symbol.



Pair #2 (PCR_07 X PCR_09)

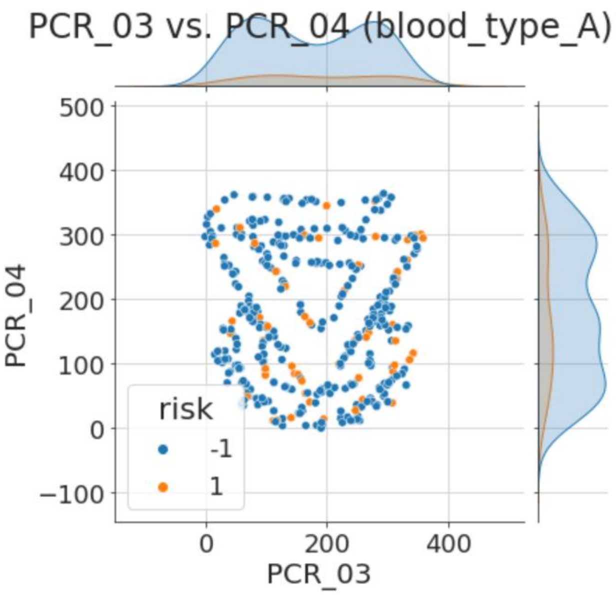
The features form an interesting structure of three separated regions.

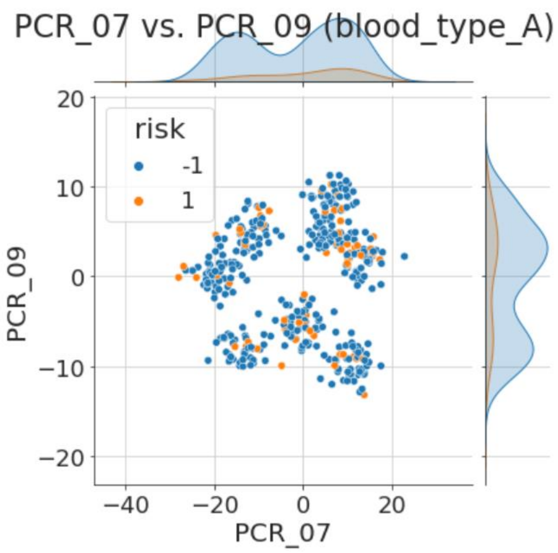
It does not seem that they explain the risk feature by themselves since every region has a high density of both values of *risk*.

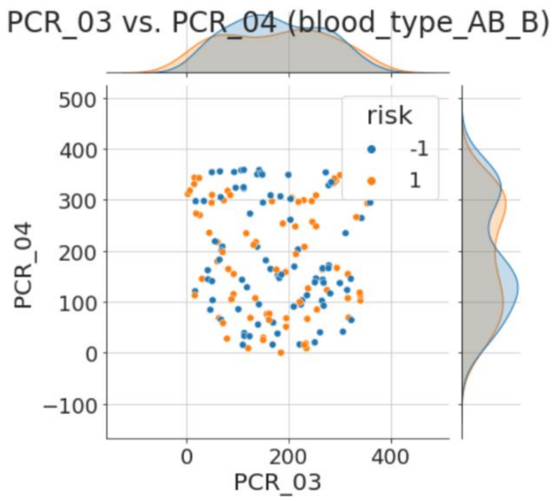


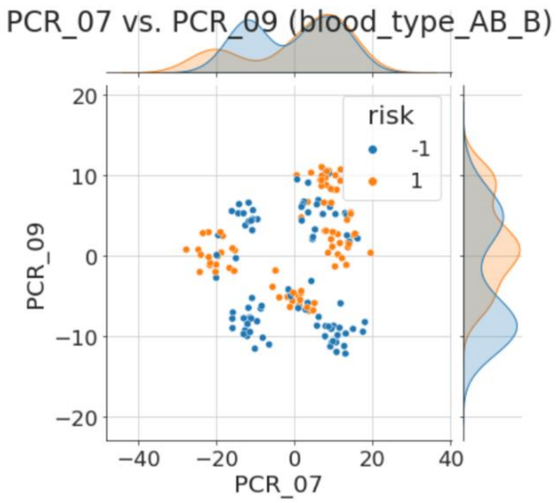
(Q18)

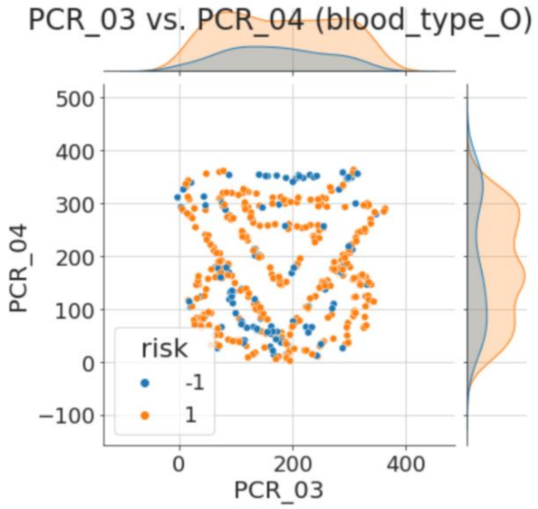
The following plots show the jointplot of the chosen features, split by blood types.

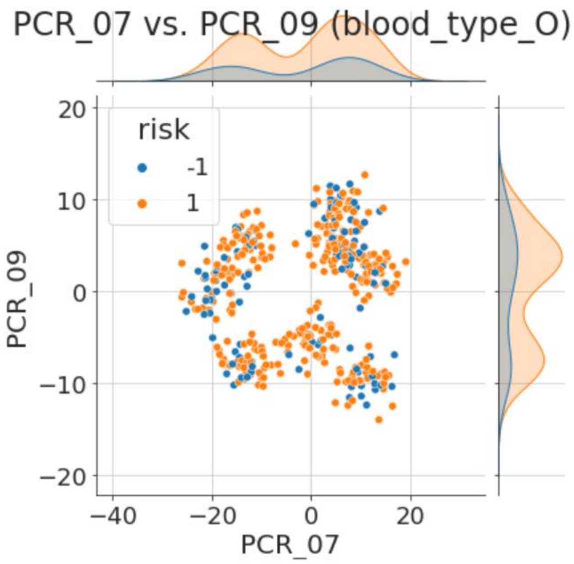












It seems that for blood type AB/B, the pair PCR_07 - PCR_09 are informative for predicting *risk*, since the 2D feature plane can be separated into clusters of fairly homogenous *risk* values.

(Q19)

No.

The correlation between PCR_07 and *risk* is 0.038636 and the correlation between PCR_09 and *risk* is 0.081810 - both are low.

Supposedly this contradicts the fact that the pair is informative for predicting *risk* for certain blood types. However, this is no contradiction: the correlation measures a linear one-dimensional dependency between the two variables (e.g. *risk* and PCR_07). No such significant one-dimensional pattern exists between each feature and *risk*, but when observing the above plot for blood type AB/B, two factors come into play: 1. We observe only blood type AB/B (so the other blood types, which may ruin the correlation, don't appear on the plot) and more importantly, 2. We allow two-dimensional dependencies of the risk on the features. This is why the pattern emerges even though the correlation of every variable by itself with *risk* is low.

(Q20)

Decision trees would be suitable.

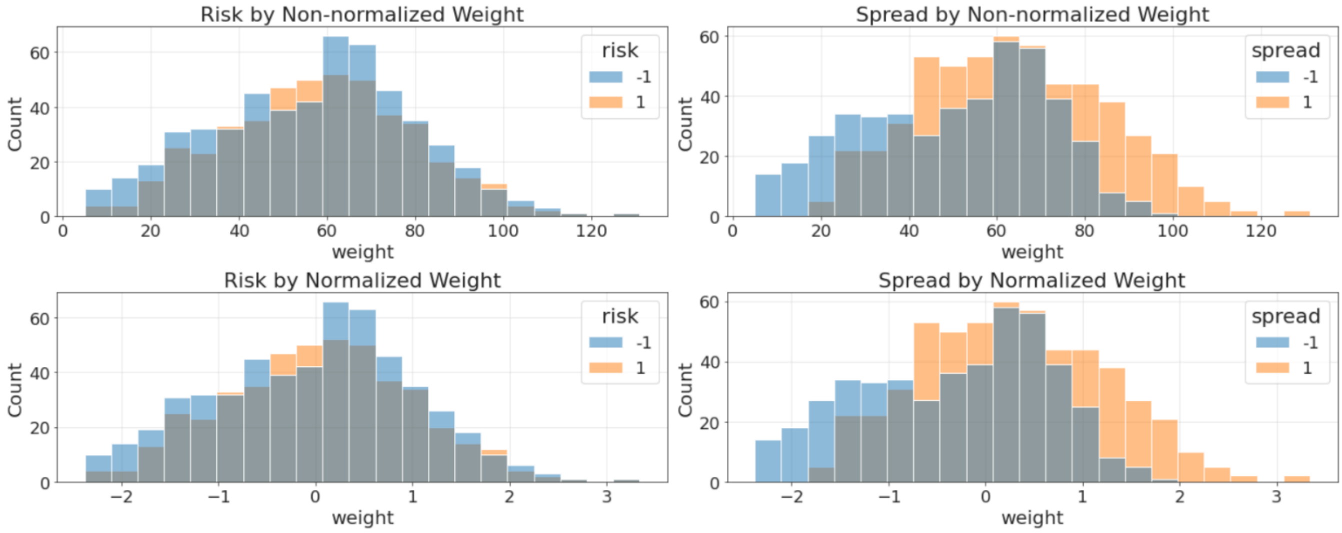
As can be seen from (Q12), and from the plots in (Q18) as well, there is a high correlation between blood type groups O, A and *risk*: blood type A have a high frequency (~80%) of *risk=-1* and blood type O have a low frequency (~20%) of *risk=-1*.

For the third blood type group, AB_B, there is an effective way to separate *risk* with PCR_07 and PCR_09, using the informative plot that was discussed in (Q18) and (Q19).

Therefore, we can create the following decision tree: first split according to the blood type group. If it is O, the risk is predicted to be 1, if it is A, the risk is predicted to be –1, and otherwise the decision tree will split according to PCR_07 and PCR_09 features, to (~8) separate regions of high and low risk according to the plot from (Q19), and then predict according to the majority in the split region.

(Q21)

Univariate histogram of the weight feature before and after normalization,



(Q22)

a. When performing forward feature selection, on iteration number $i\ = 0,1,2,..$, the number of features compared are $d_1 - i$, and therefore $d_1 - i$ models are trained.
The number of iterations will be the number of features requested to be selected i.e. d_2.
Therefore, the number of models considered in forward feature selection are

$$d_1 + (d_1 - 1) + \cdots + (d_1 - d_2 + 1) = \frac{d_2 * (d_1 + d_1 - d_2 + 1)}{2} \le 2d_1 d_2 - d_2^2 + d_2$$

That is $O(d_1 \cdot d_2)$ (since $d_2 < d_1$).

b. When performing backward feature selection, on iteration number $i\ = 0,1,2,..$, again the number of features compared and models trained is $d_1 - i$ (one model for every feature we are considering dropping).
The number of iterations will be the number of features requested to be dropped from the complete set of features, i.e. $d_1 - d_2$.
Therefore, the number of models considered in backward feature selection are

$$\begin{aligned} d_1 + (d_1 - 1) + \cdots + (d_1 - (d_1 - d_2) + 1) &= d_1 + (d_1 - 1) + \cdots + (d_2 + 1) \\ &= \frac{(d_1 - d_2) * (d_1 + d_2 + 1)}{2} \le (d_1 - d_2) * (d_1 + d_2 + 1) \\ &\le d_1(d_1 + d_2 + 1) = d_1^2 + d_1 d_2 + d_1 \le 2d_1^2 + d_1 \end{aligned}$$

That is $O(d_1^2)$.

(Q23)

The three features that were returned are:

1. weight
2. PCR_01
3. PCR_02

Two of these features were included in the chosen features from (Q6) - PCR_01 and PCR_02. However, *weight* is not the feature we chose in (Q14), which was *age*. The correlation between *weight* and *age* is 0.664 which is a moderate positive correlation.

(Q24)

It is important to perform the normalization step before performing sequential feature selection because the selection algorithm uses a greedy approach which maximizes, in each iteration, the performance of the model by selecting the most contributing feature (in forward selection; in backward selection the least helping feature is dropped).

We saw that in some models (for example in kNN), a feature with a large/small scale has a dominating effect on the classifier. If the normalization step is skipped, the effect of these features could be amplified, and the decision of the sequential selector could be skewed – either disproportionately preferring these features, or discarding them.

If normalization is done, then all the features are in the same order of magnitude, and therefore there will be no bias, and selected features will be meaningful for the classification.

(Q25)

Yes.

In each iteration of feature selection, different models may yield different results with the same subset of features, because some types of dependencies are more compatible with some algorithms than others. This means the selector could choose different features in each iteration, and end up with different final subsets.

For example, kNN may prefer features in which same-class samples are close to one another, even though they may not be linearly separable. However, if an SVM algorithm is used on the same feature, it will yield bad results, and the feature will not be chosen. Similarly, there may be a feature that a kernel SVM will have remarkable results with, while kNN fails.

(Q26)

Feature Name	Keep	New	Normalization Method	Explanation
patient_id	X	X	-	Patient id has no quantitative meaning, therefore there is no reason to make decisions according to its value.
age	V	X	Min/Max Scaling	The age of the patient may affect the spread and risk of the patient. The age has well defined bounds (for example 0-120), and a moderately balanced distribution (no crowding around the mean), so we chose min/max scaling.
sex	X	X	-	Different sexes may react in different ways to the virus and therefore it is better to take sex into consideration. However, since the original feature is a string, we replaced it with a numerical feature (M=1, F=-1).
sex (after change)	V	V	Already Normalized	Numerical sex feature, as explained above.
weight	V	X	Standardization	The weight of the patient **may** have a relation to a patient's health and reflect on the target labels. According to the univariate plot, weight is distributed approximately according to a normal distribution, and therefore Standardization is a recommended normalization method.
blood_type	X	X	-	Blood type is a categorical feature, and according to Q12, it may be a good approach to group together similar blood categories, because of similarities in reaction to the virus, and therefore we created three blood groups features (OHE) that replace the blood type feature.
blood_type_a	V	V	Already Normalized (Binary)	New feature which groups A- and A+ blood types, as explained above.
blood_type_ab_b	V	V	Already Normalized (Binary)	New feature which groups AB-, AB+, B+ and B- blood types, as explained above.
blood_type_o	V	V	Already Normalized (Binary)	New feature which groups O- and O+ blood types, as explained above.
current_location	X	X	-	In different locations (countries, cities, neighborhoods, etc.) the spread and risk may differ.

				We decided to split the location to coordinates (and therefore replace this feature), because the original feature is not a numerical value.
Latitude	V	V	Min/Max Scaling	Latitude coordinate, as explained above. Min/max scaling was chosen since the feature is bounded and there is no significant difference in orders of magnitude between the mean and the extremum values.
Longitude	V	V	Min/Max Scaling	Longitude coordinate, as explained above. Min/max scaling was chosen since the feature is bounded and there is no significant difference in orders of magnitude between the mean and the extremum values.
num_of_siblings	V	X	Min/Max Scaling	Number of siblings may be correlated to the environment the patient is found in, and therefore affect the risk and spread. The feature is bounded (for example, between 0 and 20), and not normally distributed, so we used min/max scaling.
happiness_score	V	X	Min/Max Scaling	Happiness of the patient may reflect on the mental health of the patient, which may have effect on the risk and of the patient and the environment the patient is found in and therefore affect the spread. The feature is bounded (between 1 and 10), and not normally distributed, so we used min/max scaling.
household_income	V	X	Standardization	House income may affect the environment the patient is found in, and therefore affect the risk and the spread. According to the univariate plot, household income is distributed approximately according to a normal distribution, and therefore Standardization is a recommended normalization method.
conversations_per_day	V	X	Standardization	Conversations per day may be a good approximation on the number of people a patient met every day, which may have effect on the spread. According to the univariate plot, the number of conversations per day is distributed approximately according to a normal distribution, and therefore Standardization is a recommended normalization method.
sugar_levels	V	X	Standardization	Sugar levels have a relation with the health of the patient, which may affect the risk of the patient. According to the univariate plot, number of conversations per day is distributed approximately according to a normal

				distribution, and therefore Standardization is a recommended normalization method.
Sport_activity	V	X	Min/Max Scaling	Sport activity has a relation with the health of the patient which may have an effect on the risk of the patient. The values of this feature are in a well-defined small range (1 to 5), and therefore we used min/max normalization.
Symptoms	X	X	-	*Symptoms* is not a numerical field (list of symptoms for each patient), and therefore we decided to replace it with new boolean features for every symptom which indicate if a patient had the symptom. Symptoms have a direct relation to a risk of the patient, and also may affect the spread (e.g. coughing).
Low_appetite	V	V	Already Normalized (Binary)	Single symptom feature, as explained above.
Sore_throat	V	V	Already Normalized (Binary)	Single symptom feature, as explained above.
Cough	V	V	Already Normalized (Binary)	Single symptom feature, as explained above.
Shortness_of_breath	V	V	Already Normalized (Binary)	Single symptom feature, as explained above.
Fever	V	V	Already Normalized (Binary)	Single symptom feature, as explained above.
pcr_date	X	X	-	PCR date may reflect periods of time which had more spread or risk, for example more contagious/deadly variants. However the date format is not numerical, so we changed this feature to count the number of days from the epoch (1.1.0001) to the PCR date.
pcr_date (after change)	V	V	Min/Max Scaling	Days from the epoch to the PCR test. All the dates people took PCR tests are bound from ~2020 till today (~2022) and don't have any extreme values, and therefore min/max normalization is a good approach.
PCR_01	V	X	Min/Max Scaling	All the PCR_X features are numerical properties of the PCR test, which may reveal information about the virus and therefore about the target labels. According to the univariate plot, it is bound to a range of -5 to 5 and spread fairly uniformly, and therefore min/max normalization is a good approach.

PCR_02	V	X	Min/Max Scaling	According to the univariate plot, it is bound to a range of -300 to 300 and spread fairly uniformly, and therefore min/max normalization is a good approach.
PCR_03	V	X	Min/Max Scaling	According to the univariate plot, it is bound to a range of 0 to 400 and spread fairly uniformly, and therefore min/max normalization is a good approach.
PCR_04	V	X	Min/Max Scaling	According to the univariate plot, it is bound to a range of 0 to 400 and spread fairly uniformly, and therefore min/max normalization is a good approach.
PCR_05	V	X	Min/Max Scaling	According to the univariate plot, it is bound to a range of 0 to 10 and spread fairly uniformly, and therefore min/max normalization is a good approach.
PCR_06	V	X	Standardization	According to the univariate plot, it is approximately distributed according to a normal distribution and therefore Standardization is a good approach.
PCR_07	V	X	Min/Max Scaling	According to the univariate plot, it is bound to a range of -20 to -20, and although it is not spread uniformly, there are two “centers” roughly in the middle between the mean and the edge of the range, which means min-max scaling will not result in “squeezing” the values.
PCR_08	V	X	Standardization	According to the univariate plot, it is roughly distributed according to a normal distribution and therefore Standardization is a good approach.
PCR_09	V	X	Min/Max Scaling	According to the univariate plot, it is bound to a range of -15 to -15, and although it is not spread uniformly, there are two “centers” roughly in the middle between the mean and the edge of the range, which means min-max scaling will not result in “squeezing” the values.
PCR_10	V	X	Standardization	According to the univariate plot, it is roughly distributed according to a normal distribution (albeit with a large variance) and therefore Standardization is a good approach.