236766 WET HW1 - REPORT - 208081588, 209462415

(Q1)
Load the dataset into a Pandas DataFrame. How many rows and columns are in the dataset?

A:
25 columns, 1250 rows.

(Q2)
Print the value_counts of the conversations_per_day feature (see Tutorial 01).
Copy the obtained output to your report.
Describe in one short sentence what you
think this feature refers to in the real world.
This feature's type is "ordinal". Explain briefly why.

A:

conversations_per_day	count
2	220
4	207
3	201
5	153
6	125
1	115
7	77
8	52
10	33
9	23
11	14
12	10
13	7
16	4
14	4
17	3
21	2

This feature probably refers to the number of conversations each patient makes daily (affects chances of spreading the virus).

This feature is ordinal because it is a counting of occurrences, which is naturally ordinal.

(Q3)
In your report, write a table describing each feature.
The columns must be:
a. Feature name: the name of the feature as it is written in the dataset.
b. Description: a short sentence with your understanding of the feature's meaning in the real world.
c. Type: Continuous, Categorical, Ordinal, or Other.

A:

Feature name	Description	Type
patient_id	Unique id for each patient	Ordinal
age	The age of the patient	Ordinal
sex	The sex of the patient	Categorical
weight	The weight of the patient	Continuous
blood_type	The blood type of the patient	Categorical
current_location	The current location of the patient, given be an index from a list of locations	Categorical
num_of_siblings	The number of siblings the patient has	Ordinal
happiness_score	The patient's happiness score on some scale	Ordinal
household_income	The patient's household income	Continuous
conversations_per_day	The number of conversations the patient makes daily	Ordinal
sugar_levels	The patient's sugar levels	Ordinal
sport_activity	The patient's sport activity level on some scale	Ordinal
pcr_date	The date when the patient was tested for PCR	Ordinal
PCR_01 - PCR_10	10 metrics derived from the PCR test results	Continuous (all of them)

(Q4)
Split the data randomly into a training set (80% of the data) and a test set (20% of the data). As the random_state, use the sum of the last two digits of each of your IDs (two or three IDs). The random state will ensure that you get the same split every time.
Answer: Why is it important that we use the exact same split for all our analyses?

A:
The train set should be consistent throughout the analyses because we are trying to derive info from our training set that can vary if the set changes, giving inconsistent results which will change the chosen model for out problem.

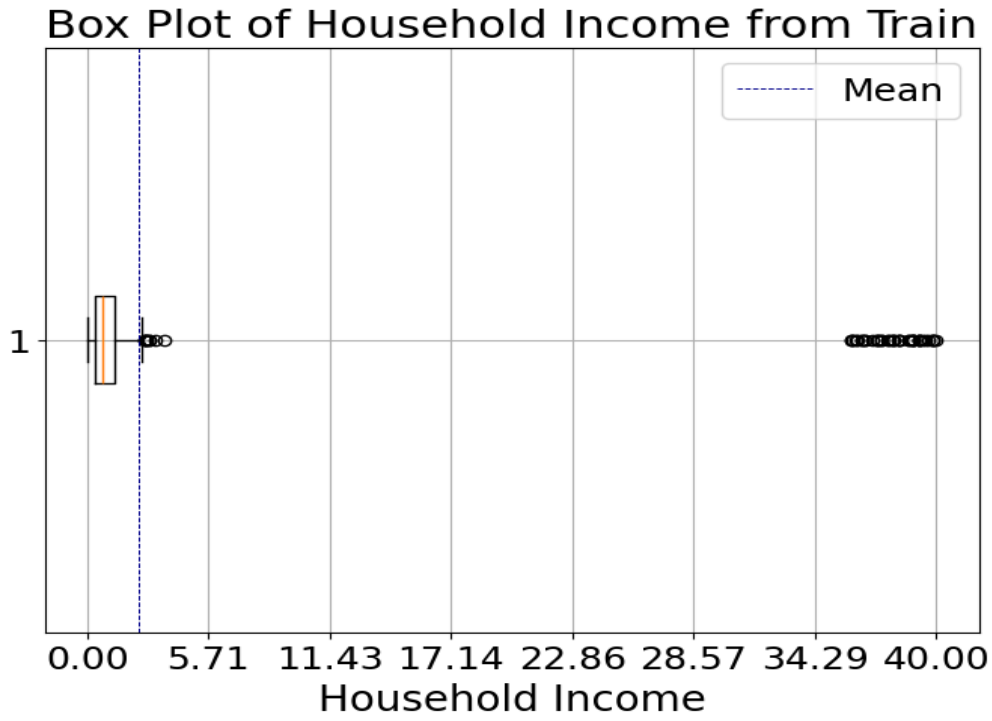
(Q5)
For both the training set and test set, report which fields have missing values and how many missing values there are. You can use Panda's function isnull().

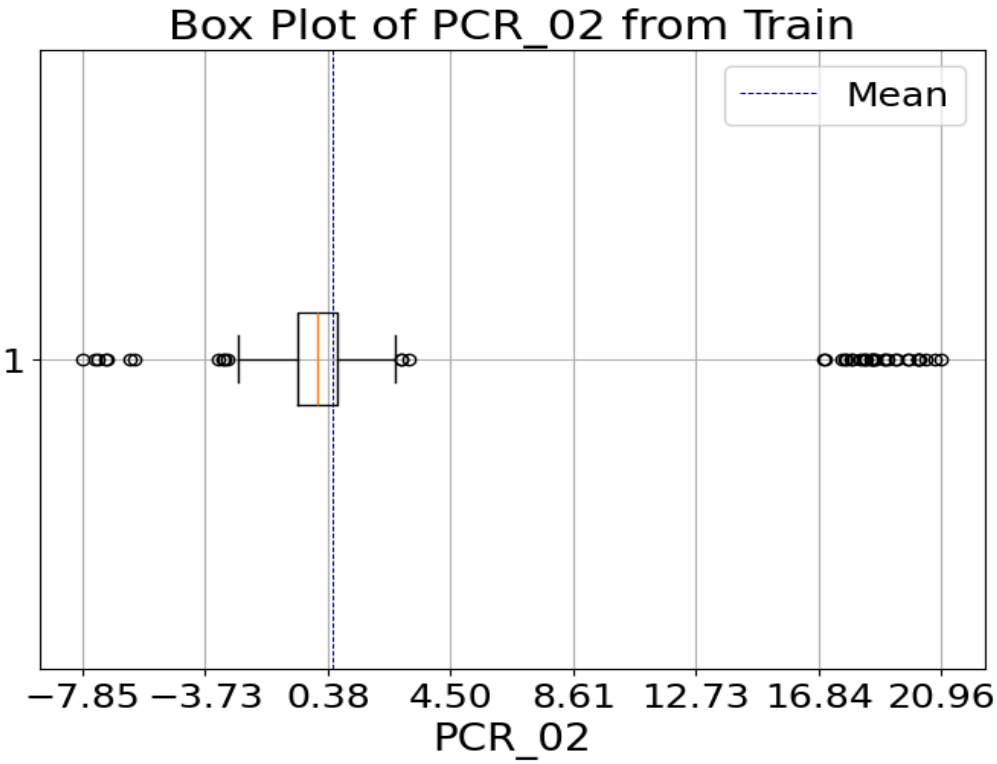
A:
In our train set there are 111 missing values in the field “household_income” (889/1000 non-null), and 63 missing values in the field “PCR_02” (937/1000 non-null).
In our test set there are 28 missing values in the field “household_income” (222/250 non-null), and 11 missing values in the field “PCR_02” (239/250 non-null).

(Q6)
Before handling missing values, it is crucial to identify outliers. For each field that contains missing values, use the provided code to create a box plot. Attach these plots to your report.
Read this explanation and answer: What are outliers? How does the box plot identify outliers? Are there any outliers in the fields you just plotted?

A:
Outliers are “extreme” values in the dataset. Namely, values that are smaller than the lower bound which is calculated using the formula: $Q1 - 1.5 \cdot IQR$, or are greater than the upper bound which is calculated using the formula: $Q3 + 1.5 \cdot IQR$.
(IQR is the interquartile range given by $Q3 - Q1$).
The box plot shows this using the whiskers extending from the box, which represent these bounds. The outliers are then marked as individual points.
As we can see, there are outliers in both fields.

Note: Matplotlib defines the whiskers bounds at the farthest datapoints in the dataset that falls inside the above mentioned bounds.


Box Plot of Household Income from Train
Mean
1
0.00
5.71
11.43
17.14
22.86
28.57
34.29
40.00
Household Income


Box Plot of PCR_02 from Train
Mean
1
−7.85
−3.73
0.38
4.50
8.61
12.73
16.84
20.96
PCR_02

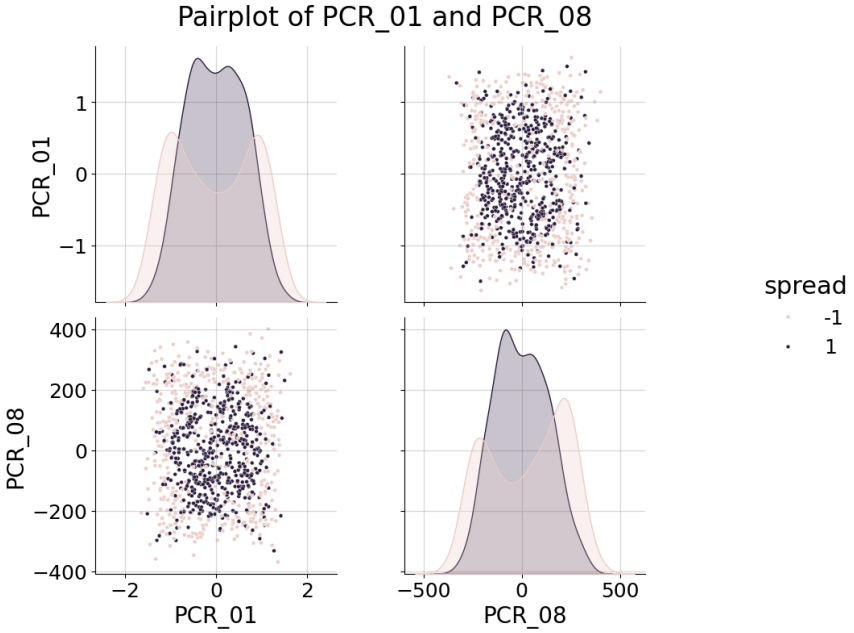
(Q7)
For each field where you found missing values, calculate the median and the mean in the training set, and report it.
If there is a significant difference between the mean and median values, explain the reason. Which filling method do you prefer to use in our case, and why?

A:
The mean in the household_income field is 2.45.
The median in the household_income field is 0.7.
The mean in the PCR_02 field is 0.557.
The median in the PCR_02 field is 0.035.

There is a significant difference between the mean and median in both fields, that is because we have many outliers in the dataset that are substantially larger than the median and not so many outliers that are smaller than the median.
We would prefer to use the median, because it better represents the dataset, as can be seen in our plots from the previous question.

(Q8)
Answer briefly: Based on the plots you created on (Task B), what pair of features is useful for predicting the spread?
Attach the seaborn.pairplot of only this pair of features to the report.

A:
The pair PCR_01 X PCR_08 is useful for precidting the spread, because it roughly separates the points with spread=1 and spread=-1. Using this, we can speculate that points that are closer to (0,0) have a higher chance for spread=1. Moreover, points with similar values tend to group.



(Q9)
Calculate and report the correlation between the features identified in (Q8) and the target variable spread. You should find a very weak correlation (close to zero). Does this result contradict your findings from (Q8)? Explain.

A:
Correlation table:

	PCR_01	**PCR_08**	**spread**
PCR_01	1	0.013074	0.027615
PCR_08	0.013074	1	-0.085300
spread	0.027615	-0.085300	1

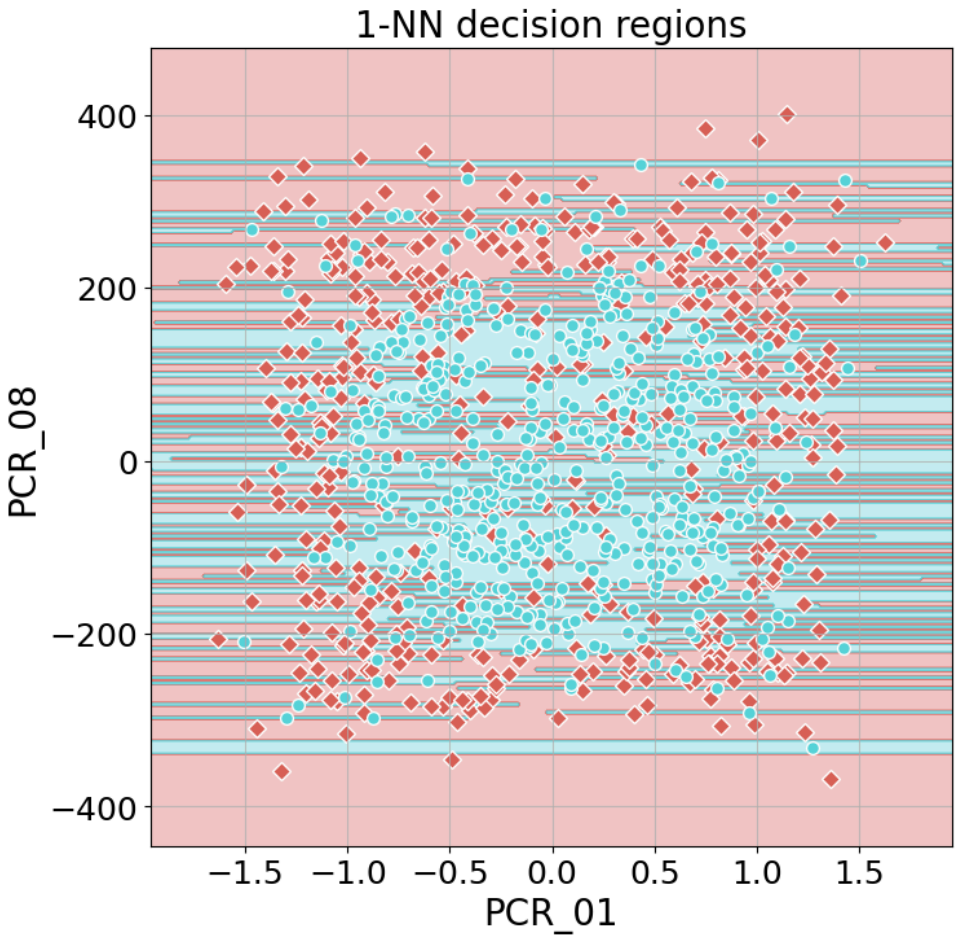
We can see that the correlation between each PCR feature and “spread” is nearly 0.
This does not contradict our findings from Q8, because we looked at the relationship between the pair (PCR-01, PCR-08) and “spread”, and not each feature separately.

(Q10)
What is the time complexity of the prediction function you wrote, applied on a single test datapoint, in terms of the number of neighbors k, the number of training datapoints m and the data dimension d? Explain. It is okay to “estimate” the complexity of python library functions. For instance, if you use np.argsort on n elements, then its complexity should be $O(nlogn)$. Use your reason and CS knowledge.

A:
For cdist(1 X d , d X m), we need to compute m times the euclidean distance between two d-dimentsional datapoints. We then estimate the complexity for this is $O(md)$.
For argpartition(1 X m, k) we know from previous courses that finding the k’th element and partitioning are both $O(m)$ and therefore this is also $O(m)$.
Then we need to collect the labels of the k-indices we found which is $O(k)$ for k array accesses.
Finally we calculate the sum of these k elements which takes $O(k)$ as well.
Overall, the time complexity of “predict” is $O(md + m + k + k) = O(md)$, (since $k \leq m$).

(Q11)
Attach the figure to your report. Specify the model’s training and test accuracies. (The plot should exhibit a bizarre behavior which we will discuss next.)

A:
The 1-NN model's train accuracy score is: 1.0.
The 1-NN model's test accuracy score is: 0.612.



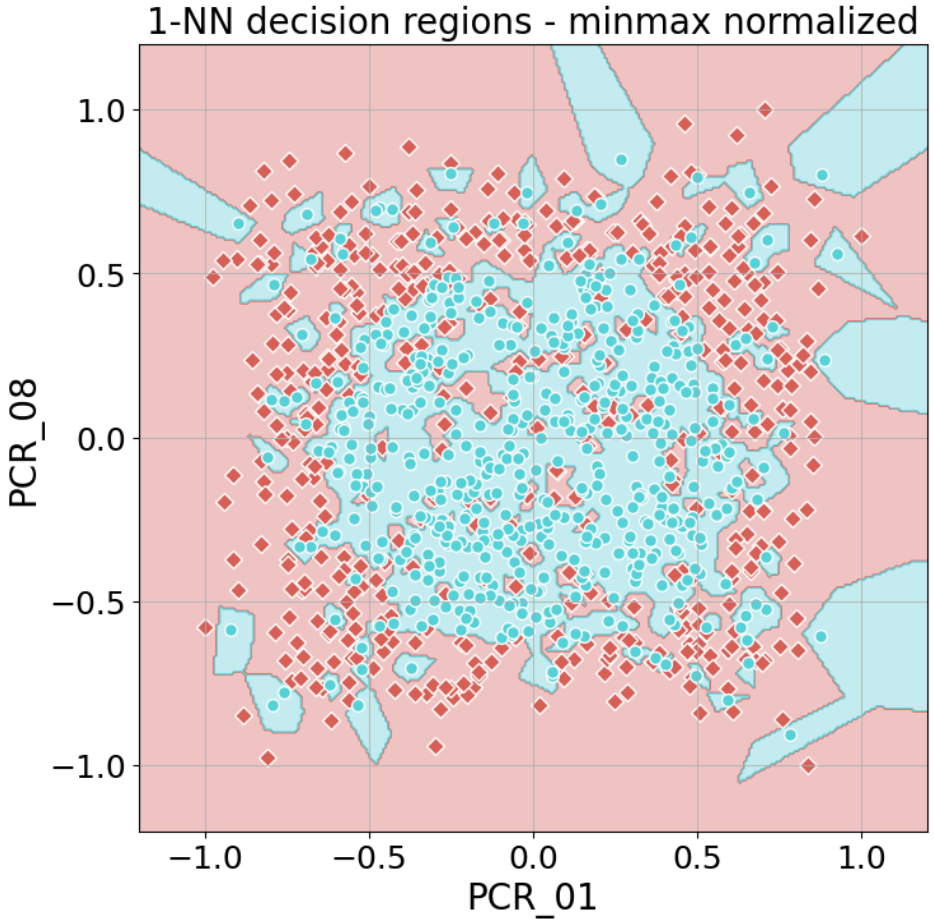
(Q12)
Use min-max scaling (between [−1,1]) to normalize the two features in the temporary DataFrame you created before, and train a new kNN model (k = 1) on the normalized dataset.
Compute the new training and test accuracies and draw the decision regions of the model. Attach the results to your report and compare them to those from (Q11) for the same k = 1 model on the raw data. Use these results to explain why normalization is important for nearest neighbor models.

A:
The min-max normalized model's train accuracy score is: 1.0
The min-max normalized model's test accuracy score is: 0.656

As we can see in the results of Q11, the division areas we get are dominated by the value of PCR_08 because its values are on a much larger scale than PCR_01 ((-400,400) compared to (-2,2)) and because we used the euclidean norm to calculate the distances between neighbors. This means, that the we give more weight to differences in PCR_08 when we calculate the

nearest neighbor of some point in the test dataset - therefore our nearest neighbors algorithm is not ideal.
This is shown in the results of the normalized model - some areas changed colors because they were previously wrongly perceived as “closer” or “further” from other points in the dataset.

This reflects in the test accuracy results which are now better.

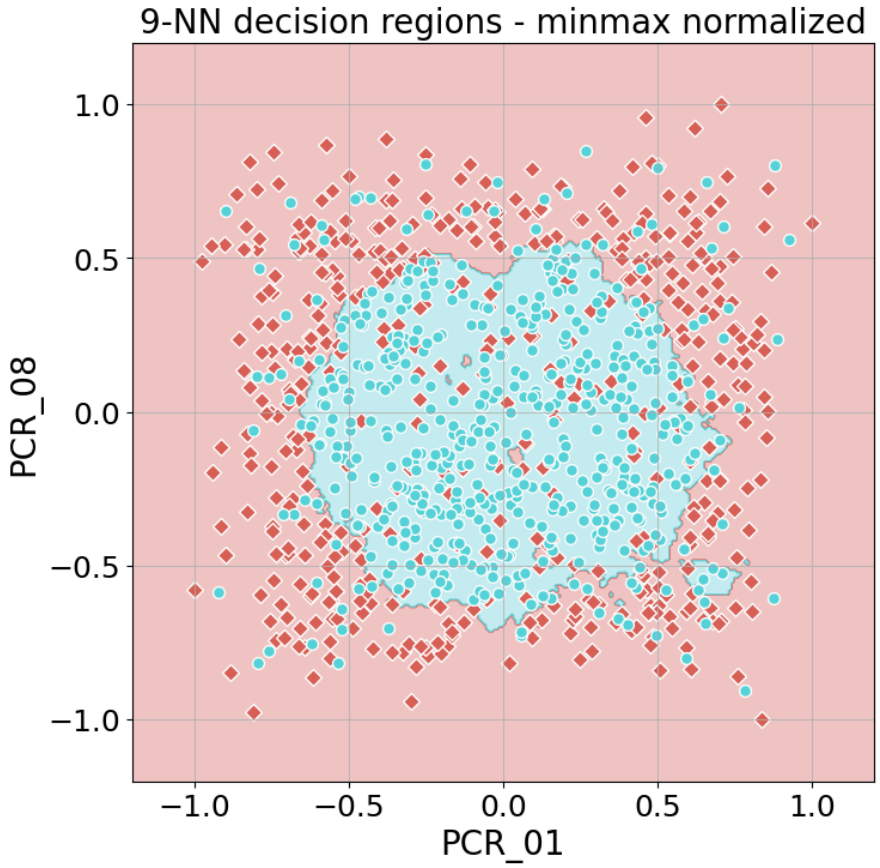


(Q13)
Using the normalized dataset, train another kNN model with k = 9. Compute the training and test accuracy and draw the decision regions of this model.
Attach the results to your report and compare them to those from (Q12).
Use these results to briefly explain the effect of k on the decision regions.

A:
The min-max normalized 9-NN model's train accuracy score is: 0.812
The min-max normalized 9-NN model's test accuracy score is: 0.716

As we can see in our new results, using 9 neighbors instead of one resulted in better test accuracy - this is because areas with a majority of similar labels are now not affected by a single

different label, as is the case with 1-NN where each datapoint has some neighborhood where the model would predict similar labels. Namely, it is less affected by “noise”.



(Q14)

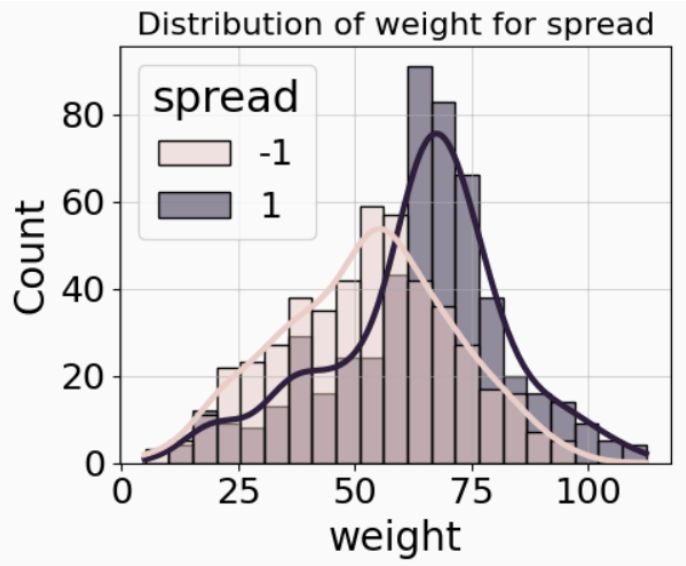
This question is general and does not deal with the given dataset. Assume a dataset with two features, one randomly sampled (i.i.d.) from a uniform continuous distribution on the range [2,5] and the other randomly sampled (i.i.d.) from a chi-squared distribution $\chi^2(k = 2)$ (see in Wikipedia).
(The labels are determined by some unknown function of these two features.)
Why is normalizing both features using min-max scaling to [−1,1] a bad idea?
Explain in detail.

A:

This is a bad idea because the two distributions are very different - one is uniform and the other is skewed and tends towards it’s minimal value. Scaling both on the same scale will changed the relationship between them, because the uniform one will not change dramatically while the chi-squared will be squeezed into a much smaller interval with the majority of values being very close to -1 (because it can get values on a large interval). This also means outliers have a lot of influence over the outcome of the normalization of chi-squared - while larger values have a very low probability of appearing, they can cause the dataset to compress in a way that misrepresents the original dataset.

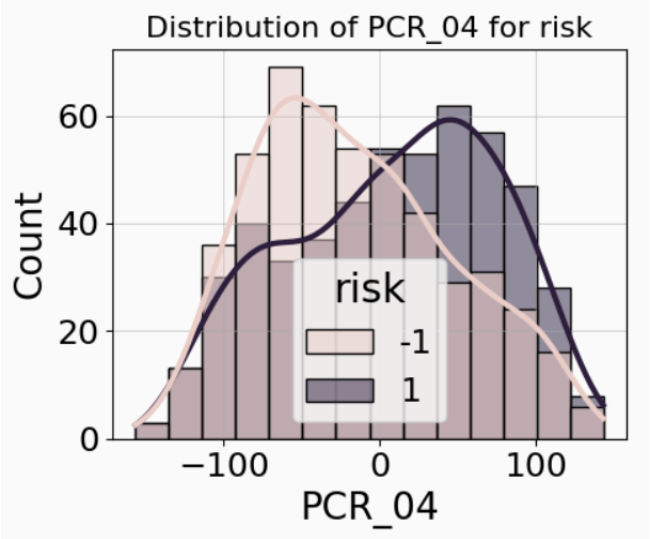
(Q15)
According to the univariate analysis, name one feature that seems informative for predicting the spread target variable (other than the 2 features from Q8).
Attach the appropriate univariate plot and briefly explain (2-3 sentences) why this plot makes you think that feature is informative.

A:
We believe the “weight” feature is informative for predicting “spread”.
That is because the histogram shows that when weight is over 60 we have about double the counts of spread=1 compared to spread=-1 for each weight group. On the other hand when weight is less that 60 we have about half the counts of spread=1 compared to spread=-1 for each weight group. This roughly partitions the dataset to two weight group with one having a higher change for spread=1 and the other having a higher chance for spread=-1.



(Q16)
According to the univariate analysis, name one feature that seems informative for predicting the risk target variable (other than the blood groups).
Attach the appropriate univariate plot and briefly explain (2-3 sentences) why this plot makes you think that feature is informative.

A:
We believe the feature “PCR_04” is informative for predicting “risk”.
That is because the histograms shows an interval in which the counts of risk=1 are higher and another interval in which the counts of risk=-1 are higher. Moreover, for the PCR values with the most counts (around -60 and around 50) we can see a great difference between the counts of risk=1 and risk=-1.
The plot is attached below:



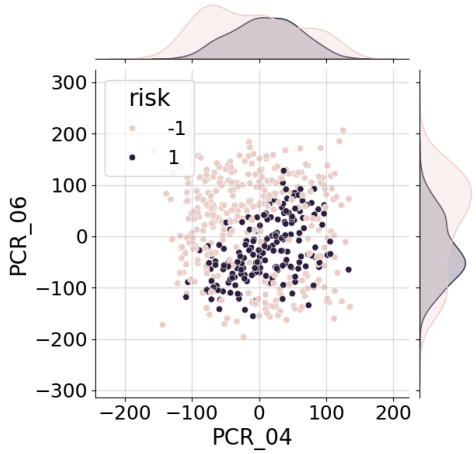
(Q17)
Split the (training) data based on the binary SpecialProperty feature created in (Task E). For each split, perform a bivariate analysis for the PCR features in the set, in relation to the risk.
According to those plots, choose a pair of PCR features that could be helpful for predicting the risk (with the partition according to the SpecialProperty).
What PCR features did you choose? And why?

A:
We believe that the pair PCR_04 X PCR_06 could be helpful for predicting the risk, because they roughtly separate the dataset to two areas - one where the majority of samples have risk=1 and the other where the majority of sample have risk = -1. And furthermore, it does so for both groups (SpecialProperty = True and SpecialProperty = False), and the partition is quite similar.

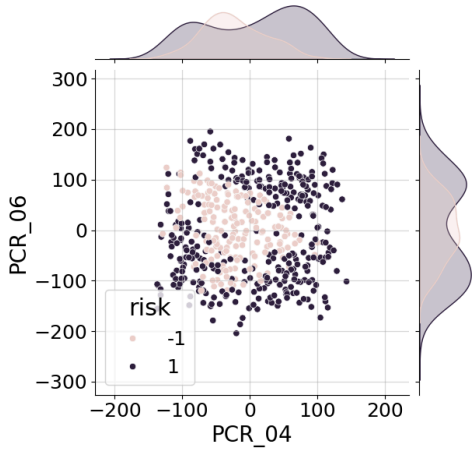
(Q18)
For the pair of PCR features you chose in (Q17), create three jointplots (see Tutorial 01), all conditioned on the risk variable. The first jointplot should include only the data in the first blood group you created in (Task E), {O+, B+}. The second jointplot should include only the data in the other blood group. The third jointplot should be for the full data, without partitioning to blood groups.
Attach the 3 resulting plots to your report. Remember to have grids, titles, and axis-labels.

A:

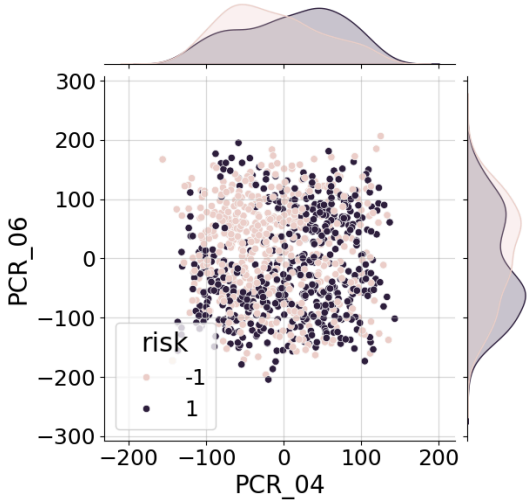
PCR_04 vs. PCR_06 (SpecialProperty = True)



PCR_04 vs. PCR_06 (SpecialProperty = False)



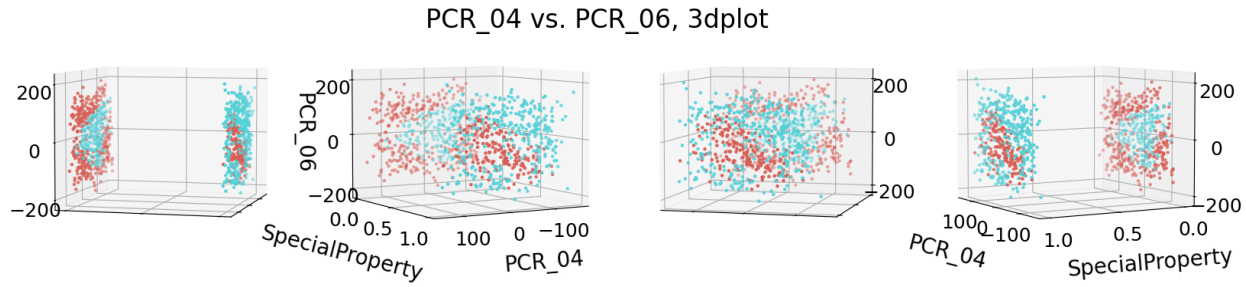
PCR_04 vs. PCR_06 (All)



(Q19)

Use the provided function plot3d to plot the pair of PCR features you chose (axes X and Z) and the SpecialProperty feature (axis Y), colored by the risk label. Make sure that the plot is clear & readable and that it has a proper title. Attach the plot to your report.

A:



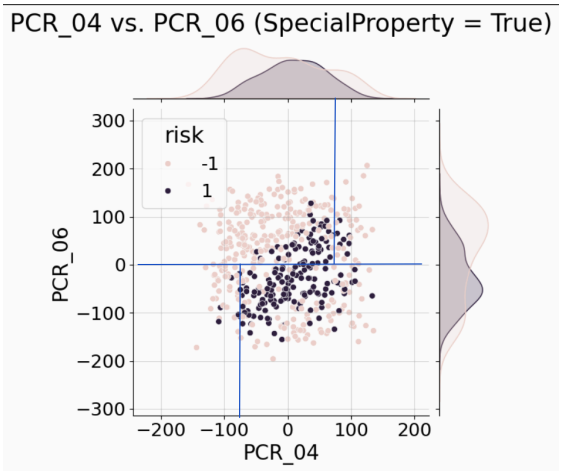
(Q20)

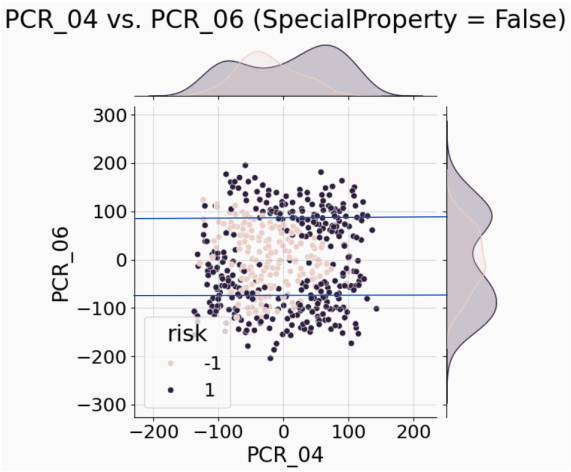
How well will a decision tree of max-depth=3 be able to fit the training data? Explain briefly.

A:

Not so well.

Based on our previous analysis, splitting the dataset based on SpecialProperty will results in two subsets each with relatively partitioned samples. For each subset, we can use a decision tree of depth 2, which separates the data using two vertical or horizontal lines. Unfortunately both subsets cannot be separated well into 3 or 4 rectangles. (See drawing).





(Q21)
How well will a decision tree of max-depth=30 be able to fit the training data?
Explain briefly.

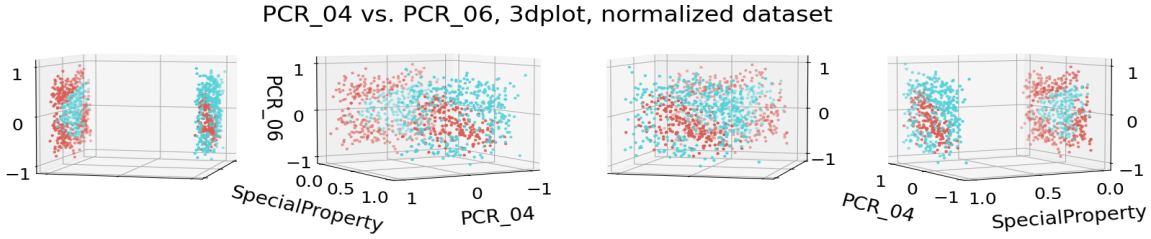
A:
Much better - probably very well.
Like in the previous question, we should first split the data based on SpecialProperty, now for both subsets we can use a decision tree with depth 29 to make our decisions. This is deep enough to be able to partition the data into decision areas with great accuracy.

(Q22)
How well will a 1-NN model be able to fit the training data? Note that in this question, a point in the training set is not considered its own neighbor (i.e., when making a prediction for a training data point, the model won't use the same point for prediction, but only the nearest point in the remaining training set).
Hint: consider the scale of the features in your answer.

A:
Not so well.
If we treat the boolean property SpecialProperty as {1,0}, we see that it's scale is a lot smaller than the other features which scale roughly in [-200,200]. This means that the nearest neighbor of a point with SpecialProperty=1 may be with SpecialProperty=0, and therefore we lose the partition we had in the two previous questions. This is especially bad since the data in each subset based on SpecialProperty has inverse labels - for SpecialProperty=1 we have samples with risk=1 at the center with risk=0 outside the center, and for SpecialProperty=0 it is the other way around.

(Q23)
What will be the effects of data normalization on your answers in (Q20), (Q21), (Q22)? Explain.

A:
For questions (Q20) and (Q21) it will not change, since both features had similar scales normalizing these features did not change the way the samples are distributed (now they simply distribute on [-1,1] instead of on [-200,200]).
For (Q22), our answer will change, since now the distance between two points in each plane (PCR_04 X PCR_06 for SpecialProperty=True/False) is smaller than 1 which is the distance between both planes (see 3dplot below). This means that the nearest neighbor will be chosen from the same plane, and this will increase accuracy greatly.



(Q24)
Write a table summarizing the data preparation process you created.
The columns of the table must be:
a. Feature name: the name of the feature as written in the dataset.
Names of new features should be meaningful!
b. New: “V” if the feature was handcrafted using other feature(s), “X” otherwise.
c. Normalization method, only for the PCR features.
d. Strategy to fill NULL values, if used.

A:

Feature name	New	Normalization method	Strategy to fill NULL values
patient_id	X	-	-
age	X	-	-
sex	X	-	-
weight	X	-	-
blood_type_is_o+,b+	V	-	-
current_location	X	-	-
num_of_siblings	X	-	-
happiness_score	X	-	-
household_income	X	-	Mean
conversations_per_day	X	-	-
sugar_levels	X	-	-
sport_activity	X	-	-
pcr_date	X	-	-
PCR_01	X	MinMax Scaling	-
PCR_02	X	Standardization	Mean
PCR_03	X	MinMax Scaling	-
PCR_04	X	MinMax Scaling	-
PCR_05	X	Standardization	-
PCR_06	X	MinMax Scaling	-
PCR_07	X	MinMax Scaling	-
PCR_08	X	MinMax Scaling	-
PCR_09	X	MinMax Scaling	-
PCR_10	X	Standardization	-