**Major HW 1 - Introduction to Machine Learning**

**Naama Shamgar: 204893952**

**Liran Cohen: 209043470**

1. The dataset has 2500 rows and 28 columns.
2. The output for num\_of\_siblings:

| 1.0 | 786 |
| --- | --- |
| 2.0 | 553 |
| 0.0 | 544 |
| 3.0 | 318 |
| 4.0 | 120 |
| 5.0 | 34 |
| 6.0 | 9 |
| 7.0 | 1 |
| 8.0 | 1 |
| 9.0 | 1 |

We think this feature refers to the number of siblings each person in the dataset has.

We believe this feature to be an ordinal variable, since it is composed of different “categories” (integers between 0 to 9) and has an intrinsic order.

| Feature Name | Description | Type |
| --- | --- | --- |
| patient\_id | The patient’s ID number | Other |
| Age | The patient’s age | Ordinal |
| Sex | The sex of the patient (male or female) | Categorical |
| Weight | The weight of the patient | Continuous |
| blood\_type | The patient’s blood type | Categorical |
| Address | The address of the patient | Other |
| Current\_location | Where the patient is right now | Other |
| Num\_of\_siblings | The number of siblings the patient has | Ordinal |
| Happiness\_score | A number describing how happy the patient is - the higher the number, the happier the patient. | Ordinal |
| Household\_income | The income of the patient’s family/household | Continuous |
| Conversations\_per\_day | How many conversations with other people the patient has each day | Ordinal |
| Sugar\_levels | The patient’s sugar level | Ordinal |
| sport\_activity | A number describing how much physical activity the patient does - the higher the number, the more active a patient is. | Ordinal |
| Symptoms | The symptoms the patient is experiencing | Other |
| Pcr\_date | The date when the patient’s PCR test took place | Other |
| PCR\_1 | Measure of some substance in the patient’s PCR sample. | Continuous |
| PCR\_2 | Measure of some substance in the patient’s PCR sample. | Continuous |
| PCR\_3 | Measure of some substance in the patient’s PCR sample. | Ordinal |
| PCR\_4 | Measure of some substance in the patient’s PCR sample. | Continuous |
| PCR\_5 | Measure of some substance in the patient’s PCR sample. | Continuous |
| PCR\_6 | Measure of some substance in the patient’s PCR sample. | Continuous |
| PCR\_7 | Measure of some substance in the patient’s PCR sample. | Continuous |
| PCR\_8 | Measure of some substance in the patient’s PCR sample. | Continuous |
| PCR\_9 | Measure of some substance in the patient’s PCR sample. | Continuous |
| PCR\_10 | Measure of some substance in the patient’s PCR sample. | Continuous |

1. Patient\_id - since the values are unique, it cannot be categorical or ordinal. Also, the difference between different values is abrupt, therefore it’s not continuous.

Age - this feature is made up of different “categories” (ages), that have an intrinsic order, therefore it is ordinal.

Address - There is no order to the different values of this feature (since it’s a string), therefore it isn’t continuous or ordinal. There also aren’t “categories”, therefore it is not categorical.

Current\_location - same as address, there is no order between the different values, and there are no “categories”.

Num\_of\_siblings - this feature is made up of different “categories” (integers between 0 and 9) that have an intrinsic order.

Happiness\_score - this feature is made up of different “categories” (integers between 1 and 9) that have an intrinsic order.

Conversations\_per\_day - this feature is made up of different “categories” (integers between 0 and 20) that have an intrinsic order.

Sugar\_levels - again, this feature is made up of different “categories” (integers between 40 and 159) that have an intrinsic order.

sport\_activity - this feature is made up of different “categories” (integers between 0 and 4) that have an intrinsic order.

Symptoms - this feature is a string, therefore it isn’t continuous or ordinal (no order). Furthermore, for each sample this field holds several values, therefore it isn’t categorical.

Pcr\_date - Since this feature holds several numerical fields (month, year, date), it can’t be categorized as any of the options (ordinal, categorical or continuous).

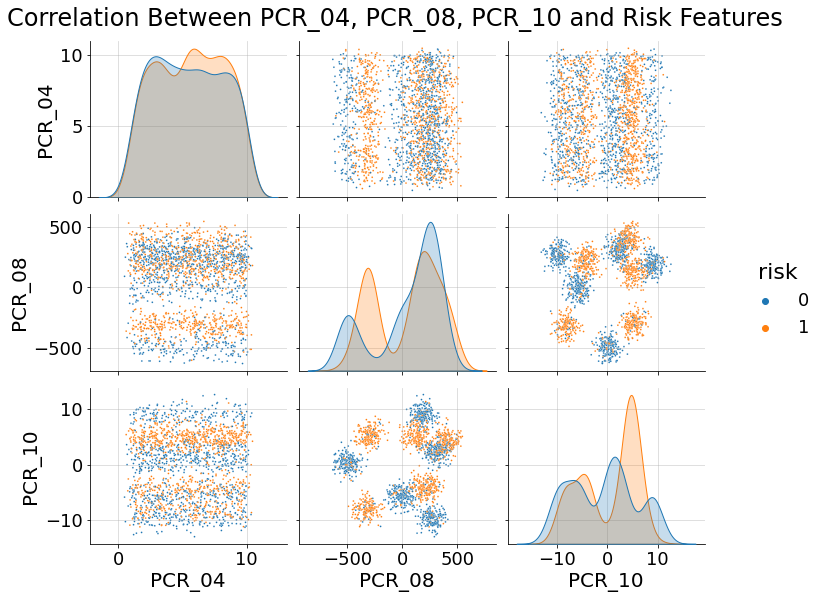
PCR\_3 - this feature is made up of different “categories” (integers between 0 and 9) that have an intrinsic order.

1. In our data preparation we use different values we deduce from the data - median of a certain feature, the maximal value of a feature, etc. If we split the data differently every time, we could potentially receive different values every time and that could damage the data preparation and processing.
2. The correlations are:

| PCR\_04 | 0.023026580152301208 |
| --- | --- |
| PCR\_08 | -0.011683685828004397 |
| PCR\_10 | 0.09580951519896994 |

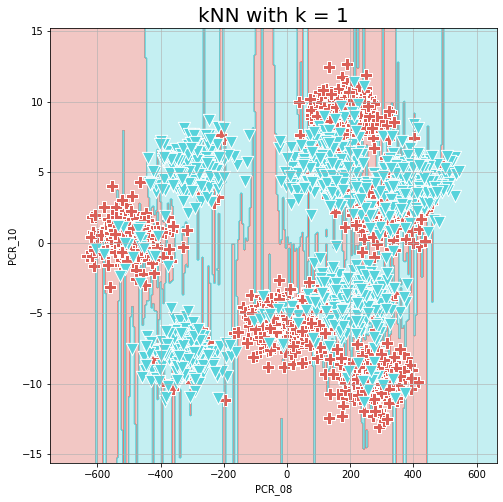
Seeing as a value of 1 or -1 indicates a perfect linear correlation between 2 variables, we can deduce none of those features are highly correlated with the risk.

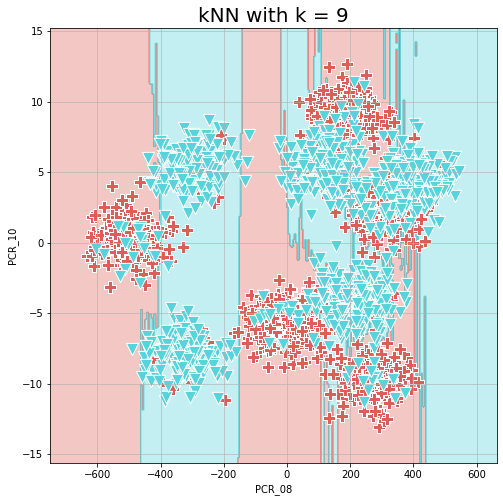
1. Below is the pairplot describing the correlations requested:



In this pair plot, we can see that out of the 3 features the 2 that can be the most helpful to predict ‘risk’ are PCR\_08 and PCR\_10. We can see in the plot that when using these 2 features together, the samples are somewhat close to separable. However, any 2 other features will give us no clear correlation (as we can see in the plot, the High and Low risk samples are fairly mixed and would be very difficult to separate). This settles with our answer to the last question, because in question 6 we looked at the correlation between risk and each feature separately. In this question, we look at the correlation between risk and a pair of features, which gives us more information.

1. Below are the visualizations for the decision boundaries of the kNN models:

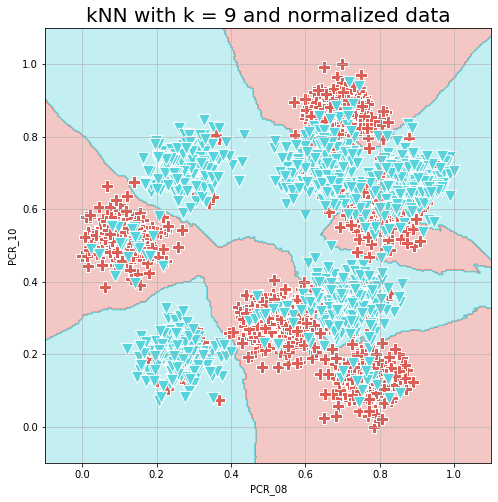




The score for the model with k = 1 is 1.0.

The score for the model with k = 9 is 0.827.

1. The k = 1 model has better accuracy. This perfect precision comes from the fact that the closest point to each point is the point itself, therefore the model is deciding each point’s label solely based on its own label. Since we trained and tested on the same set of points, we got 100% accuracy (for predicting we use the actual labels).
2. With k = 1 we see some overfitting - the boundaries are less smooth. Since the model with k = 9 takes into account more points for each area, we get smoother boundaries that would fit the general case better.
3. Below are the decision boundaries for the normalized data:



The training accuracy is 0.865.

We can see that the model that used the normalized data achieved higher accuracy, and we can also see in the visualization smoother decision boundaries. This is attributed to the normalization of the data - previously, PCR\_08’s values were in a much wider range than PCR\_10, so the model regarded those as more important (that fact is also visible in the decision boundaries for k = 9 with unnormalized data). Now, however, the boundaries are not only smoother but are also across both axes, showing the model now takes into account PCR\_10 just as much.

1. If a feature is normally distributed, then its expectancy is 0 and most of the values would be concentrated between -1 and 1. However, some values would reside outside of this range, some of them having large values. If we normalize this feature using min-max scaling, the minimal and the maximal value would be of the extreme values, and would cause the majority of the samples to move closer to 0 even though the extreme values are few and shouldn’t affect the data too much. Doing this would essentially distort the data, giving the few extreme values more influence than the majority of the values that are around 0. kNN specifically is a distance dependant model, so if we use min-max scaling we would greatly influence distances between the samples and therefore damage kNN’s performance.
2. Since there are 8 different blood types that appear in our dataset, the length of the vector is 8 bits (one bit for every blood type).
3. We can extract useful information from the ‘Symptoms’ feature by splitting it into 5 different features - one for every symptom. Each sample will have 1 value if that patient had the corresponding symptom, and 0 otherwise. We used the get\_dummy function, as if to use OHE, but every sample can get 1 value in more than one feature or 0 in all of them.
4. Firstly, we converted the “Sex” feature to numerical values using 1/0 values instead of OHE, as it makes more sense. This feature could be relevant, as our target variables may be influenced by gender. We also converted ‘Spread’ and ‘Covid’ (our target variables) in the same manner, as to remain with only numerical features.

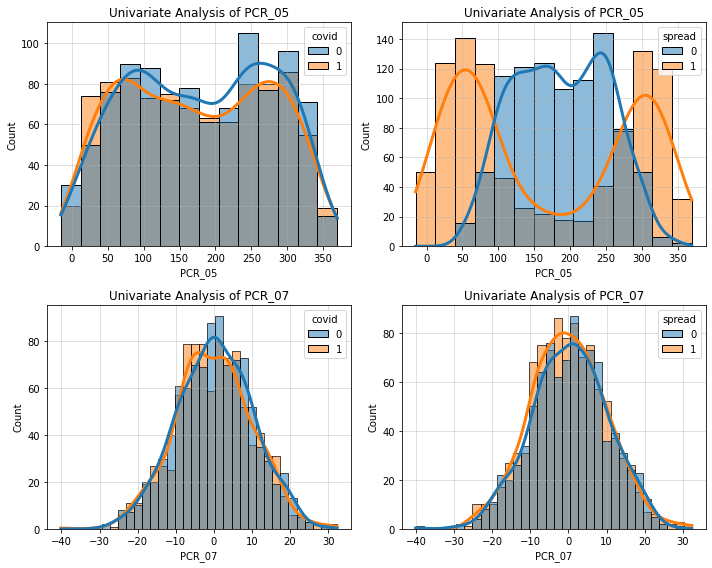
Next, we split pcr\_date into day, month and year, to create 3 separate numerical features. We think this feature is important because certain times of the month or specific months can correlate to the target variables.

We also split the “current\_location” feature into latitude and longitude. We think this feature could be important since specific areas might be more prone to the spread of covid.

Since there is no reasonable way to turn the “Address” feature into numerical values (using OHE on any part of the variable, for example, will result in an enormous number of bits), and since it is mostly a unique feature (very few people will share an address), we decided to drop this feature.

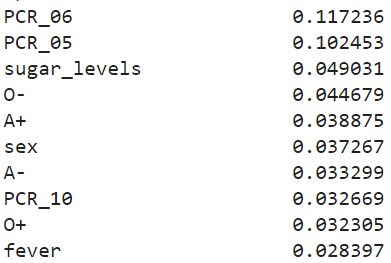
“Blood\_type” and “Symptoms” were already converted to numerical values in earlier questions.

1. Below are the histograms we generated for PCR\_05 and PCR\_07, regarding the “Spread” and “Covid” features:



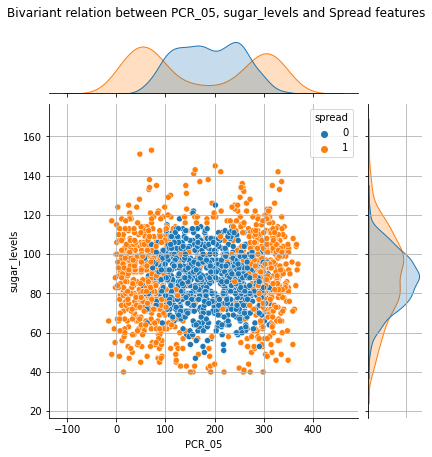
In the histogram for PCR\_05 and Spread, we can see that in almost every single column, the majority of the samples point to a specific label - specifically, values of PCR\_05 under 100 and over 300 would highly indicate high spread, while values between 100 and 250 would indicate low spread. Therefore,PCR\_05 would be somewhat reliable at predicting Spread. In the other 3 histograms, we can see that the columns are mostly “mixed” - each range of values for PCR\_07 has more or less the same amount of samples for both values of the target variables, and the same is true for PCR\_05 and Covid. From this we can deduce that PCR\_05 and PCR\_07 would not work well at predicting Spread and Covid, other than PCR\_05 and Spread (as we’ve described earlier).

1. As we can see in the output of the code, the 10 most correlated features to Spread are as follows:



We excluded the Spread feature itself, as it doesn’t provide any additional information (no reason in using a variable to predict itself).

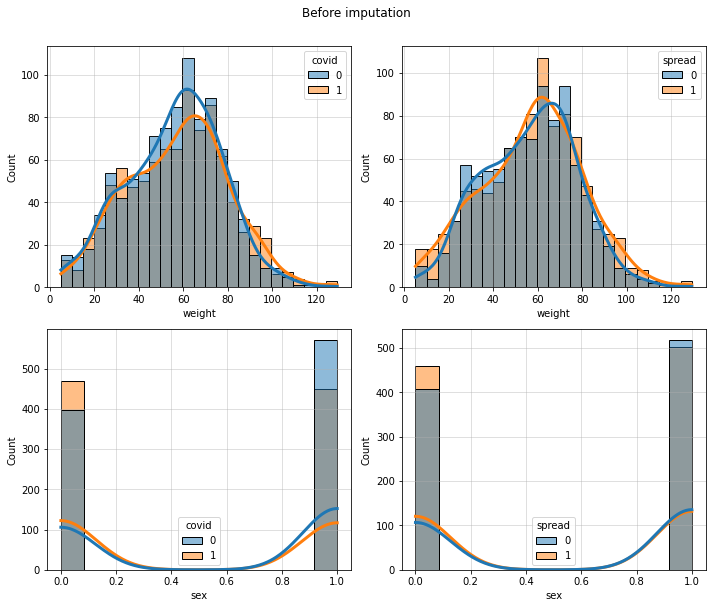
1. We attempted to find the 2 features that best explain the ‘Spread’ label. The method we used was drawing a pairplot for every 2 variables out of the 3 highest correlated ones, and determining which plots showed a distribution closest to separable. In the pair plots we received, we saw that in itself, PCR\_05 showed a distribution close to separable in the values for “Spread”, therefore we chose it as one of the features used. Furthermore, joined together with sugar\_levels we could see a distribution close to separable - with one main middle area for Low spread. Below is the jointplot for PCR\_05, sugar\_levels and Spread:



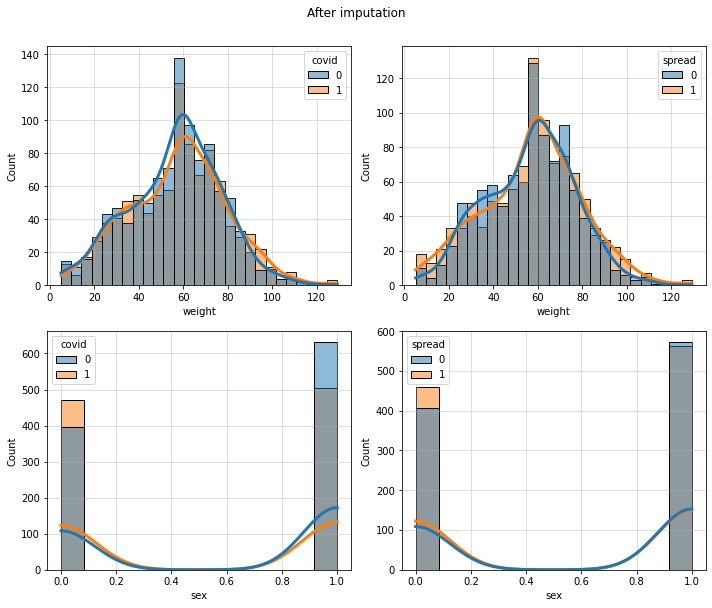
1. There is no question 17 :)
2. Advantage: This method is very easy to implement - only requires the calculation of the median/mean.

Disadvantage: Using this method could distort the distribution of the variable.

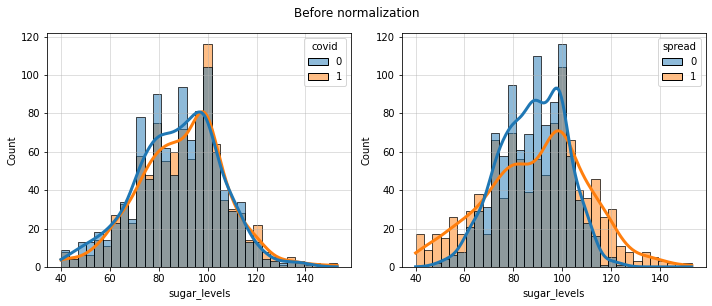
1. In using the “missing category” method we gain a way to represent the missing values - instead of ignoring them or assigning a value to the missing info, we derive information from the fact that data is missing. Furthermore, since we are not assigning a “real” value to the missing values, we aren’t distorting the data.
2. We chose to impute the missing values from ‘Happiness\_score’ using the ‘Frequent category imputation’ method. We wanted to choose a method fitting an ordinal variable, which is a kind of categorical variable. Since ordinal variables are essentially categorical with an order to them, we chose this imputation method which is somewhat similar to the median imputation method, only more relevant to categorical variables. Since only 4.5% of the values are missing for this feature, using this method probably wouldn’t disturb the distribution too much, and wouldn’t cause us to lose the samples or ignore them.
3. Univariate analysis visualization before filling Null values:

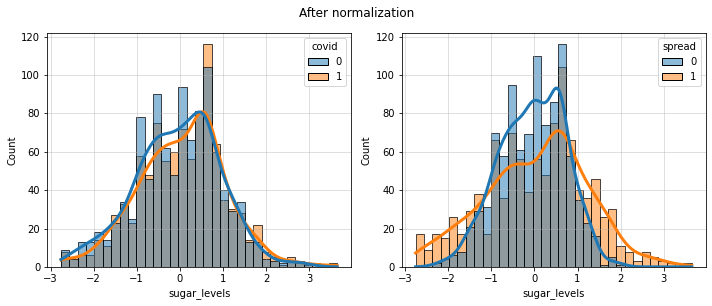


After filling Null values:



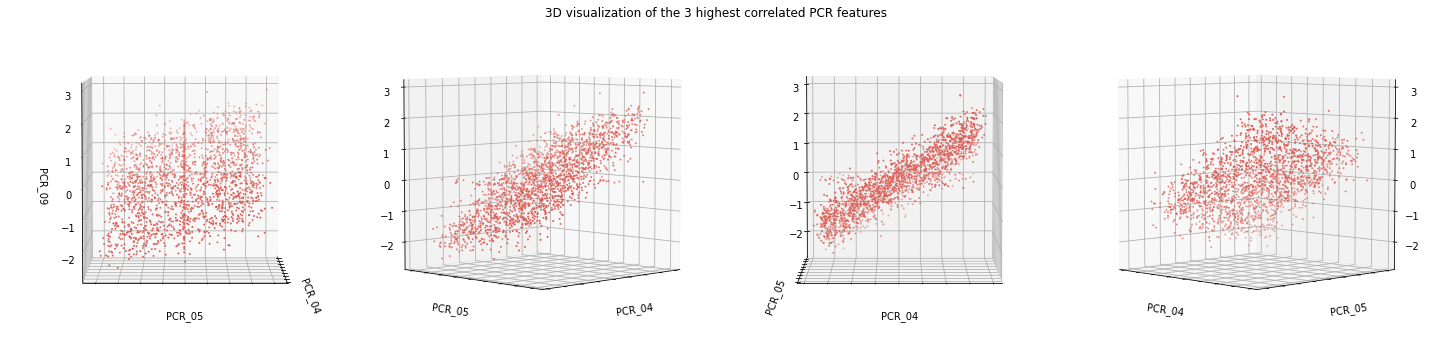
1. Below are the histograms for ‘sugar\_levels’ before and after normalization:





1. From the pairplot, we can deduce that the feature that interacts most strongly with PCR\_01 is PCR\_02. We can see in the pairplot that for every value of PCR\_01, we get a specific range of values where the value of PCR\_02 will be found. However, we cannot remove one of those features from our dataset - from a value of PCR\_01 we get several different ranges for PCR\_02 (there is no linear correlation between them), and as such, any combination of values for PCR\_01 and PCR\_02 might predict different labels for our target variables. Therefore, we cannot remove one of those features without damaging the prediction process.

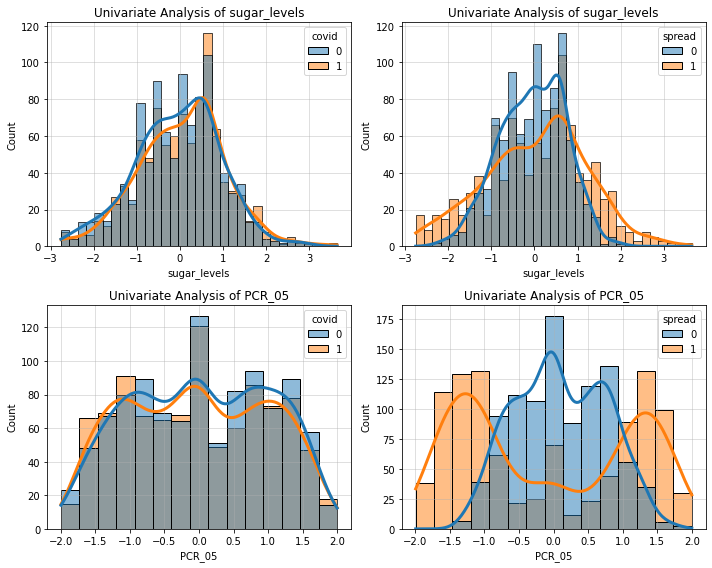
1. Below we attach the 3d plot for PCR\_04, PCR\_05, PCR\_09:



1. We can see a clear linear correlation, which is the strongest between PCR\_04 and PCR\_09. Furthermore, we can see the shape of a plane.
2. By looking at the correlations we calculated earlier and the graphs and 3d plots we have, there is a clear linear correlation between PCR\_04 and PCR\_09. Therefore, we can remove either one of those without it damaging our prediction too much - since one feature determines the other. We can see that PCR\_09 is more highly correlated to PCR\_05 than PCR\_04 is to PCR\_05, therefore the best feature to remove without damaging our predictions too much would be PCR\_09.
3. Using the sequential feature selector, the features we got are ‘PCR\_05’ and ‘Sugar\_levels’’. These are the same features we found in question 16.
4. In our feature selector we used a kNN classifier. for such a classifier, normalizing the data improves the results - a feature wouldn’t be viewed as having higher importance only because its values are higher (for example, if one variable has values in the range of (0,10), and another in (-100, 100), since the distances for this feature are greater they might be considered more important for the model). Generally, we would like all features to have equal weight in the decision making process, so we would like them all to have values in more or less the same range. Furthermore, missing values would interfere with the model, therefore it’s important to perform imputation before using the feature selector.
5. We will present our feature selection for the target variables:

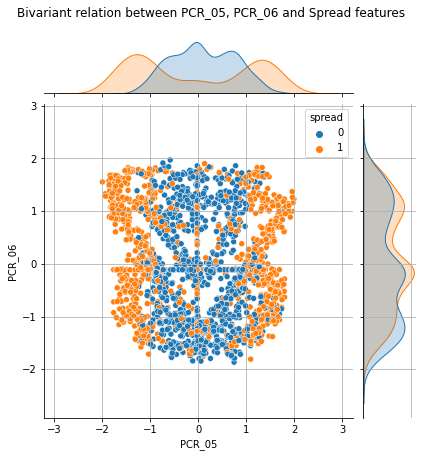
Spread

Firstly, we used our univariate analysis. In the analysis, we noticed correlations between two features and the spread target variable - sugar levels and PCR\_05:



As we can see in the histograms and as we’ve described in question 14, there is a very high correlation between PCR\_05 and spread (values of PCR\_05 below -1 and above 1.25 (after normalization) indicate High spread, while values between -0.5 and 1 indicate a low spread). Furthermore, in the histogram for sugar\_levels we can see that values under -1.8 or over 1.2 indicate high spread.

Next we drew a pairplot for the 10 most highly correlated features to spread. We could see that PCR\_06 combined with PCR\_05 give a fairly separable distribution - meaning together they could be beneficial at predicting the spread variable:



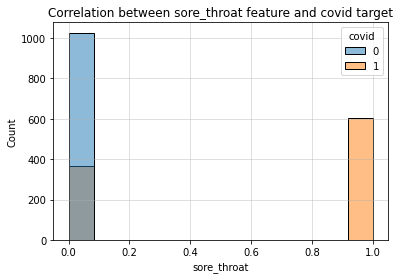
We found no linear relations between any 2 features (except any that we previously saw in this report), so this step did not cause us to drop any features.

The top 10 most highly correlated features to spread were: PCR\_05, PCR\_06, pcr\_year, O-, sugar\_levels, A+, sex, A-, pcr\_month, PCR\_10.

Finally, we used the sequential feature selector to pick the 3 best features for predicting the covid target variable using a greedy algorithm. We got the features PCR\_05, sugar\_levels (like we saw in question 25) and low\_appetite.

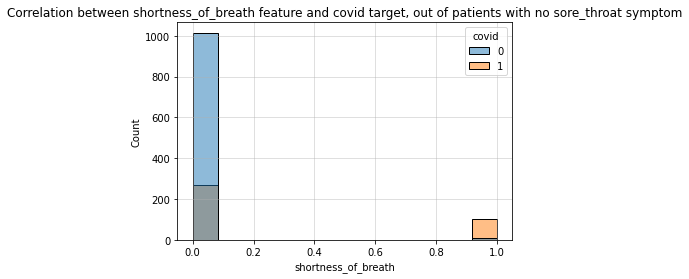
Covid

Firstly, we used our univariate analysis. In the analysis, we noticed correlations between the sore\_throat feature and the target variable covid:



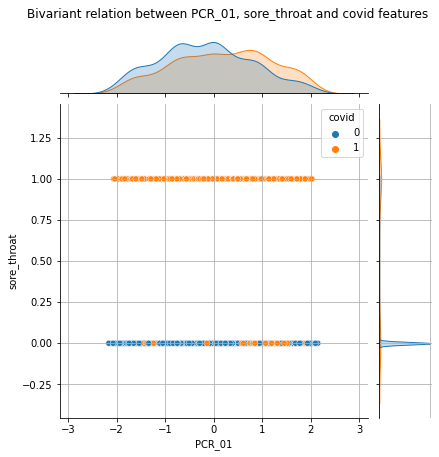
As we can see in the histogram, all patients who show the sore\_throat symptom are positive for covid, and the majority of those who don’t show it are negative for covid.

Another interesting observation is, that for patients without a sore\_throat symptom, we see a high correlation between covid variable and shortness\_of\_breath symptom:



We can see that for those patients, shortness of breath will show positivity for covid almost certainly, and the majority of patients without shortness of breath (and without sore throat) will be negative for covid.

Next we drew a pairplot for the 10 most highly correlated features to covid. We saw that sore\_throat has a close to linear correlation to PCR\_1 (with sore\_throat being the determining factor):



Therefore we’ve decided to drop the PCR\_01 feature.

The top 10 most highly correlated features to covid were: sore\_throat, shortness\_of\_breath, sport\_activity, cough, PCR\_01, PCR\_09, PCR\_04, sex, A+, O+.

Finally, we used the sequential feature selector again in order to pick the 3 best features for predicting the covid target variable using a greedy algorithm. We got the features cough, shortness of breath and sore throat.

1. The Table:

| Feature name | Keep | New | Imputation method | Normalization method | Explanation |
| --- | --- | --- | --- | --- | --- |
| age | X | X | Frequent category | Standard | Ordinal variable, therefore we used frequent category imputation (a method recommended for categorical variables). We normalized with standard normalization because this feature has a big range of values. Finally, we dropped this feature since it showed no clear correlation to any target variable. |
| sex | V | X | Frequent category | Min-max | Categorical variable - used frequesnt category imputation. A “boolean” feature - normalized with min-max (doesn’t change the values). We chose to keep this feature since it shows some correlation to spread and covid. |
| weight | X | X | Median | Standard | Continuous variable - used median imputation (small percentage of the values were missing) and standard normalization (better for continuous variables). We decided to drop this feature since it showed no clear correlation to any target variable. |
| num\_of\_siblings | X | X | Frequent category | Min-max | Ordinal variable with a small range of values - frequent category imputation and min-max normalization. We decided to drop this feature since it showed no clear correlation to any target variable. |
| household\_income | X | X | Median | Standard | Continuous variable - median imputation and standard normalization. We decided to drop this feature since it showed no clear correlation to any target variable. |
| conversations\_per\_day | X | X | Frequent category | Min-max | Ordinal variable with a small range of values - frequent category imputation and min-max normalization. We decided to drop this feature since it showed no clear correlation to any target variable. |
| sugar\_levels | V | X | Frequent category | Standard | Since this is an ordinal variable with a large range of values - we treated it as if it’s continuous and used standard normaization. Since it’s ordinal, we used frequent category imputation. We kept this feature since it showed high correlation with spread (alone and together with PCR\_05) |
| sport\_activity | V | X | Frequent category | Min-max | Ordinal variable with a small range of values - frequent category imputation and min-max normalization. We kept this feature since it showed some correlation to covid. |
| PCR\_01 | X | X | Median | Standard | continuous variable - median imputation and standard normalization. We decided to drop this feature since it is linearly correlated to sore\_throat. |
| PCR\_02 | X | X | Median | Standard | continuous variable - median imputation and standard normalization. We decided to drop this feature since it showed no clear correlation to any target variable. |
| PCR\_03 | X | X | Frequent category | Min-max | Ordinal variable with a small range of values - frequent category imputation and min-max normalization. We decided to drop this feature since it showed no clear correlation to any target variable. |
| PCR\_04 | V | X | X | Standard | continuous variable - median imputation and standard normalization. We kept this feature since it showed some correlation to covid. |
| PCR\_05 | V | X | Median | Standard | continuous variable - median imputation and standard normalization. We kept this feature since it showed high correlation to spread (alone and paired with sugar\_levels or PCR\_06) |
| PCR\_06 | V | X | Median | Standard | continuous variable - median imputation and standard normalization. We kept this feature since it showed high correlation to spread when paired with PCR\_05. |
| PCR\_07 | X | X | Median | Standard | continuous variable - median imputation and standard normalization. We decided to drop this feature since it showed no clear correlation to any target variable. |
| PCR\_08 | V | X | X | Min-max | continuous variable - median imputation and standard normalization. We decided to keep this feature as it is correlated to risk. |
| PCR\_09 | X | X | Median | Standard | continuous variable - median imputation and standard normalization. We decided to drop this feature since it showed no clear correlation to any target variable. |
| PCR\_010 | V | X | X | Min-max | continuous variable - median imputation and standard normalization. We decided to keep this feature as it is correlated to risk. |
| A+ blood\_type | V | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to keep this feature as it showed some correlation to spread and covid. |
| A- blood\_type | V | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to keep this feature as it showed some correlation to spread. |
| AB+ blood\_type | X | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to drop this feature as it showed no clear correlation to any target variable. |
| AB- blood\_type | X | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to drop this feature as it showed no clear correlation to any target variable. |
| B+ blood\_type | X | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to drop this feature as it showed no clear correlation to any target variable. |
| B- blood\_type | X | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to drop this feature as it showed no clear correlation to any target variable. |
| O+ blood\_type | V | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to keep this feature as it showed some correlation to covid. |
| O- blood\_type | V | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to keep this feature as it showed some correlation to spread. |
| symptom: cough | V | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to keep this feature as it showed some correlation to covid. |
| symptom: fever | X | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to drop this feature as it showed no clear correlation to any target variable. |
| symptom: low\_appetite | V | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to keep this feature as it showed some correlation to covid. |
| symptom: shortness\_of\_breath | V | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to keep this feature as it showed high correlation to covid among samples with a value of sore\_throat = 0. |
| symptom: sore\_throat | V | V | X | Min-max | “Boolean” variable (similar to sex) - min-max normalization. No missing values - no imputation. We decided to keep this feature as it showed high correlation to covid. |
| pcr\_day | X | V | Frequent category | Standard | Ordinal variable with a large range of values - frequent category imputation and standard normalization. We decided to drop this feature since it showed no clear correlation to any target variable. |
| pcr\_month | V | V | Frequent category | Standard | Ordinal variable with a large range of values - frequent category imputation and standard normalization. We decided to keep this feature as it showed some correlation to spread. |
| pcr\_year | V | V | Frequent category | Standard | Ordinal variable with a large range of values - frequent category imputation and standard normalization. We decided to keep this feature as it showed some correlation to spread. |
| current\_location\_ latitude | X | V | Median | Standard | continuous variable - median imputation and standard normalization. We decided to drop this feature since it showed no clear correlation to any target variable. |
| current\_location\_ longitude | X | V | Median | Standard | continuous variable - median imputation and standard normalization. We decided to drop this feature since it showed no clear correlation to any target variable. |
| happiness\_score | X | X | Frequent category | Min-max | Ordinal variable with a small range of values - frequent category imputation and min-max normalization. We decided to drop this feature since it showed no clear correlation to any target variable. |
| patient\_id | X | X | X | Standard | No missing values - no imputation. A large range of values - standard normaliztion. We decided to drop this feature since it showed no clear correlation to any target variable. |
| blood\_type | X | X | X | X | Split using OHE. Missing values were treated as a vector of all zeroes - no blood type had a value of 1. |
| address | X | X | X | X | Was dropped since we weren’t able to convert it into a numerical feature. |
| current\_location | X | X | X | X | Split into 2 continuous features, then was imputed and normalized. |
| symptoms | X | X | X | X | Was split into seperate symptom features and then dropped to avoid duplication. |
| pcr\_date | X | X | X | X | Was split into day, month and year and then dropped to avoid duplication. |