**Introduction to Machine Learning**

**Major HW 1**

Submitted by: Sofia Blyufer 321128019 and Dor Rozen 318965365

1. See code.
2. In the table below, we can see for each attribute a proposed meaning and a proposed type of attribute.
3. See in the table above.
4. See code.
5. See 2.
6. The strings discussed are “Job”, “CurrentLocation” and “Address” and “Self\_declaration\_of\_Illness\_Form”.   
   Job – we have decided to not extract any information from this feature for two reasons. The first, there is more than 25% missing data for this attribute. The second, there are too many unique values (620 out of 3000) and is not informative enough.  
   Current Location – location coordinates can be important to identify risk area. As the data already numerical, which is easiest for further analysis, we have decided to split the data into two attributes – X and Y (x and y coordinates separately).   
   Address – we have seen fit to drop this attribute, as the location information is already numerically convenient in “CurrentLocation”. Address is an ID feature and therefore might only harm our modelling. Using the same reason, we have decided to drop “ID” now.  
   Self-Declaration of Illness Form (symptoms) – we decided to split into the 14 different types of features, columns of 1 and 0 (symptom exists or not). We are aware it will be a lot of new information with risk of overfitting and long running times and will later find the most relevant features.
7. Chart, histogram

   Description automatically generatedAs previously mentioned, job, address and ID were deleted altogether. Additionally, we have dropped “PCR\_11” and “PCR\_15” as they both have above 80% missing values, and imputation would be too inaccurate in such a case. The rest of the features were chosen to be imputed as there is enough data to draw conclusions based on the its distribution.   
   For example, the histogram below is of the attribute “MedicalCarePerYear”. It can be approximated well to a normal distribution. Therefore, for such features we think imputation is relatively straightforward.  
   For other features that had enough data we also decided to impute, using other methods, as will be described below.
8. We have handled our features according to 3 types:
9. See code.
10. Let us show BMI as an example. First, we have transformed all values into z-score values. Then, we removed any rows that contained z-score below or above 3 (in other words ). The process can be seen in the plots below, that show (from left to right): BMI raw train data box plot, data transformed into z-scores and data after outliers removal. For all numeric data the process was the same.  
    Chart, scatter chart

    Description automatically generated
11. Step per year was normalized like all other numeric data, into z-score. The graph below represents the distribution of “StepsPerYear” and confirms a z-score transformation will be convenient (the mean is relatively representative).   
    Chart, histogram

    Description automatically generated
12. For numeric data, we used the advantage of z-score transformation, which can normalize any type of distribution and is sensitive to outliers, unlike min/max.  
    Additionally, for better learning, we decided to convert all categorical data to numeric data. Henceforth, “Virus” became 0 for anything that wasn’t “covid” and 1 for “covid”. We kept a code line for another splitting, 0 for not sick, 1 for covid and 2 for other diseases, in case it might have us with learning in future assignments.   
    Other data, such as “Risk”, “SpeadLevel” and “BloodType” was also converted numerically. We checked that all possible categories exist in train (that we aren’t missing categories that were split only into test).
13. Before analysing all correlations, we decided to choose the most relevant symptoms amongst the 14 given. The correlation map below shows correlations for all 14 symptoms, as well our three target labels.   
    Graphical user interface, chart, surface chart

    Description automatically generated  
    As we can see, no high correlations were generated. We decided to perform a decision tree on these features, to see if new information would arise there. The importance values can be found below. We chose “New\_loss\_of\_taste\_or\_smell”, ‘Nausea\_or\_vomiting” and “Low\_appetite”.   
    The correlation table to the features left is presented below, including the chosen symptoms.  
    Chart

    Description automatically generated with medium confidence

To understand the correlations, we plotted histograms and plots of highly correlated features. The pairs we decided to check were those with correlation 0.8 and above.

For two pairs we’ve received perfect correlation: steps per year (not normalized) vs. age group and number of cousins vs. age group. The plots below confirm these numbers.

Chart, scatter chart

Description automatically generated

Therefore, by keeping the feature of age group, number of cousins and steps per year do not add any new data. From the plots we can also see that there is no need of manipulation and taking a combination of the data (look the same).

Before showing the next features, it should be noted that as previously explained, missing data was filled with mean. As a result, peaks in the mean are observed. Consequently, those effects slightly “ruin” correlations by creating a “cross”. For example, below is a plot of household expenses of parking tickets per year vs. studying per day. On the left is the original data, and on the right is the data normalized to z-score values, after filling missing data and outlier removal. For that reason, we also look at the plots before filling the data. Nevertheless, the correlation is worse after adding the “cross”, and therefore we are only being stricter. The correlation of the pair after filling the missing data is 0.91.

Chart, line chart

Description automatically generated

1. In addition to the correlations explained above, we have also used the following methods for feature selection: decision tree (as wrapper method) and manual inspection of scatter plots as univariate and bivariate methods.   
   As we have 3 target labels, we performed 3 decision trees, one for each target. Eventually, the importance arrays were added into the final table below (in section 15) to help choose the most important features. The best indicator for the decision tree was the sum of importance for each feature, as we need the same features for modelling for all 3 targets.   
   By eliminating correlated features, choosing them according to highest importance, we were left with features that weren’t redundant correlation-wise and were not highly important on decision trees. Those we decided to inspect manually using univariate and bivariate methods. Let us first notice that high correlation of risk and some features was found, but has already been noticed by the decision tree importance and therefore will not require any further treatment.