COVID-19 SIMULATION REPORT
DATA PROCESSES ASSIGNMENT

Authors:	María Ayuso Luengo, Doga Cengiz, Javier Gallego Gutiérrez and Pablo Hernández Carrascosa
Degree:	Master's Programme in Data Science
Date:	January 9, 2022

Contents

1 Introduction

Along this document, we present the results of the work detailed in the Project plan. We will cover the most interesting aspects of the data analysis performed as well as the results of the prediction models created. The data analysis was developed in *Python* and the prediction models were developed using KNIME. Everything can be seen in the zip file provided or in `https://github.com/javiegal/covid19-simulation`.

2 Data analysis

This section will cover all the different parts related with the processing and analysis of the original data set.

2.1 Variables examination

The data set consisted initially of 2054 instances and 13 different variables that are explained in Table 1.

2.2 Data preprocessing

After studying the meaning, type and expected value of the data set variables, we had to process them and remove null, empty and abnormal values. We explain the process followed step by step in the following lines.

Attribute	Values expected	Description	Type
ID	Integer	Patient's ID	Numeric
AGE	Integer	Patient's age	Numeric
SEX	MALE, FEMALE	Patient's sex	Categorical
DAYS_HOSPITAL	Integer	Days in hospital	Numeric
DAYS_ICU	Integer	Days in ICU	Numeric
EXITUS	YES, NO	Exitus	Categorical
DESTINATION		Destination after being admitted in ER	Categorical
TEMP	Double	Temperature	Numeric
HEART_RATE	Integer	Heart rate	Numeric
GLUCOSE	Integer	Blood glucose	Numeric
SAT_O2	Integer	Oxygen saturation	Numeric
BLOOD_PRES_SYS	Integer	Systolic blood pressure	Numeric
BLOOD_PRES_DIAS	Integer	Diastolic blood pressure	Numeric

Table 1: Variable description

1. We started the preprocessing transforming binary variables ("EXITUS" and "SEX") to numerical variables. They were mapped the following way:
   - "EXITUS": "NO" $\mapsto$ 0 and "YES" $\mapsto$ 1.
   - "SEX": "FEMALE" $\mapsto$ 0 and "MALE" $\mapsto$ 1.
2. Then, we looked at the missing values.

```
--------------MISSING VALUES--------------
AGE                   4
SEX                   2
DAYS_HOSPITAL         0
DAYS_ICU              0
EXITUS               41
DESTINATION        1383
TEMP                  0
HEART_RATE            0
GLUCOSE               0
SAT_O2                0
BLOOD_PRES_SYS        0
BLOOD_PRES_DIAS       0
dtype: int64

--------------DATAFRAME SHAPE--------------
(2054, 12)
```

   Since there are 2054 observations in total, and 1382 are null in variable "DESTINATION", we decided to drop it. We can also see that variable "EXITUS" has 41 null values. We deleted the rows were "EXITUS" was null since it was our target variable and it does not make sense to impute those values.

   After that, we proceeded to impute the age and sex missing values. First, we imputed the missing values of "SEX" by the mode. In this case, there were only 2 missing values so this imputation did not introduce noise or modify these variables distribution. On the other hand, we decided to impute variable "AGE" by the median. We used this estimator instead of the mean because, as we will see shortly, 'AGE" had some abnormal values.

3. We then studied the descriptive statistics of our numeric columns.

```
--------------DATAFRAME DESCRIPTION--------------
              AGE  DAYS_HOSPITAL     DAYS_ICU        TEMP   HEART_RATE
count  2013.000000    2013.000000  2013.000000  2013.00000  2013.000000
mean     70.842524       8.119722     0.353701    28.65077    71.639344
std      20.462241       6.205003     2.178214    15.24003    41.378233
min      15.000000       0.000000     0.000000     0.00000     0.000000
25%      57.000000       4.000000     0.000000    35.40000    64.000000
50%      68.000000       7.000000     0.000000    36.40000    84.000000
75%      98.000000      10.000000     0.000000    36.90000    98.000000
max     189.000000      98.000000    36.000000    40.10000   593.000000

           GLUCOSE       SAT_02  BLOOD_PRES_SYS  BLOOD_PRES_DIAS
count  2013.000000  2013.000000     2013.000000      2013.000000
mean      1.812221    74.091406       84.199205        48.696970
std      20.640189    37.337955       67.363599        44.293107
min       0.000000     0.000000        0.000000         0.000000
25%       0.000000    82.000000        0.000000         0.000000
50%       0.000000    93.000000      115.000000        65.000000
75%       0.000000    96.000000      137.000000        79.000000
max     448.000000    99.000000      772.000000       845.000000
```

In this table we see some unusual values that could be a result of errors in data collection. For example, the maximum age is 189, which we know must be an error.

4. We decided to check the highest values for variable "AGE".

```
--------------HIGHEST AGE. 10 ELEMENTS--------------
         SEX EXITUS    AGE  DAYS_HOSPITAL  DAYS_ICU  TEMP  HEART_RATE
ID
2050  FEMALE     NO  189.0           11.0       3.0   0.0         0.0
2049  FEMALE    YES  106.0            5.0       0.0  38.2        89.0
2048  FEMALE     NO  105.0            4.0       0.0  36.4        74.0
2047  FEMALE    YES  102.0            5.0       0.0  36.5        83.0
2046  FEMALE    YES  101.0            2.0       0.0  36.8        84.0
2045    MALE    YES  100.0            2.0       0.0  36.6        65.0
2044    MALE     NO  100.0            3.0       0.0  36.6        70.0
2040  FEMALE     NO   99.0            3.0       0.0   0.0        90.0
2042  FEMALE     NO   99.0            7.0       0.0  37.3        92.0
2043  FEMALE    YES   99.0            4.0       0.0   0.0         0.0

      GLUCOSE  SAT_02  BLOOD_PRES_SYS  BLOOD_PRES_DIAS
ID
2050      0.0     0.0             0.0              0.0
2049      0.0    98.0           143.0             63.0
2048      0.0    98.0           169.0             97.0
2047      0.0    94.0           150.0             65.0
2046      0.0    95.0           110.0             65.0
2045      0.0    84.0           144.0             80.0
2044      0.0    94.0             0.0              0.0
2040      0.0    92.0             0.0              0.0
2042      0.0    92.0           149.0             73.0
2043      0.0     0.0             0.0              0.0
```

Since the age value 189 looks like an error and all diagnoses are 0 for this individual, we decided to remove this instance.

5. We repeated the process for the variable "DAY_HOSPITAL", but in ascending order.

```
--------------LESS DAYS IN HOSPITAL. 10 ELEMENTS--------------
         SEX EXITUS   AGE  DAYS_HOSPITAL  DAYS_ICU  TEMP  HEART_RATE  GLUCOSE
ID
66      MALE     NO  36.0            0.0       0.0   0.0         0.0      0.0
1002    MALE     NO  68.0            0.0       0.0  36.5        80.0      0.0
```

```
254      MALE      NO  47.0            0.0        0.0   0.0          0.0      0.0
1003   FEMALE      NO  68.0            0.0        0.0   0.0        103.0      0.0
606    FEMALE     YES  59.0            0.0        0.0  36.2        110.0      0.0
1407   FEMALE     YES  77.0            0.0        0.0  35.5         67.0      0.0
1051   FEMALE      NO  69.0            0.0        0.0  37.5        114.0      0.0
1225   FEMALE      NO  74.0            0.0        0.0  36.5         88.0      0.0
941    FEMALE      NO  66.0            0.0        0.0   0.0          0.0      0.0
1748   FEMALE     YES  98.0            0.0        0.0  36.4         78.0      0.0

      SAT_O2  BLOOD_PRES_SYS  BLOOD_PRES_DIAS
ID
66       0.0             0.0              0.0
1002    98.0           184.0             84.0
254      0.0             0.0              0.0
1003    70.0           132.0             81.0
606     96.0            60.0             40.0
1407    93.0           142.0             93.0
1051    95.0           143.0             83.0
1225    84.0           150.0             70.0
941     40.0             0.0              0.0
1748    50.0           137.0             54.0
```

We can see that many of the instances with a zero value also have all diagnosis equal to 0. For this reason we decided to look at these rows and see if there was a clear pattern for variable "EXITUS":

```
--------------EXITUS INFO ABOUT ELEMENTS WITH ALL DIAGNOSIS = 0--------------
count      361
unique       2
top         NO
freq       299
Name: EXITUS, dtype: object
```

We see that there is not a clear pattern for the target variable. However, we were not able to impute these zero values because we would add more noise than information to the data set. We decided to remove all these rows.

6. We moved on to "HEART_RATE":

```
--------------HIGHEST HEART RATE. 10 ELEMENTS--------------
         SEX EXITUS   AGE  DAYS_HOSPITAL  DAYS_ICU  TEMP  HEART_RATE  GLUCOSE
ID
186      MALE      NO  44.0            3.0        0.0  36.1        593.0      0.0
2052   FEMALE      NO  68.0            6.0        6.0  36.8        190.0      0.0
1542   FEMALE     YES  98.0            1.0        0.0  38.1        170.0      0.0
1412   FEMALE     YES  77.0            2.0        0.0  37.2        167.0      0.0
1396   FEMALE     YES  77.0            1.0        0.0  36.8        156.0      0.0
1249   FEMALE      NO  74.0            1.0        0.0   0.0        150.0      0.0
280    FEMALE     YES  48.0            0.0        0.0   0.0        145.0      0.0
780      MALE      NO  63.0            6.0        0.0  36.5        145.0      0.0
435      MALE      NO  54.0            7.0        0.0  38.3        143.0      0.0
1176   FEMALE      NO  72.0           17.0        0.0  36.8        140.0      0.0

      SAT_O2  BLOOD_PRES_SYS  BLOOD_PRES_DIAS
ID
186     97.0           136.0             88.0
2052    98.0             0.0              0.0
1542    95.0           126.0             70.0
1412     0.0            95.0             63.0
1396    98.0           135.0             80.0
1249    99.0           163.0             81.0
280     88.0            80.0             39.0
780     89.0             0.0              0.0
435     90.0           143.0             89.0
1176    80.0           112.0             71.0
```

There was one instance with 593 heart rate value. We decided to remove it.

7. We repeated the process for variable "BLOOD_PRES_SYS":

```
--------------HIGHEST BLOOD_PRES_SYS. 10 ELEMENTS--------------
         SEX EXITUS   AGE  DAYS_HOSPITAL  DAYS_ICU  TEMP  HEART_RATE  GLUCOSE
ID
23      MALE     NO  27.0            2.0       0.0  36.3        76.0      0.0
1892  FEMALE    YES  98.0            2.0       0.0   0.0         0.0      0.0
1240    MALE     NO  74.0           11.0       0.0  36.6       107.0      0.0
1950  FEMALE     NO  98.0            3.0       0.0   0.0        98.0      0.0
1850    MALE     NO  98.0            6.0       0.0  37.4        73.0      0.0
1716    MALE     NO  98.0            9.0       0.0  38.6        80.0      0.0
1731  FEMALE     NO  98.0            7.0       0.0   0.0       108.0      0.0
1346    MALE     NO  77.0            1.0       0.0  36.7        70.0      0.0
563   FEMALE     NO  57.0            7.0       0.0  37.6       103.0      0.0
1675  FEMALE     NO  98.0            5.0       0.0  36.6       102.0      0.0

      SAT_O2  BLOOD_PRES_SYS  BLOOD_PRES_DIAS
ID
23      99.0           772.0             90.0
1892    93.0           199.0             90.0
1240    88.0           198.0             86.0
1950    80.0           196.0             88.0
1850    98.0           196.0             89.0
1716    95.0           195.0             74.0
1731    85.0           193.0             94.0
1346    98.0           192.0             91.0
563     96.0           191.0             92.0
1675    99.0           190.0             87.0
```

Again, one instance had a value of 772, which must be an error. We removed it.

8. After that, we studied "BLOOD_PRES_DIAS":

```
--------------HIGHEST_BLOOD_PRES_DIAS. 10 ELEMENTS--------------
         SEX EXITUS   AGE  DAYS_HOSPITAL  DAYS_ICU  TEMP  HEART_RATE  GLUCOSE
ID
1798  FEMALE     NO  98.0           15.0       0.0  36.1        85.0      0.0
196     MALE     NO  45.0            6.0       0.0  37.7        99.0      0.0
1728    MALE    YES  98.0            3.0       0.0  35.0       119.0      0.0
1755  FEMALE     NO  98.0            2.0       0.0  36.8        76.0      0.0
831     MALE     NO  64.0           12.0       0.0  36.7       121.0      0.0
42      MALE     NO  32.0            2.0       0.0  36.8        95.0      0.0
1912  FEMALE     NO  98.0            5.0       0.0  37.2       100.0      0.0
1534  FEMALE     NO  98.0           13.0       0.0  36.9        76.0      0.0
159     MALE     NO  43.0            1.0       0.0  36.1       115.0      0.0
352     MALE     NO  51.0            9.0       0.0  37.0        83.0      0.0

      SAT_O2  BLOOD_PRES_SYS  BLOOD_PRES_DIAS
ID
1798    96.0           166.0            845.0
196     94.0           108.0            741.0
1728    74.0           145.0            127.0
1755    95.0           150.0            120.0
831     80.0           173.0            114.0
42      95.0           160.0            110.0
1912    90.0           170.0            110.0
1534    98.0           183.0            109.0
159     94.0           166.0            109.0
352     93.0           150.0            108.0
```

There was one instance with a value of 845 and another with 741. We removed them too.

9. Finally, we checked the amount of zero values each column had:

```
--------------NUMBER OF VALUES EQUAL TO ZERO--------------
SEX                 0
EXITUS              0
AGE                 0
DAYS_HOSPITAL      14
DAYS_ICU         1575
TEMP               81
HEART_RATE         59
GLUCOSE          1628
```

```
SAT_O2              36
BLOOD_PRES_SYS     364
BLOOD_PRES_DIAS    364
dtype: int64

--------------DATAFRAME SHAPE--------------
(1647, 11)
```

We can see that 1628 observations out of 1647 of "GLUCOSE" were zero. For that reason, we removed this variable. We can see there were still many zeroes in "BLOOD_PRES_SYS" and "BLOOD_PRES_DIAS", which may be a problem.

2.3 Exploratory analysis

As we did in Section 2.2, we will go step by step with the exploratory analysis we performed.

1. First, we show an histogram for the variables related with diagnosis in Figure 1. As we stated previously, there are some zero values in some variables, especially in those related with blood preasure.

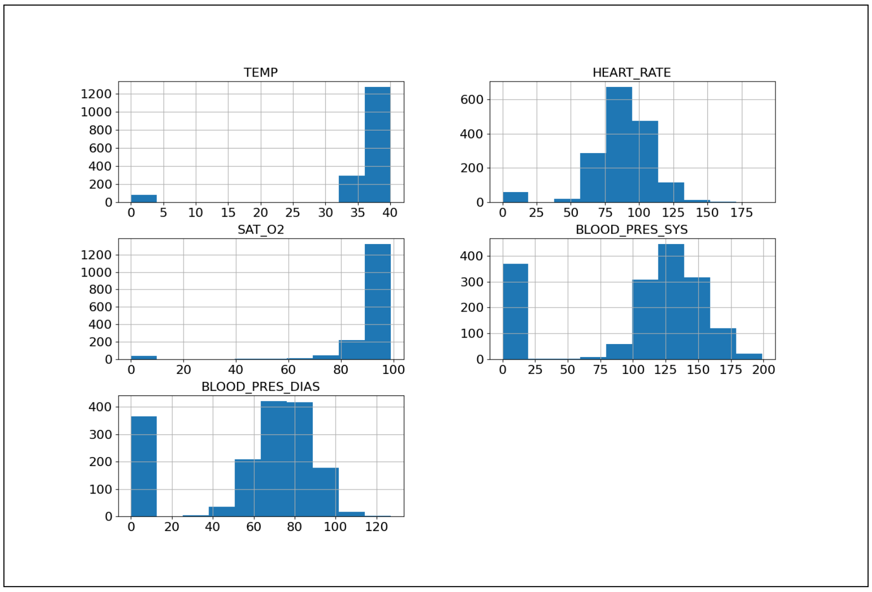

Figure 1: Histograms for diagnosis variables.

2. Then, we got the histogram shown in Figure 2. It divides the results by sex. We can observe there are slightly more men than women in our data set, but the proportions for variable "EXITUS" are almost the same.

3. Other interesting information is provided by box plots. Figures 3 and 4 show different results for individuals with value "YES" or "NO" for variable "EXITUS". We can see how individuals who died present older ages and worse oxygen saturation values.

4. Figure 5 shows the distribution for variable "AGE". The proportion of success change dramatically from the age of eighty in advance.

2.4 Correlation analysis

After the exploratory analysis performed in the previous section, we did a correlation analysis. Figure 6 shows a correlation heatmap for our data set. We can see that most of the variables are uncorrelated. The most outstanding information the heatmap gives us is the following:

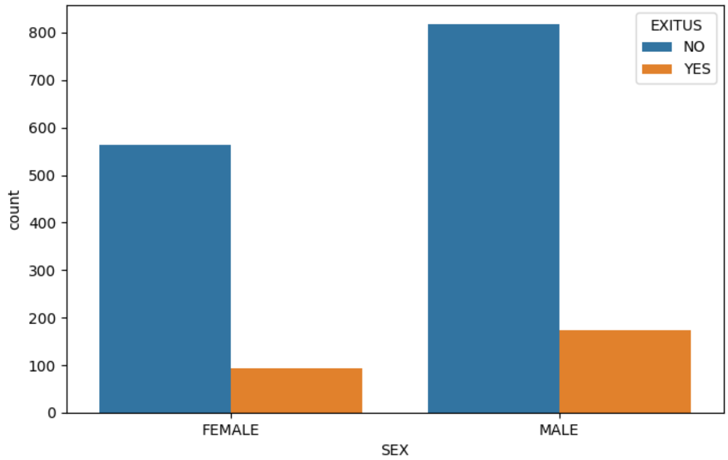

Figure 2: "EXITUS" per sex.

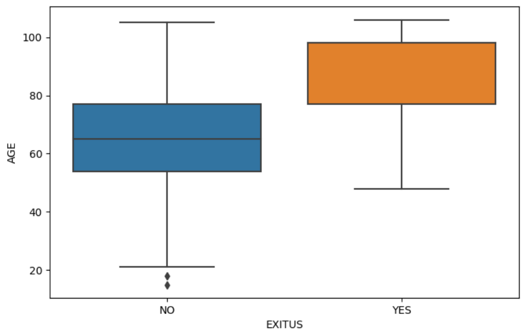

Figure 3: Box plot for "AGE".

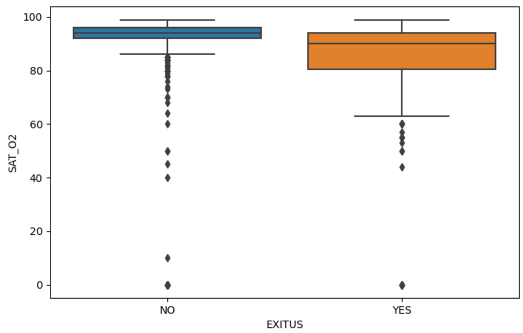

Figure 4: Box plot for "SAT_O2".

- Systolic blood pressure and diastolic blood pressure are highly correlated. Even though we are not doctors, it seems to make sense.
- Small positive correlation between "EXITUS" and "AGE". In Section 2.2 we established value 1 for "EXITUS" value equal to "YES", so age is related with mortality. It looks intuitive and agrees with the results obtained in Section 2.3.
- Small negative correlation between "EXITUS" and "SAT_O2". Bad oxygen saturation is slightly related with mortality, as we could suspect.
- Small positive correlation between "DAYS_HOSPITAL" and "DAYS_ICU".

2.5 Survival curves

The last task of the data analysis part had to do with getting survival curves from the data. Figure 7 shows a Kaplan-Meier plot [1]. We can see how the probability of survival decreases as an individual stays more days in hospital. This is something intuitive, but, as we were not able to get any more information from it, we decided to divide the survival curves in two: one for those individuals that get ICU treatment and another one for those without it. The result is shown in Figure 8. We can see now how ICU increases the probability of survival a little bit.

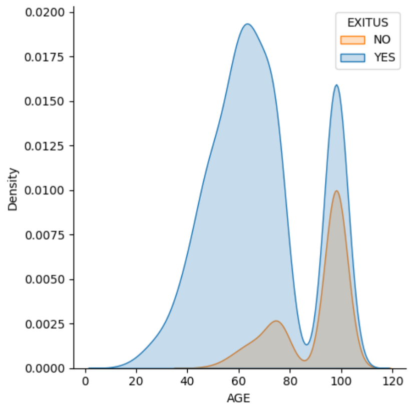

Figure 5: Distribution for variable “AGE”.

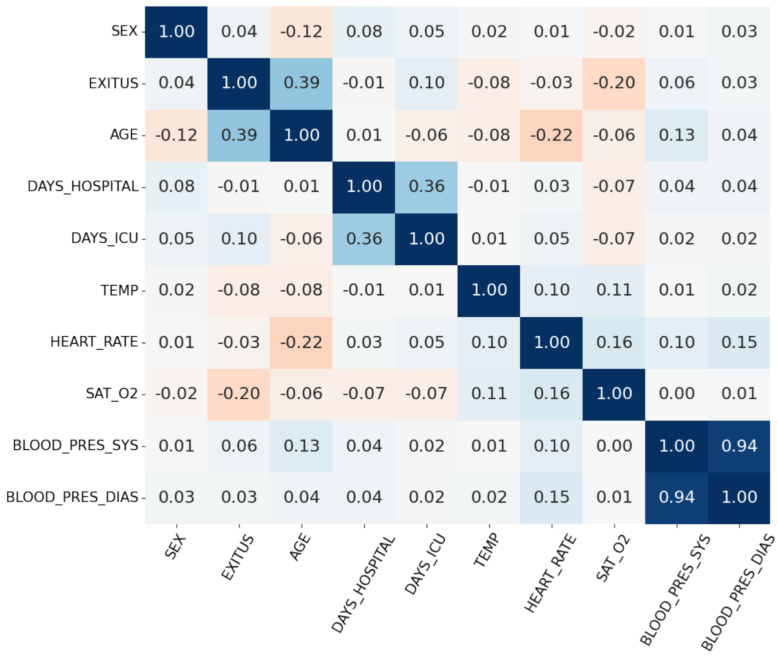

Figure 6: Correlation heatmap

3 Prediction models

In the KNIME workflows below, some common nodes, such as CSV Reader, Rule Engine, Normalizer, X-Partitioner, X-Aggregate, and Scorer, were used for the different problems. Firstly, CSV reader node was used to read the data set. After that, Rule Engine node was used to change the type of the variables. SEX variable was changed from nominal to numeric the same way we did in Section 2.2 and the “DAYS_ICU” variable was changed from numeric to nominal only for the “DAYS_ICU” prediction problem (Section 3.3). Normalizer node was used to normalize the data between 0 and 1 for MLP model. For logistic regression, Z-score normalization was applied. The size of the data set is not too big, therefore, cross validation technique was selected for the partitioning with fold = 10. Nodes called X-Partitioner and X-Aggregator were used to apply cross validation. For regression problems, Numeric Scorer node was used and for classification problems, the Scorer node. The other prediction algorithm specific nodes, learner and predictors, were selected according to the problem.

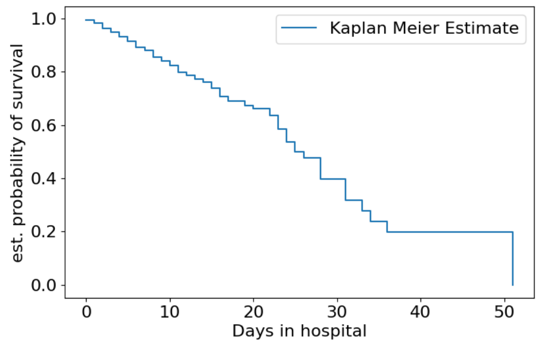

Figure 7: Survival curve.

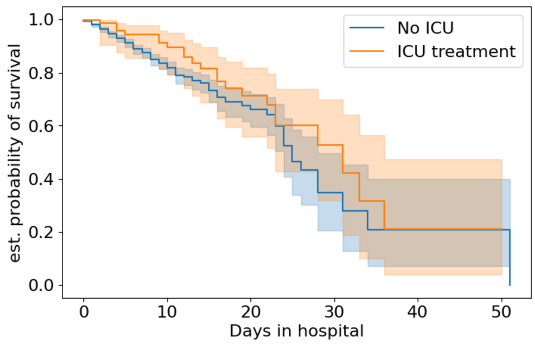

Figure 8: Survival curves per treatment.

3.1 Predicting "EXITUS" in patients

One of the goals of the project was to predict the value of the "EXITUS" variable using some given data. Multilayer perceptron, logistic regression, and decision tree methods were used to classify the patients. For the MLP workflow and logistic regression workflow, Normalizer node was used. Corresponding Learner and Predictor node were used for each workflow by choosing the predicting column. Figures 9, 13 and 14 show the workflow of each model.

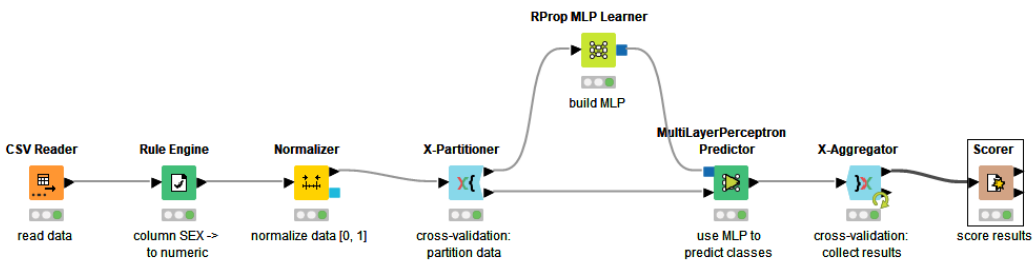

Figure 9: Workflow of the Multilayer Perceptron model.

Figure 10 shows the confusion matrix for the MLP model. As it can be seen, the model predicts 182 records as "No" while they need to be "Yes". On the other hand, Figure 11 shows the confusion matrix for the decision tree model. It has relatively similar numbers of false negatives and false positives, and both are higher than the records that are correctly classified as "Yes". Finally, Figure 12 shows the confusion matrix for the logistic regression model. Similarly to the MLP model, the number of false negatives is very high.

		Predicted	
		No	Yes
Actual	No	1332	48
	Yes	182	85

Figure 10: Confusion matrix of the MLP model.

		Predicted	
		No	Yes
Actual	No	1258	122
	Yes	148	119

Figure 11: Confusion matrix of the decision tree model.

		Predicted	
		No	Yes
Actual	No	1348	32
	Yes	204	63

Figure 12: Confusion matrix of the Logistic Regression model.

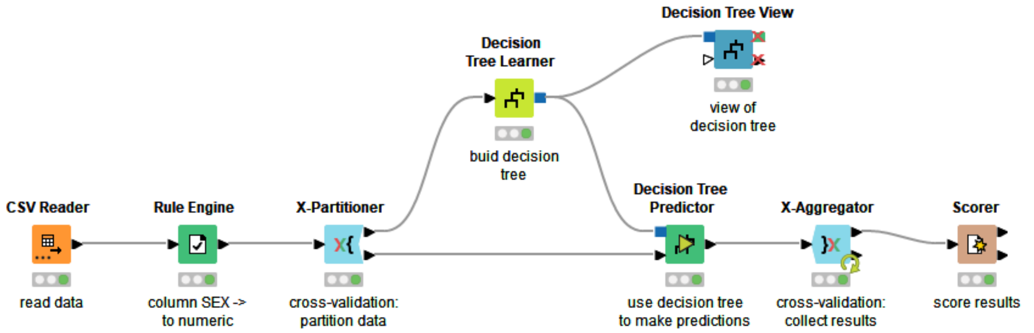

Figure 13: Workflow of Decision Tree.

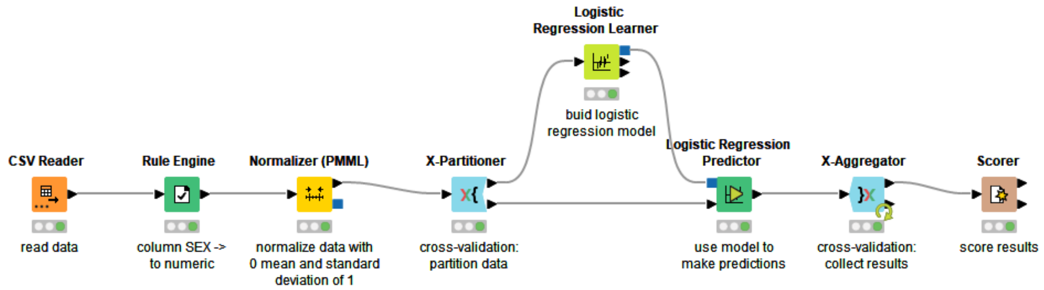

Figure 14: Workflow of Logistic Regression model.

In Table 2, the evaluation measures of the three different algorithm mentioned above are shown. While highest accuracy was obtained by using logistic regression, the best recall ("YES") was obtained with the decision tree model, which means the decision tree has better performance on predicting who will die due to COVID-19.

Measures	MLP	Decision Tree	Logistic Regression
Accuracy	0.860	0.836	0.968
Recall (NO)	0.965	0.912	0.977
Recall (YES)	0.318	0.446	0.236
Precision (NO)	0.880	0.895	0.869
Precision (YES)	0.639	0.494	0.663
F1-measure (NO)	0.921	0.903	0.920
F1-measure (YES)	0.425	0.469	0.348
Cohen's kappa	0.356	0.372	0.287

Table 2: MLP, Decision Tree and Logistic Regression evaluation results for predicting "EXITUS".

3.2 Predicting the number of days in hospital

Another business goal set in the project plan was to predict the number of days that a patient will have to stay in hospital. Two different machine learning algorithms were used for this problem: multilayer perceptron and linear regression. Firstly, the variables that cannot be obtained at the moment that patient gets to the hospital, such as "EXITUS" and "DAYS_ICU", were not selected in the transformation tab of the configuration window. Then, the common nodes mentioned before were used to change the type of the variables, partitioning and to get the score results. For MLP model, RProp MLP Learner node was used by selecting the class variable "DAYS_HOSPITAL", and MultiLayerPerceptron Predictor node to get the predictions. Another workflow for the same problem was created by using the Linear Regression Learner and Regression Predictor. To see the results, Numeric Scorer was used by selecting the corresponding columns as target columns or prediction column.

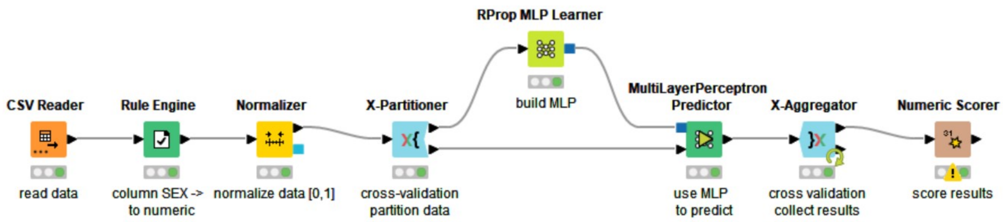

Figure 15: Workflow of Multilayer Perception - Days in hospital model.

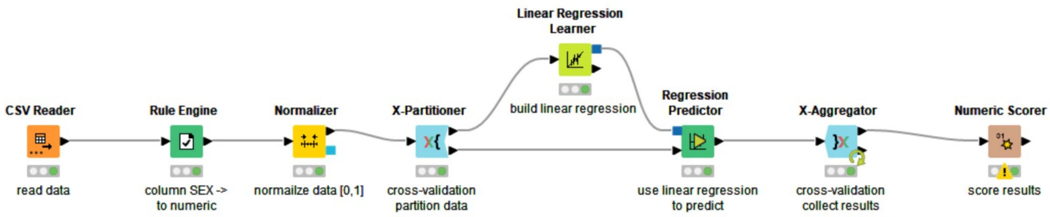

Figure 16: Workflow of Linear Regression - Days in hospital model.

Measures	MLP	Linear Regression
R^2	0.019	0.004
MAE	0.077	0.078
MSE	0.012	0.012

Table 3: Evaluation results for days in hospital.

3.3 Predicting ICU

The last goal we are trying to achieve with prediction models has to do with predicting if the patient will have to be placed in the ICU or not. For this problem, "DAYS_HOSPITAL" and "EXITUS" variables were removed by using CSV Reader. Rule Engine was used to change the type of the target column, "DAYS_ICU". We converted "DAYS_ICU" values bigger than 0 to the label "YES" and those equal to 0, to the label "NO". A decision tree algorithm was used for this classification problem.

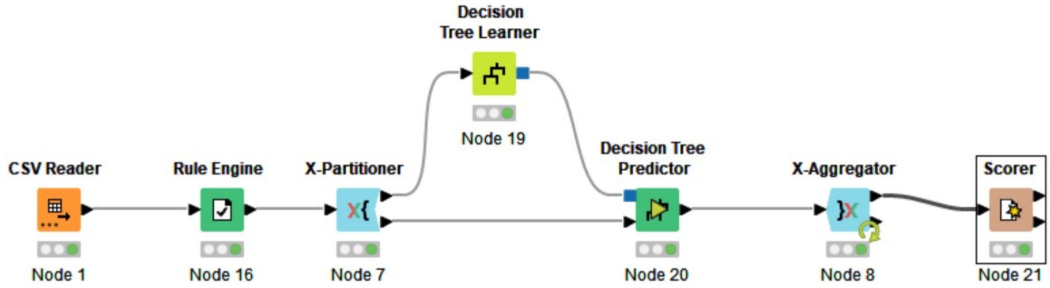

Figure 17: Workflow of Decision Tree (ICU prediction).

		Predicted	
		No	Yes
Actual	No	1533	42
	Yes	68	4

Table 4: Confusion matrix of the Decision Tree (ICU prediction).

Table 5 shows the evaluation results of the "DAYS_ICU" decision tree model. As it can be seen in the table, the algorithm works well on detecting the "NO" class, because the data set is unbalanced and it has many records with label "NO". On the other hand, the recall of "YES" is too low which means detecting the patient who will have ICU treatment is done poorly.

Measures	Decision Tree
Accuracy	0.933
Recall(NO)	0.973
Recall(YES)	0.056
Precision(NO)	0.958
Precision(YES)	0.087
F1-measure(NO)	0.965
F1-measure(YES)	0.068
Cohen's kappa	0.035

Table 5: Decision Tree evaluation results for predicting DAYS_ICU.

4 Conclusions

After all the work done, we obtained some useful conclusions. They can be divided the same way this report is divided: data analysis conclusions and prediction models conclusions.

Data analysis shows us how some variables are not particularly useful, because they have so many empty or zero values or they are uncorrelated with our target variable. However, in Sections 2.3 and 2.4 we saw that age and oxygen saturation keep some relation with survival. Moreover, in Section 2.5, we were able to get Kaplan-Meier survival curves for our data and we saw how ICU treatment makes individuals survive longer.

For the prediction models, the first thing we can say is that the data set is quite unbalanced. Therefore, it makes the prediction of two classes more difficult. Techniques for balancing data can be used to improve the results. The SMOTE node from KNIME was used but the results did not improve. As the data set was simulated, that is, it is not real, it does not make any sense to create more fake data. Another way to get more data is by asking to these or other hospitals to supply more records of the minority class. A different solution was thought for balancing the data, deleting rows of the majority class until classes are balanced but by doing that, we will obtain an extremely small data set, which is not good.

Also, the main goal depends on many other features of the human body which are not in the data set, for example, breathing problems, heart diseases, etc. By expanding the data set including new useful variables, the results could be significantly improved.

References

[1] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American Statistical Association*, vol. 53, no. 282, pp. 457–481, 1958. DOI: 10.1080/01621459.1958.10501452. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1958.10501452. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1958.10501452.

[2] M. L. Waskom, "Seaborn: Statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021. DOI: 10.21105/joss.03021. [Online]. Available: https://doi.org/10.21105/joss.03021.

[3] The pandas development team, *Pandas-dev/pandas: Pandas 1.3.3*, version v1.3.3, Sep. 2021. DOI: 10.5281/zenodo.5501881. [Online]. Available: https://doi.org/10.5281/zenodo.5501881.

[4] T. A. Caswell, M. Droettboom, A. Lee, *et al.*, *Matplotlib/matplotlib: Rel: V3.4.3*, version v3.4.3, Aug. 2021. DOI: 10.5281/zenodo.5194481. [Online]. Available: https://doi.org/10.5281/zenodo.5194481.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2.

[7] C. Davidson-Pilon, "Lifelines: Survival analysis in python," *Journal of Open Source Software*, vol. 4, no. 40, p. 1317, 2019.

[8] C. Bielza Lozoya and P. Larrañaga Múgica, *Data-driven computational neuroscience : machine learning and statistical models*, eng. Cambridge: Cambridge University Press, 2020, ISBN: 978-1-108-49370-3.