Project report for the Data Science Course

Project 10: Predicting mortality for
COVID-19 patients [COVID]

Primary Topic: DM, Secondary Topic: TS
Course: 2020-1B – Group: 83 – Submission Date: 2021-01-10

Arwan Credoz
University of Twente
a.g.j.s.g.credoz@student.utwente.nl

Lucas Trabuc
University of Twente
l.y.trabuc@student.utwente.nl

ABSTRACT

This paper aims at training a binary classifier which predicts the mortality of patients tested positive with COVID-19. Other researchers attempted the same task, fitting a logistic regression model using data from what they considered to be the best 3 features from the patients' reports. However, we use a different dataset containing more features and set our goal to predict as early as possible, with the highest accuracy, the outcome of each patient. Through our choice of data pre-processing techniques, feature selection and model hyper-parameter tuning, we obtain a logistic regression model capable of predicting on average 10 days and 17 hours beforehand with an average accuracy of 81% the outcome of a patient. Beside its accuracy, the amount of false-negative predictions, which correspond to patients predicted as surviving but who actually succumb, is low to none. This comforts us in the fact that this model could be used by actual medical personnel without risking misdiagnosing a patient as not in danger.

KEYWORDS

Machine Learning | COVID-19 | Mortality Prediction

1 INTRODUCTION

The COVID-19 pandemic of 2020 is one of the biggest sanitary disasters of the last decades. It has been especially challenging for the medical personnel. Because of the recent nature of this event, it is difficult to establish a precise diagnosis on the health condition of the contaminated patient. This study aims at creating a tool to predict the mortality of patients using machine learning models trained on blood sample biomarkers. Such a classifier could allow medical staff to foresee a patient's situation and better allocate their time and resources to attend certain patients before they enter a critical state. The dataset consists of reports of 375 patients with 75 biomarker measurements, collected between January 10th, 2020 and February 18th, 2020 at the Tongji Hospital in Wuhan, China. Every patient has a varying amount of reports from their admission time to their discharge, showing the evolution of their health status.

In order to achieve our goal we go through several steps starting with data pre-processing with the aim of dealing with missing values and standardization. Secondly, since the dataset contains 75 features, another task will be focused on the feature selection to determine which features are the most significant and helpful in order to make the predictions. In addition to the feature selection step, we also decide how to constitute our training and testing datasets. After splitting the whole data into a training and testing set, there are multiple ways of modifying our training set. For example, we could treat each report as an input, or only select the latest report values for each patient. As for the testing set, since we want to make a classifier that foresees an outcome as early as possible, we keep the first report of each patient only (see Figure 3). Next, we implement several machine learning models on the selected features of the pre-processed data. Our objective is to evaluate the performances of these models on this classification problem and identify the best fitting one, leading to the best accuracy as well as earliest prediction, while manipulating different types of datasets, models and hyperparameters.

2 RELATED WORKS

In Yan et al. (2020) [1] as well as in Zhou et al. (2020) [2], only three biomarker features have been used: lactic dehydrogenase (LDH), lymphocyte and high-sensitivity C-reactive protein (hs-CRP). Other models have already shown great results at predicting the mortality rate of COVID-19 patients, such as the ones from Yan et al. (2020) [1], which managed to reach 90% accuracy at predicting the mortality of individual patients more than ten days in advance of their passing or their discharge, using the dataset containing only 3 features.

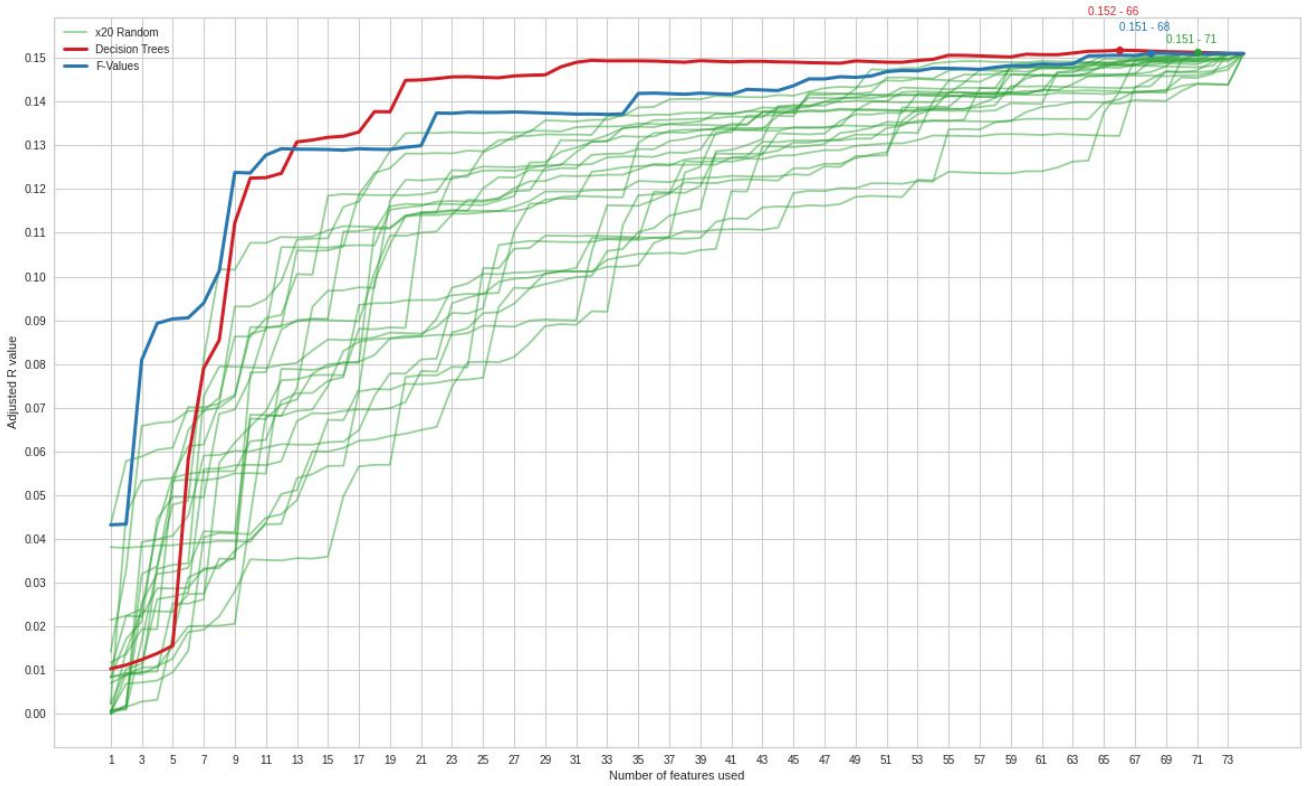


Figure 1-a. A comparison of the evolution of the adjusted R-squared value from a regression model with different sequences of features.

We want to determine which features are the most significant for the classification amongst the 75 available, to maximize the chance of higher accuracy. For this, different criteria exist, such as F-values and Gini index. The first one is used in a one-way ANOVA, the second one in randomized decision trees. As discussed in the next parts, this feature selection step will lead us to use different biomarkers than [1] and [2].

3 APPROACH

About the data

The data includes a sheet of 6120 reports, including 375 patients, each report containing the report time, admission time, discharge time, outcome and list of biomarker values. It is to be noted that only a few biomarker values are present in each report. Indeed, most of the features have a majority of null values, which goes up to more than 95% of null values for the “Interleukin 10” feature. In total, the dataset consists of 85% of null values. It is depreciated to work with sparse data, as only a few machine learning algorithms work with null values amongst their inputs (e.g. Naive Bayes classifier), we will thus have to come up with a way of handling these missing values

Data Processing for feature selection

There are two follow ups to this step, a feature selection process and a machine learning process. This means that the data pre-processing will slightly differ for each process. Regarding the feature selection step, we do not need to split the data into a training and testing set, and can use all the data to find out which features are more relevant. The first task is to fill in missing values. As stated above, we can not afford to remove reports or features containing null values since all of them do. Therefore we decided to do mean imputation, by filling in every missing value by the mean of its corresponding feature over all reports. This method has the merit of being easy to implement and preserves the mean of the data. However, it is to be used only as a last resort as it can lead to a change in the relationship between variables as well as an underestimate of the standard deviations errors.

Regarding the machine learning process, it is crucial to split the data into a training and testing set **before** doing

the standardization and mean imputation (not doing so can lead to a better yet unfair accuracy, or in other words, the model would overfit). Hence each set will be standardized separately and will have their own means added. Doing so helps to keep the testing dataset completely unrelated to the training one. Otherwise, the testing data would affect the training data and bias the fitting of our model. Splitting the dataset before pre-processing it therefore contributes to an optimal training of the future models.

Data standardization

In order to have data with values compatible with most machine learning algorithms as well as feature selection algorithms, we have to standardize it. We noticed by visualizing the data that some values reach over a 1000, whereas some features only range between 0 and 1. Hence, standardization is done for each feature by removing the mean and scaling to unit variance to remove the unbalanced effect that such disparities could add to the fitting of a regression classifier or multilayer perceptron classifier.

Features selection

First, we want to have an idea of the individual relevance of each feature. Several methods exist to achieve such a task. The first one we chose is the ANOVA method that computes the F-values for all features, allowing us to express the linear dependency between the features (Figure 4-a).

Another method of feature ranking is the random decision trees. They are useful to determine features' importance. Using impurity measurement based on the Gini Index, features that are able to create leaves with low impurity are the most discriminative and therefore the most useful to our model (Figure 4-b).

By doing an Ordinary Least Squared (OLS) regression, β_0 and β_1 are chosen in the linear equation $y = \beta_0 + \beta_1 x$ to minimize the square of the distance between the predicted values and the true values. We obtain a variety of fit-statistics, in particular the adjusted R-squared values (see Figures 5-1 and 5-b and Equations 1 and 2). We added features one by one, starting with the highest scoring ones and created and fitted a model every time, while storing the adjusted R-squared values. We also plotted the evolution of the adjusted R-squared values for randomly ordered features (20 random sequences) to display the efficiency of our previous feature selection methods (see Figure 1-a). As we can see, both the f-ANOVA and random decision tree plots indicate a maximum adjusted R-squared value when all the features are considered, meaning that no features are redundant or detrimental. That is why we used all the features for the classification models (see Figure 7 for features pairwise correlation).

Data Visualization

Before training our models, we have to make sure the data is balanced between the labels. An unbalanced dataset (a dataset where class ratios are not equal) can lead to a decrease in testing accuracy. As we can see in Figure 2, our data has fairly balanced classes. We will repeat this process on the training and testing dataset after splitting the data. However since their ratios are not completely equal, we will do a stratified cross-validation to keep this unbalance in each fold.

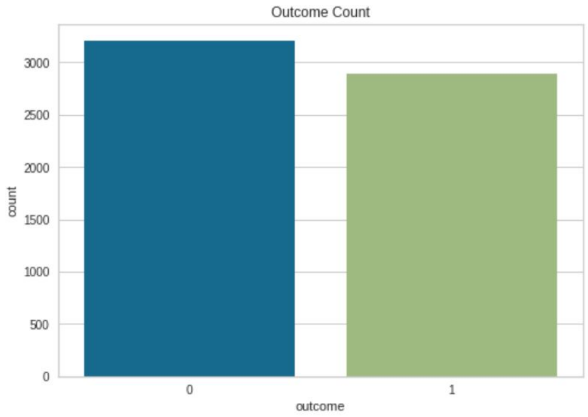


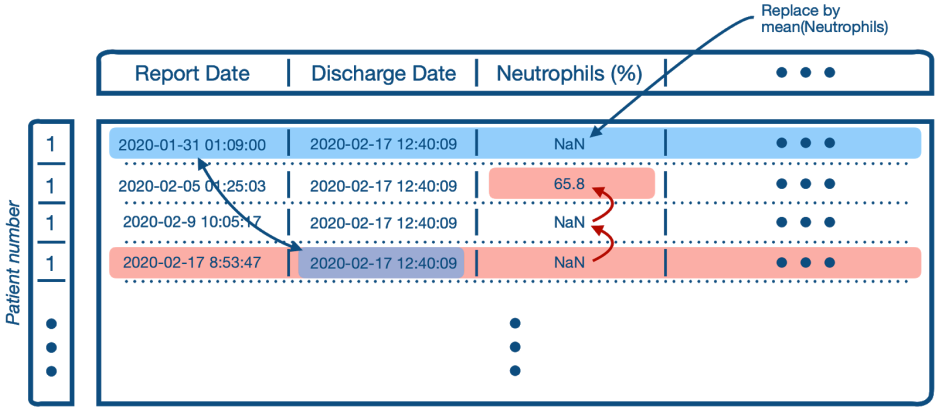
Figure 2. Data balance between the labels (the outcome '0' meaning the patient survived, and '1' succumbed)

Preprocessing for Machine Learning

Reports selection

If the choice of the learning algorithm is important, we also have to decide the samples we will feed to our models. In our case, we want to create a model able to predict if a patient is going to succumb from the covid or not with the highest accuracy. But we also want it to make the prediction as early as possible, so that the model would be more applicable in a real case scenario. In case of null values, we fill them using mean imputation as described in Figure 3.

The second crucial criteria, and probably the most important, is that we absolutely want to minimize the number of patients diagnosed as healthy by our model when they're actually going to succumb from the COVID, which corresponds in our case to the number of false



When considering the latest values (in red): we look at the last report of a patient in order to get the most recent value from each feature. In case of an "NaN" value, we will look at the previous reports (of the same patient) until finding a non-"NaN" value, and use this one (we can compare this to a medical practitioner having access to the previous reports of his patient). There are times where no values are found, in that case we impute the mean of the column values, all patients combined. Using this method on the testing set lets us predict the outcome of a patient with great accuracy (see Table 1-a and Table 1-b) but not ahead of time since we use the latest values.

This is another strategy (in blue). We only consider the features from the earliest report to make the earliest predictions. However since there are no previous reports from the same patient to look at in case of a "NaN" value, we impute the mean of all the columns where there's a missing value. When limiting our data intake by only looking at the first report, it allows us to predict several days in advance (interval between Report Date and Discharge Date), with more or less confidence, the outcome of patients. We eventually reach the best prediction accuracy using this technique on the training set when using the first reports in the testing set.

Figure 3. Diagram example of dataset transformation

positives (FP). Indeed, if we were to put our classifier in practice, the worst scenario would be that it told the medical staff that some patients will survive whereas they actually will succumb.

Therefore we decided to use the first report of each patient in the training phase as well as in the testing phase.

This is done in order to achieve the earliest predictions possible. This model resulted in the best performances (while testing with the first reports) compared to the other models presented in the Discussion section. Probably because of the similarity between the training data and testing data.

4 EXPERIMENTS

We decided to use the following machine learning models because of their high performance in classification tasks. We coupled this with a 5 folds cross-validation process in order to verify that their results are not due to particular splits of data. Models are compared on the mean accuracy returned from the cross-validation (see Figure 6), and the proportion of FP.

Machine Learning Models

Logistic Regression: With a squared L2 penalty as a regularization parameter tweaked to 0.025, the model reaches 81% average accuracy with 2% of FP.

Naive Bayes (Gaussian): The results are not satisfying as the model didn't reach 80% of accuracy for the test (72% on average), and the proportion of FP stayed relatively high, around 50%.

Decision Tree: Because of the high number of parameters, we decided to not put any depth limit. Both impurity criteria (Gini and Entropy) gave similar results, 77% average accuracy and 15% FP.

KNeighbors: The models showed better performances when increasing the number of neighbors to use as classification boundaries. Setting the threshold to 10

neighbors made it reach a 77% average accuracy for 12% FP.

Linear SVM: The model increased in performance by tweaking its generalization parameter, which is a squared L2 penalty. Setting it to 0.025 made the model reach 84% accuracy for 2% FP.

Neural Network (MLP): The model is composed of two fully connected layers of 10 neurons each. We settled on the *tanh* activation function and a learning rate of 10^4. It reaches 82% average accuracy and 2% of FP.

Hence, the Logistic Regression and the MLP network showed the best results, with an average accuracy **above 81%** (reaching **89%** on some splits), and a proportion of **FP between 0% and 2%** (see Figure 8). Also, because we use the earliest reports of the patients, our models make these predictions on average **10 days and 17 hours** in advance, or in other words, before the discharge date.

5 DISCUSSIONS

Our first approach concerning the data pre-processing was wrong. Indeed, we first did the pre-processing steps, consisting of standardization and filling the missing values with mean imputation, and then splitted the data between the training set and the testing set. However, as discussed in the "Approach" section concerning the data pre-processing, doing so induces the data in the training set to influence the data in the testing set, which is a case of data leakage.

Concerning the adjusted R-squared plots giving us information about feature importance. For the first plot in Figure 1-b, when we used the latest report for each patient, some features' values were abnormal (in reports where the patients were in a critical state) making them way more significant than other features. That is why the adjusted R-squared value maxed out when only few features were considered. However, now that we use the earlier reports as data, the new adjusted R-squared plot (see Figure 1-a) made us implement all the features (as the adjusted R-squared value was maximum when all the features were considered), because no feature is drastically more significant than the others in early reports, which makes the problem we want to solve harder.

Other datasets configurations we tried:

Train: Latest values / Test: Latest values

Our first model was taking as input the latest report for each patient, for the training and for the testing (see first method in Figure 3). This data contains more information than the earliest report, because it is when the condition of the patient worsens that the features become more significant. As a consequence, the model easily performed 100% accuracy on the test dataset. Such a model would not be useful since it is only applicable in very late situations. That is why we decided to use earlier reports for the next models, so that it can predict several days in advance, making it applicable to more interesting situations.

Train: First four reports / Test: First report

Similar to the previous model, but in this case we used the first four reports of each patient in the training phase, so that we could eventually collect more "real" data for each patient in the sense they were not imputed using mean imputation.

Train: All reports / Test: First report

Another model we tried, where we considered all the reports for the training set. This however has not resulted in good scores, probably because of the quantity of imputed data (85%) which made the majority of the data redundant.

Generalization parameter

For the SVM and the Logistic Regression, the squared L2 regularization parameter helped to reduce overfitting for both models. Without it, the Logistic Regression reaches 88% accuracy, but its number of False Positives increases (15%), which is something we wanted to avoid.

Limitations

The main flaw is the sparsity of the data. Because 85% of it has been filled by us using mean imputation, it reduces features' variance.
Also, the period of 10 days foreseen predictions is imposed by the data, which doesn't provide earlier reports to make it possible to predict earlier in time.

6 CONCLUSIONS

This paper compares the performances of different Machine Learning methods applied on a topical subject, COVID-19 mortality. Using a dataset consisting of biomarkers values, we showed that the logistic regression the multi layers perceptron managed to reach up to **89% accuracy** (82% accuracy on average using a stratified cross-validation) and **0% of False Positive** (inferior to 2% on average) on this dataset, making these predictions on average more than **10 days and 17 hours in advance**.

REFERENCES

[1] Yan et al. 2020. An interpretable mortality prediction model for COVID-19 patients. https://doi.org/10.1038/s42256-020-0180-7
[2] Zhou et al. 2020. Do not forget interaction: Predicting fatality of COVID-19 patients using logistic regression arXiv:2006.16942v

APPENDIX

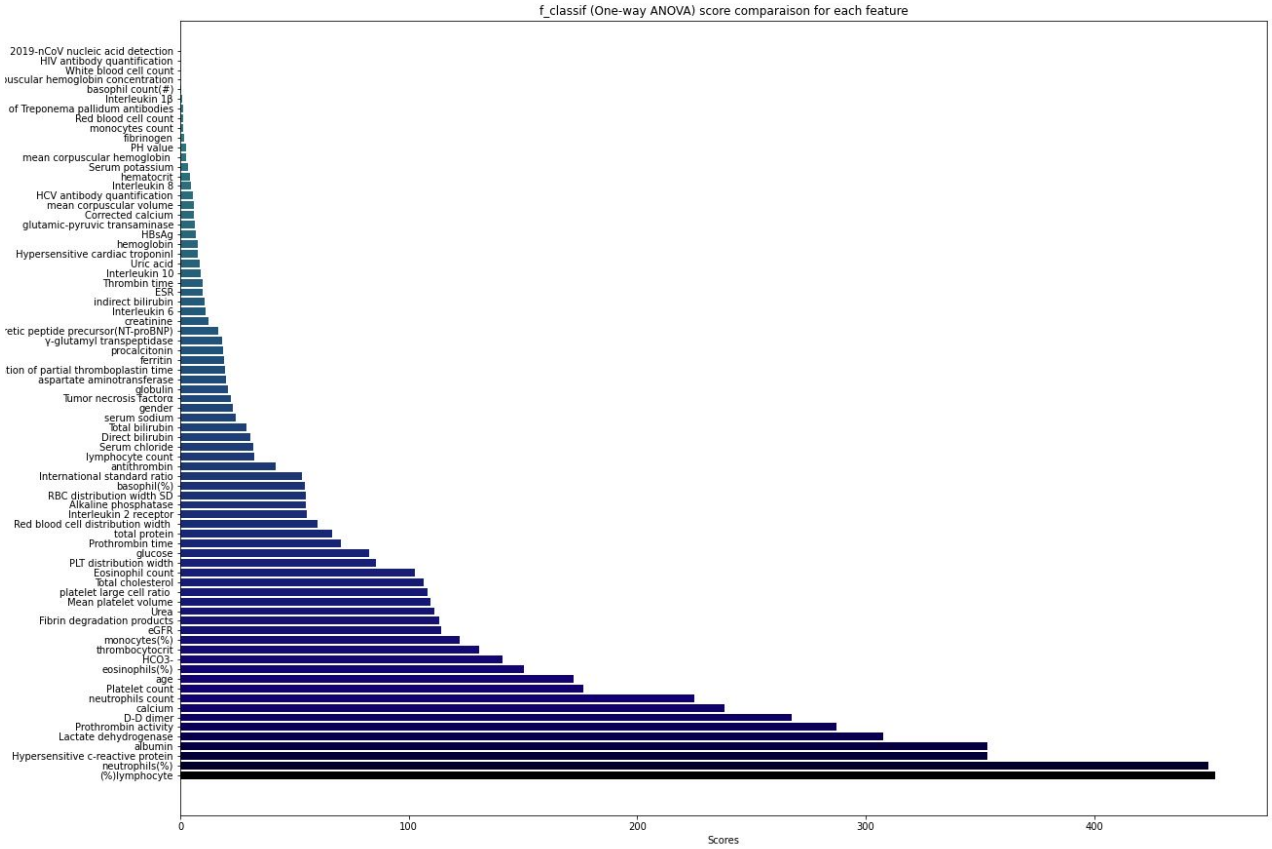


Figure 4-a. A ranking of the features based on one-way ANOVA. We rank the features from lowest to highest score based on a one-way ANOVA, used to discern statistically significant differences between the means of the features.

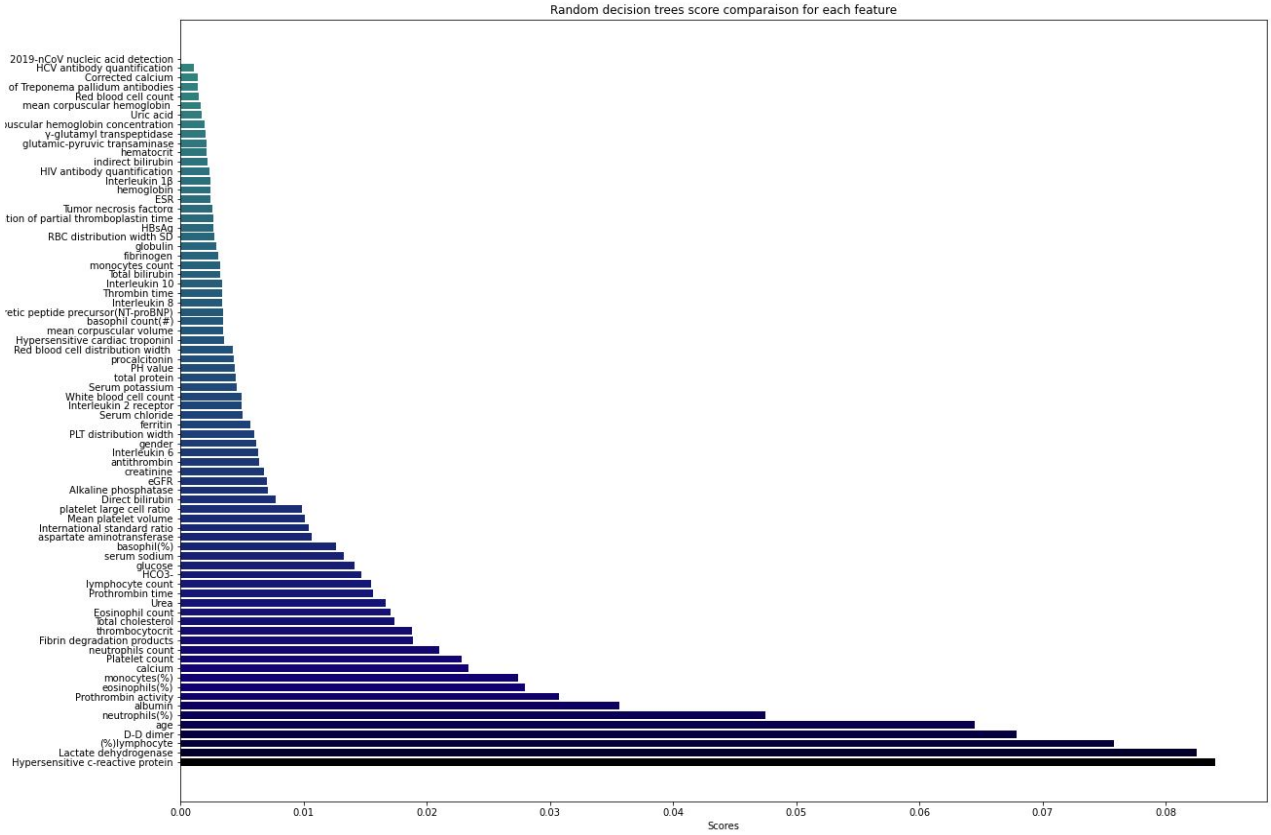


Figure 4-b. A ranking of the features based on 100 randomized decision trees. Here the features are ranked from lowest to highest score (using the Gini impurity criterion) based on the result of 100 randomized decision trees (a.k.a. extra-trees) fitted on various sub-samples of the dataset.

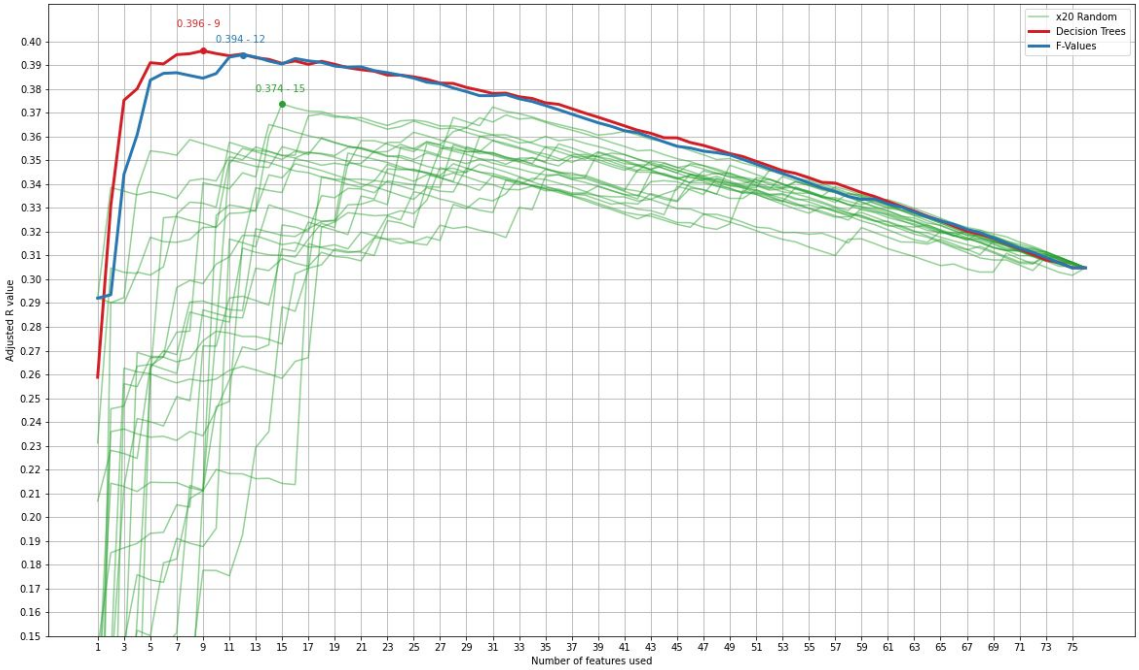


Figure 1-b. A comparison of the evolution of the adjusted R-squared value from a regression model with different sequences of features when using data from the latest report of each patient. We can see that unlike in Figure 1-a, this comparaison does show a clear benefit of choosing a limited amount of specific features. Here, we see that the highest adjusted R-Squared value is reached when using the first 9 features of the random decision trees selection (see Figure 4-b). Adding more features seems to only add in complexity without adding more beneficial information to the classifier. We can explain this by the fact that features are more distinct in the final reports: the health condition of a patient will have either stabilized or worsened at the end of his hospital stay, the feature values will now more clearly represent each outcome, therefore making it easier to spot useful features.

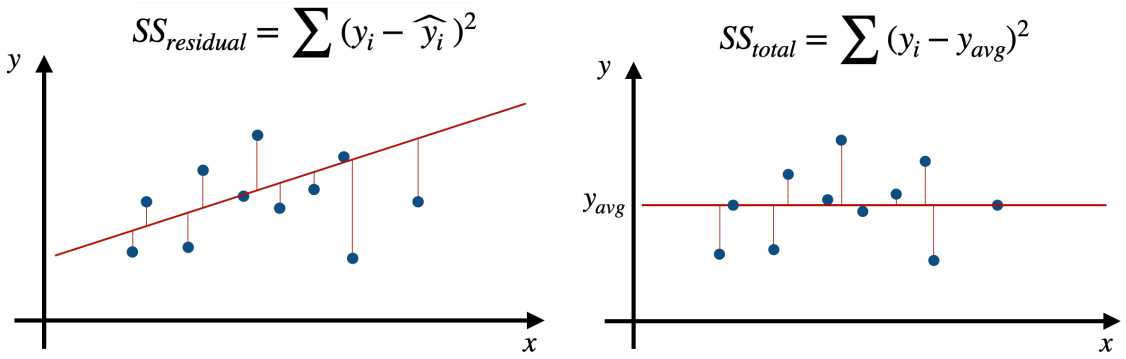


Figure 5-a. Diagram of Residual Sum of Squares ($SS_{residual}$) Figure 5-b. Diagram of Total Sum of Squares (SS_{total})

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} \quad (1)$$

$$R^2_{adjusted} = 1 - \frac{(1-R^2)(N-1)}{N-p-1} \quad (2)$$

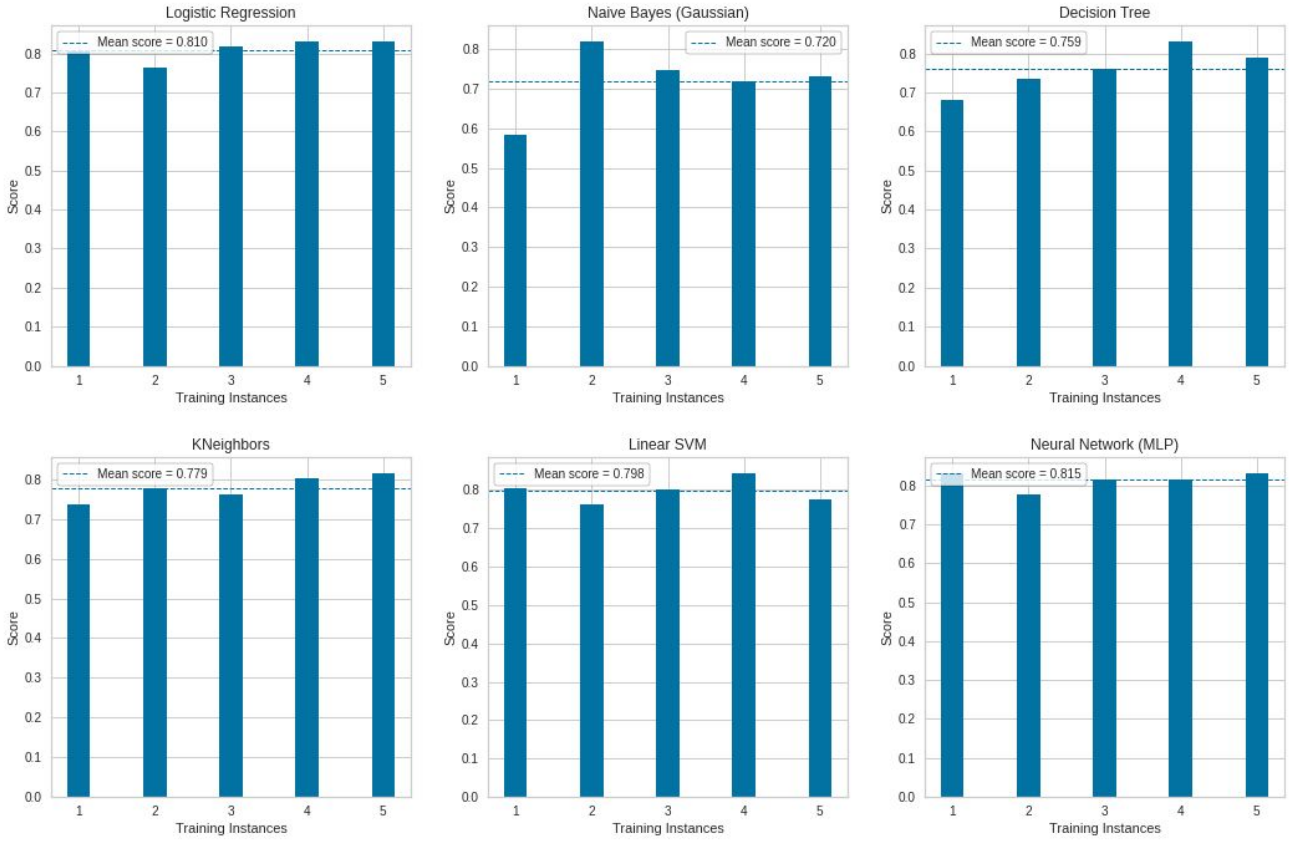


Figure 6. Cross-validation on Stratified K-Fold. This way the folds preserve the percentage of samples for each class. Since the data has a slight imbalance in class ratio, this way of analyzing results is more accurate.

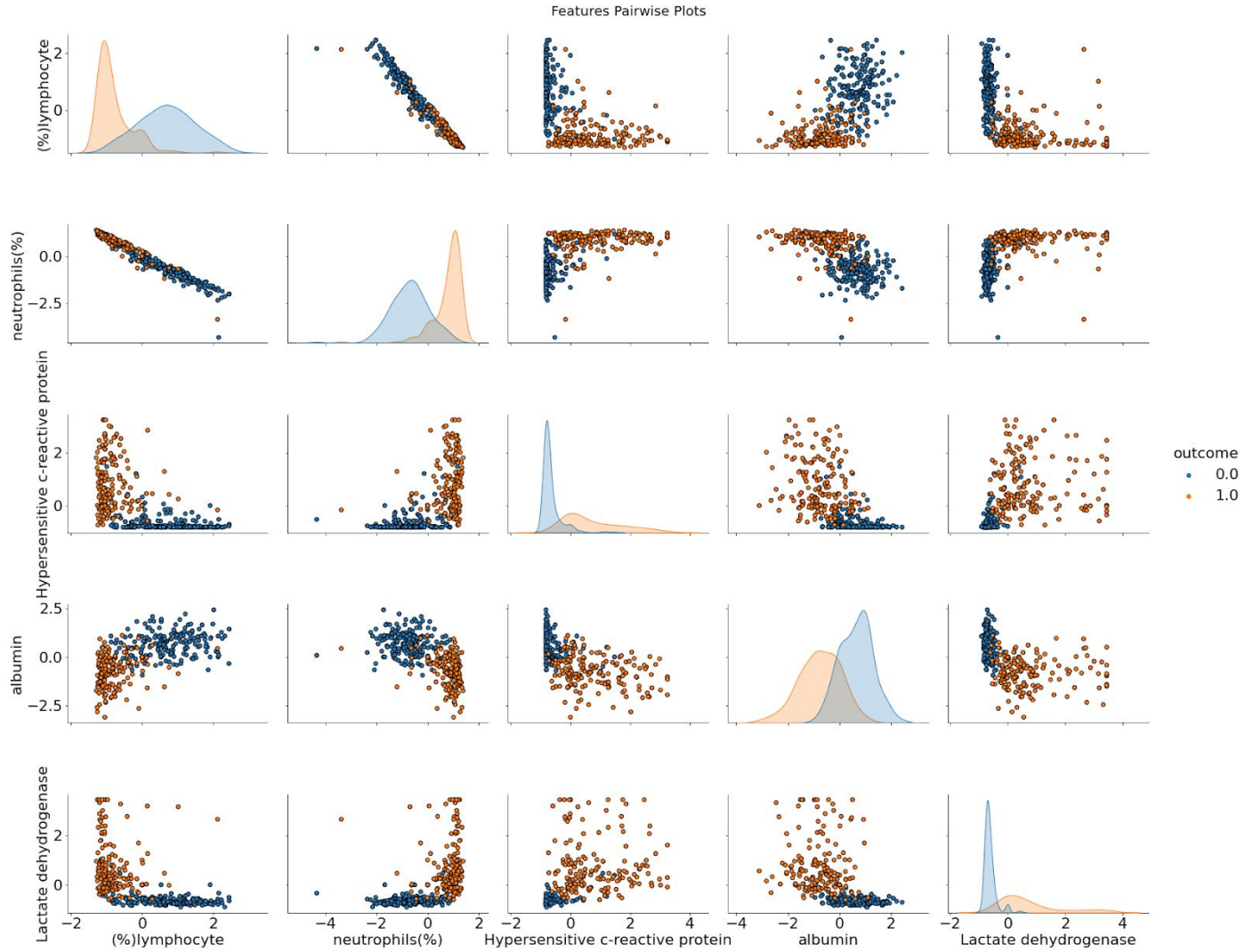


Figure 7. Plot of pairwise relationship between the top 5 features with distinguishable labels. The scatter plots show the correlation between two features as well as distinguishing classes. The plots on the diagonal are the univariate distributions. (outcome being 0 means the patient survives)

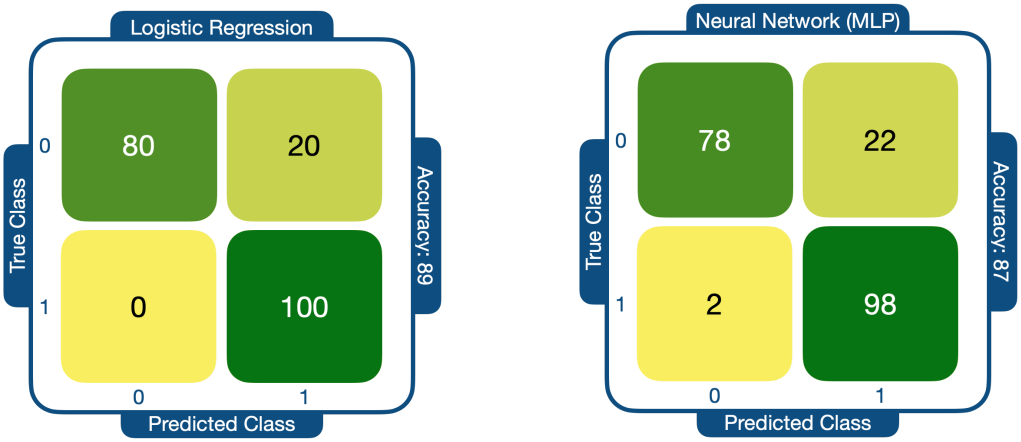


Figure 8. Confusion matrices for the Logistic Regression (left) and the MLP (right) models. (Using the first report in the training and testing sets)

Table 1-a. Classification report of logistic regression using all features to fit the model. Model fit with training and testing set using the latest values method (see Figure 3).

	Precision	Recall	f1-score	support
0 (alive)	0.94	0.96	0.95	50
1 (deceased)	0.95	0.93	0.94	44
accuracy			**0.94**	94
macro avg	0.95	0.95	0.95	94
weighted avg	0.95	0.95	0.95	94

Table 1-b. Classification report of logistic regression using top 9 features to fit the model (see Figure 4-b). Model fit with training and testing set using the latest values method (see Figure 3).

	Precision	Recall	f1-score	support
0 (alive)	1	0.98	0.99	51
1 (deceased)	0.98	1	0.99	43
accuracy			**0.99**	94
macro avg	0.99	0.99	0.99	94
weighted avg	0.99	0.99	0.99	94