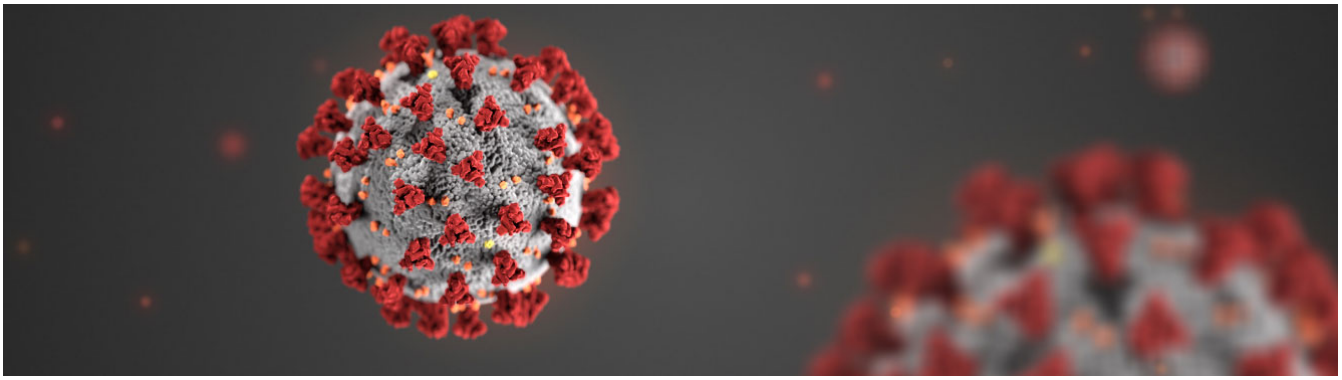
Report for the Data Science Course

Project Title: **Predicting mortality for COVID-19 patients [COVID]**
Primary Topic: DM, Secondary Topic: IENLP
Course: 2021-1B – Group: 63 – Submission Date: 2022-02-06

Sonia Litwin
s.litwin@student.utwente.nl

Zuzanna Posłuszna
z.m.posluszna@student.utwente.nl



ABSTRACT

For more than two years, the world has been struggling with a coronavirus pandemic that is taking millions of lives [1]. As a result of the pandemic and the high infection rate, it has been necessary to identify high-risk patients in advance. However, an effective method to accurately predict mortality among high-risk patients has not been invented. Research made by Yan et al. (2020) and Zhou, Chen and Lei (2020) shows that we can predict mortality of COVID-19 patients with more than 90% accuracy and also 10 days before. They also emphasize that this can be done with only three biomarkers: lactic dehydrogenase (LDH), lymphocytes, and high-sensitivity C-reactive protein (hs-CRP). [2]. Therefore, a biomarker-based mortality prediction model for COVID-19 patients and machine learning models were created based on the available data and compared based on the applicable performance measures. The best one was selected.

KEYWORDS

COVID19, pandemic, machine learning, prediction models, death, mortality, patients, biomarkers.

1. INTRODUCTION

The emergence of a severe form of atypical pneumonia in December 2019 surprised not only China but also the world. A hectic race began between scientists, who began to search for the causative agent of this mysterious disease and ways to combat it, and the growing epidemic in the countries. On February 11, the disease was named COVID19 and three days later the pathogen was named SARSCoV2 [3]. COVID19 has become the biggest health problem in the world. Month after month, the number of deaths increased. We also experienced each successive wave and variant of coronavirus, which was associated with successive huge numbers of deaths. The world is facing many clinical problems and high mortality rates. Therefore, predicting mortality and identifying predictors is an important aspect for patients with COVID19 [4]. Idea to solve this problem is to execute experiments on the data set provided with the use of different DM techniques and preprocessing methods. Created machine learning models could be a precursors for harder lockdown or support vaccine distribution strategies, as well as help to separate high-risk COVID-19 patients. Models made it possible to answer the research question: Which of the created machine learning models fits the best in prediction model for the mortality of Covid-19 patients, based on biomarkers?

2. BACKGROUND AND RELATED WORK

2.1 Related work

Many of the researchers have evaluated the effect of machine learning in prediction of mortality, but just some of them decided to use biomarkers to it. Riccardo Doyle in his

article [5] decided to use machine learning to predict mortality among COVID patients19. He examined associations based on 25 variables including symptoms, comorbidities and demographic information. Tawsifur Rahman and more [6] designed a prediction model of high mortality risk for both COVID-19 and non-COVID-19 patients based on age and some biomarkers. Rahila Sardar and more [7] developed AI-based algorithms that were used to predict a COVID-19 patient's survival or death. It was based on a publicly available dataset. These data sets consist of clinical parameters and protein profile data of hospital-admitted COVID-10 patients. “RandomForest” as the classification technique achieved a maximum accuracy of 89.47% with the highest sensitivity. [8] Therefore, it is necessary to study the best fitting model of machine learning to make a prediction model for the mortality of Covid-19 patients, based on specific biomarkers.

2.2 Background

Machine learning is a type of artificial intelligence (AI) that allows computers to learn without being explicitly programmed. Machine learning focuses on developing computer programs that can change when exposed to new data. This article describes the basics of machine learning and how to use Python to implement simple machine learning algorithms. [9]

2.2.1 Tools

Specific tools used in the project come from scikit-learn library. It is a free machine learning library for Python. There are various algorithms such as Support Vector Machine, Random Forests, k-neighbours, and also supports numerical and scientific Python libraries such as NumPy and SciPy. [10]

Imputing missing values - Imputation refers to the process of replacing missing or null values in a dataset with a specific value. [11]

SimpleImputer - SimpleImputer is a scikit-learn class that can aid with missing data in predictive model datasets. It substitutes a placeholder for the NaN values. [12]

KNNImputer - Using k-neighbors, imputation is used to fill in missing values. The mean value from k-neighbors discovered in the training set is used to impute each sample's missing values. If the features that neither sample lacks are similar, the samples are near. [13]

IterativeImputer - Imputing missing values in a round-robin fashion by modeling each feature with missing values as a function of other attributes. [14]

Standardization and normalization - The two most examined scaling strategies are Normalization and Standardization. Normalization regularly implies rescales the values into an extent of [0,1]. Standardization ordinarily implies rescales information to have a cruel of and a standard deviation of 1. [15]

StandardScaler - Transforms the data so that the mean is 0 and the standard deviation is 1. That is, standardize the data. Standardization is useful for data with negative values. Place the data in a standard normal distribution. [16]

MinMaxScaler - transform features by scaling each feature to a given range. This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one. where min, max = feature_range. [17]

Models:

LogisticRegression() - Machine Learning regression algorithm that is used to predict the probability of a categorical dependent variable. In calculated relapse, the subordinate variable may be a twofold variable that contains information coded as 1 (yes, victory, etc.) or (no, disappointment, etc.). In other words, the calculated relapse show predicts P(Y=1) as a work of X. [18]

SVC() - The Linear Support Vector Classifier (SVC) strategy applies a straight part work to perform classification and it performs well with a huge number of tests. In the event that we compare it with the SVC show, the Linear SVC has extra parameters such as punishment normalization which applies 'L1' or 'L2' and misfortune work. [19]

SelfTrainingClassifier() - This class permits a given directed classifier to operate as a semi-supervised classifier, permitting it to memorize from unlabeled information. It does this by iteratively foreseeing pseudo-labels for the unlabeled information and including them to the preparing set. The classifier will proceed emphasizing until either max_iter is come to, or no pseudo-labels were included to the preparing set within the past emphasis. [20]

DecisionTreeClassifier() - classification could be a two-step process, learning step and prediction step. Within the learning step, the show is created based on given preparing information. In the prediction step, the model is used to predict the response for given data. [21]

ExtraTreeClassifier() - may be a sort of ensemble learning method which totals the comes about of different

de-correlated choice trees collected in a “forest” to output it's classification result. [22]

2.2.2 Datasets

The two datasets held allowed the creation of a biomarker-based mortality prediction model for COVID-19 patients. The first of the sets was taken between 10.01.2020, and 18.02.2020, at Tongji Hospital, Wuhan, China. This was a training set, and this collection includes 375 patients, 174 of whom have died. Many biomarkers were collected from these patients. The second set was also taken from Tongji Hospital, Wuhan, between 19.02.2020 and 24.02.2020, and was used as test data and included 110 patients, 13 of whom died. The file that served as training (time series 375 preprocess en.xlsx) consists of data on age, sex, time of admission of the patient and outcome (death/discharge) and time of death/discharge. In the remaining 74 columns we can find information on the biomarkers used. The test file (time series test 110 preprocess en.xlsx) contains the same patient data, but different biomarkers. [2]

3. APPROACH

The approach to solve the research question was as follows.

3.1 Understanding and cleaning

Firstly we began with loading and then understanding and cleaning the data. Data has been visualized with head(), shape() and info() commands. It has been noticed that there are many Nan (not a number) values in the data set. Additionally test and training sets contained columns that were considered unnecessary for the purpose of covid mortality prediction. Those columns included 'PATIENT_ID', 'RE_DATE','Admission time' and 'Discharge time' for both training and test data sets. With the drop() command those columns were deleted.

	PATIENT_ID	RE_DATE	age	gender	Admission time	Discharge time
0	1.0	2020-01-31 01:09:00.000000	73	1	2020-01-30 22:12:47	2020-02-17 12:40:09
1	NaN	2020-01-31 01:25:00.000000	73	1	2020-01-30 22:12:47	2020-02-17 12:40:09
2	NaN	2020-01-31 01:44:00.000000	73	1	2020-01-30 22:12:47	2020-02-17 12:40:09
3	NaN	2020-01-31 01:44:59.999999	73	1	2020-01-30 22:12:47	2020-02-17 12:40:09
4	NaN	2020-01-31 01:56:00.000001	73	1	2020-01-30 22:12:47	2020-02-17 12:40:09

Figure 1 Unnecessary columns in train dataset

It has been observed that biomarkers available in the datasets follow different scales. Literature research has shown that most of the biomarkers should follow Gaussian distribution with a very few outliers [23]. Therefore it has been decided to focus mainly on standardization and including normalization as an additional experiment.

With the use of isfinite() function from numpy library the dataset was checked for infinite values.

3.2 Imputing missing values

The performance of a machine learning model not only depends on the model and the parameters but also on the variables we give to the model. Because most machine learning models only accept numerical variables, it is necessary to preprocess the categorical and Nan variables. Usually it is necessary to convert categorical variables and Nan’s to numbers so the model is able to process and extract valuable information from our datasets.

Fortunately there were no categorical variables that would have been necessary to replace. However, as many values were missing in the dataset imputers had to be used before proceeding with data scaling. The imputers used were SimpleImputer, KNNImputer and IterativeImputer.

For SimpleImputer two strategies were tested. First one was to impute missing values by calculating the mean of the column. Second one was to fill all the missing values with a constant equal to negative 1 or 0.

For KNNImputer and IterativeImputer default parameters were used.

3.2 Standardization and Normalization

Some machine learning algorithms are especially sensitive to feature scaling and or variables distribution. Data normalization and standardization is therefore used to make machine learning models less sensitive to the scale of features. This allows ML algorithms to better adjust weights and therefore leads to a more accurate model.

Standardization is used for the data variables that follow normal distribution. By standardizing the values the statistics of data distribution become: mean = 0 and standard deviation = 1. To perform standardization Python sklearn library StandardScaler() function can be used. [24]

Normalization is a different type of scaling that is performed for datasets that don't follow normal distribution. With normalization values are shifted and scaled in a way that they range between 0 and 1. [25] To perform standardization Python sklearn library MinMaxScaler() function can be used.

After imputing the missing values 4 training datasets resulting from 4 imputing methods had to be standardized and normalized before building the models.

3.3 Feature selection

In the article mention at the beginning of the report 3 biomarkers were selected for training the machine learning models for covid-19 mortality prediction. Those biomarkers were: *'Lactate dehydrogenase', '(%)lymphocyte' and 'Hypersensitive c-reactive protein'*. Before proceeding with models training and therefore choosing the biomarkers to predict covid mortality it has been decided to explore how other biomarkers available in data correlate with the outcome (death or living).

To achieve this correlation matrix (Appendix I) for all the features was calculated. It was discovered that apart from *'Lactate dehydrogenase', '(%)lymphocyte' and 'Hypersensitive c-reactive protein'* also *neutrophils(%)* ; *age, interleukin 2 receptor, urea, neurophilis count, D-D dimer* and *Fibrin degeneration products* show high positive correlation with the outcome.

For all the biomarkers identified, scatter plots were created (Appendix II) to examine further the relations in the dataset.

Table below shows the correlation between biomarkers and outcome.

Table 1. Caption goes above table

Biomarker	**Outcome correlation**
Lactate dehydrogenase	0,62
(%)lymphocyte	-0,72
Hypersensitive c-reactive protein	0,65
nerophils(%)	0,73
age	0,59
interleukin 2 receptor	0,47
urea	0,49
neurophilis count	0,61
D-D dimer	0,64
Fibrin degeneration products	0,60

It can be seen that the 3 chosen biomarkers indeed represent a high correlation with the covid mortality in patients, therefore it has been decided to follow the steps of researchers and continue with building models based on those clinical parameters.

However the knowledge of the other biomarkers that correlate with outcome can be possibly used to further improve the models.

3.4 Building Models

To train the models, the training data set was split with the use of train_test_split() function with X predictors set to 'Lactate dehydrogenase','(%)lymphocyte','Hypersensitive c-reactive protein' and Y to 'outcome'.

From sklearn library 5 different models have been chosen to test and compare on the dataset: LogisticRegression(), SVC(), SelfTrainingClassifier(), DecisionTreeClassifier() and ExtraTreeClassifier().

All the models were firstly trained on the data that was standardized and standardized + normalized afterwards.

At the end models were evaluated in the terms of accuracy of covid 19 prediction.

4. EXPERIMENTS

Experimental part consisted of training the models with the use of datasets that were differently preprocessed, standardized and normalized.

Firstly, the LinearRegression model was trained on the datasets that were imputed with different methods and standardized. LinearRegression was also tested on the dataset that underwent normalization as an additional step but it didn't impact the accuracy of the model.

From all the imputing methods KNNImputer was shown to create a model with the highest accuracy of 0,97. Other results are shown in the table below.

Table 2. LinearRegression evaluation

Imputer	Accuracy
SimpleImputer (mean)	0,64
SimpleImputer (const=0)	0,54
KNNImputer	0,97
IterativeImputer	0,57

As KNNImputer performed significantly better that other methods it was decided to test other ML models with dataset preprocessed with this technique. Results of medals evaluation are shown in the table below. For decision tree models the actual trees can be found in Appendix III and IV.

Table 3. ML models evaluation with KNNImputer

Model	Accuracy
BayesianRidge()	0,76
SVC()	0,97
DecisionTreeClassifier()	0,97
ExtraTreeClassifier()	0,95

It can be seen that KNNImputer models achieve a high accuracy in prediction of covid-19 mortality.

All models were evaluated on the test sets created from the training dataset that was available for this project.

LinearRegression model with KNN imputation was also evaluated on KNN imputed test dataset available for the project achieving 0,98 accuracy as shown in the classification report below.

```
Accuracy: 0.98
              precision    recall  f1-score   support

           0       1.00      0.97      0.98       132
           1       0.94      1.00      0.97        58

    accuracy                           0.98       190
   macro avg       0.97      0.98      0.98       190
weighted avg       0.98      0.98      0.98       190
```

Figure 2 Classification report for LinearRegression()

5. DISCUSSION

As presented in the experiments section preprocessing of the data is a crucial step in machine learning. Especially the method of handling the missing values impacts the performance of the models tested. KNN imputer has been shown to bring the best results for the given dataset. Nearest Neighbor Imputation is indeed widely used for preprocessing data and as it uses the nearest values to fill missing information it's more accurate than simple imputation methods.

For the models Classification and Regression methods have been shown to perform the best on the given dataset achieving the accuracy of 0,97 and high precision. As compared to the results of *" An interpretable mortality prediction model for COVID-19 patients"* [26] it is a good result. However the accuracy of models based on SimpleScaler and IterativeImputer is disappointing and possibly could be improved with changes in parameters.

It has been proven that biomarkers in the dataset follow normal distribution as the accuracy of models did not change with additional data normalization.

For the given models limitations that apply are strongly connected to the dataset with many missing values. Any imputation of those involved addition of artificial data into the dataset and therefore raises questions about the models measured accuracy.

6. CONCLUSIONS

To conclude, the answer to the research question is not straight forward mainly because of the amount of factors that can have an impact on the models. It can be said that more complicated methods like SVC() perform as good as

LinearRegression() on the given dataset. Most likely the significant difference in performance could be noticed on the much larger datasets. In this scenario advanced methods would most likely perform better.

It has been observed that the data quality and it's preprocessing is even more important than the chosen model as during the works on the project firstly only the SimpleImputer was used and the results of 0,64 accuracy were not acceptable for real life scenarios as the mortality would be predicted only a bit better than a flip of a coin.

With proper data preprocessing models described in this report with over 90% accuracy could be used in a real hospital setting.

Complete code for this project can be found at Github: https://github.com/DSSL28/DataScience.git

REFERENCES

[1] David Adam, The pandemic's true death toll: millions more than official counts, 18 January 2022, https://www.nature.com/articles/d41586-022-00104-8

[2] Brenda Voorthuis & Karin Groothuis-Oudshoorn, Description of the project COVID19 introduction, Data Science

[3] Hengbo Zhu, Li Wei & Ping Niu, The novel coronavirus outbreak in Wuhan, China, 02 March 2020, https://ghrp.biomedcentral.com/articles/10.1186/s41256-020-00135-6

[4] Mohammad M Banoei, Roshan Dinparastisaleh, Ali Vaeli Zadeh, Mehdi Mirsaeidi, Machine-learning-based COVID-19 mortality prediction model and identification of patients at low and high risk of dying, 2021 Sep 8, https://pubmed.ncbi.nlm.nih.gov/34496940/

[5] Richard Doyle, Machine Learning-Based Prediction of COVID-19 Mortality With Limited Attributes to Expedite Patient Prognosis and Triage: Retrospective Observational Study, 2021 Oct 15, https://pubmed.ncbi.nlm.nih.gov/34843609/

[6] Tawsifur Rahman and more, Mortality Prediction Utilizing Blood Biomarkers to Predict the Severity of COVID-19 Using Machine Learning Technique, 8 July 2021, https://www.mdpi.com/2075-4418/11/9/1582

[7] Rahila Sardar and more, Machine Learning Assisted Prediction of Prognostic Biomarkers Associated With COVID-19, Using Clinical and Proteomics Data, 20 May 2021, https://www.frontiersin.org/articles/10.3389/fgene.2021.636441/full

[8] Rahila Sardar and more, Machine Learning Assisted Prediction of Prognostic Biomarkers Associated With COVID-19, Using Clinical and Proteomics Data, 20 May 2021, https://www.frontiersin.org/articles/10.3389/fgene.2021.636441/full

[9] Introduction To Machine Learning using Python - Geeksforgeeks. URL https://www.geeksforgeeks.org/introduction-machine-learning-using-python/

[10] Python Libraries - Medium. URL https://medium.com/analytics-vidhya/python-libraries-d73859384c43

[11] Impute missing data values in Python – 3 Easy Ways! - Askpython. URL https://www.askpython.com/python/examples/impute-missing-data-values

[12] Handle Missing Data with Simple Imputer - Geeksforgeeks. URL https://www.geeksforgeeks.org/ml-handle-missing-data-with-simple-imputer/

[13] Sklearn.impute.KNNImputer - Scikit-learn. URL https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html

[14] sklearn.impute.IterativeImputer - Scikit-learn. URL https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html

[15] Normalization vs Standardization — Quantitative analysis - Towardsdatascience. URL https://towardsdatascience.com/normalization-vs-standardization-quantitative-analysis-a91e8a79cebf

[16] When to use standard scaler and when normalizer? - DataScience. URL https://datascience.stackexchange.com/questions/45900/when-to-use-standard-scaler-and-when-normalizer

[17] sklearn.preprocessing.MinMaxScaler - Scikit-learn. URL https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[18] Building A Logistic Regression in Python, Step by Step - Towardsdatascience. URL https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8

[19] lassification Example with Linear SVC in Python - Datatechnotes. URL https://www.datatechnotes.com/2020/07/classification-example-with-linearsvm-in-python.html

[20] sklearn.semi_supervised.SelfTrainingClassifier - Scikit-learn. URL

https://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.SelfTrainingClassifier.html
[21] Decision Tree Classification in Python - Datacamp. URL https://www.datacamp.com/community/tutorials/decision-tree-classification-python
[22] Extra Tree Classifier for Feature Selection - Geeksforgeeks. URL https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/
[23] Gaussian distribution - Sciencedirect. URL https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/gaussian-distribution
[24] Data normalization in Python - Educative. URL https://www.educative.io/edpresso/data-normalization-in-python
[25] Feature Scaling for Machine Learning: Understanding the Difference Between Normalization vs. Standardization - Analyticsvidhya. URL https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/
[26] Yan, L., Zhang, H-T., Goncalves, J., Xiao, Y., Wang, M., Guo, Y., ... Yuan, Y. (2020). An inter- pretable mortality prediction model for COVID-19 patients. Nature Machine Intelligence, 2, 283-288. doi:10.1038/s42256-020-0180-7