Covid-19 prediction of fatality

Hanna Foerster
Department of Computer Science
Durham University
Durham, UK
hanna.foerster@durham.ac.uk

***Abstract*—The spread of covid-19 has put a strain on the hospitals and medical sector all over the world. As the number of covid-19 patients rise and medical facilities encounter shortages in ICU capacities, it is inevitable to develop a triage system for covid-19 patients based on evidence of how vulnerable patients are. This paper aims to predict on an individual level which covid-19 cases are the most likely to end fatally and therefore should be prioritized for hospital admission or a space at ICU.**

Keywords—Covid-19, fatality, machine learning

I. Introduction

The SARS-Cov-2 virus is responsible for the global covid-19 pandemic which changed the lives of billions of people and has claimed more than 2.9 million lives. The pandemic has caused a big strain on the medical facilities and front-line hospital staff. On top of the usual patients from other illnesses and accidents, hospitals must further accommodate many Covid-19 patients. This has caused the medical staff to be overworked and the ICUs to be overwhelmed with the number of patients. Even patients that have illnesses unrelated to Covid-19 have been affected by not getting the required treatment or getting treated too late because of this incredible strain on the health sector.

It is not easy to judge which covid-case is most likely to end fatal. There are many ways the disease can progress and many different factors contributing to its predicament. Moreover, the likeliness of death often must be decided in a short amount of time by overworked physicians, who cannot always get the full picture of severity. Furthermore, it can sometimes happen that many not as severe cases are admitted just before a big wave of even more severe cases arrive at the hospital if there is no clear formula to go by to judge fatality.

Therefore, it is of utmost importance to gain a better understanding of the risk group. This kind of analysis will be beneficial to incorporate into the existing triage system. The triage system implemented now does not have any specific orders regarding covid-19 patients. Because of this either too many covid-19 patients are hospitalised causing other patients in need of treatment to be turned away or too little covid-19 patients are hospitalised causing deaths at home that could have been prevented at the hospital. In times when hospitals are flooded with patients and resources are limited it is necessary to develop a risk assessment method in order to prevent deaths by prioritising the most at-risk patients.

II. Methodology

A. Data

The data set that was used is from a database on github (https://github.com/beoutbreakprepared/nCoV2019) [1] which records real-time case information. The data provides individual-level information of 2676311 cases dating from the start of the Covid-19 outbreak in December 2019 to the 17th May 2020. While other datasets have been mainly focused on providing aggregated case counts per geographic location, this dataset has accumulated more detailed data for each individual case. Variables provided include age, sex, dates for the onset of symptoms, admission in hospital, confirmation of the virus, travel history and death or discharge. In addition to that several variables indicate geographical position: country name, name of region and city the case occurred, latitude and longitude. Further variables give an insight about the background of the case: symptoms, lives_in_Wuhan, reported market exposure, additional information, chronic disease, outcome and source of the data.

B. Data Preparation

By analysing the data, I soon found that there were many missing values, so I selected to only work with case reports that had valid entries for outcome, sex, age and date of confirmation of the covid-19 disease. This gave me 33531 data points to work with. As a next step I picked only the relevant features, dropping variables about geographical location except for longitude and latitude and columns about the data source and notes that cannot be easily interpreted. All data with less than 200 and less than 5000 valid values for numerical and categorical data respectively were dropped as well. I kept more of the numerical data, because interesting variables such as date of hospital admission did not have as many entries, but I thought might turn out to be interesting.

The outcome feature amongst other things denotes whether a patient died of covid. This feature was used to compute the binary target variable ‘deceased’, and the outcome feature was dropped instead. In addition, three more columns were created or changed. One column called ‘symptoms_binary’ which is True if a date is given for ‘onset_symptoms’ or a symptom is listed in the feature

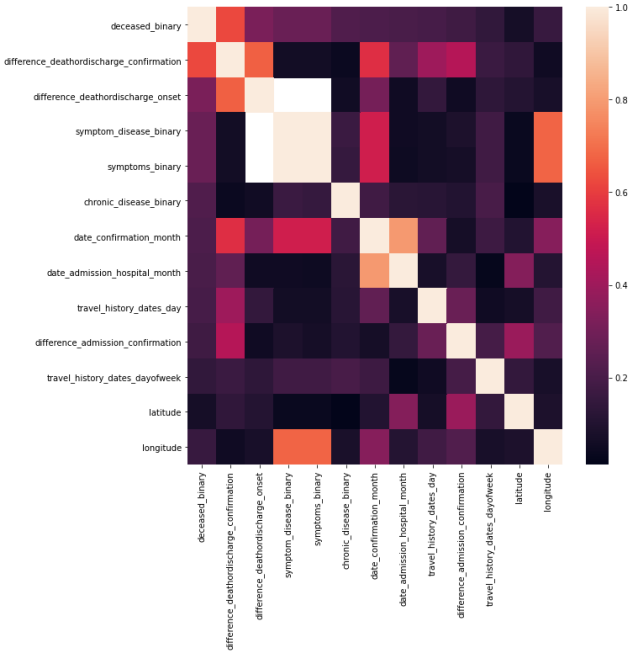

Fig. 1. Correlation map of best numerical features.

‘symptom’. To another column called ‘chronic_disease_binary’ all rows which had a chronic disease listed in ‘chronic_disease’ were additionally listed as True. Furthermore, a new column was created from these two new columns combining symptoms and chronic disease. This feature was set to True if the patient had symptoms or had a chronic disease and False otherwise.

For all date variables, the date was split into the components, year, month, day and day of week. Moreover, the difference to each of the other dates was computed, e.g., the time elapsed between the onset of symptoms and hospitalisation or between travelling and confirmation of covid. Since this left me with too many numeric features, feature selection by the absolute value of the pearson correlation was applied. First the best 13 features correlating to the target feature were selected and then if the correlation between a pair of features was more than 0.8 the feature with more missing values was dropped. (Fig. 1)

For the categorical variables, I sorted all age data into categories of age groups in steps of 10. A lot of the age data was given as a range and some of these ranges did not fit into only one specific age group. In that case I took the age group that lay in the middle of the range. The feature importance for the categorical data was also computed (Fig. 2), but since there were not many features, all of them were used.

The columns that were left after all this preprocessing are the following: ‘age’, ‘sex’, ‘chronic_disease_binary’, ‘symptoms_binary’, ‘symptom_disease_binary’, ‘travel_history_binary’, ‘difference_deathordischarge_confirmation’, ‘difference_deathordischarge_onset’, ‘difference_admission_confirmation’, ‘date_confirmation_month’, ‘date_admission_hospital_month’, ‘travel_history_dates_day’, ‘travel_history_dates_dayofweek’, ‘date_admission_hospital_day’, ‘date_death_or_discharge_dayofweek’, ‘latitude’, ‘longitude’.

After splitting the data into training set and test set at a ratio of 8:2, each set was pre-processed through a pipeline. The pipeline split the data into numerical and categorical data. The numerical data was imputed in the first step. For this the iterative imputer from the sklearn library, using the k-nearest-neighbour regressor as the estimator function, was applied. The iterative imputer is still classed as an experimental estimator in sklearn, but provides better estimations than simple imputations, such as imputing each cell with the mode of the column values. It uses the strategy of imputing missing values in each feature by modelling each feature with missing values as a function of the other features. Using the nearest neighbour algorithm as estimation function, each instance was compared with the nearest instances, to impute the missing cells. As a second step, all the numeric features were standardized to have zero mean and variance 1. This kind of normalization is necessary since the Euclidean distances between instances are calculated by the sum of distances between features of instances. If the range for one feature was much bigger than another, the overall distance would be dominated by this one feature.

For the categorical data, the features were label encoded and imputed before the data was split into test and training set, since this was necessary for calculating the feature importances. Hereby, an iterative imputer was used with the random forest classifier as an estimator. Then, in the pipeline the categorical data was one-hot encoded.

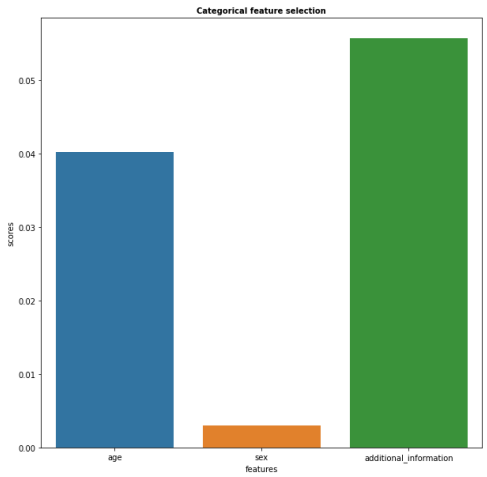

Fig. 2. Feature importance categorical variables

C. Machine Learning Algorithms and training

The algorithms that were used to evaluate the data are Logistic Regression, K-nearest-neighbour and Random Forest Classification. These three algorithms were chosen because they are conceptually very different, and I anticipated that they would give good prediction results after reading other papers that used these algorithms. Logistic Regression (LR) is a statistical model to classify an instance based on its features using a logistic function. In this research the model used the features mentioned above to classify a case as ‘Deceased’ or not. K-nearest neighbour (KNN) is an algorithm that classifies an instance by comparing it to k nearest neighbours. In this case the outcome of k nearest cases, measured by Euclidian distance between features of the cases, were considered to classify a case as ‘Deceased’ or not. Lastly, Random Forest Classification (RF) outputs the mode of classes obtained by multiple decision trees at training time. Each decision tree in my problem has internal nodes representing a rule such as true or false for a binary feature and the leaf nodes represent the two classification options ‘Deceased’ or ‘not Deceased’.

A grid-search approach was used to determine the best hyper-parameters for each algorithm. For LR the best parameter C between the values 2 to 7 and the best solver between ‘lbfgs’ and ‘liblinear’ were analysed. The parameter C is the inverse regularisation parameter and is defined as C=1/lambda, where a bigger lambda means that a bigger regularisation penalty is applied to the model to reduce the risk of overfitting. For KNN, the best hyper-parameter for the number of neighbours was determined between 4,5 and 6. The RF algorithm had a more complex grid of parameters, so a randomized grid search was run to find the area of best parameters for number of trees (between 200 and 2000), bootstrap (True or False), minimum number of samples required to split an internal node (2,4,6) and maximum depth of the tree (5,10,20,30,40). Upon determining the best area for each of these parameters a grid search was initiated to determine the best parameter for training the data with the model.

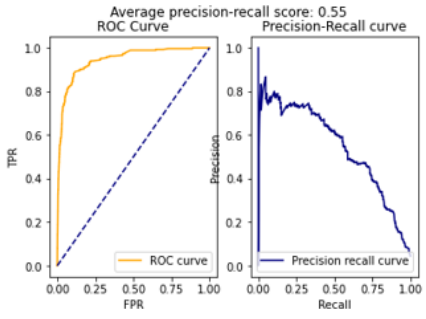

Fig. 3. ROC curve and Precision-Recall curve LR

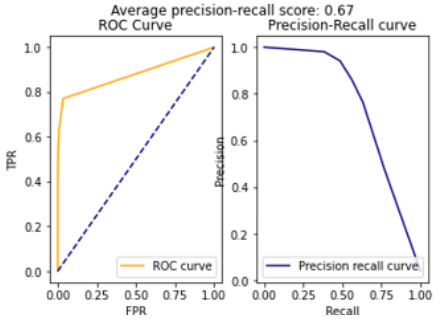

Fig. 4. ROC curve and Precision-Recall curve KNN

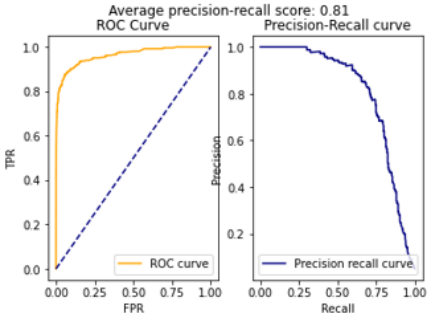

Fig. 5. ROC curve and Precision-Recall curve RF

III. Results

A. *Description of results*

The death rate in the test set was 0.04% (Fig. 11). Hence the data we are dealing with is highly imbalanced towards 'not Deceased' such that accuracy is not a good measure for the performance of the model. However, Sensitivity (also called recall) is a good measure, since it measures the proportion of true positives that the model has predicted correctly. In our model it is the measure of to what percentage cases labelled as 'Deceased' were predicted as 'Deceased' in each model. The Positive predictive value (also called precision) is calculated by the proportion of true positives in all positively predicted values, so in our case how many of the as 'Deceased' predicted labels were actually labelled as 'Deceased'. The F1 score is the harmonic mean of precision and recall and takes on the value 1.0 for perfect predictions and value 0.0 if either precision or recall has the value 0. Since it combines both precision and recall, this will be the measure we will primarily be focusing on. It is calculated with the following formula:

$$2 * \frac{precision * recall}{precision + recall}$$

The models performed in the range of accuracy between 0.97 and 0.98 (Fig. 6). The best precision recall score was attained by RF and a comparison between the ROC curves and Precision-Recall curves of the models also underline this result (Fig. 3 ,4 ,5). The f1-scores for predicting death were 0.48, 0.67 and 0.78 for each of LR, KNN and RF. Further measures can be checked in figure 6. An analysis of the weights of each feature used for LR and RF gives an insight into the importance of features. For LR the most important features turned out to be age, all age groups above 60 were weighted high. Moreover, symptoms were valued highly, latitude and the number of days between travelling and confirmation, onset and confirmation and confirmation and admission were also important. (Fig. 7) For RF the most important features were longitude, latitude, age and days between travel, onset and confirmation. To gain more insight the feature importance was calculated by the permutation importance function from the eli5 library. The permutation importance is calculated by the decrease in accuracy when a feature value is randomly shuffled. This similarly rated symptoms, dates and age highly. (Fig. 8)

Another version of the algorithm was calculated including the additional information feature. This increased the

model	Classification report (test set) (with additional Information)	MSE (w/a. Info.)	Classification report (test set)	MSE
LR	precision recall f1-score support 0 0.97 1.00 0.98 6441 1 0.74 0.32 0.45 266 accuracy 0.97 6707 macro avg 0.86 0.66 0.72 6707 weighted avg 0.96 0.97 0.96 6707	0.031	precision recall f1-score support 0 0.97 0.99 0.98 6441 1 0.71 0.36 0.48 266 accuracy 0.97 6707 macro avg 0.84 0.68 0.73 6707 weighted avg 0.96 0.97 0.96 6707	0.031
KNN	precision recall f1-score support 0 0.98 1.00 0.99 6441 1 0.92 0.58 0.71 266 accuracy 0.98 6707 macro avg 0.95 0.79 0.85 6707 weighted avg 0.98 0.98 0.98 6707	0.019	precision recall f1-score support 0 0.98 1.00 0.99 6441 1 0.83 0.56 0.67 266 accuracy 0.98 6707 macro avg 0.91 0.78 0.83 6707 weighted avg 0.98 0.98 0.98 6707	0.022
RFC	precision recall f1-score support 0 0.98 1.00 0.99 6441 1 0.95 0.55 0.70 266 accuracy 0.98 6707 macro avg 0.97 0.78 0.85 6707 weighted avg 0.98 0.98 0.98 6707	0.019	precision recall f1-score support 0 0.99 1.00 0.99 6441 1 0.87 0.70 0.78 266 accuracy 0.98 6707 macro avg 0.93 0.85 0.88 6707 weighted avg 0.98 0.98 0.98 6707	0.016

Fig. 6. Classification report of machine learning models with variable 'additional information' and without

	target	feature	weight
0	1	age_90+	2.660328
1	1	age_80-89	2.260592
2	1	age_70-79	1.771835
3	1	age_60-69	1.269086
4	1	symptoms_binary	0.926694
5	1	latitude	0.430550
6	1	difference_confirmation_travel	0.280390
7	1	difference_admission_confirmation	0.262784
8	1	age_50-59	0.261421
9	1	difference_confirmation_onset	0.183941
10	1	chronic_disease_binary	0.151141

Fig. 7. Feature weights LR

	feature	weight	std
0	symptoms_binary	0.053217	0.004600
1	difference_confirmation_travel	0.029259	0.002258
2	age_70-79	0.022765	0.003836
3	age_20-29	0.018598	0.006017
4	chronic_disease_binary	0.017371	0.000832
5	age_80-89	0.017091	0.002162
6	difference_confirmation_onset	0.016551	0.003435
7	age_30-39	0.013960	0.004756
8	age_90+	0.010588	0.001536
9	difference_admission_confirmation	0.010132	0.003149
10	age_10-19	0.008867	0.004087

Fig. 8. Feature importance LR

precision-recall score by 0.01% for LR and RF and gave some additional insight about feature importance. The additional information that was rated highly was related to travelling history at certain corona-hotspots, such as Maharashtra. Some other additional information was about chronic disease, such as leukaemia causing death (Fig.9,10). The feature 'additional information' however was not taken into the standard set of features to classify instances, because many missing values were found in the column and imputing lead to some not interpretable information.

B. Interpretation

The results indicate that elderly people are more at risk. This supports the findings from Zhang et al. [2] who observed that there is a strong association with age and the severity of Covid-19 infection. Moreover, travel associated features have been seen as important, especially if the travel was from a hotspot area. These findings correspond with the policy of travel bans to countries with high infection rates that many countries have taken. The geographical features longitude and latitude were also deemed as important in the RF, in the LR however, only latitude was weighted highly. In a study conducted by Sun, Zhibin et al. [3] the relationship between geographical location and the spreading speed of Covid-19 was conducted. By limiting the cases to cases in China that were under the same lockdown measures, the study has shown that altitude and latitude have an impact on covid spread. The findings from the feature importance in the LR correspond to the findings in this study, since a faster spread of cases would cause more cases of covid and hence increase the absolute number of deaths in this area.

C. Model evaluation and Limitations

The results indicate that RF performed the best and LR the worst in predicting death of covid patients. All models however have limitations, given the nature of the dataset. First, the dataset had many missing values for features such as symptoms, chronic disease and additional information. Therefore, they had to be generalized to a binary value and imputed for some cells or taken out of account. Other similar studies have further had more specific data about the patients' conditions, such as medical record, medical images, lab results and comorbidities. Moreover, only the rows that did not have missing values for sex, age, outcome and date of confirmation were taken and this subset is very geographically biased. 85% of the cases are from India. An

	target	feature	weight
0	1	additional_information_Detected post mortem	3.864544
1	1	additional_information_P27	3.775751
2	1	additional_information_First death in India, R...	3.645879
3	1	additional_information_C771689	3.498684
4	1	additional_information_C543744	3.481768
5	1	additional_information_He was from Munger. He ...	3.391582
6	1	additional_information_Died within 12 hrs of a...	3.373916
7	1	additional_information_C665068	3.292354
8	1	additional_information_C356521	3.261318
9	1	additional_information_C943553	3.250437
10	1	additional_information_C464028	3.242282

Fig. 9. Feature weights LR with addtl. information

	feature	weight	std
0	travel_history_dates_day	0.063118	0.004733
1	symptoms_binary	0.062033	0.003110
2	additional_information_Travelled from Maharashtra	0.061571	0.009151
3	difference_deathordischarge_confirmation	0.051995	0.005279
4	age_70-79	0.048907	0.001818
5	age_20-29	0.035186	0.004843
6	age_30-39	0.031191	0.004145
7	difference_admission_confirmation	0.027204	0.003837
8	date_admission_hospital_month	0.025145	0.001895
9	age_60-69	0.024548	0.002865
10	chronic_disease_binary	0.022994	0.001372

Fig. 10. Feature importance LR with addtl. information

example of this bias is the high prediction power of the additional information about travel from Maharashtra. This high prediction power only arises because the number of deaths that were analysed is so small. This raises doubt about the out of sample prediction power, because the training and test data is unlikely to be representative.

```
Overall:
Number of positive instances:  266
Number of instances:  6707
Positive rate:  0.039660056657223795
```

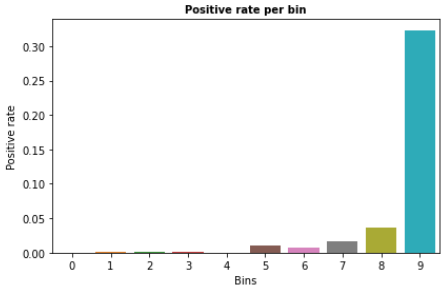

Fig. 11. Death rate per bin in test set

D. Conclusion and lesson learnt from the assignment

It can be concluded that a machine learning approach to solve the problem of prioritisation for hospitalisation is helpful and will alleviate some of the stress that medical staff are facing in the Covid-19 pandemic. However, as this study shows, more precise data on the patients' medical condition is needed to yield more accurate results. This supports the findings of Li et al. [4] who have also found that better machine learning prediction results can be achieved if this dataset was used alongside other medical resources. Nevertheless, it can be noted that key findings in this dataset classify people aged 60 and above, people who have travelled in hot spot areas and people with preconditions or symptoms with higher risk of fatality.

From this assignment I have learned to work with a big dataset and pre-process data in such a way that it can be used to train machine learning models. In addition to that I have learned to apply machine learning algorithms on the data and then measure these results in appropriate ways to be able to judge the performance of the model. Moreover, I learned about different data visualisation methods and studied other academic articles to be able to write this article in a similar format.

REFERENCES

[1] Xu, B., Gutierrez, B., Mekaru, S. et al. "Epidemiological data from the COVID-19 outbreak, real-time case information" Scientific Data vol.7 (2020): 106. doi:10.1038/s41597-020-0448-0

[2] Zhang Chi, Qin Ling, Li Kang et al. , "A Novel Scoring System for Prediction of Disease Severity in COVID-19" Frontiers in Cellular and Infection Microbiology vol.10 (2020): 318. doi:10.3389/fcimb.2020.00318

[3] Sun, Zhibin et al. "Impacts of geographic factors and population density on the COVID-19 spreading under the lockdown policies of China." The Science of the total environment vol. 746 (2020): 141347. doi:10.1016/j.scitotenv.2020.141347

[4] Li Yun, Horowitz Melanie Alfonzo, Liu Jiakang et al. ," Individual-Level Fatality Prediction of COVID-19 Patients Using AI Methods" Frontiers in Public Health vol. 8 (2020): 566. doi:10.3389/fpubh.2020.587937