Covid-19 predictive Model in Brazil

Pedro Rodrigues

7/22/2020

Contents

Introduction

Coronavirus Disease 2019 (COVID-19) is a viral respiratory disease that was declared as Pandemic by the World Health Organization (WHO) in March 11^{th}. Since then it has spreading quickly. As of mid July there still hasn't been found any definitive treatment or any vacine for the disease. Acttualy there's little established knowledge concerning Sars-Cov-2. This exploratory analysis and predictive model come to try and help with that. Shed some light regarding the risk and prognosis of some Covid cases. Trying to create a system to choose cases that should be treated with bigger pryority pior to agravating in symptoms, leading to a premature especial attention.

Methodology

Data

We began with some open datasets on the notifications of coronavirus cases in Brazil. Available on the openDatasus website there is one dataset for each brazilian state, in which they added notitifications arising

from the *e-SUS NOTIFICA* (e-(Unique Health System) NOTIFY), which was developed to record suspected Covid-19 and Influenza Syndrome cases. There is over 2.5 Gb worth of data, so there was quite a bit of data cleaning to do.

Data Cleaning

Begining with multiple csv files, and a lot of noise in my datasets I began with developing a function that would read all the files; filter to get only confirmed cases that had a definitive ending to then, patients who got cured, and patiets who passed away; select the usefull variables, in my understanding:

```
## [1] "id"                 "notification_date"  "first_symptom_date"
## [4] "conditions"         "sex"               "state"
## [7] "age"                "case_evolution"
```

And finally add to a single dataset, the one we're going to use throughout this analysis: clean_dataset. After that I translated the variable names and factors used in the datasets from portuguese to english, making it easier to use in this report. Then I changed the Conditions column into a tidyer format, by taking each possible condition into it's own column, using factors that sad if it was present or not. And if summary variable that sad which any of the comorbidities were present. And finally I filtered out some weird ages, by setting the max age as 110 years old, since I had some over 300 years old polluting the dataset. After that I just exported a csv file and uploaded a compressed version of it into my GitHub. There you have the final version of my dataset:

```
## 'data.frame':    397048 obs. of  15 variables:
##  $ id                     : Factor w/ 395493 levels "0003sYEqHO","001Ndyf0dw",..: 298726 61007 380425
##  $ notification_date      : Factor w/ 172027 levels "2020-01-28T04:00:00.000Z",..: 1220 60 1200 2281
##  $ first_symptom_date     : Factor w/ 749 levels "2020-01-01T03:00:00.000Z",..: 188 172 190 223 204
##  $ sex                    : Factor w/ 3 levels "female","male",..: 1 1 1 1 2 1 2 2 1 2 ...
##  $ state                  : Factor w/ 28 levels "","ACRE","ALAGOAS",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ age                    : int  28 39 47 40 23 52 42 67 30 73 ...
##  $ case_evolution         : Factor w/ 2 levels "decease","cure": 2 2 2 2 2 2 2 2 2 2 ...
##  $ diabetes               : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ cardiac                : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ respiratory            : Factor w/ 2 levels "no","yes": 1 1 1 1 1 2 1 1 1 1 ...
##  $ renal                  : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ immunosuppression      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ pregnant               : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ chromosomal_abnormality: Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ comorbidities          : Factor w/ 2 levels "no","yes": 1 1 1 1 1 2 1 1 1 1 ...
```

Data Exploration

General

Now we can begin the data exploration, let's see the dimensions:

```
## [1] 397048     15
```

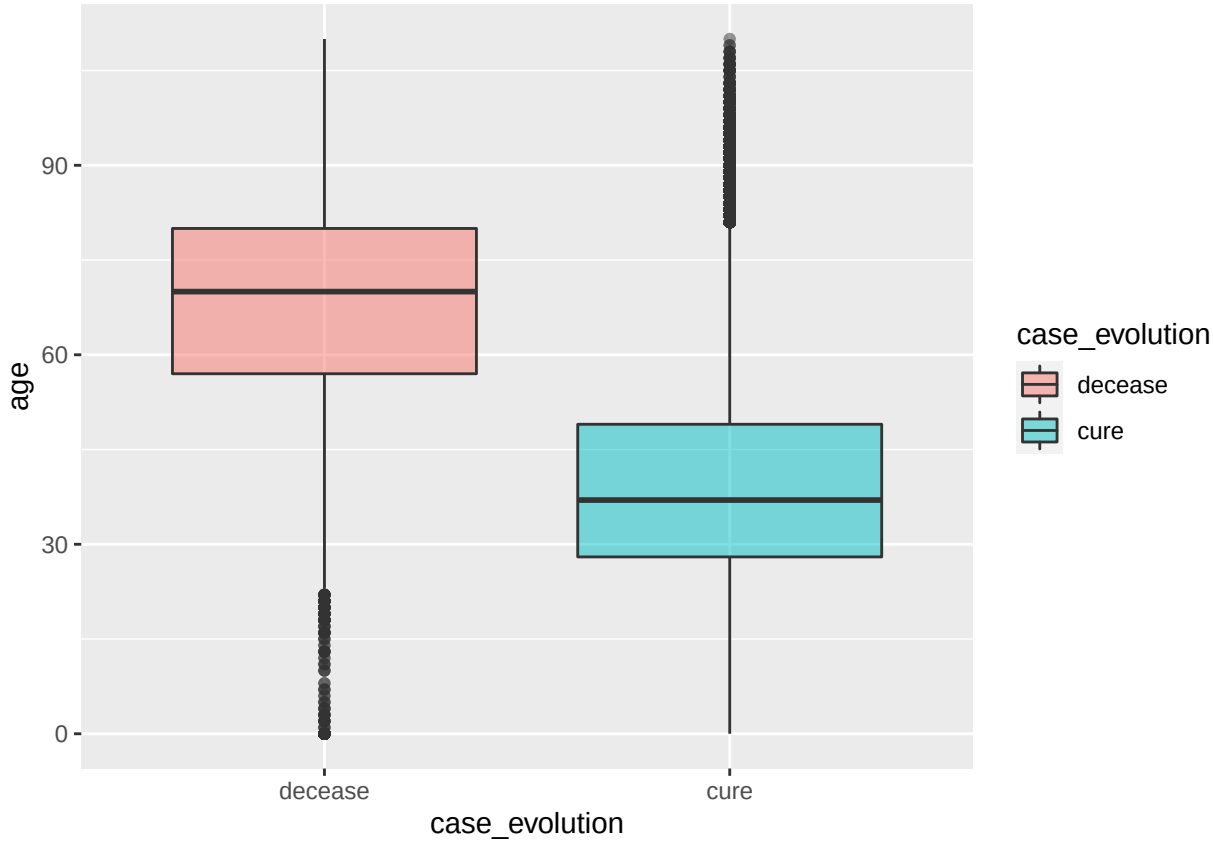
First thing we note is how badly imbalanced the classes are:

case_evolution	total	percentage
decease	7278	0.018
cure	389770	0.982

We are going to deal with this basically with tuning, using Balanced Accuracy as our main Performance Metric and in the final model we also undersample, by randomly excluding some cured observations

Age

Here we look to see if there is any relationship between the age and the case evolution, simply a boxplot:



There, it is notable that the median age amongst the deceased is quite higher than amongst the cured group. Which is something we are going to use in the models we are going to train.

State

Took a look if there was any connections between the state and the case evolution. Since there are 28 states, there will be just the percentages of cure and deceased cases ahead:

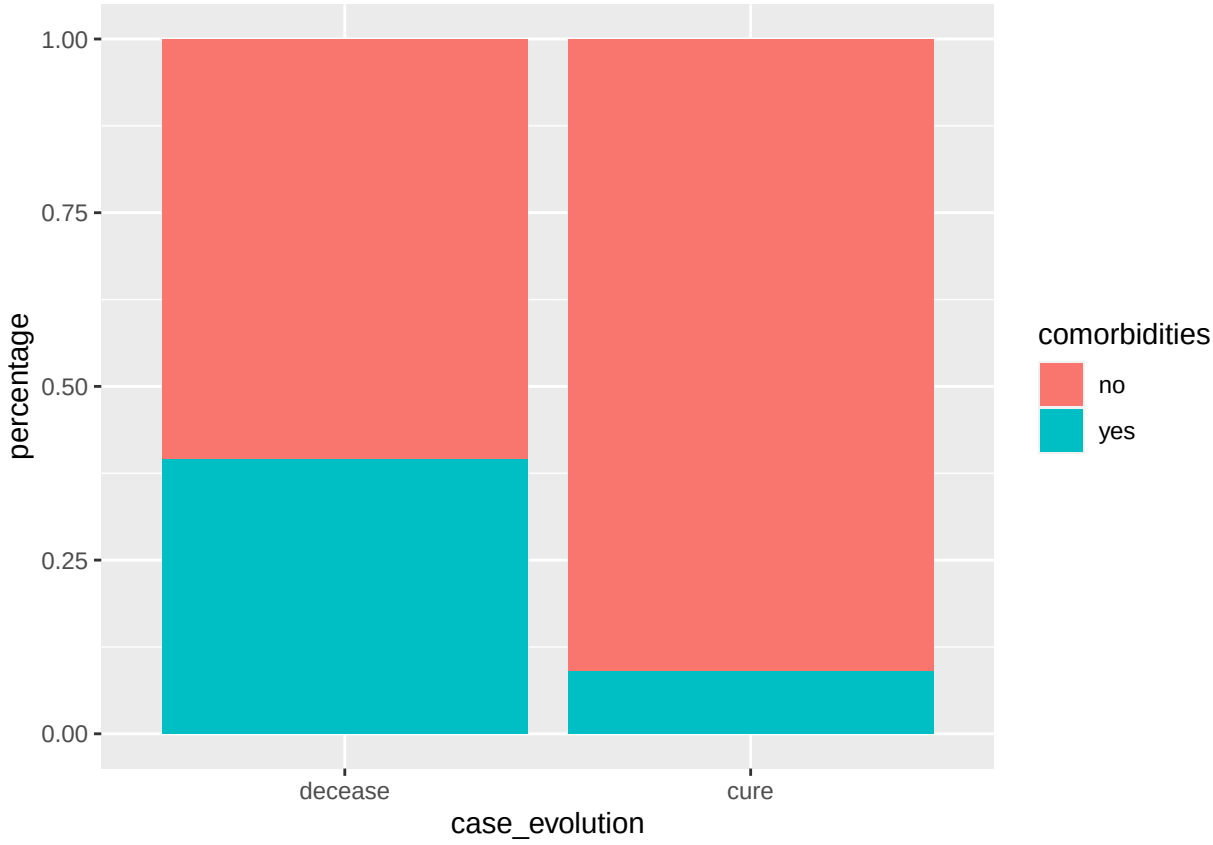
As noticible, there is little to no variation, I don't think it was going to be usefull.

state	deceased	cured
	0.810	0.190
ACRE	0.039	0.961
ALAGOAS	0.034	0.966
AMAPÁ	0.056	0.944
AMAZONAS	0.006	0.994
BAHIA	0.015	0.985
CEARÁ	0.021	0.979
DISTRITO FEDERAL	0.020	0.980
ESPÍRITO SANTO	0.013	0.987
GOIÁS	0.012	0.988
MARANHÃO	0.024	0.976
MATO GROSSO	0.017	0.983
MATO GROSSO DO SUL	0.001	0.999
MINAS GERAIS	0.013	0.987

state	deceased	cured
PARÁ	0.019	0.981
PARAÍBA	0.000	1.000
PARANÁ	0.020	0.980
PERNAMBUCO	0.000	1.000
PIAUÍ	0.041	0.959
RIO DE JANEIRO	0.006	0.994
RIO GRANDE DO NORTE	0.000	1.000
RIO GRANDE DO SUL	0.000	1.000
RONDÔNIA	0.014	0.986
RORAIMA	0.000	1.000
SANTA CATARINA	0.038	0.962
SÃO PAULO	0.001	0.999
SERGIPE	0.000	1.000
TOCANTINS	0.056	0.944

Comorbidities

The comorbidities, as I inicially suspected, are much more present in the deceased group than in the cured one:



We can also see the presence of each comorbidity:

case_evolution	comorbidities	diabetes	cardiac	respiratory
decease	0.396	0.236	0.119	0.081
cure	0.090	0.036	0.028	0.023

renal	immunosuppression	chromosomal_abnormality	pregnant
0.040	0.028	0.010	0.001
0.004	0.006	0.002	0.002

Model 1

This was basically a ensemble between: - A linear model to estimate the probability of death, based on the age of the patient, following a model of the sorts:

$$p(age) = P(case\ evolution = decease \mid age = age) = \alpha + \beta age$$

- Some bias calculation for each one of the comorbidities that was calculated trough conditional probability, using the formula:

$$p(comorbidity) = P(case\ evolution = decease \mid comorbidity = yes) = \frac{P(evolution = decease)}{P(comorbidity = yes)}$$

This biases were then added to the probability of death estimated by age, leading to the final probabilities. Then, using a function to take the Balanced Accuracy and the test set we tuned for the best cutoff value for p, than we used that value on the final model, trained in the remaining set, and the final results on the validation set.

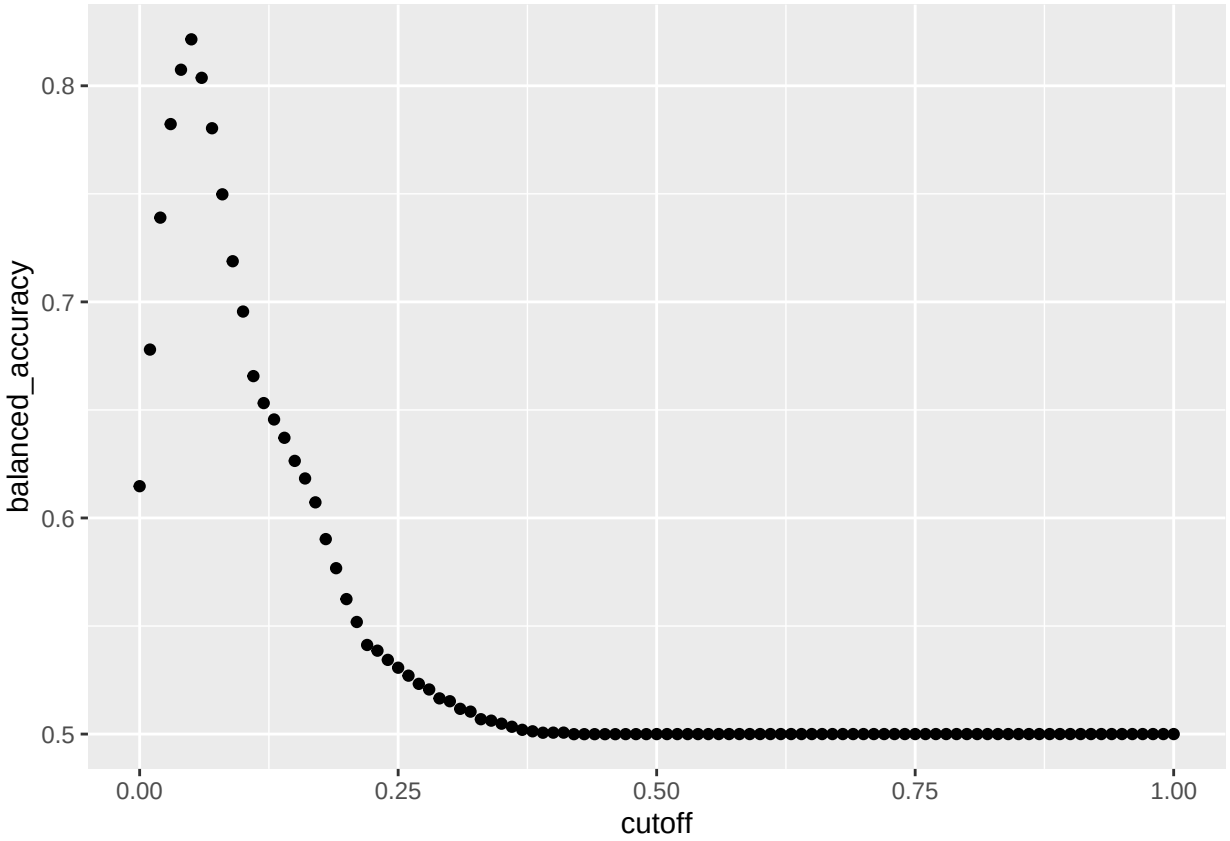
Model 2

Due to the size of the dataset, low power of my machine and class imbalance, I decided to reduce my train and test data sets by ignoring some of the cured cases, a technique called under-sampling. I did by taking all the deceased cases, than sampling the same amount of cured cases. Thus I had a perfect balanced dataset, and quite smaller as well. So I could easily train a Generalized Linear Model, a k-Nearest Neighbors, and a Decision Tree. Than I made two ensembles: - In the first one, I had then afirming decease if any of the three sad decease. Or only sad cure if every one of the three sad cure. - In the second one it would go by marjority. Compared the two in the test set, and the first one was more successfull, so thats the one I used on the Validation test set to get the final results.

Results

Model 1

After tuning for the best cutoff value, I found that 0.05 would give me the best balanced accuracy:



After that I retrained using the bigger dataset, and got the final result in the validation dataset:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  cure decease
##    cure    31882     107
##    decease  7095     621
##
##                Accuracy : 0.819
##                  95% CI : (0.815, 0.822)
##     No Information Rate : 0.982
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.118
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8180
##             Specificity : 0.8530
##          Pos Pred Value : 0.9967
##          Neg Pred Value : 0.0805
##              Prevalence : 0.9817
##          Detection Rate : 0.8030
##    Detection Prevalence : 0.8057
##       Balanced Accuracy : 0.8355
```

```
##
##        'Positive' Class : cure
##
```

Model 2

After training the methods on the test sets, I got this balanced accuracies:

X	model	balanced_acc
1	Generalized Linear Model	0.846
2	k-Nearest Neighbors	0.838
3	Decision Tree	0.837
4	Ensemble - 1	0.844
5	Ensemble - 2	0.839

And in the validation test set, this were the results:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  cure decease
##    cure    31665     100
##    decease  7312     628
##
##                Accuracy : 0.813
##                  95% CI : (0.809, 0.817)
##     No Information Rate : 0.982
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.115
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8124
##             Specificity : 0.8626
##          Pos Pred Value : 0.9969
##          Neg Pred Value : 0.0791
##              Prevalence : 0.9817
##          Detection Rate : 0.7975
##    Detection Prevalence : 0.8000
##       Balanced Accuracy : 0.8375
##
##        'Positive' Class : cure
##
```

Comparing to model 1 - final we improoved a little in regards to balanced accuracy, and in sensitivity, which, considering the nature of the data, is more important than the overall accuracy. Even though it wasn't the improovment I was expecting I still consider it valid, and satisfied me.

Conclusion