COVID-19 Risk Prediction | DL Course project

Meir Nizri – 312237563

Netanel Indik – 311300784

## Abstract

This article contains a final report on a machine learning project whose purpose is to build a model that, given a COVID-19 patient's current symptom status and medical history, will predict whether the patient is in high risk or not. In particular, the main measure that will determine the quality of the model is to successfully predict a high percentage of all those at high risk, while minimizing the percentage of low risk the model falsely predicted they are at high risk.

This report contains a description of the dataset on which the model is practicing and what steps were taken to prepare it for that purpose. The report describes three different types of models: simple logistic regression, logistic regression with a neural network and convolutional neural network, the steps in constructing each of them and comparison between the models based on their final results.

## Introduction

Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. Most people infected with COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness.

During the entire course of the pandemic, one of the main problems that healthcare providers have faced is the shortage of medical resources and a proper plan to efficiently distribute them. They have been in the dark, failing to understand how much resource they can assign for their patients when even in the very next week COVID-19 curve could sway very unpredictably. In these tough times, being able to predict what kind of resource an individual might require at the time of being tested positive or even before that will be of great help to the authorities as they would be able to procure and arrange for the resources necessary to save the life of that patient.

Using the model described in this report healthcare providers will be able to prioritize patients effectively and thus reduce mortality rates.

## Dataset

The dataset for this project obtained from [Kaggle](https://www.kaggle.com/tanmoyx/covid19-patient-precondition-dataset). It was provided by the Mexican government ([link](https://www.gob.mx/salud/documentos/datos-abiertos-152127)). This data set contains a huge number of anonymized patient-related information including pre-conditions. The analysis of this project will not necessarily apply for other countries; However, it will give a general picture that serves the purpose of the model.

The columns the dataset contains are: id, sex, age, patient type, entry date, date symptoms, intubated, pneumonia, pregnancy, diabetes, copd (Chronic obstructive pulmonary disease), asthma, inmsupr (immune system is suppressed), hypertension, cardiovascular (heart or blood vessels related disease.), renal chronic, other disease, obesity, tobacco, contact other covid, icu (had been admitted to an Intensive Care Unit) and death. Dataset consists of 563,201 unique values.

### Data cleaning and preparations

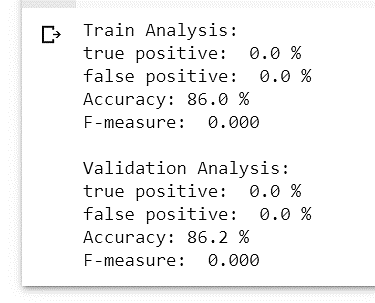
After analysis of the data a few actions were taken in order to make the data usable.

1. All patients who haven't tested positive for COVID were deleted.
2. For features which have many conclusive values all rows with inconclusive value were filtered. (affected features: sex, pneumonia, diabetes, copd, asthma, inmsupr, hypertension, other\_disease, cardiovascular, obesity, renal\_chronic, tobacco).
3. For features which have a very few conclusive values the entire feature deleted. (affected features: contact\_other\_covid, icu, pregnancy, intubed).
4. Excessive features deletion. (affected features: id, covid\_res, entry\_date, date\_symptoms).
5. All the data values modified to ones and zeroes to get it converted to One Hot Encoded data. The prepared data consists of 218,902 unique values and 15 columns.
6. The entire data is divided into three groups: training set (80%), validation set (14%) and test set (6%)

Even after cleaning and preparing the data set it's still unbalanced due to only 14% of the patients died. we will address this issue going forward.

## Project development process

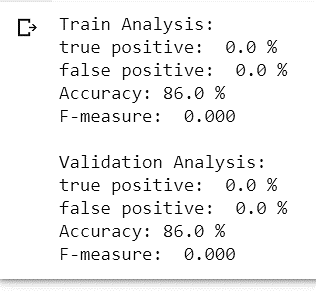
This section describes the construction process of three different models that all attempt to correctly predict the risk of Covid-19 patients: a simple logistical model, a neural network and CNN.



### Simple Logistic Regression

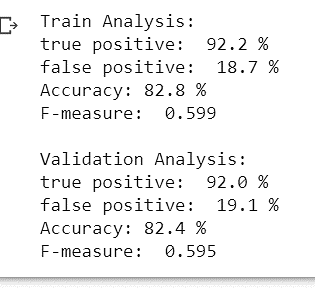
Logistic regression with 10k steps and a learning rate of 0.001. As can be seen, the prediction success rate was 86% but this fact alone does not tell the whole story. A quick look at the results will show that the training and validation models predicted only 0% true positive. A possible reason for the poor results is the data imbalance.

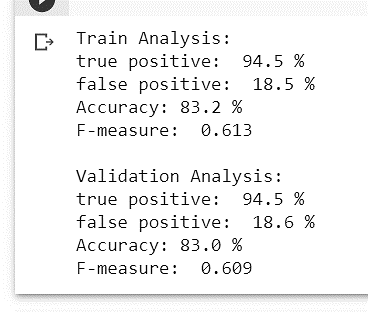
Two methods have been tried to get better results and address the data imbalance. The first method is to synthesizes new objects to the data using the SMOT technique and the second method is Loss Function Interventions technique, in which the loss function is calculated using the “tf.nn.weighted\_cross\_entropy\_with\_logits” function, where in the "pos\_weight" parameter the value is . Both methods did greatly improve the results, however the second method was more effective and was therefore chosen.

Simple logistic regression with 10k steps, a learning rate of 0.001 and with the c value as described yielded respectable results 0f 92% among those at high risk predicted right and only 19% of those at low risk predicted wrong.

### Logistic Regression with Neural Network

*First attempt*: simple logistic regression model with a hidden layer of 10 neurons. The model ran 10k steps with a learning rate of 0.001. The results are 86% accuracy, but again only 0% true positive, which means the model always predicts that there is no risk. This problem arises due to the imbalance of the data. As mentioned above, an efficient solution to this problem is to use the "Loss function interventions" technique.

*Second attempt*: logistic regression model with a hidden layer of 10 neurons. The loss function in this attempt was calculated with the “tf.nn.weighted\_cross\_entropy\_with\_logits” function, where in the "pos\_weight" parameter the value was . The model ran 10k steps with a learning rate of 0.01. As can be seen in the result, 92.2% of the true positive predicted successfully while only 19.1% of those with low risk falsely predicted at high risk. These results are very similar to the results of the simple logistical regression described above. In the next attempt, all sorts of different methods were added in order to improve the quality of the model.



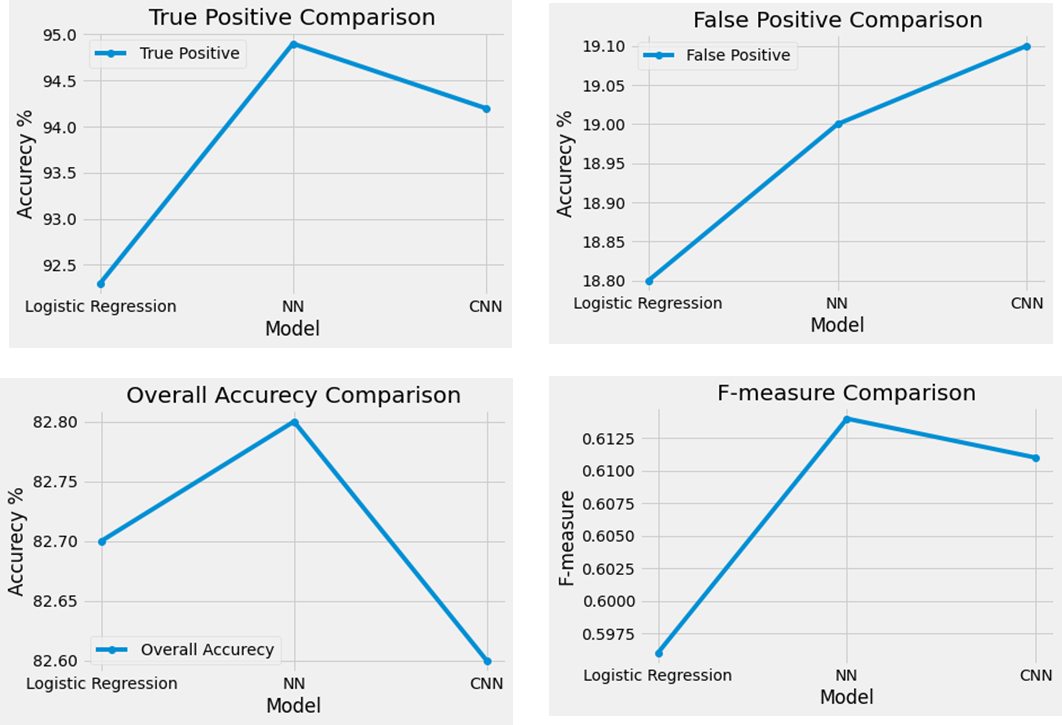
*Third attempt*: Logistic regression model with two hidden layers of 10 and 4 neurons respectively. Using Loss function interventions technique, Ridge regularization and Adam Optimizer. The model ran 50k steps with a learning rate of 0.01. The results are 94.5% accuracy among the high-risk cases and only 18.6% wrong among the low-risk cases. The F-Measure rate is 0.609, the best compared to the other models.

### Convolutional neural network

A relatively simple version of CNN. Contains one convolution layer with four filters (neurons) ending in max pooling and then two fully connected layers. The loss function was calculated with the “tf.nn.weighted\_cross\_entropy\_with\_logits” function. Here too the optimizer is Adam. The results of this model are 93.3% of the true positive predicted successfully while only 18.9% of those with low risk falsely predicted at high risk.

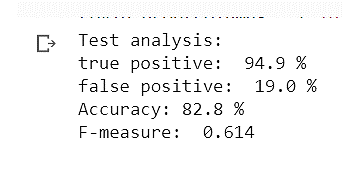
## Models Comparison

In order to select the best model, a comparison was made between the three models described in this report: simple logistic regression, logistic regression with a neural network and Convolutional neural network. the main measure that will determine the quality of the model is to successfully predict a high percentage of all those at high risk, while minimizing the percentage of low risk the model falsely predicted they are at high risk.

The differences between the models can be seen using the following diagrams created using the matplotlib library.

It can be seen that the model with highest true positive, accuracy and F-measure is the NN model by a relatively small gap and is therefore the final model. There is indeed a slight advantage to a simple logistic regression of 0.2% at the false positive rate, but this advantage is not significant enough given that the NN model predicts 3% more at the true positive rate. Another significant advantage of the NN model is that it is simple and easy compared to CNN so it requires less memory and faster.

## Project description

The final model for this project is Logistic regression model with two hidden layers of 10 and 4 neurons respectively. The loss function calculated using Loss function interventions technique, Ridge regularization and Adam Optimizer. The model ran 50k steps with a learning rate of 0.01.

The whole process of preparing and cleaning the data, together with the code for the various models and the code that runs them and returns results, can be found at the following link to Colab notebook:

<https://colab.research.google.com/drive/197KLi2FsIEiZfP9OaF1MW1MztsKiHnwC?usp=sharing#scrollTo=pF2PtHEC79OF>

## Conclusion

In summary the best model manages to classify a high percentage of 95% critical patients successfully while classifying about one-fifth of the mild ones incorrectly. The difficulty in predicting optimally may be due to the uncertain nature of the corona virus as well as the quality of the dataset.