Universidad Carlos III de Madrid

COVID-19 DETECTION

Final Project

MACHINE LEARNING

PABLO REYES MARTÍN NIA: 100409333
CLAUDIO SOTILLOS PECEROSO NIA: 100409401

Index

1. Definition and motivation of the problem

After this long quarantine caused by the Coronavirus, and now that researchers have made more advances in the investigation and the knowledge of the virus, we have set ourselves the challenge of investigating by our own with the personal objective of ending up knowing more about this virus that has affected the entire world in the last months.

1.1 Goal definition

Our main objective is to predict if an individual has had the COVID depending on the certain features that it presents. The features in which we will emphasize will be detailed in the next section.

For solving this problem, we will implement a supervised learning approach creating different types of detectors and evaluating them using ROC curves (AUROC evaluation).

1.2 Is Machine Learning a good approach?

We think that machine learning could be a good approach for finding these patterns in the features that the individuals present, being able to classify new individuals as positive or negative according to their status.

For solving this problem, we have decided to make a hybrid approach using techniques from Machine Learning and Statistical Signal Processing (plus some algorithms from Numerical Methods).

1.3 Machine Learning Process

Our machine learning process will consist on a data mining exploration. First, we collect the data from citizens from Spain. Since the data has been collected using formularies sent to our contacts, we will obtain an unbalanced dataset with respect to the amount of people that have recovered from the COVID. We will put special focus on the Madrid population since the majority of the instances are from this province.

We had to carry out a lot of data cleaning because some of the instances were not well defined (some people didn´t understand correctly certain questions of the questionnaire), for that reason we have cleaned the data.

Once the data has been cleaned, we transit to the second phase based on the selection of attributes (we will use the Weka tool, using the “select attribute” option) to get our optimal model.

Apart from selecting the more relevant attributes, we will divide our dataset in train and test. Also, we will balance the data so we can increase the percentage of prevalence.

After finishing the data treatment, we will proceed with the elaboration of detectors. We will explain how the detector works and give some mathematical feedback, but we won't talk about this very deeply.

For checking which detector is the best one for this data we will use the AUROC evaluation criteria. The best detector will be used for creating a detection model.

Finally, we will check the quality of our model by using the test data which we have previously separated for the deployment.

2. Data and Data Gathering

In this section we will explain how we have gathered the data and also give a little bit of insight on the attributes that we have decided to use for this study.

2.1 How we have gathered the data

The Data Gathering has been collected by generating a Google Form and broadcasting it to our families, friends, people close to our neighborhood, etc; via Email and WhatsApp.

Initially, we thought on going to health centers within our reach to make the survey, however this would take us a long time that we don't have, and also the data that we obtain wouldn't be very varied.

The formulary structure contains several questions to gather the information of interest for our study. The answer to each question will constitute an attribute of our data and each people that fills the questionnaire will be an instance for our database.

We truly trust in the answers of the people but some of them had no sense such that we have classified as noise for our compilation or we have format them directly. Other ones made sense, but they had misspellings which we had to correct or other kinds of errors like, for instance, if we asked for an integer number and someone answered "zero", we replaced this value by an integer (0).

It also should be pointed out that we have collected instances from Spanish population, but as the answers that take more contribution to our data came from Madrid, we have decided to focus mainly on this city to analyze just the citizens who live in the capital of Spain.

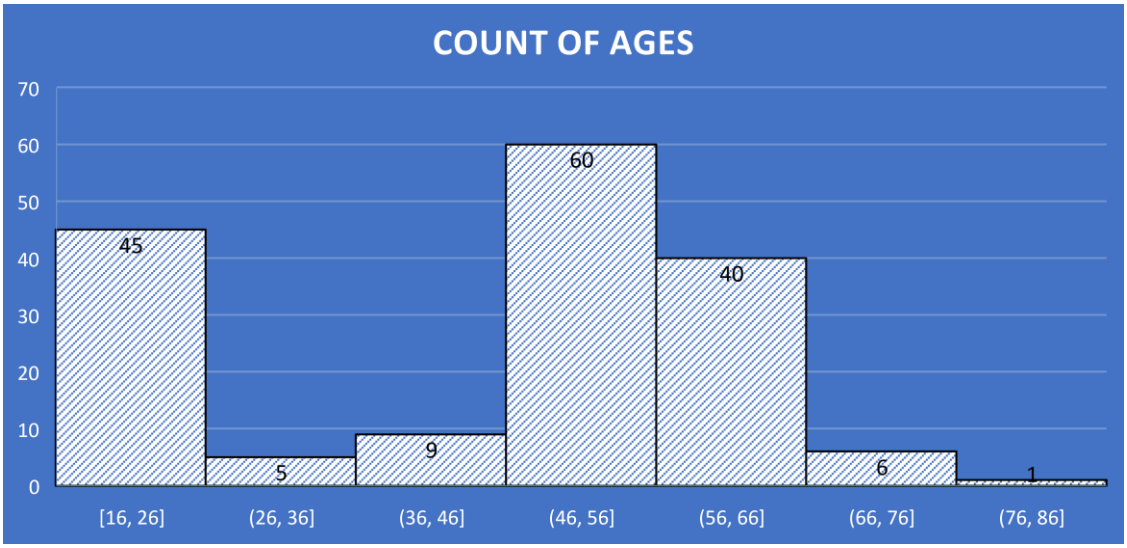
The resulting dataset that we have obtained consist of 211 instances, with 26 positive instances affected by COVID. It is a quite well result considering we just were two researchers, so we think that using Google Form for gathering data is a very powerful tool.

2.2 Attribute description

We didn´t ask very personal questions in the questionnaire. Also, we would like to give the credit of the selection of attributes to a doctor which is an acquaintance of ours.

The attributes that we have gathered are the next ones:

- *Age (Numeric):* We have obtained a range of values from 16 years old up to 83 years old. That means that we don´t have any information with respect children



- *Gender (Categorical):* There are more or less the same amount of men as women. (men = 71, women = 95).
- *Province (Categorical):* We have obtained instances of 14 different provinces of Spain, however the 78% of the instances are from Madrid.
- *Main Symptom (Categorical):* What we want people to answer in this section is which is the most common symptom that they have experimented.
  The symptoms that we decided to ask are: Dry cough, fatigue, fiber, smell disturbances, taste disturbances, diarrhea and others. Of course, there are people in the dataset which haven´t presented symptoms.
- *Duration of symptoms (Categorical):* We generalize the duration time of the symptoms in four categories; 1 week, 2 weeks, 3 weeks, and more than 3 weeks.
- *Symptoms starting Month (Categorical)*: This is the month in which they started noticing the previous mentioned symptoms (January, February, March, April, May).
- *Pneumonia diagnosed (Categorical):* We ask for the pneumonia as an extra attribute, differentiating it from the other symptoms considering that it is the most significant symptom of this virus.
- *Hospital (Categorical):* We ask the people if they have been hospitalized. If the individual has gone to the hospital and has been hospitalized (yes), hasn´t been hospitalized (no) or if hasn´t gone to the Hospital (hospitalize. (Yes/No/None).

- *Co-Habitants (Numeric):* This attribute provides information about the number of cohabitants in the same dwelling.
- *Co-Habitants with symptoms (Numeric):* Cohabitants which have presented symptoms during a period of time.
- *Co-Habitants with COVID-19 (Numeric):* Cohabitants that had been diagnosed with COVID.
- *PCR (Categorical):* Test which provides information about if the individual is currently infected by COVID. If it is positive means that the person has the COVID actually, negative if he/she has not suffered it and none if the person has not made the test.
- *Serology (Categorical):* Test which provides information about if the person has been infected by COVID or not. The labels are the same as in PCR test (positive, negative, none). What this test does is to detect antibodies which have been produced in response to the COVID. The two kinds of antibodies produced are IgM (this is the first antibody that is produced when the body fights an infection) and IgG (generated later than the IgM, but last much more in order to provide immunity during a longer period of time) [1]. In our questionnaire, having answered implies having tested positive for any of these antibodies.

Test results			Clinical Significance
RT-qPCR	IgM	IgG	
+	–	–	Patient may be in the window period of infection.
+	+	–	Patient may be in the early stage of infection.
+	+	+	Patients is in the active phase of infection.
+	–	+	Patient may be in the late or recurrent stage of infection.
–	+	–	Patient may be in the early stage of infection. RT-qPCR result may be false-negative.
–	–	+	Patient may have had a past infection, and has recovered.
–	+	+	Patient may be in the recovery stage of an infection, or the RT-qPCR result may be false-negative.

We attach this photo [2] for giving more feedback about the two main tests that are carried out for the COVID detection. It shows the different consequences of the combination of the test results.

- *COVID (Categorical):* Just indicates if a person has been diagnosed COVID or not. It is the target of our supervised learning study (what we want to predict).

There are several issues apart from cleaning and formatting data, such as the unbalanced proportion between the people who has COVID compared with the people who hasn´t, the unfamiliarity with the distributions of each attribute, and attributes that not contribute to the prediction of target. We will solve these issues in the next section.

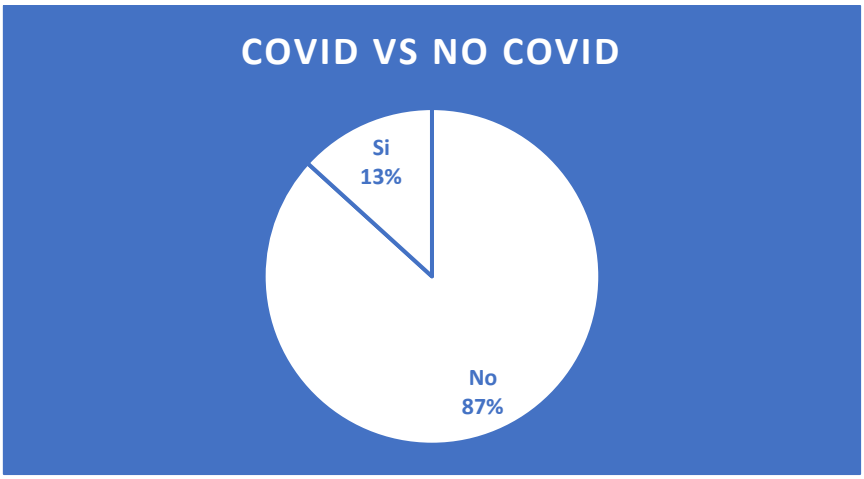
3. Data Pre-Processing

First of all, we thought if standardizing the data was a good idea. We have decided not to do this because we consider that scaling is useful for datasets with very varied data units. Since this isn´t the case of our data, we won´t scale the data.

In the former subsections we will explain how we have solved the problem of COVID prevalence and the selection of different kinds of attributes.

3.1 Data Balance

In this section we are going to focus on balancing the data because the positive labels in COVID are out of proportion with respect to the negative ones.



The balance can be made by two ways but one of them is going to be discarded because of being useless [3].

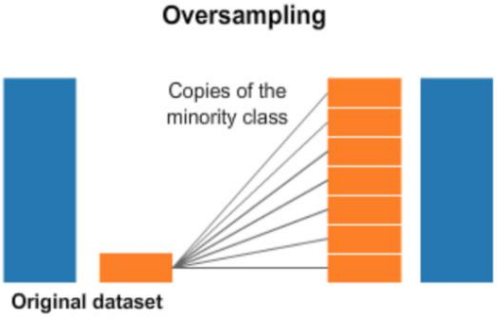
-Oversampling (discarded)

-Undersampling

3.1.1 Oversampling

The oversampling has the purpose of making copies from the sample which has less proportion of data. This technique is going to be discarded because having repeated instances will be useless to have a good train model and we want to have diverse data, not a monotonous one (with repeated instances).

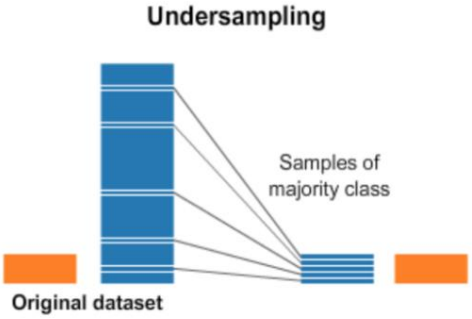
That is the reason why we discard this method.



3.1.2 Undersampling

The undersampling has the goal of deleting instances from the class which has the most proportion of data (the negative one). This technique is going to be useful since we can remove several instances which can be considered as "noise".

An example of these are the ones in which the tests (PCR and Serology) have not been performed, they don´t present symptoms and they haven´t had COVID. We have checked that more than the 50% of the instances are of this kind, so we can delete some of them (we consider these as redundant data).



UNDERSAMPLING PROPORTIONS

We will make three undersamplings:

- *1º Sampling:* 20% Positives - 80% Negatives
- *2º Sampling:* 28% Positives - 72% Negatives
- *3º Sampling:* 30% Positives - 70% Negatives

For the first sampling we will just eliminate some instances which we consider noise (No symptoms + None PCR + None Serology + No COVID) up to we reach the 20% of positives samples.

In the second one we will eliminate all the instances that are considered "noise". After eliminating all these instances, we get a 28% of Positives.

Lastly, besides removing the same noise instances as before, we will delete some instances which present:

- *No symptoms + Negative PCR + None Serology + No COVID* → We delete these instances because if an individual has a negative PCR means that he hasn´t the COVID currently and if he hasn´t perform the Serology test, we can´t know if he has had the COVID previously.
- *No symptoms + None PCR + Negative Serology + No COVID* → We delete these instances because although an individual has no antibodies (Negative Serology), he could be currently infected but since as the PCR test hasn´t been performed we don´t have much information from this kind of instances.

We would have liked to generate bigger positive proportions, but since we have a reduced dataset it hasn´t been possible.

3.2 Attribute Selection

The attribute selection is a relevant process for the prediction of our goal. For doing this we have followed some algorithm techniques to select just a few properties and forget about other ones which don´t contribute to the target prediction. We have taken three known algorithms in Machine Learning to select our attributes.

- Correlation Neyman-Pearson

- Information of Gain Ratio

- GreedyStepwise

3.2.1 Neyman-Pearson Correlation

The Neyman-Pearson correlation measures the lineal dependence between an attribute of our data and the target.

$$r_{x,y} = \sum \frac{(x_i - E[X])(y_i - E[Y])}{(n-1)\sigma_x \sigma_y}$$

$$x \equiv Arbitrary\ attribute$$

$$y \equiv COVID\ attribute$$

To carry out this algorithm, each attribute of our dataset is compared with the target goal (COVID attribute) to measure its dependence. We have made a simulation in Weka to get the most related attributes with respect to the target.

In general, the attributes have some correlation with the COVID but does not take too much correlation to state that exists a linear dependence. For that reason, we have gathered a set of attributes which, some of them take some dependence respect to the class label and other ones are practically independent.

SET 1

Serology

Symptoms

Age

Pneumonia

Co-Habitants with Symptoms

Duration of Symptoms

```
        Correlation Ranking Filter
Ranked attributes:
 0.5817   13 PCR
 0.5488   14 Serology
 0.393    12 C.C.Covid
 0.281     6 D.Symptoms
 0.2698    8 Pneumonia
 0.2668    9 Hospital
 0.2139    7 S.S.Month
 0.2066   11 C.C.Symptoms
 0.1925    5 Symptoms
 0.1584    2 Sex
 0.0983    1 Age
 0.0927    3 Province
 0.0784    4 District
 0.0305   10 Co-Habitants
```

3.2.2 Gain Ratio Information

The gain ratio information of certain attribute measures how an attribute may separate the labels of the target.

If the gain ratio is maximum for an attribute, implies that it maximizes the separation between classes. We must find out these attributes just if we want to create a tree or another kind of model.

$$\max GR(A_i) = \frac{\max G(A_i)}{\sum_{j=1}^{nv(A_i)} \left(\frac{n_{ij}}{n} log_2 \frac{n_{ij}}{n}\right)}$$

$$GR(A_i) \equiv Gain\ Ratio\ of\ an\ attribute$$

$$G(A_i) \equiv Gain\ of\ an\ attribute$$

To implement this algorithm selection, we also have simulated a cross-validation gain ratio information in Weka to show how each attribute separates the classes of the target.

PCR and Serology are the best ones separating classes. Our second set is structured by attributes that maximize the gain ratio:

SET 2

Serology

PCR

Age

Pneumonia

Co-Habitants with COVID

Duration of Symptoms

average merit	average rank	attribute
0.271 +- 0.023	1.3 +- 0.46	13 PCR
0.25 +- 0.019	1.7 +- 0.46	14 Serology
0.165 +- 0.018	3 +- 0	12 C.C.Covid
0.109 +- 0.009	4.5 +- 0.67	9 Hospital
0.098 +- 0.039	5.4 +- 2.29	6 D.Symptoms
0.082 +- 0.017	6.3 +- 0.9	8 Pneumonia
0.066 +- 0.022	7.3 +- 2.37	1 Age
0.061 +- 0.01	7.8 +- 0.6	5 Symptoms
0.053 +- 0.003	9 +- 1.18	4 District
0.048 +- 0.008	9.4 +- 0.66	7 S.S.Month
0.02 +- 0.005	11 +- 0.77	2 Sex
0.013 +- 0.005	11.8 +- 0.75	3 Province
0.005 +- 0.015	13.1 +- 0.83	11 C.C.Symptoms
0 +- 0	13.4 +- 0.92	10 Co-Habitants

3.2.3 Correlation Based Feature Selection

This attribute evaluator measures how an attribute is correlated with the class and at the same time measures how is correlated with the rest of the attributes. The attribute which contains the highest rank will be the one who is highly correlated

with the class label but less correlated with the rest of attributes. That is implemented to avoid multicollinearity or dependencies among attributes.

Using different search methods in Weka, we conclude our last set of attributes.

SET 3

Age

Sex

Symptoms

Hospital

Co-Habitants with COVID

PCR

4. Modelling

Our model is not going to be based on classical algorithms as KNN, SVM or Linear Regression because we thought it would be too difficult to implement them. What we thought is creating a detector to check if a person has the COVID or not, based on two multidimensional Gaussians with their respective mean and variance covariance matrix.

One will represent the information of the people infected with COVID and the other will represent the uninfected ones.

To do that we consider that the data is going to be divided into train (dataset with which we create the detector and we estimate the parameters) and test (dataset for evaluating the model). The splitting is going to be random.

Once we have our train set, we are going to construct the model for this set. For doing this, first we split the train set in two datasets:

-The positive COVID instances

-The negative COVID instances

After obtaining the two datasets, we are going to create the detector such that we distribute our data according to two multidimensional gaussian distributions (Positive COVID / Negative COVID) whose parameters are going to be the base of the classification problem.

Summarizing, the modelling process has three different steps:

- *Creation of Multidimensional Gaussian processes*

- *Estimation of parameters*

- *Creation of detector*

4.1. Creation of Multidimensional Gaussian Processes

In this subsection we will classify the instances into positive and negative. We will assume that each dataset of positive and negative instances follows a normal distribution with mean and variance covariance matrix:

$$YES \sim N(m_{yes}, V_{yes}) \qquad NO \sim N(m_{no}, V_{no})$$

$$m_{target} \equiv Expected\ value\ of\ the\ target$$

$$V_{target} \equiv Variance\ Covariance\ Matrix\ of\ the\ target$$

"YES" represents the instances which have the COVID and "NO" the analogous instances. The mean of each distribution is going to be a vector of p dimensions, being p the number of attributes and the variance covariance is going to behave as a matrix of pxp dimensions, where the diagonal of the matrix is going to represent the variance of each attribute and the rest of the elements will be the covariances among different attributes. Representation of the parameters:

$$m_{target} = [m_{A1}, m_{A2}, m_{A3}, \dots, m_{Ap}] \qquad V_{target} = \begin{bmatrix} Var_{A1} & \cdots & Cov_{(A1,Ap)} \\ \vdots & \ddots & \vdots \\ Cov_{(Ap,A1)} & \cdots & Var_{(Ap)} \end{bmatrix}$$

$$m_{Ap} \equiv Expected\ value\ of\ an\ attribute$$

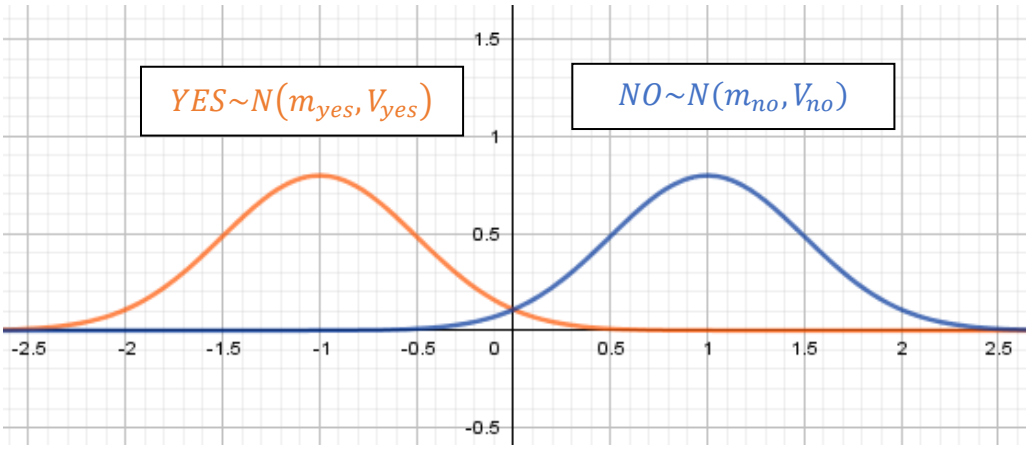
$$V_{Ap} \equiv Variance\ of\ an\ attribute$$

$$Cov_{(Ap,Aj)} \equiv Covariance\ of\ an\ attribute\ with\ other\ one$$

Knowing that both distributions are Gaussian we present the formula of the multidimensional Gaussian.

Suppose a random variable X follows a Gaussian distribution, then the probability of a random variable is:

$$X \sim N(m, V) <=> f(X) = \frac{1}{(2\pi)^{P/2}\sqrt{|V|}} e^{(X-m)^T V^{-1}(X-m)}$$



These Gaussians are not the real ones (not obtained with our real data), they are just an analytic visualization of how we want that Gaussians to behave.

Therefore, each Gaussian will store the information of each target distribution as we have mentioned at the beginning of the statement.

4.2 Estimation of Parameters

First, we consider that we want to create two gaussian distributions, in which each one will belong to a representation of positive or negative instances. But before creating this gaussians we must estimate their parameters.

Once the dataset is splitted (positive and negative sample) and their Gaussians are defined, we want to take some modifications for each attribute to estimate the mean and the variance of each attribute.

There are few attributes that there is no problem to obtain its mean and variance because of being numeric, the age for instance.

But, what about the ordinal ones?

The estimation of the mean and the variance for this kind of attributes is a little bit different:

-First each label of the attribute is discretized. For instance, the labels of Serology (positive, negative or none) are mapped into numeric values such that this feature can follow a uniform distribution.

$Serology = \{+, -, None\} \rightarrow \{0,1,2\}$

-Once the labels are discretized, we obtain their proportion to know how much they contribute to the attribute.

$$proportion = \{P(+), P(-), P(None)\}$$

Eventually, we have their proportions and the labels converted into numeric values, we can create a uniform function:

$$f(x) = \begin{cases} p(+)\ if\ x = 0 \\ p(-)\ if\ x = 1 \\ p(None)\ if\ x = 2 \\ 0\ otherwise \end{cases}$$

And once the distribution is created, we just have to estimate the parameters (mean and variance of each attribute).

$$E[X] = \sum_{i=1}^{n} x_i p(x_i)$$

$$Var[X] = \sum_{i=1}^{n} x_i^2 p(x_i) - E[X]^2$$

The rest of the attributes are discrete values (age, co-habitants with COVID, …) therefore the function is a discrete uniform and we can estimate the parameters (mean and variance) following the above formulas.

Apart from the variance and the expected value, we also need the covariances between variables, which is the dispersion between attributes.

$$COV[X,Y] = E\big[X - E[X]\big]\big[Y - E[Y]\big]$$

$$being\ X\ and\ Y\ random\ variables\ of\ different\ attributes$$

Of course, assuming that all attributes are discretized.

These are the necessary estimations to construct the distribution of the samples (positive samples and negative samples), considering that for developing a multivariate normal distribution it is needed its mean and the variance-covariance matrix, composed by the diagonal of variances of each attributes and covariances among attributes as explained in the above Section 4.1.

The estimation of parameters for both distributions provides a kind of confidence interval about the location where each observation can be pointed in an $\mathbb{R}^{pxp}$ space being p the number of attributes presented in an arbitrary set.

The estimation of parameters provides a representation of how both gaussians behave in a $\mathbb{R}^{pxp}$ space. The mean of each distribution is the centroid of the train observations and the variance-covariance matrix is the dispersion of observations around the centroid of each distribution.

4.3 Generation of Detector (Model)

First of all, we want to clarify that this section is going to be a general explanation of how to construct a detector with the data that we have at this point. This means that we will have to perform this process for all the combinations of the three sets of attributes and the three undersampling proportions, making a total of 9 detectors.

Each distribution follows a hypothesis, that in our dataset is the target goal. Then we assign our null hypothesis as the observations of positive COVID and the other for the negative COVID:

$$H_0: NO$$

$$H_1: YES$$

For the creation of the detector we will need to compute two main kind of probability distributions:

- *A priori Probability* $\{P_H\ (h)\}$ → Discrete distribution which quantifies the probability of each o hypothesis (independently of the observations).
- *Likelihood Probability* $\{P_{X|H}\ (x|h)\}$ → Probability of the observations given the hypothesis. We assume a collection of distributions over the random variable X.

The priori probability is the proportion of observations that follows one hypothesis over the total observations of the dataset. In this case, we define two priori distributions.

$$p(Yes) = \frac{N_{H1}}{N_{H1}+N_{H0}} \qquad p(No) = \frac{N_{H0}}{N_{H0}+N_{H1}}$$

$$N_{H1} \equiv number\ of\ positive\ samples \qquad N_{H0} \equiv number\ of\ negative\ samples$$

For the likelihood we want to proof it how to compute it.

Let X be a random variable with k observations:

$$X = \{x_1, x_2, \dots, x_k\}$$

Assuming that the number of observations of each hypothesis is k, we say that the likelihood can be expressed as the joint distribution of the observation, that being independent, can be computed by the product of the normal observations:

$$p(X|Yes) = p\left(\{x\}^{k_{yes}}|m_{yes}, V_{yes}\right) = \prod_{i=1}^{k_{yes}} \frac{1}{(2\pi)^{P/2}\sqrt{|V_{yes}|}} e^{(x_i - m_{yes})^T {V_{yes}}^{-1}(x_i - m_{yes})}$$

$$p(X|No) = p(\{x\}^{k_{no}}|m_{no}, V_{no}) = \prod_{i=1}^{k_{no}} \frac{1}{(2\pi)^{P/2}\sqrt{|V_{no}|}} e^{(x_i - m_{no})^T {V_{no}}^{-1}(x_i - m_{no})}$$

Now that we have computed the two main ingredients, we are ready to build our "general" detector (we say general because as you vary the threshold you obtain different detectors).

We consider both likelihood proportions, but when an observation is introduced to our model, it may be classified as a positive or negative sample. Actually, the detector compares both likelihood proportions, and if it is greater or lower than a certain threshold, the observation is detected as negative or positive sample, respectively.

We know that if the likelihood proportion is greater than other, the observation will be detected as the proportion with highest rank:

$$P\ (x|No) \ \begin{matrix} >^{d=0} \\ <_{d=1} \end{matrix}\ P\ (x|Yes)$$

**That is a principle to create the detector*

At the same time, we can consider that the detector decides by a certain threshold.

The creation of the detector is given by the Likelihood Ratio Test (LRT):

$$If\ \frac{P\ (x|Yes)}{P\ (x|No)} > \mu \rightarrow Decide\ 1 \qquad If\ \frac{P\ (x|Yes)}{P\ (x|No)} < \mu \rightarrow Decide\ 0$$

Being μ a given threshold.

$$\frac{P\,(x|Yes)}{P\,(x|No)} \begin{matrix} >^{d=1} \\ <_{d=0} \end{matrix} \mu$$

➔ $(After\ some\ Algebra)$→

$$(x_i - m_{yes})^T {V_{yes}}^{-1}(x_i - m_{yes}) - (x_i - m_{no})^T {V_{no}}^{-1}(x_i - m_{no}) \begin{matrix} >^{d=1} \\ <_{d=0} \end{matrix} \mu'$$

$Being\ \ \mu' = 2\log(\mu) + \log|V_{no}| - \log|V_{yes}|.$

As we are in a gaussian case (both probability distributions are Gaussian) we will have to compute the ratio between the two gaussian distributions, and after applying some algebra you end up obtaining the above formula.

Actually, μ' can take different values and for each of these values we would obtain a different detector.

By obtaining the *Probability of Detection* (also known as True Positive probability) and the *Probability of False alarm* (also known as False Positive Probability) from each of these detectors (different thresholds) we could end up plotting a ROC curve. We will detail this process more in the next subsection.

	Actual Positives	Actual Negatives
Positive Predictions	True Positives (TP)	False Positives (FP)
Negative Predictions	False Negatives (FN)	True Negatives (TN)

4.4 Drawing the Roc Curve

The generation of the ROC curve is simple once the detector is elaborated. It is useful for checking how well classifies according to several thresholds. For computing the ROC curve, we will need to compute the following probabilities from the detector:

-Probability of Detection (True Positives): *The positive instances that are classified as positive.*

-Probability of False Alarm (False Positives): *The positive instances classified as negative.*

4.4.1 Probability of Detection (True Positives)

The probability of detection are all the instances that have COVID and they are classified as such. Basically, the proportion of instances well classified for H_0:

$$P_D = P(D = Yes|H_1 = Yes)$$

Regarding the detector, the probability of detection of an observation x can be expressed as the surface of the likelihood probability of the observation which we know that is a positive instance p(x|Yes), whose region domain is the one that belongs to all positive instances:

$$P_D = \int_{\min(\chi_{Yes})}^{\max(\chi_{Yes})} P(x|Yes)$$

$$\chi_{Yes} = \{x| \frac{P(x|No)}{P(x|Yes)} - \mu' >^{d=1} 0\}$$

4.4.2. Probability of False Alarm (False Positives)

The probability of false alarm are all the instances that have COVID and they are classified as instances that don´t have. Basically, the proportion of instances wrongly classified with respect H_0:

$$P_{FA} = P(D = Yes|H_0 = No)$$

In this case, the probability of the false alarm computes the surface of the likelihood probability of negative instances p(x|No) preserving the domain used in the previous section 4.4.1 Probability of Detection.

$$P_{FA} = \int_{\min(\chi yes)}^{\max(\chi_{yes})} P(x|No)$$

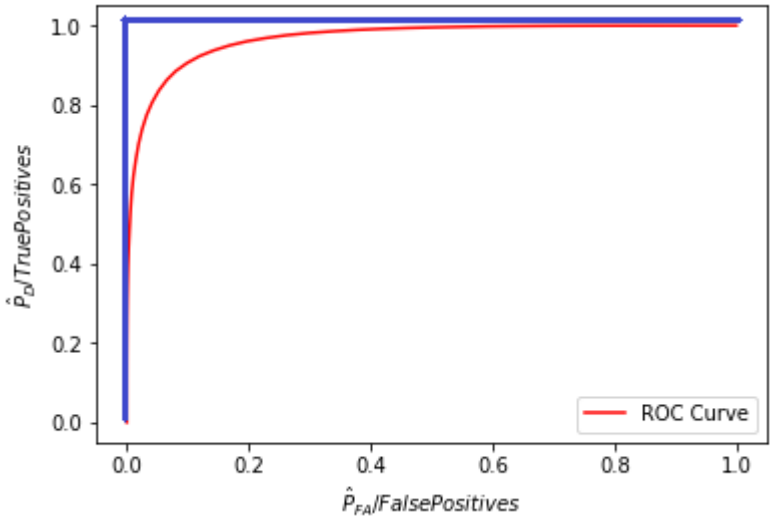
$$\chi_{Yes} = \{x| \frac{P(x|No)}{P(x|Yes)} - \mu' >^{d=1} 0\}$$

5. Evaluation

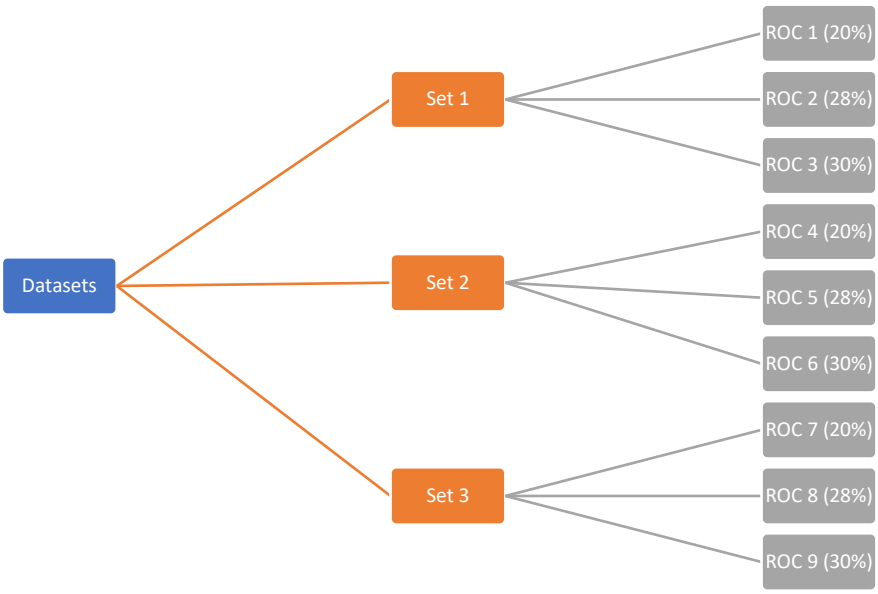
In our study we will have to obtain a ROC curve for each of the combinations mentioned at the beginning of section 4.3 (a total of 9 curves). Doing this we can find out which combination of attributes and which undersampling proportion generates the best detector.

Here is an example of how a ROC curve looks like (red curve).

The ideal ROC curve would be the one which approaches more to the blue curve.



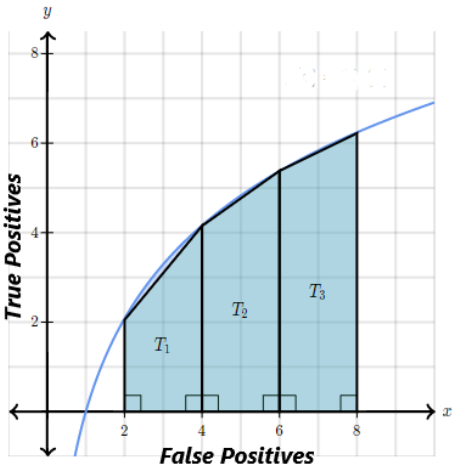
Here is a schedule of the nine ROC curves obtained from the different datasets.



5.1. Computing the Area Under the ROC Curve (AUROC)

In this subsection we will explain how the area of the ROC curve (AUC) is computed.

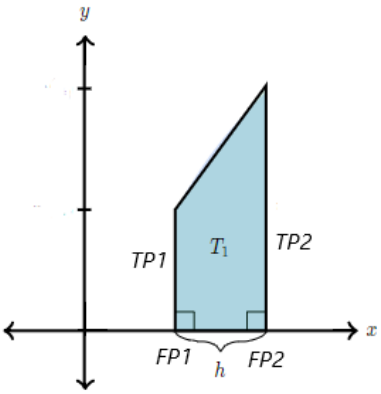
For computing this mathematically, since we don´t have the formula of the ROC curves we can´t obtain their area by normal integration. We will have to resort to a Numerical Methods technique called the Trapezoidal Rule [4].



This rule computes a very accurate approximation of a definite integral. It consists basically on dividing the curve into trapezoids of the same base length, computing their area and summing all the areas of the trapezoids.

Imagine that the photo on the right is a ROC curve, then we will have to divided into "n" trapezoids (T_1, T_2, …, T_n) and make the sum of these. The more trapezoids we use, the better the approximation we obtain.

Below we attach the formula for the computation of the integral through this rule. Clarify that TP_1 and TP_2 are the lengths of the lateral sides of the trapezoid and the value of H (height of the trapezoid) will be always the same.



$$H = FP_2 - FP_1$$

$$AUC_j = \frac{H}{2} * \sum_{i=1}^{N-1} TP_{i1} + TP_{i2}$$

With this technique we can compute the areas under the ROC curve, being able to compare them.

5.2. Bias reduction of ROC Curves

In this section we want to unbias as much as possible each ROC Curve that we create for each proportion sampling. For carrying out this process, we will assume that our dataset is larger than the one that we have.

As we said in the before sections, we are splitting the data in train (to generate the model and the representation of the ROC Curves) and test that are the observations used to deploy the model.

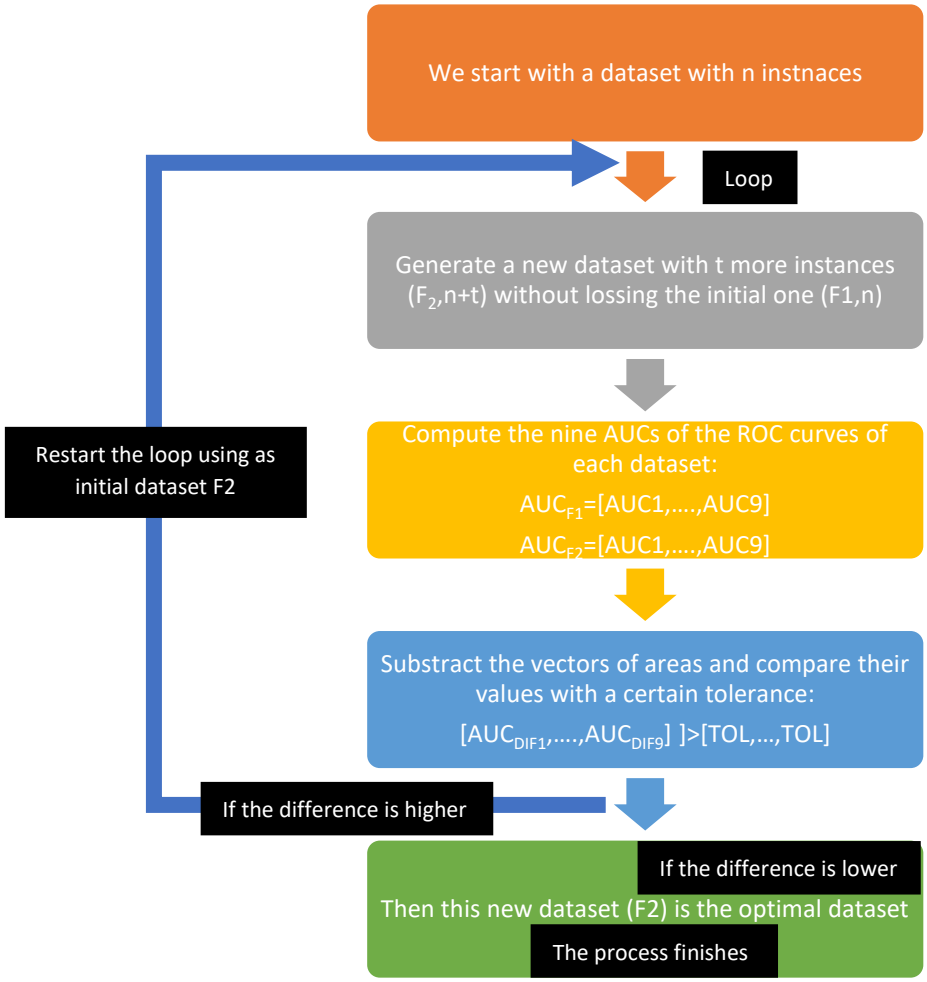
When we say "unbiasing" it means, getting more robust curves which have less error talking in terms of their performance. This process will have the following steps:

- We create k folds, starting with a dataset of n train instances (F_1) and creating a new one (F_2) adding some instances more (n+t). At the end of each fold, we keep the dataset with more observations (F_2) and becomes the new F_1 in the next fold.

- For each fold, we would compute the nine ROC curves of the datasets F_1 and F_2.

- If the difference of areas of the ROC surfaces is lower than a certain tolerance, the optimization will be stopped, returning a dataset with more instances that the initial train dataset.

- If the difference is greater than the tolerance, then the algorithm will continue up to finding a difference lower than the tolerance.



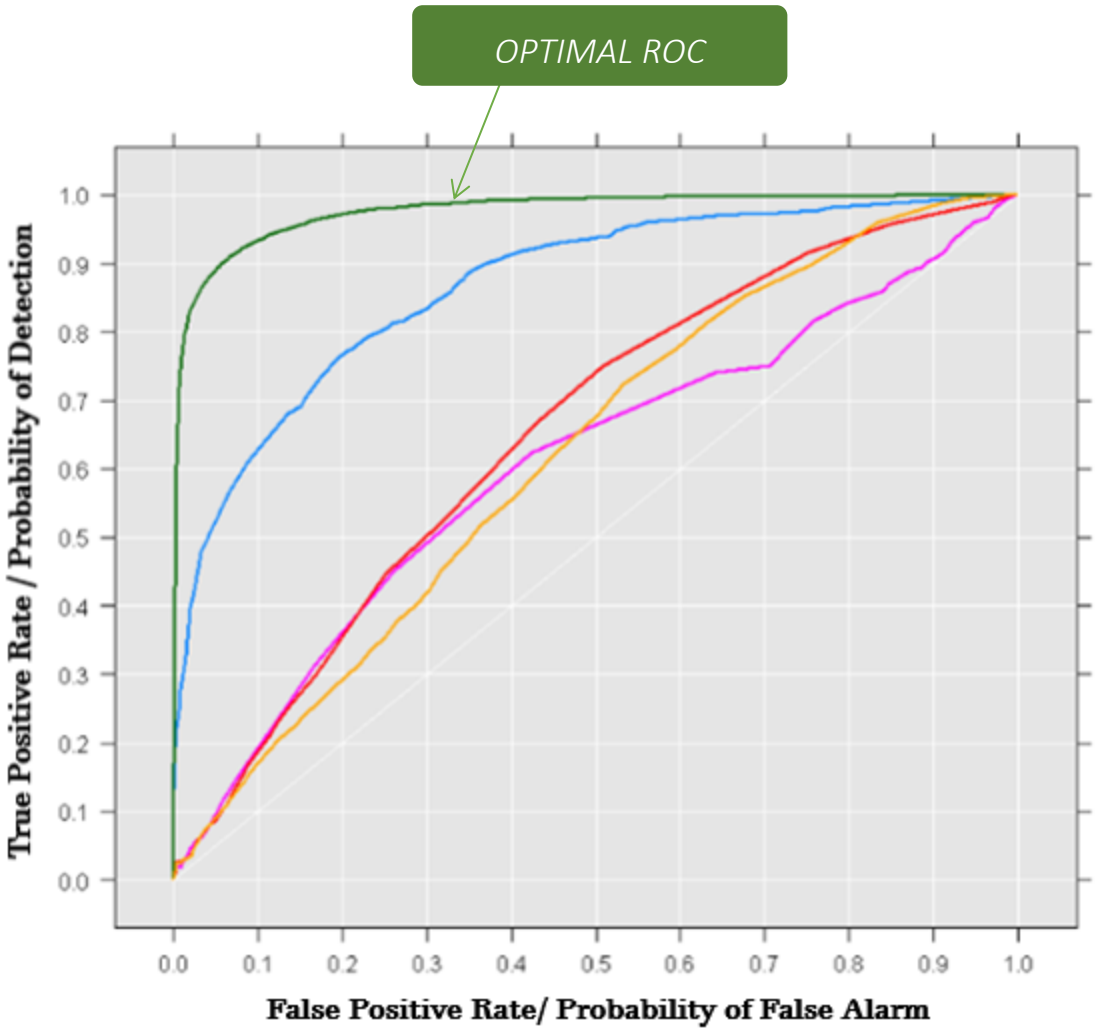
If a priori doesn´t find a fold that fulfills the stopping condition, this means that we will have to add more instances up to we find the optimal dataset (which produces the most robust curve).

5.3. Optimal ROC curve

When the unbiased train is computed, we will have to start the process of the selection of the best ROC curve which is, in fact, the one which belongs to our best model.

At this point, this process is almost done. We just have to find the ROC with the maximum area among the nine areas of the unbiased dataset (these areas have already been computed in the previous section). That is why more area provides better capacity for classifying a true positive.

$$ROC_{OPTIMAL} = max\{AUC_1, AUC_2, AUC_3, AUC_4, AUC_5, AUC_6, AUC_7, AUC_8, AUC_9\}$$



Imagine that the above plot, is the result of plotting the ROCs of the best dataset [5]. Now we must find the detector which works better in this situation.

Taking into account that we are talking of a very harmful virus (especially for old people), we want to have a very small false negative rate (deciding that the patients isn´t infected when it is infected). This equals to increase the true positive rate (detect not infected when actually is not infected). $TPR = 1 - FNR$

The ideal detector in the above photo, would be one with TPR between 0.9 and 0.95, having a very low FNR as well as a quite small FPR (we also are interested in reducing the type one error). Imagine we choose to design a detector with TPR 0.9; then, for finding the threshold of this detector, we just have to solve the below integral and find the value of μ′.

$$P_D = \int_{\mu'}^{\max(X_{yes})} P(x|Yes) = \ 0.9$$

Our definitive detector will be the one with that threshold. We will explain its functioning in the next section.

6. Deployment

In this final section, we will use our model generated in the previous section to simulate how it would work. The process done up to now is briefly summarized in the following steps:

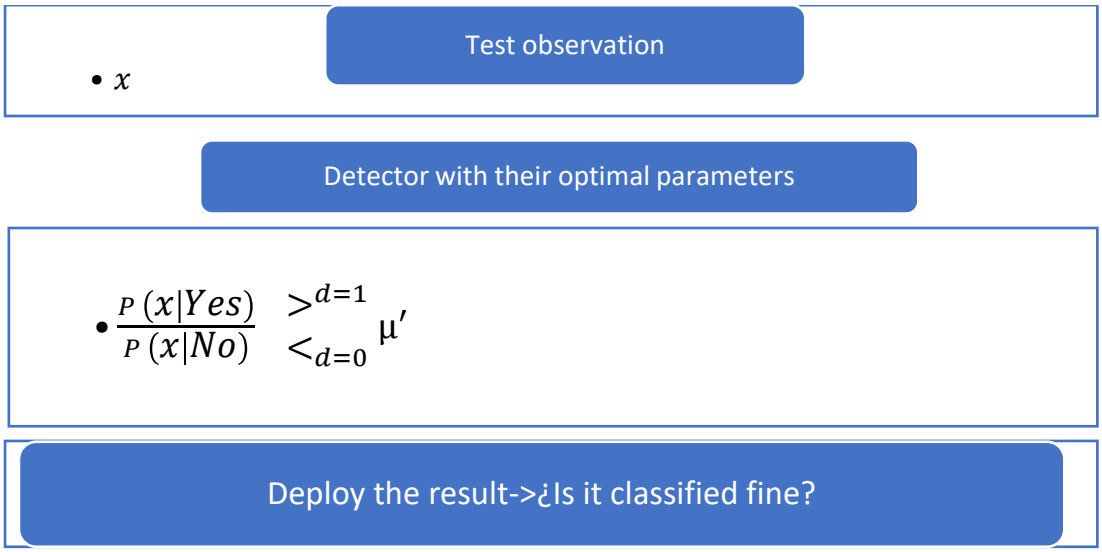
- Use the optimal set of attributes.

- Use the optimal balance of data.

- Choose the optimal ROC curve and parameters (means, variance covariances, AUROCs).

- Take the threshold for an arbitrary Probability of Detection that you decide (keeping in mind that for this case in particular we principally want to reduce the number of false negatives).

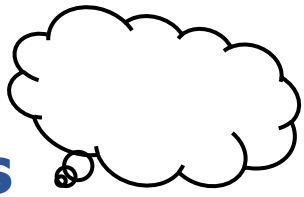
We just take a new instance (x) from the test dataset and we use the optimal parameters described above to classify the instance:



We don´t have enough data for carrying out this process and checking if it really works. However, if the deployment fails, we will try to review again our data gathering, set of attributes balance, generation and evaluation of model… to improve the model until it works fine.

There may be other better approaches for solving this problem, but we think that ours is an approach which makes sense. This is the best approach that we have been able to develop with our actual knowledge.

Another thing to point out, is that if the process were unfeasible because of the difficulty of the problem, we should confirm that is not possible solving if a person may have COVID or not.



Personal Conclusions

CLAUDIO

After performing this study, besides of having increased my knowledge with respect the COVID-19, I have learnt how to schedule a problem from scratch. I personally think that the approach that we have done is feasible, but we would need to increase our dataset. It would take some time for carrying out the development of the optimal detector, but once obtained, it could work as well as a quick test.

PABLO

I would like to implement all this by means of a university practice, but I think that the level contents mentioned in this study are too tough to execute this problem according to our Machine Learning knowledge. It has been interesting carrying out a theoretical supervised learning oriented to solve a social-health problem and perhaps it may be impossible to get a reliable answer introducing these automatization techniques.

References

[1] ‘Blood Test: Immunoglobulins (IgA, IgG, IgM) (for Parents) - Nemours KidsHealth’. https://kidshealth.org/en/parents/test-immunoglobulins.html (accessed Jun. 02, 2020).

[2] ‘Why Do We Need Antibody Tests for COVID-19 and How to Interpret Test Results’, *Diazyme Laboratories, Inc*. http://www.diazyme.com/covid-19-antibody-tests (accessed Jun. 02, 2020).

[3] W. Badr, ‘Having an Imbalanced Dataset? Here Is How You Can Fix It.’, *Medium*, Apr. 20, 2019. https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb (accessed Jun. 02, 2020).

[4] ‘Understanding the trapezoidal rule (article)’, *Khan Academy*. https://www.khanacademy.org/math/ap-calculus-ab/ab-integration-new/ab-6-2/a/understanding-the-trapezoid-rule (accessed Jun. 02, 2020).

[5] ‘What is a ROC Curve and How to Interpret It’, *Displayr*, Jul. 05, 2018. https://www.displayr.com/what-is-a-roc-curve-how-to-interpret-it/ (accessed Jun. 02, 2020).