# Table of contents

## *ABSTRACT*

The main objective of the project is to create a machine learning model, particularly a classification model (Garg, 2018), that will learn the pattern available in the dataset about COVID-19 survey in Brazil (Instituto Brasileiro de Geografia e Estatística, 2021), and later will give probabilities for new observations (information about new people) given to it.

The first step of the study was the cleaning of the data and making some transformations (removal of rows or columns, normalization, missing value imputation and etc.) (Pyle, 1999), so that it is possible to train different models on it. The problems of missing values, outliers, and oversampling were solved. All this was done using different metrics and graphical illustrations to find out insights about the data (the details are in the provided Jupyter Notebook). All this was done for one purpose: to be available to train a good model that can give us necessary predictions in the feature.

After preprocessing the data, 3 different models were created` Random Forest Classifier, K Nearest Neighbor Classifier and Gradient Boosting Classifier, various metrics were got for them, and later the models were compared, so that the best model for making our final evaluations was obtained. There are surely many more machine learning models that could be created and evaluated, but it was decided to use only the mentioned 3 in the scope of the project, because they are using 3 different approaches and that would be more than enough to find the best one and use it to meet the general goals of the project.

In the end it was concluded that "Random Forest Classifier" gives the best results when trained on the dataset, and with the help of it, it was possible to obtain the probability of being infected with coronavirus for each person present in the dataset. Each tree in the algorithm gives a prediction on whether the patient is infected or not, and the model simply counts the sum of predictions of being infected and divides it to the total number of predictions, thus getting the estimate of the probability.

A very good outcome of the project was obtained, as the model allowed to make high-quality predictions (particularly it gives a probability that the given survey-taker is infected with COVID-19) with accuracy varying somewhere

between 80 and 90 percent. So we ran the trained model on the whole dataset and calculated the probability of being infected for each person. Therefore, the created model can be a huge helping tool in the world of medicine, as with the help of it, Coronavirus can be diagnosed much more quickly, so more efficient treatments can be given to patients and many lives can be saved.

## *1.0 INTRODUCTION*

COVID – 19 has been a huge disaster for the whole world. Breaking in our lives in the beginning of 2020 it changed all the aspects of it. Economics of all the countries suffered hugely, and the population of the people decreased a lot, because nothing was known about the virus, people were not ready for it which led to more than 4 million deaths.

Although vaccination has already started, it is still very important to be able to diagnose COVID-19 in early stages so that some help could be given to people and reduce the spread of the virus as early as possible

The motivation for this project is to create a machine learning algorithm with the help of a data with information about COVID-19 symptoms and some other economical and regional factors from Brazil [2][3], which will take as an input all that information and will give a probability whether that person is likely to test positive or not. All that will help to diagnose the virus at an early stage and can help governments to define strategies that will reduce the bad impact from the virus on the country and its population.

The project will be executed by the following steps:

1) Translation of dictionaries about the data from Portuguese to English (the dictionaries are provided for different months, just like the datasets, but they will be combined into one final dictionary, that will include information about all the necessary variables),

2) Exploratory data analysis on the data, to get insights about it and get fully acquainted to select the necessary variables for the model (Section 3.1 - 3.2),

3) Creation of 3 different machine learning models – Random Forest Classifier, K Nearest neighbor, XG boost classifier (Section 3.3),
4) Comparison of the results of the created models and final model creation (Section 4.1 - 4.2),
5) Overview of the results (Section 5).

The whole project will be done in Python. Everything in details is shown in the provided Jupyter Notebook file.

# *2.0 MODEL CREATION*

## 2.1 DATA EXPLORATION

The information used for the project contains data about COVID-related surveys taken with people from Brazil for 7 months – from May 2020 to November 2020. The total dataset from all months contains answers to 148 different questions and 2650459 different observations (each observation is an interview with a single person). Therefore, the data is enormous, and some cleaning steps will be taken to keep all the necessary rows and columns and to be able to create the best possible model that will make accurate predictions taking an already existing data as an input.

During the exploration, the first thing was to create a graph of the distribution of people from different federation units:

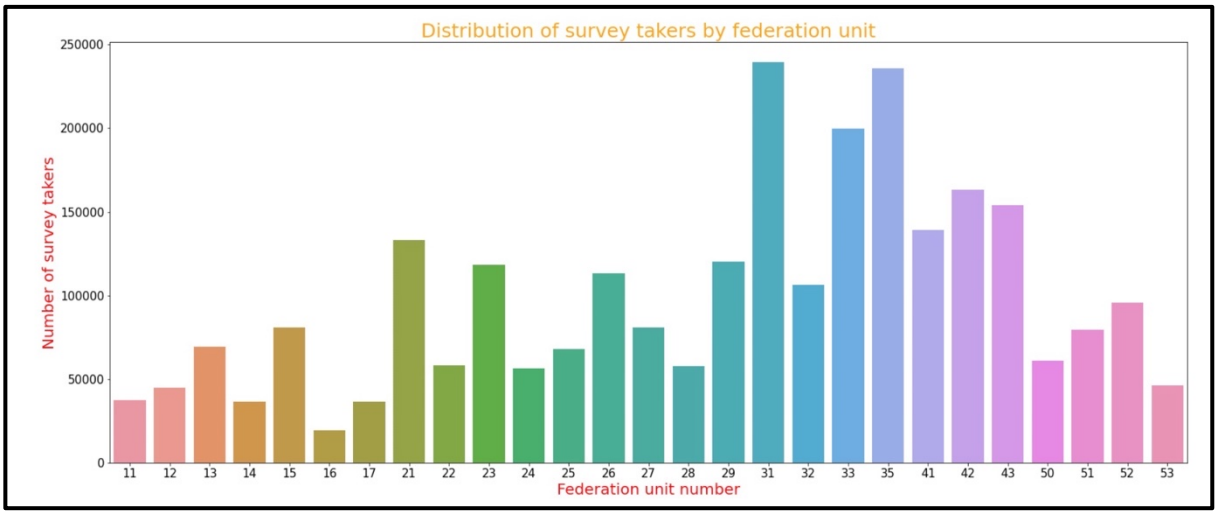

Figure 1. Distribution of survey takers by federation unit (federation unit names are listed in Appendix 1)

From this graph it was found that most of the people are from federation units 31 and 35, which are Minas Gerais and Sao Paulo.

The next step was to find out what is the number of people from each of the 4 area types available in the dataset:

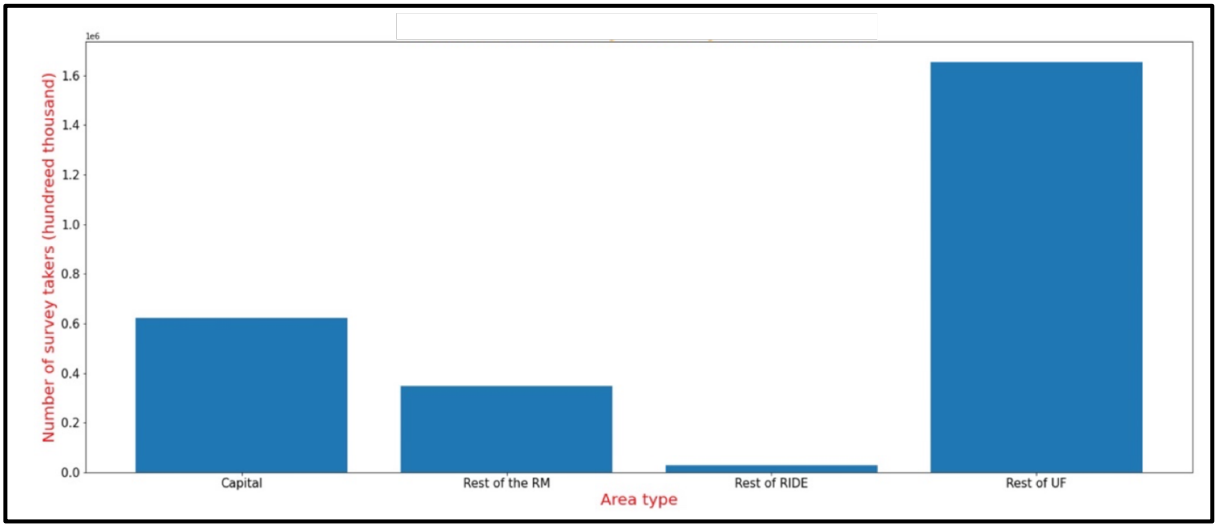


Figure 2. Distribution of survey takers by area type

Almost 600.000 people are from capitals of federation units, almost 400.000 are from Rest of the RM (Metropolitan Region, excluding the capital), approximately 50.000 people are from Rest of RIDE (Integrated Economic Development Region, excluding the capital), and most of the people, almost 1.600.000 are from Rest of UF (Federal Unit, excluding the metropolitan region and RIDE). It can be seen that high occurrence of people is present from 3 of the 4 area types, so this factor can surely be used in the analysis later, as we have enough variance.

A graph about the distribution of people based on home condition and some statistics about symptoms (included in Jupyter Notebook file) was created in order to get some more relevant information. From those graphs it was noted that most of the people who have taken the survey are people responsible for the household. Symptom statistics show that very little number of people have had symptoms. Logically thinking, symptoms might help a lot for this project when making

predictions about a person being infected or not with COVID-19, therefore in feature oversampling might be needed to be able to train the data in the right way.

Besides the material presented previously, people distribution by age and by sex was also inspected, and the results are shown in Figure 3:

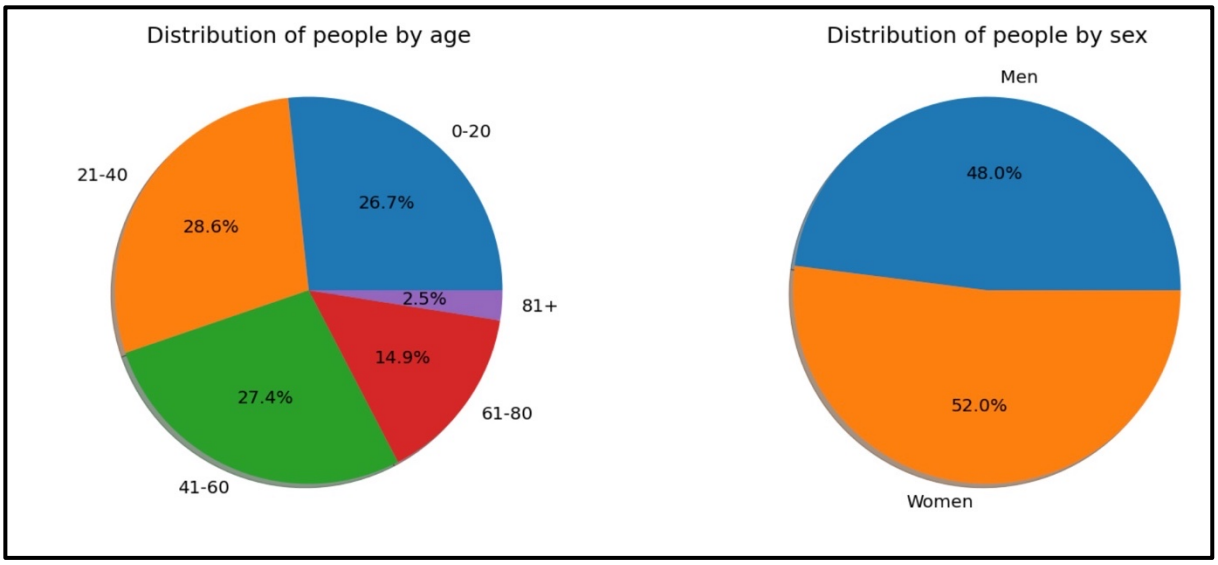


Figure 3. Distribution of survey takers by age and gender

Most people are in age group of "21-40", and number of men and women taking the survey is almost equal.

The results from the 3 types of COVID tests were also inspected to discover the variance between them:

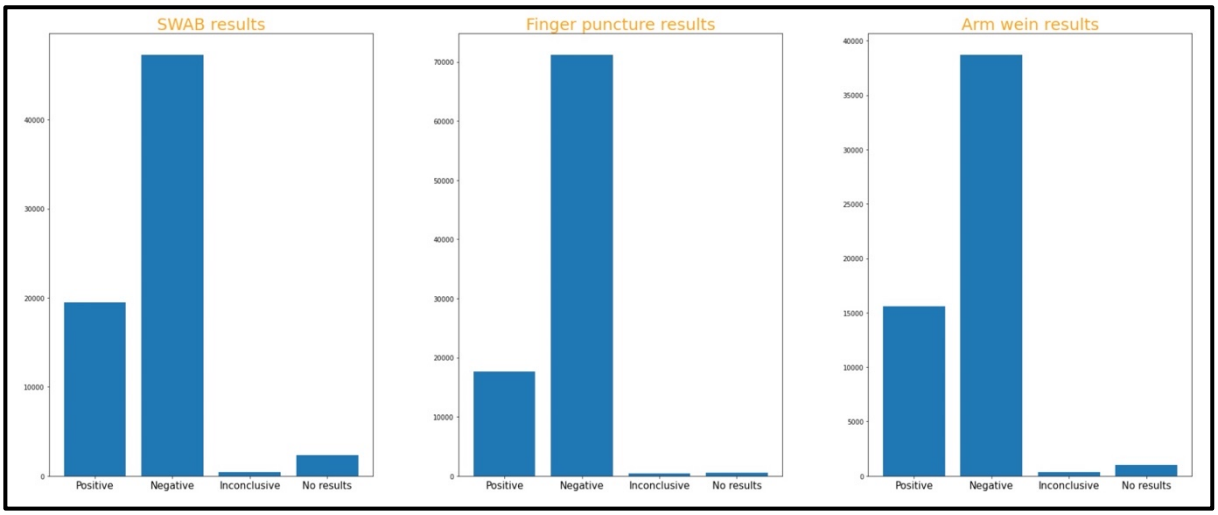


Figure 4. Distribution of results from 3 types of tests

Most of the test results are negative, but there is also solid number of positive tests, therefore this will allow to create a good model.
Later, some graphs regarding the reasons for people being away from work were created.

## 2.2 DATA CLEANING AND PREPROCESSING

After finishing the data exploration, the next step was cleaning the data. This is one of the most important steps in the analysis, as creating a model with an appropriate data will lead to much better results. Problems with the data like missing values, oversampling or outliers can hurt the model a lot, and even some "bad" rows might not let to create a good model. That is why this is very important step and must be done in a detailed way.

First, only the rows in which there was a test result from at least one type of test were retained, so that later the model could be trained using this data. After filtering, the data remained with 185,921 rows. Later the final model will be run on that rows too and predictions for being infected will be given for all the survey-takers in the dataset.

Then, it was the turn to choose only the features that could be useful for creating the models. Therefore, all the variables in the dictionary were covered and the table in Appendix 2 mentions the reasons for keeping or eliminating certain variables (all the results are made logically and not mathematically, also we have stored some variables in the same row as they would have the same values in the table). After removing the unnecessary variables, 34 variables (variables with label "Yes" label in the table) were left and 3 more variables where test results are stored.

If test results from the 3 types of tests are not different (meaning one is positive and one is negative), and a results from at least one test is available, then we will give "1" value to positive tests and "2" value to negative tests. We will also add the value "0" to the rows, when we have different results from different tests, or when the result from any test is not determined at all. Rows with "0" values will later be removed from the dataset, as they won't provide any help when training the model in the future. A single column for test results was created, thus the variables have 35 columns left in the dataset.

Next step is dealing with missing values. There are many different tactics for replacing the missing values in the dataset` they can be replaced by the mean value of the particular columns, or with the previous value or next value in the column. For this case, the substitution with mean values was selected, as distinct data rows are not connected with each other and there will be no use in replacing the missing values with the values from the previous or next rows.

34 variables are still too much, therefore correlation matrixes were created to find out which variables are more correlated with the target variable and remove unnecessary ones. As there are many variables, 2 correlation matrixes were created to be able to clearly observe the results:

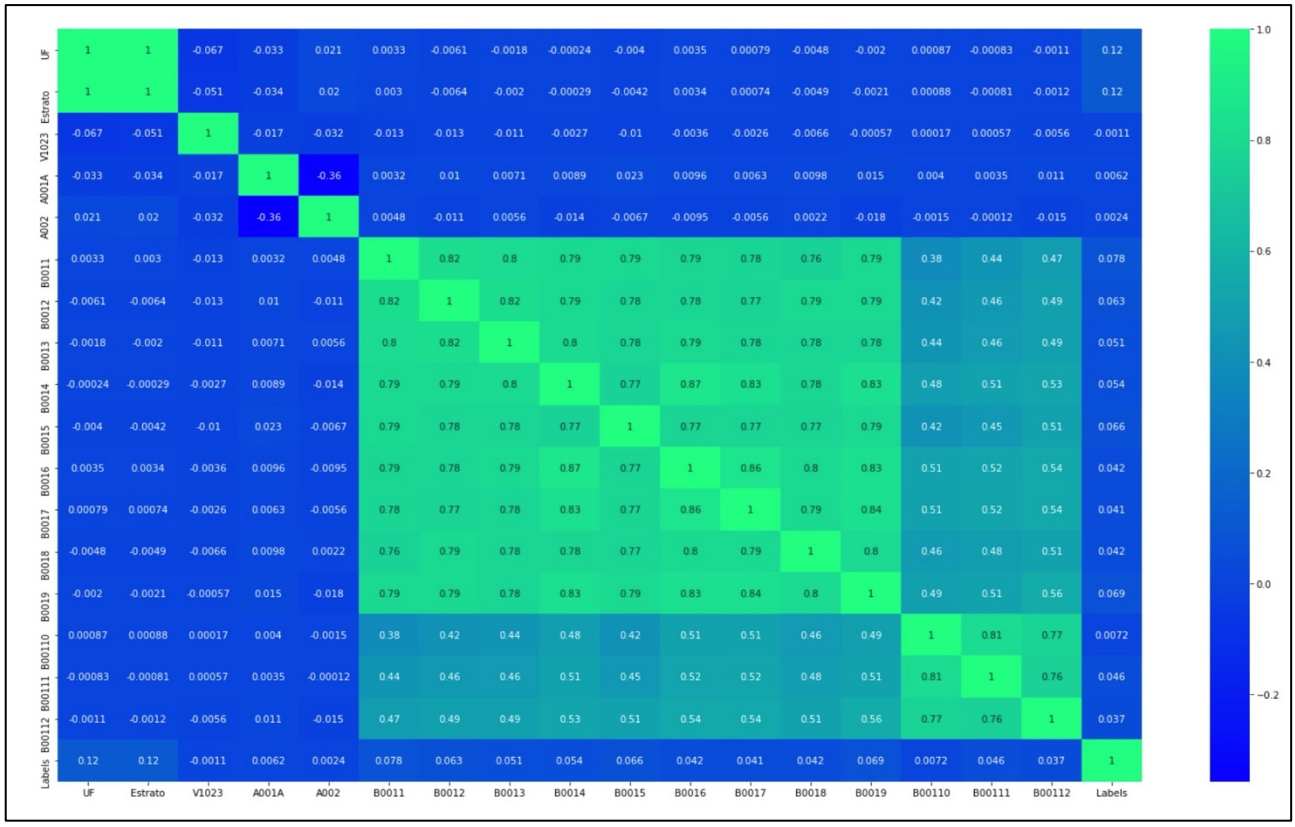


Figure 5. Correlation between first group of variables

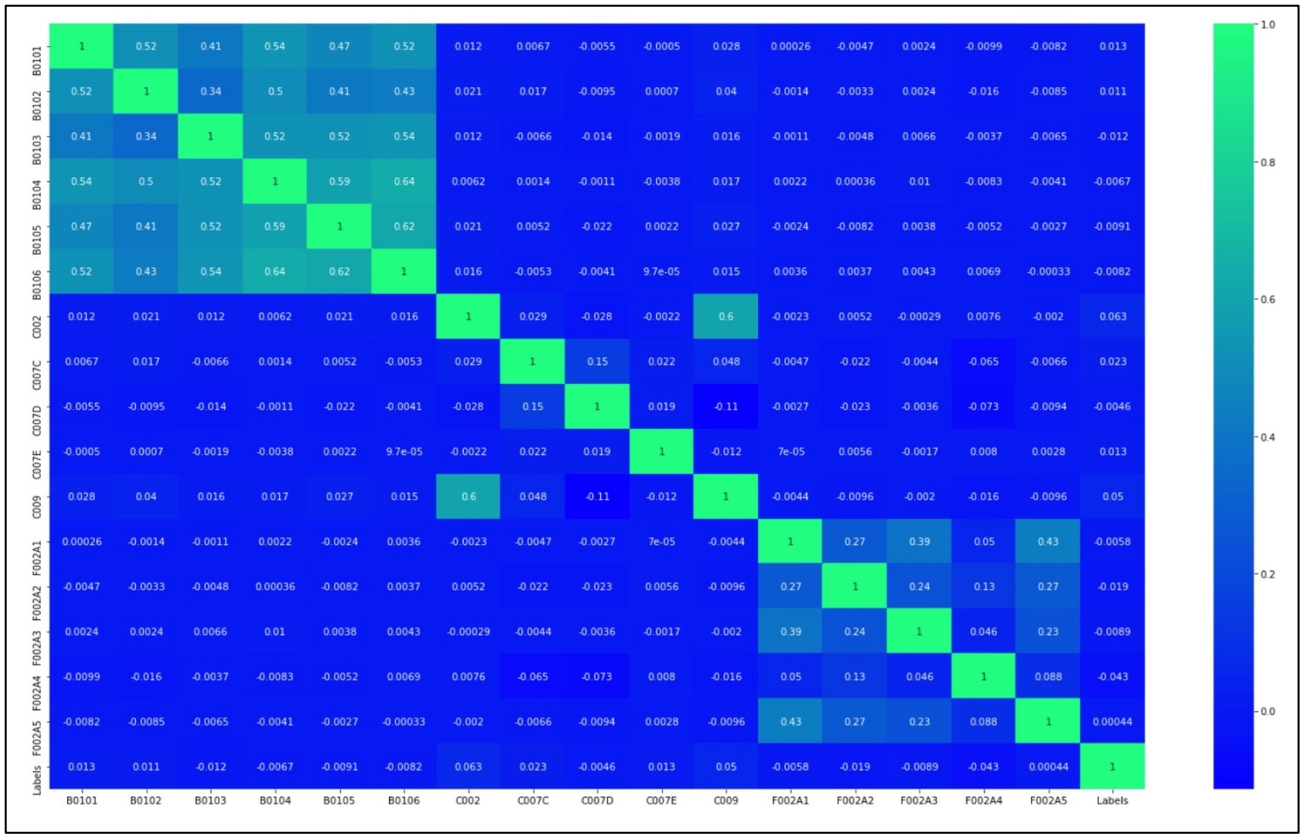


Figure 6. Correlation between second group of variables

Correlation is counted with the help of Python's Pandas library, which counts Pearson's correlation (Rodgers and Nicewander, 1988). There are some variables that have very low correlation with the target variable and they will not help in creating the models, as they are not connected with the target variable, therefore they were removed, as they would not give any help when creating the model. The variables that have correlation less than 0.01 were removed, so there are only 19 features left. We could have defined other thresholds, even bigger ones to remove more features but not to put the whole analysis under risk and remove necessary variables we have actually defined very low threshold, so some important variable, even if it has low correlation with the target variable, won't be removed.

There is also a need to solve outlayer problems for numerical variables there might be some values in some columns that are too far from mean values, so they need to be removed. In order to do that, box-plots and distribution plots were used. They will help to detect those far values and remove unnecessary rows from the dataset. Box-plots didn't help to detect any outlayers, but with distribution plots some problems were detected. The unnecessary rows were removed and

got the normal distribution for each variable (all the plots are presented in the Jupyter notebook file).

Another danger is oversampling. This is a situation when one of the values of the label is presented too many times compared to the others, so it is a compulsory step for the project. If the checking is done in the dataset, it can be said that the negative tests have more values than the positive tests, but the positive tests also have a solid number of values, so there is no problem with sampling and the project can move on to creating the models.

The final data has 178.226 rows, 18 columns representing variables, and one column presenting the label (predictive variable). In the next steps the data will be spitted to training and testing sets, so that the created machine learning models can be trained on the train datasets, and later their performance can be evaluated on the testing sets.

## 2.3 CREATION OF DIFFERENT MODELS

Three models were created: Random Forest Classifier, K Nearest Neighbor and Gradient Boosting Classifier (PEDREGOSA et al., 2011). The three chosen model use different techniques for making predictions and the purpose is to compare the performances of those 3 models and use the best one for the final evaluations.

Before creating the models, an empty dictionary was created, where each models contains information about different metrics about the performance of the model. Some predictions were made on whether the particular person is infected with COVID or not and different metrics of performance based on those predictions were calculated. For evaluation of the models,5 metrics were picked:

- **accuracy score**, which is equal to true predictions divided by the number of all the observations,
- **precision**, which is equal to true predicted negative test cases divided by the number of all negative test guesses,
- **recall (sensitivity)**, which is equal to true predicted negative cases divided by all the number of negative cases,

- **specificity**, which is equal to the number of true positive test guesses divided by the number of positive cases
- **negative** predictive value, which is equal to the number of true positive test guesses divided by the number of all positive test guesses.

Some functions to automate those processes were also created. First, a Random Forest Classifier model was created, and the following results from the model were observed:

- Accuracy: 0.81
- Precision: 0.84
- Recall | Sensitivity: 0.93
- Specificity: 0.44
- Negative predictive value: 0.68

The results were double checked by doing cross-validation on the dataset and got the following scores:

Scores: [0.79358133 0.83254313 0.83604994 0.82847524 0.8202553]
Random forest average score: 0.8221809870594614

The check passes normally, so Random Forest is a pretty good classifier as it gives results overall higher than 82%.

The confusion matrix of the results (1 for negative tests, 2 for positive):

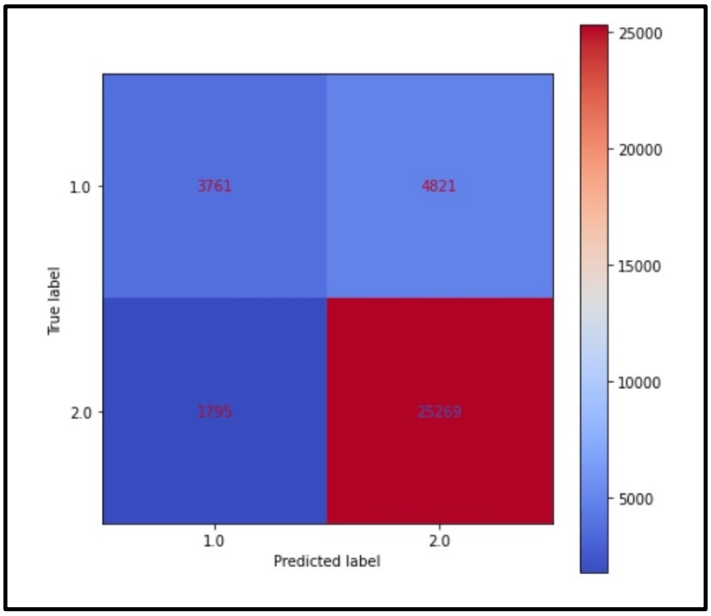


Figure 7. Confusion matrix of Random Forest Classifier

The top left cell in the confusion matrix shows the number of people that are infected with coronavirus and the model actually predicted that they are infected. The top right cell shows the number of people that are infected with coronavirus, but the model has predicted that they are not infected. The bottom left cell of the matrix shows the number of people that are actually not infected but have not been predicted so by the model. And the bottom right cell shows the number of people that are not infected with COVID-19 and the prediction for them is correct.

The next model was K Nearest Neighbor and got the following results:

- Accuracy: 0.77
- Precision: 0.82
- Recall | Sensitivity: 0.9
- Specificity: 0.36
- Negative predictive value: 0.53

Again, cross-validation was performed to check the results:

Scores: [0.75074342 0.79231309 0.78978819 0.77724786 0.77542432]

KNN average score: 0.7771033763870309

KNN also performs great, but it is a little bit worse than Random Forest Classifier. The overall score for the KNN is 77%.

The results from the confusion matrix:

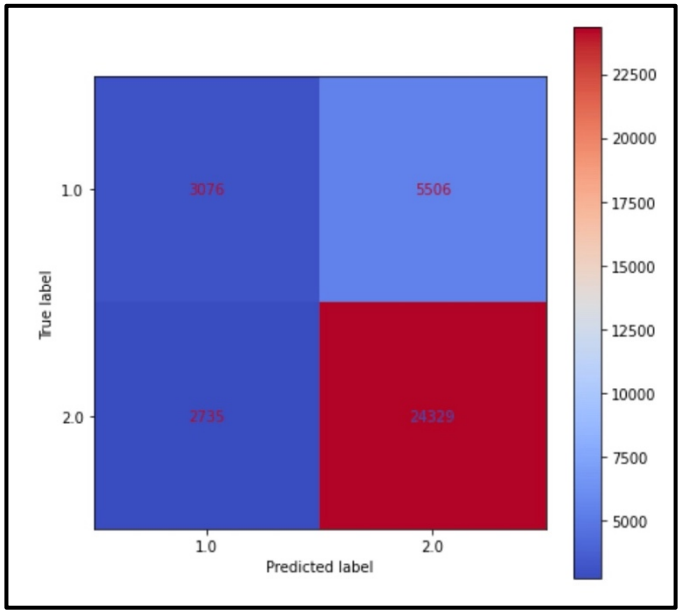


Figure 8. Confusion matrix of KNN Classifier

The last model created was Gradient Boost Classifier. The following result were obtained:

- Accuracy: 0.77
- Precision: 0.77
- Recall | Sensitivity: 0.99
- Specificity: 0.09
- Negative predictive value: 0.69

The accuracy of the predictions:

Scores: [0.78283678 0.77393744 0.76978538 0.76675551 0.76821434]
Gradient Boosting Classifier average score: 0.772305889539717

Gradient boosting gave a slightly worse results than KNN, again a little bit higher than 77%.

Confusion matrix of the results:

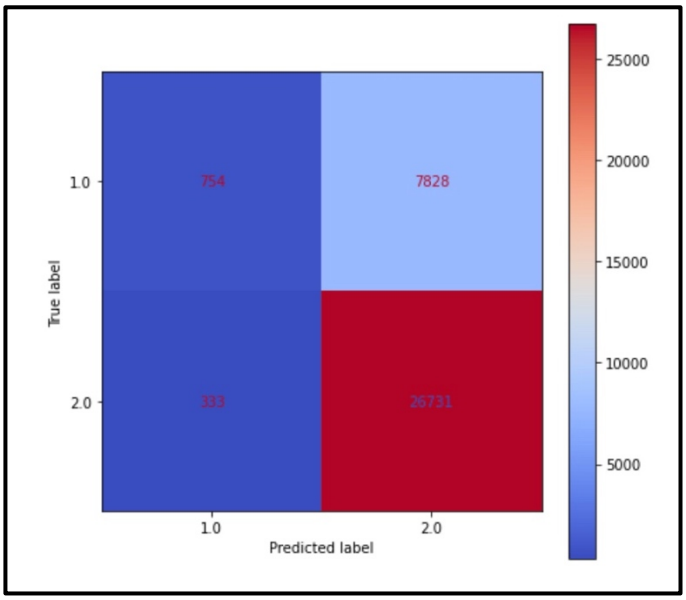


Figure 9. Confusion matrix of Gradient Boosting Classifier

The process regarding the models is done and the next step was to compare the results of them.

## 3.1 COMPARISON OF RESULTS OF DIFFERENT MODELS

To compare the models, a table was created in the dictionary in which the values regarding the performance of the models were kept:

| | Accuracy | Precision | Recall \| Sensitivity | Specificity | Negative predictive value |
|---|---|---|---|---|---|
| **Random Forest Classifier** | 0.81 | 0.84 | 0.93 | 0.44 | 0.68 |
| **KNN** | 0.77 | 0.82 | 0.90 | 0.36 | 0.53 |
| **Gradient Boosting Classifier** | 0.77 | 0.77 | 0.99 | 0.09 | 0.69 |

Figure 10. Performance metrics of the 3 models

Random Forest Classifier gives the best result in the most important metric: accuracy. It is also the leader in precision and specificity. Gradient Boosting is slightly better by recall and negative predictive value, but it has terrible results for specificity. Therefore, Random Forest classifier seems to be the best choice for creating the final model.

Random Forest is one of the best classification models available now. It uses different techniques, like bagging and boosting for generating different decision trees and finding out very accurate predicted values. The final model will be created using that type of classification technique and will give probabilities of being infected with coronavirus for all the people in our initial dataset.

## 3.2 FINAL MODEL CREATION

First the model will be trained again, but this time with hyperparameter tuning to find out the parameters that will provide the highest accuracy score for the valid data. Default parameters were used in the first model and here they are:
RandomForestClassifier(criterion='gini', min_samples_split=2, n_estimators=100, random_state=0)

After performing this step, the best model can be gained with the following parameters:

RandomForestClassifier(criterion='entropy', min_samples_split=3, n_estimators=200, random_state=0)

The next step is to fit this model on the entire dataset and give back the same dataset to see how well it predicts the values and what accuracy score is obtained.

A very high accuracy score of around 0.88 is obtained, which again shows that Random Forest Classifier is one of the best models available now. Here is the confusion matrix for the final model:

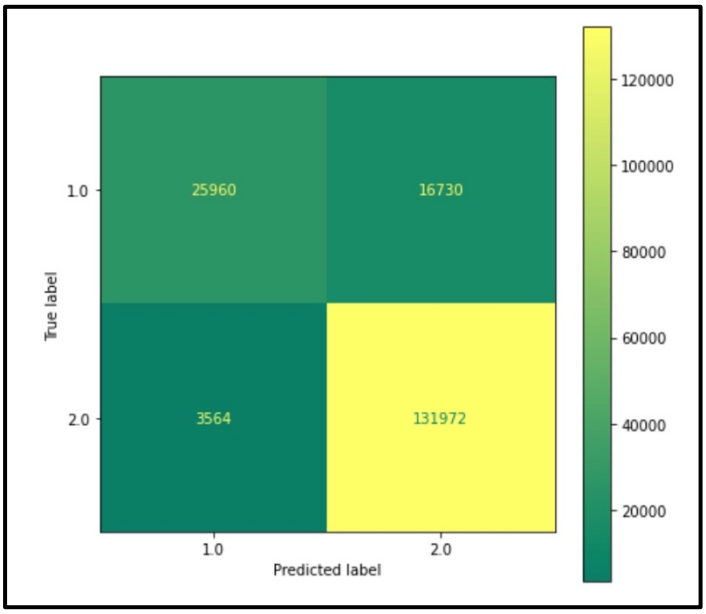


Figure 11. Confusion matrix of final model

After this, probabilities for each person in the dataset can be predicted and they will be added it as a new column called "Probability". Each value in that column shows the probability that the particular person in the dataset is infected with COVID-19.

Here is how the final dataset looks. It includes all the features that were used for model creation and the probability of being infected:

| | UF | Estrato | B0011 | B0012 | B0013 | B0014 | B0015 | B0016 | B0017 | B0018 | B0019 | B00111 | B00112 | B0101 | B0102 | C002 | C007C | C007E | C009 | Probability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 11 | 1110011 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 1 | 2 | 2.0 | 2.0 | 2.0 | 35.0 | 1.0 | 48.0 | 0.400833 |
| **1** | 11 | 1110011 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2.0 | 2.0 | 2.0 | 34.0 | 1.0 | 4.0 | 0.654583 |
| **2** | 11 | 1110011 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2.0 | 2.0 | 2.0 | 19.0 | 1.0 | 33.0 | 0.493633 |
| **3** | 11 | 1110011 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2.0 | 2.0 | 2.0 | 19.0 | 1.0 | 33.0 | 0.493633 |
| **4** | 11 | 1110011 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2.0 | 2.0 | 2.0 | 19.0 | 1.0 | 33.0 | 0.493633 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **0454** | 53 | 5310220 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2.0 | 1.0 | 1.0 | 2.0 | 1.0 | 0.0 | 0.567048 |
| **0455** | 53 | 5310220 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2.0 | 2.0 | 1.0 | 8.0 | 1.0 | 0.0 | 0.157333 |
| **0456** | 53 | 5310220 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2.0 | 2.0 | 2.0 | 19.0 | 1.0 | 33.0 | 0.332237 |
| **0457** | 53 | 5310220 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1.0 | 1.0 | 2.0 | 19.0 | 1.0 | 33.0 | 0.154862 |
| **0458** | 53 | 5310220 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2.0 | 1.0 | 2.0 | 19.0 | 1.0 | 33.0 | 0.374234 |

Figure 12. Portion of final dataset

## *4.0 OUTCOME AND DISCUSSION*

Predicting COVID is very complicated task, because the virus is new to everyone and it is difficult to find the most important variables. Even the doctors not yet have clear image of what are the symptoms of the virus or what pre-symptoms it might have. That is the reason that COVID-19 was such a huge disaster for the whole world and now many health companies are still working on creating efficient vaccines that will help people to stay away from the virus [10].

As there is not much information about COVID-19 yet, machine learning is the thing that comes on the first place. By using, data scientists might be able to find insights about the data that are not visible by human eye and find the most important variables that can hint us that the person is infected with COVID-19. Therefore, the model created in this project can be a huge help for the whole world to stand even stronger against the Coronavirus, as it will take some information about the patients as an input and will return the probability for each person to be infected with Coronavirus. In this case, people who receive high probability, can take tests at an early stages and in case they are infected, they can isolate early not to infect other people and break the chain of the virus.

The initial step of the project was just to create some charts and graphs to get acquainted with the data. Later came filtering some of the features by logic, because they surely don't have any connection with COVID-19. After it, some mathematical tactics were applied to keep only the most important ones. Particularly, correlation of the features was used with the target variable. After feature selection, 3 different models were created, and Random Forest classifier was the one with the best performance, the accuracy was 81%. Later hyperparameter tuning was applied to it, and the final model was created, which is a tool that can work with any amount of data. It returns the probability that the person is infected just by getting the necessary information from each row. It can be a powerful tool for preventing a large spread of the virus and detecting COVID in early stages.

The model created had very high accuracy (around 80%) and by doing surveys not only in Brazil but in other countries too, data scientist will be able to achieve even better results for future projects that will lead to an even higher accuracy.

# *REFERENCES*


GARG, R., 2018. *7 Types of Classification Algorithms.* [online] India, Raipur: Analytics India Magazine. Available from: https://analyticsindiamag.com/7-types-classification-algorithms/ [Accessed: 3.06.2021].

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATISTICA, 2020. *Datasets.* [online] Brazil: GovBR. Available from: https://www.ibge.gov.br/estatisticas/downloads-estatisticas.html?caminho=Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_PNAD_COVID19/Microdados/Dados [Accessed: 25.05.2021].

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATISTICA, 2020. *Dictionaries.* [online] Brazil: https://www.ibge.gov.br/estatisticas/downloads-estatisticas.html?caminho=Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_PNAD_COVID19/Microdados/Documentacao [Accessed: 25.05.2021].

PYLE, D., 1999. *Data Preparation for Data Mining*. Los Altos, California: Morgan Kaufmann Publishers.

PEDREGOSA et al., 2011. *Scikit-learn: Machine Learning in Python.* [online] Available from: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html [Accessed: 20.06.2021]

PEDREGOSA et al., 2011. *Scikit-learn: Machine Learning in Python.* [online] Available from: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html [Accessed: 20.06.2021]

PEDREGOSA et al., 2011. *Scikit-learn: Machine Learning in Python.* [online] Available from: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html [Accessed: 20.06.2021]

RODGERS, J. L., and NICEWANDER, W., A., 1988. *Thirteen Ways to Look at the Correlation Coefficient.* The American Statistician: Taylor & Francis, Ltd.

WORLDOMETER, 2021. *Covid-19 Coronavirus Pandemic.* [online] United States of America. Available from: https://www.worldometers.info/coronavirus/ [Accessed: 10.06.2021]

WORLD HEALTH ORGANIZATION, 2020. *Coronavirus disease (COVID-19): Vaccines.* [online] Switzerland, Geneva. Available from: https://www.who.int/news-room/q-a-detail/coronavirus-disease-(covid-19)-vaccines?adgroupsurvey={adgroupsurvey}&gclid=CjwKCAjwsNiIBhBdEiwAJK4khrSWzWcY7QDtilOsaiQGC6NstqnOP8c5PtdIpxqei64tNtxLqnOr7RoC7AcQAvD_BwE [Accessed: 30.06.2021]