**Building a Model for Covid-19 Diagnosis Based on Symptoms**

A DISSERTATION

Submitted to



Amity University, Kolkata

For the partial fulfilment of the award of the degree of

**BACHELOR OF SCIENCE IN STATISTICS (HONS.)**

By

Bijoy Kumar Dey

Enrolment No. A90555918017

Under the supervision of: **Prof. Soumya Banerjee**

MAY 2021

Amity Institute of Applied Sciences

Amity University Kolkata

**Major Arterial Road (South East), Action Area II, Newtown**

**Kolkata – 700017**

(India)

**TABLE OF CONTENTS**

|  |
| --- |
|  |
| **ABSTRACT** |
| **CHAPTER ONE: INTRODUCTION** |
| **CHAPTER TWO: OBJECTIVE** |
| **CHAPTER THREE: DATA** |
| **CHAPTER FOUR: METHODOLOGY** |
| **CHAPTER FIVE: RESULTS** |
| **CHAPTER SIX: CONCLUSION** |
| **REFERENCES** |
|  |

Page: 1

Abstarct

Classification problem plays an important role in the machine learning domain. ML models such as Logistic Regression, K nearest Neighborhood, Decision Tree, Ensemble Learning like Random Forest, Boosting and Xgboost are commonly used to solve classification problems. Among this Algorithms Decision Tree, XGBoost, Logistic Regression are most used.

I have used here Decision Tree and XGBoost for my project. In this case I use Israeli Ministry of Health who publicly released data for all individuals who were tested for covid-19 via RT-PCR assay of a nasopharyngeal swab. Here all the individuals’ symptom is given and also a particular individual positive or negative that is also given. On the basis of these symptoms we can predict that the individual is positive or negative by building a suitable statistical model. Here I take the data from November 2019 to September 2020. In this data, there are some NA values and I fill up the missing values by prediction of NA values using another available values.

After that the data is split into two parts: Train and Testing. Then I build a XGBoost model and Decision Tree model on Train Data and test the model under the test data. Comparing these two models I select one appropriate model with greater accuracy rate.

Introduction

The coronavirus-2019 disease caused by the sars-cov2 continues a critical and urgent threat of world. As in October 2020, the overall number of patients who are COVID-19 positive has exceeded 39500000, in greater than 180 countries and more than 1110000 people have died because of COVID-19.

This pandemic builds a challenge on medical systems worldwide, including sharp increases of demands for hospital beds, oxygen and shortage of medical equipment, when many health workers themselves are affected. The most preferable diagnosis test for COVID, using reverse transcriptase polymerase chain reaction test (RT-PCR), is a long term process. This test contributed to increased infection rates very quickly.

In this time we need such a process that can identify covid-19 positive patients very quickly. So now at this time we need such a model that use symptoms ,such as however, most previous models were based on data collected from hospitalized patients, thus are not effective in screening for COVID in the whole population.

Selecting a proper algorithm is vital for modeling any data driven problem. There have been handful article on covid-19 diagnosis based on symptoms by Yazeed Zoabi, Shira Deri-Rozov and Noam Shomron (2021). In this article they use machine learning to predict the covid-19 positive and negative. Here they use only the boosting algorithm.

Page: 2

Objectives of the

The classification goal is to predict that a patient is covid-19 positive or negative on the basis of his/her symptoms. Building a model which can fulfill the classification goal with more accuracy is the main objective of my project.

DATA

The Israeli Ministry of Health released datasets of all individuals those who were tested for COVID-19 via RT-PCR assay of a nasopharyngeal swab. At the first months of the COVID condition in Israel, all laboratory tests for COVID were performed according to the criteria determined by the IMH (Israeli Ministry of Health). The criteria implemented on the study period always included the presence of clinical symptoms, possible exposure of individuals who are confirmed to have COVID, certain geographic areas, the risk of complications if infected. Except for a small minority who were tested under surveys among health workers, all individuals who tested had indications for testing. Thus, there was no apparent bias regarding the majority of the subjects in the dataset which are used in this study; these contrasts with previous studies and for which such bias was drawback. In addition, all Negative and Positive COVID cases this data were confirmed by RT-PCR assay.

The list describes each of the dataset’s features which are used by the model:

A. Basic Questions:

1. Sex (female/male).

2. Age >=60 years (false/true)

B. Symptoms:

3. Headache (T/F).

4. Sore throat (T/F).

5. Fever (T/F).

Page: 3

6. Shortness of breath (T/F).

7. Cough (true/false).

C. Other information:

8. Known contact with an individual confirmed to have COVID-19(false/True).

Methodology of Project

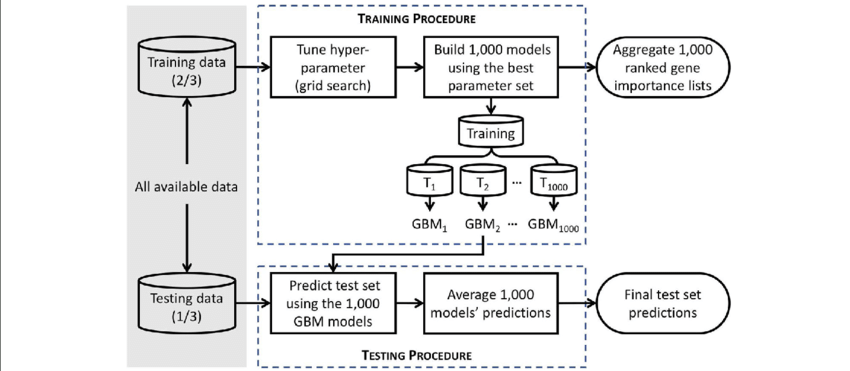
Xgboost model full form is extreme Gradient Boosting model. The xgboost model, though, basically refers to engineering goal to push limit of computational resources for boosted tree algorithms which is the reason for most used of this algorithm.

Xgboost model is an ensemble learning algorithm. Sometimes, it may not be sufficient to rely upon the results of just one machine learning model. Ensemble learning offers systematic solutions to combine predictive power of multiple students. The resultant is a single model which gives the aggregated output from several models. Xgboost models that form ensemble, also be known as base learners, could be either from same learning algorithm or different learning algorithm. Bagging methods and boosting methods are two widely used ensemble learners. Though these two techniques can be used with several statistical models, the most predominant usage with decision trees.

Features of Xgboost:

1. It can be run on both single and distributed systems.
2. XGBoost is used in supervised learning.
3. It supports parallel processing.
4. It Takes care of outliers to some extent.
5. Efficient memory management for large datasets exceeding RAM.
6. Xgboost has a variety of regularizations which helps in reducing over fitting.
7. It has auto tree pruning that means Decision tree will not grow further after certain limits internally.
8. It can handle missing values.
9. Xgboost is used in supervised learning (regression and classification problems).

1. It has inbuilt Cross-Validation.



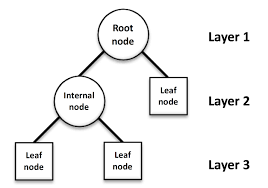
Page: 4

The decision tree is tree shape diagram or chart that helps determine a course of action or show a statistical probability. The chart is called a decision tree due to its resemblance to the namesake plant, usually outlined as an upright or a horizontal diagram that branches out. Beginning from the decision it (called a "node"); each "branch" of decision tree which represents possible decision or outcome, or reaction. The furthest branches on the tree represent the end results of a certain decision pathway and are called the "leaves".

The decision tree is graphical depiction of a decision and every potential outcome or result of making that decision. Individuals deploy decision tree in variety of situations, from something simple and personal to more complex industrial, scientific, or microeconomic undertakings.

By displaying a sequence of steps, a decision tree give people an effective and easy way to visualize and understands the potential effect of a decision and its range of possible outcomes. The decision tree also helps people identify every potential option and weigh each course of action against the risk and reward so that each option can yield.

An organization can deploy decision tree as a kind of decision support machine. The structured model allows the reader of the chart to see how and why one would choice may lead to next, with the use of the branches indicating mutually exclusive options. The structure can allow users to take problem with multiple possible solutions and to display these solutions in simple, easy-to-understand format that also shows the relationship between different events and decisions.



Page: 5

I use Decision Tree and XGBoost algorithm with R programming in this project. At first the dataset is imported in R-studio using read.csv. Here all variables have two or three factors like yes/no. So I convert the entire yes /no to 0/1 like that.

I check the data structure after that and notice that some variables are character and some are integer. So I convert all the variables to factors. Then I check the percentage of non-covid in dataset as non-covid case is more than covid case and I need such a model so that its confusion matrix accuracy rate is higher than the percentage of non-covid case.

Then I handle the NA values of datasets. I use here prediction method for handling this NA values. This method takes all the NA values as a test dataset and takes others where all numeric values are available, as a train dataset. Then we predict the NA values using the model which built using the train dataset (available dataset). I use “Mice” and “VIM” library function of R for this NA values prediction. This method is easy to interpret and predicted values are most probably near to the accurate values.

Then I realize that I will use Decision Tree for building a model since here all variables are factor and this is a classification problem. But Decision Tree has sometime over fitting problem so I use another algorithm of ensemble learning named XGBOOST and this XGBOOST algorithm give us a model with highest accuracy rate and this also give us a graph in which we can see which of the variables are important for this model from the dataset.

Before using the models I split the dataset in 70:30 ratios that means 70% of dataset is train data by which I can build the model and 30% of dataset is test dataset by which I can test my model accuracy.

For decision Tree, I use “party” library function. Here I also take minimum tree 10000 and get the tree model for the train dataset. After that I test the model on test data and get confusion matrix but it has over fitting problem so I use another model XGBOOST.

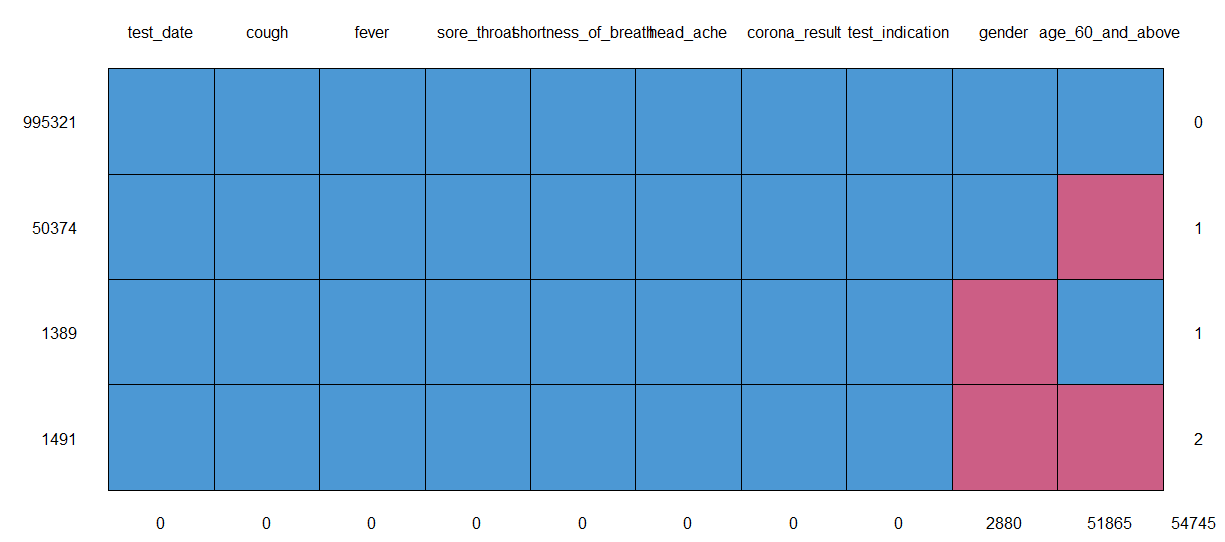
For XGBOOST algorithm, I use “xgboost”, “magrittr”,”Matrix” library functions. Here I need to convert my dataset into matrix form as XGBOOST run on matrix datatype. Here total no. of iteration is 100, maximum depth 6, eta value 0.2. Then I plot the graph of important variables. After that I find the confusion matrix and calculate the accuracy rate of model on test dataset. At last I save my XGBOOST model and I choose XGBOOST as best model for this project.

Page: 6

Results

At first I find that there is 943141 no. of negative case and 105434 no. of positive case in my dataset. That means in my dataset there is 89.94%. So I need a model that gives me a confusion matrix (as this is a classification problem) with greater than 89.94% accuracy.

Now my dataset have some missing values. I use prediction method for this. My missing value plot for my dataset is:



Now I write a code in R for predict the missing value:-

“

impute<-mice(data[,2:10],m=3,seed=0)

print(impute)

barplot(table(data$gender))

head(impute$imp$gender)

tail(impute$imp$gender)

barplot(table(data$age\_60\_and\_above))

head(impute$imp$age\_60\_and\_above)

tail(impute$imp$age\_60\_and\_above)

Page: 7

newdata<-complete(impute,2)

“

This code predict the missing values using logistic regression and this code give me a 3 set of predicted values. I need to choose one set of predicted value from that which is good for this dataset and I choose this set after seeing the bar plot of those variables.

Then the dataset is split in 2 part : 70% for train and 30% for test , by using this code :

“

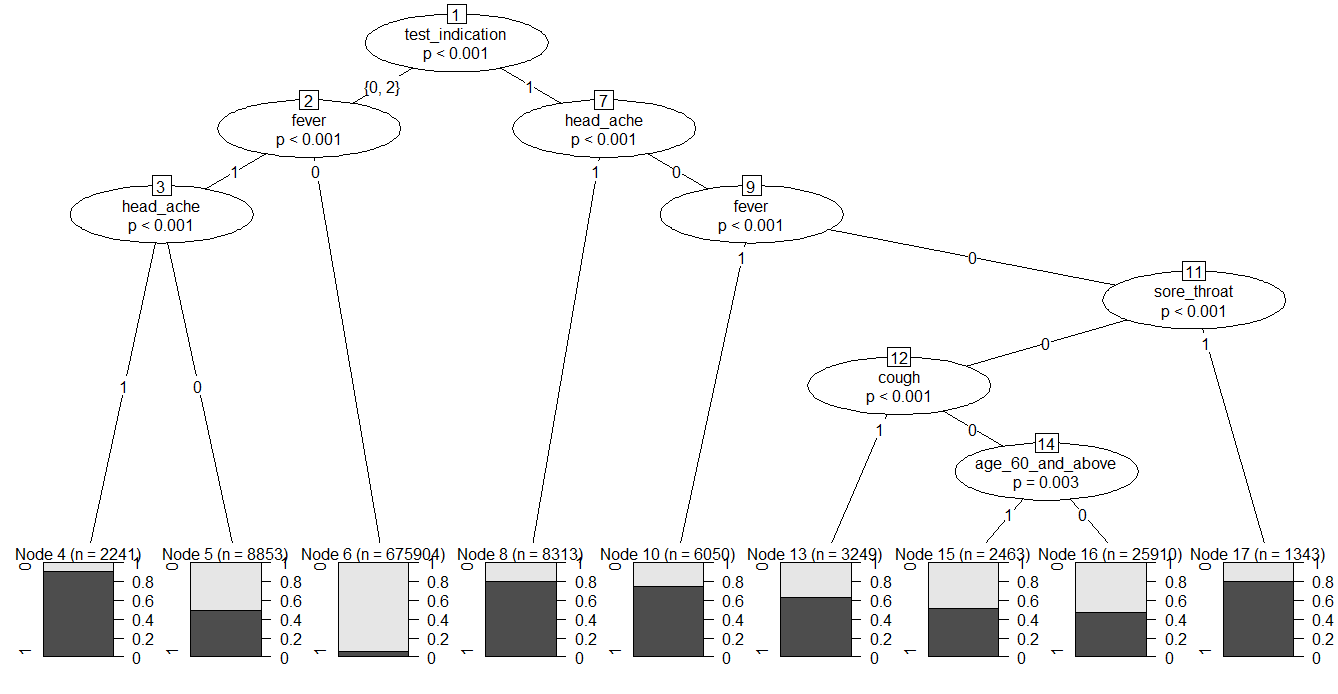
set.seed(0)

split<-sample(2,nrow(newdata),replace=T,prob=c(0.7,0.3))

train<-newdata[split==1,]

test<-newdata[split==2,]

“

Now I plot the Decision Tree on the train dataset with minimum split 10000. I got a decision tree in R :-

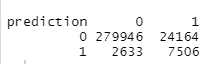
After getting this decision tree model, dependent variable is predicted and calculates the confusion matrix since this problem is a classification problem and I got the confusion matrix for train data like this:



Page: 8

Here accuracy rate is 91.5%. Formula of accuracy rate =

(True positive + true negative)/ (total observation). For Test dataset the confusion matrix of decision tree algorithm is:-



Here accuracy rate 91.47%.

After that I think I fit another algorithm on this data and comparing this two I will decide which one I will choose. So I choose XGBOOST algorithm since it has so many advantages which I discuss before. Now for this I first convert all my dataset into matrix format since XGBOOST support only this format.

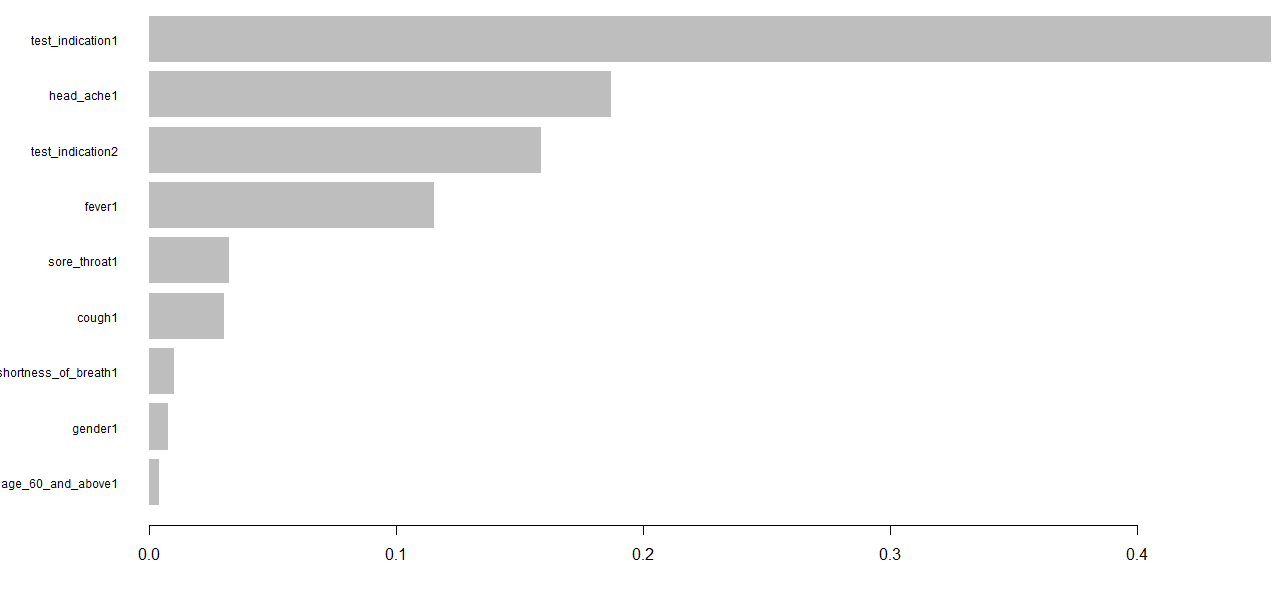
“

xmatrix<-xgb.DMatrix(data=trainx,label=trainy)

xmatrix\_t<-xgb.DMatrix(data=testx,label=testy)

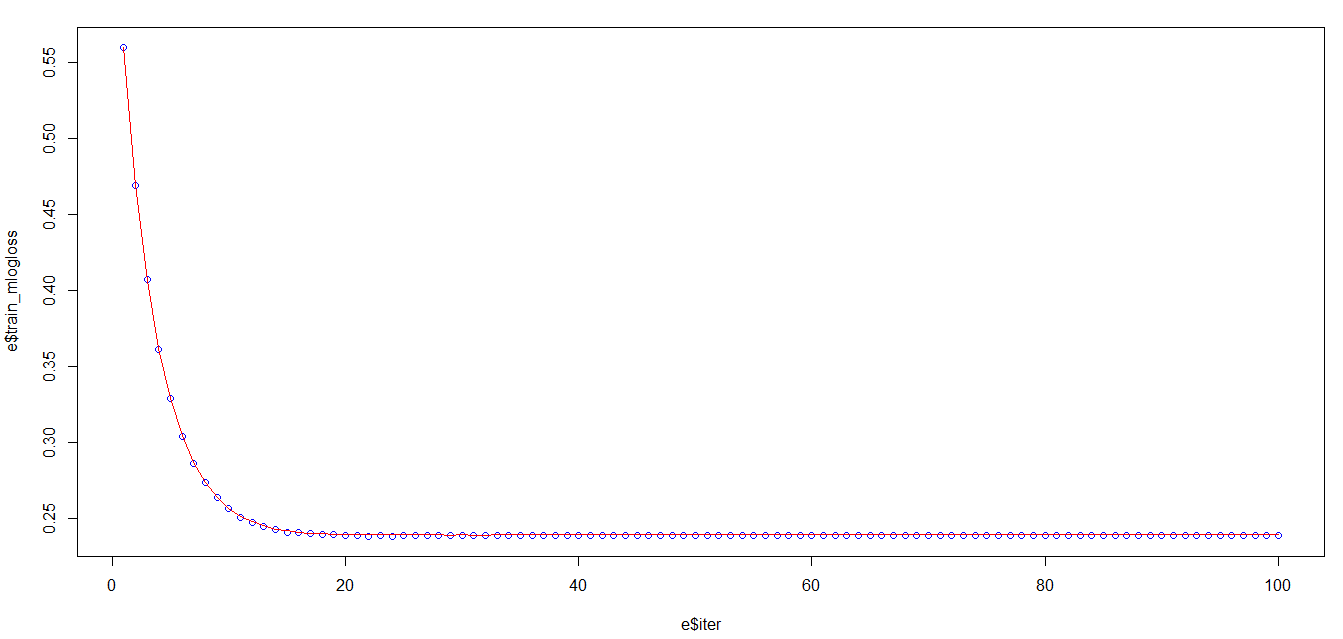
“

After converting, I run This XGBOOST model on the train data with eta value 0.2, depth of the decision tree 6 and no. of iteration is 100. Then XGBOOST model give us a graph from which we must see which variables of the data are important:

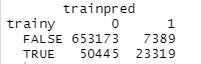


Now A graph is drawn on Iteration vs. log loss and here I plot a scatter plot with blue color where I take the log loss on train data and draw a line on the same graph on log loss of the test dataset of red color. This graph says us that this model which is building on train data will fit on test data well or not. The graph is:

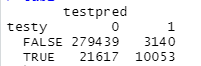
Page: 9



Here red curve fill the dots very well that means XGBOOST algorithm fit well. Now the confusion matrix of train data of XGBOOST model is:



Now the accuracy rate of the confusion matrix is 92.1241% and the confusion matrix of test data of XGBOOST model is:



Now the accuracy rate of the confusion matrix is 92.12249%. After that I save the model in a .rds file so that anyone can run this model behind any app or data structure.

I have studied two models namely Xgboost and decision Tree and I choose here XGBOOST model as a perfect model for my project because first think Decision tree’s confusion matrix accuracy rate is less than XGBOOST confusion matrix accuracy rate. Secondly Decision tree has a over fitting problem so that sometime decision tree model run on the dataset with good accuracy rate but when we run this behind any data structure then it does not function well. Thirdly XGBOOST algorithm handles a bias dataset very well with good accuracy rate.

Conclusion

So XGBOOST model is chosen as a best fitting model for my project and I save my model in .rds file so that anyone can run this behind any data structure so easily.

Page: 10

Reference

COVID-19 Government Data: <https://data.gov.il/dataset/covid-19> (2020)

Xgboost algorithm: <https://www.mygreatlearning.com/blog/xgboost-algorithm/>

Xgboost in R: <https://www.analyticsvidhya.com/blog/2016/01/xgboost-algorithm-easy-steps/>

Decision tree in R: [https://www.tutorialspoint.com/r/r\_decision\_tree.htm#:~:text=Decision%20tree%20is%20a%20graph,Data%20Mining%20applications%20using%20R.](https://www.tutorialspoint.com/r/r_decision_tree.htm%23:~:text=Decision%20tree%20is%20a%20graph,Data%20Mining%20applications%20using%20R.%20)

Article on COVID-19 Diagnosis using AI: [https://www.sciencedirect.com/science/article/abs/pii/S0092867420305511](https://www.sciencedirect.com/science/article/abs/pii/S0092867420305511%20)