****

**Predictive Analysis**

**Mini-Project report on**

**TOPIC: EARLY PREDICTION OF SEPSIS**

**Submitted by:**

Yushant Tyagi | TYA03 | T194062

**Guided by:**

Prof. Vaishali Wangikar

**School of Computer Engineering and Technology**

MIT Academy of Engineering

Alandi, Pune-412105

**CERTIFICATE**

This is to certify that the mini project entitled “**Early Prediction of Sepsis**” has been carried out by Yushant Tyagi (TY), Amal Sutone (TY) , Adhiksha Thorat (TYB) in the partial fulfilment of the requirements for the course Predictive Analysis in the sixth Semester of Degree of Engineering in School of Computer Engineering and Technology, MIT Academy of Engineering.

Prof. Vaishali Wangikar

Course Instructor

**ACKNOWLEDGEMENTS**

We would like to express our gratitude to the School of Computer Engineering and Technology of MIT Academy of Engineering, who encouraged us to do this mini project in the course of Predictive Analysis : Early Predictions of Sepsis

We wish to express our profound thanks to our project guide Prof. Vaishali Wangikar for their full cooperation, guidance and advices for the betterment of our project.

We are grateful that such interesting courses are included in the curriculum and for the proper support and guidance.

Yushant Tyagi | TYA03

**MINI PROJECT TITLE**

The Prediction of Sepsis before the disease takes over the patient and leads to death. All the attributes consisting of the patients present condition help us in predicting the state of his health after a couple of hours.

**Early Prediction of Sepsis.**

**Requirements :**

Hardware requirements :

* Processor
* RAM
* Display
* Memory Storage

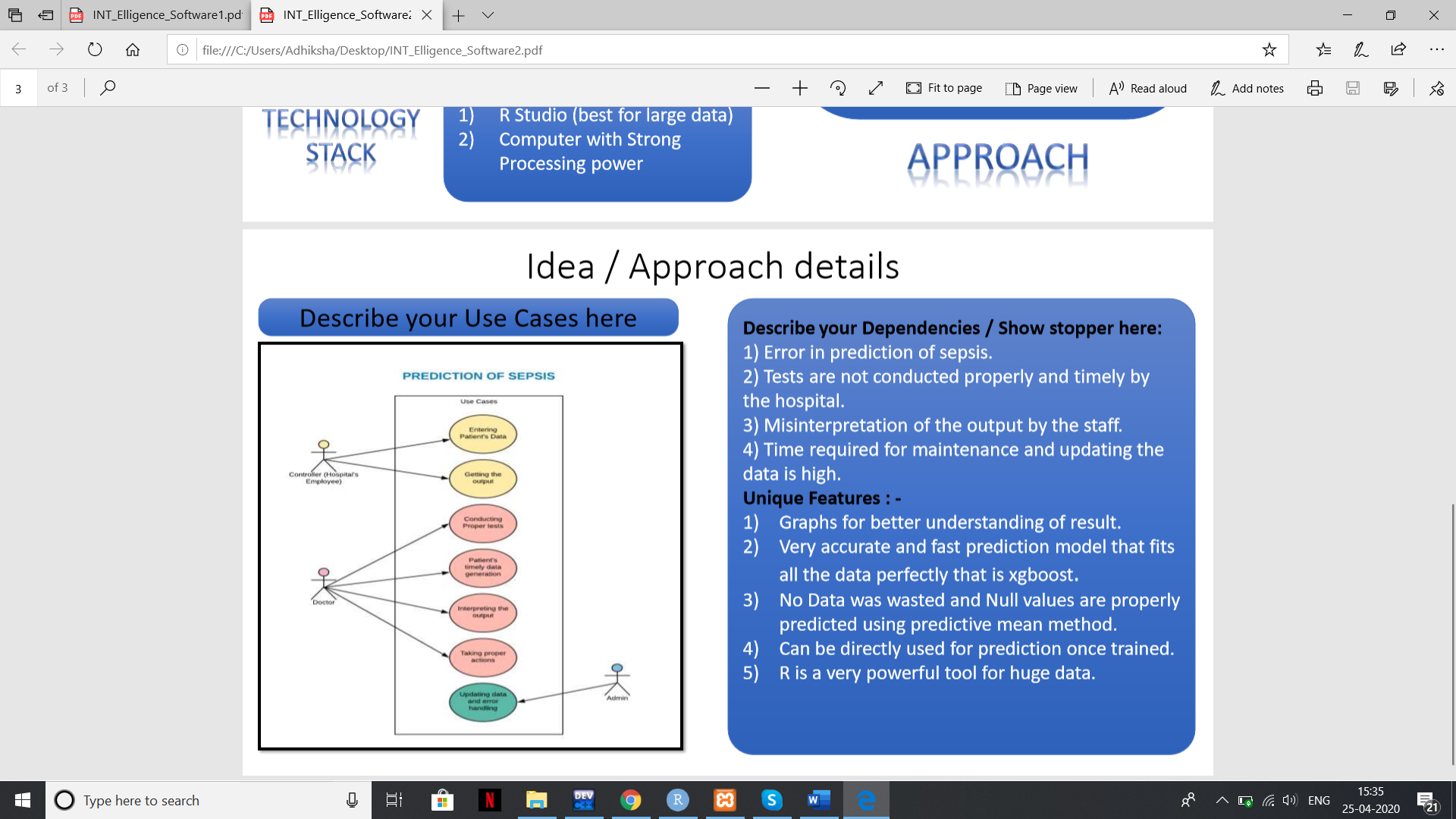
Software requirements :

* R Studio
* R > 3.6.2
* Xampp

**Description:-**

A trained model in R to predict the presence of Sepsis which is a disease that can be life threatening. Hence we have come up with a solution that will predict Sepsis in the least amount of time and to the most of its accuracy. The system uses all the training data that is provided on the official website with all the important parameters to be considered for the prediction. No data is wasted or unused. This helps us achieve the two most important aspects :1) Accuracy of prediction 2) Shortest Time required.

**ARCHITECTURE DIAGRAM OF PROJECT**



**Dataset Link:-**

<https://physionet.org/content/challenge-2019/1.0.0/>

**CODE:**

**Sepsis.R (Analysis Script)**

#dataset = read.csv("p000001.psv", sep = "|", header = TRUE, stringsAsFactors = FALSE)

#dataset= dataset[-1,]

#for (i in c(2:500)) {

# noofzeros = 6- nchar(i)

#s2=paste(replicate(noofzeros,"0"),collapse = "")

#s1="p"

#s3=".psv"

#s1=paste(s1,s2,as.character(i),s3,sep="")

#datasets\_t=read.csv(s1, sep = "|", header = TRUE, stringsAsFactors = FALSE)

#datasets\_t=datasets\_t[-1,]

#dataset=rbind2(dataset,datasets\_t)

#}

#test\_set=dataset

#write.csv(dataset,file="compiled\_data.csv")

dataset=read.csv("compiled\_data.csv")

#install.packages("mice")

#dataset=dataset[0:100,]

dataset=dataset[-1,]

glimpse(dataset)

summary(dataset)

str(dataset)

for (i in c(1:ncol(dataset))) {

dataset[,i] <- as.numeric(as.character(dataset[,i]))

}

library(mice)

md.pattern(dataset)

imputed\_Data <- mice(dataset, m=1, maxit = 1, method = 'pmm',seed='500')

dataset <- complete(imputed\_Data,1)

#write.csv(dataset,"compiled\_data.csv")

library(caTools)

set.seed(123)

split = sample.split(dataset$SepsisLabel, SplitRatio = 8/10 )

training\_set=subset(dataset,split==TRUE)

test\_set=subset(dataset,split==FALSE)

############################MULTIPLE REGRESSION###################

ML<-lm(formula= SepsisLabel~HR+O2Sat+Temp+SBP+MAP+DBP+Resp+BaseExcess+HCO3+FiO2+pH+PaCO2+SaO2+AST+BUN+

Alkalinephos +Calcium +Chloride +Creatinine + Glucose +Lactate +Magnesium +Phosphate+

Potassium +Bilirubin\_total + Hct +Hgb + PTT+ WBC +Fibrinogen+ Platelets + Age+ Gender+ Unit1

+HospAdmTime+ICULOS

,data=training\_set)

pred\_mul\_reg=predict(ML,test\_set[,-41])

pred\_mul\_reg<- ifelse(pred\_mul\_reg>0.5,1,0)

confmatrix<- table(pred\_mul\_reg,test\_set[,41])

accuracy\_mulreg<- (confmatrix[1,1])/length(test\_set[,41])

accuracy\_mulreg

summary(ML) ##### Adjusted R squared is 0.0801

plot(ML)

######BACKWARD ELIMINATION (SETTING THE SIGNIFICANCE LEVEL AS 0.05)###################

ML<-lm(formula= SepsisLabel~HR+O2Sat+Temp+MAP+Resp+FiO2+

Alkalinephos +Creatinine +Phosphate+

Fibrinogen+ Platelets + Age+ Gender+ Unit1 +ICULOS

,data=training\_set)

pred\_mul\_reg=predict(ML,test\_set[,-41])

pred\_mul\_reg<- ifelse(pred\_mul\_reg>0.5,1,0)

confmatrix<- table(pred\_mul\_reg,test\_set[,41])

accuracy\_mulreg<- (confmatrix[1,1])/length(test\_set[,41])

accuracy\_mulreg

summary(ML) ##### Adjusted R squared is 0.0801

plot(ML)

######final accuracy of Multiple Regression is 0.97785 #####################

#install.packages('xgboost')

library(xgboost)

classifier\_xgb=xgboost(data=as.matrix(training\_set[,-41]),label=training\_set$SepsisLabel,nrounds=300)

y\_pred=predict(classifier\_xgb,newdata=as.matrix(test\_set[,-41]))

y\_pred0 = (y\_pred <=0.3)

y\_pred1=(y\_pred>=0.7)

final\_pred<- ifelse(y\_pred<0.5,0,1)

final\_table=table(final\_pred,test\_set[,41])

total\_correct\_predicts=final\_table[1,1]+final\_table[2,2]

accuracy=total\_correct\_predicts/nrow(test\_set)

ggplot()+geom\_point(aes(x=index(test\_set[,41]),y=test\_set[,41]),color="green")

+geom\_point(aes(x=index(final\_pred),y=final\_pred),color="red")+xlab("index")+ylab("SespsisLabel")

+ggtitle("Original(Green) VS Predicted(Red)")

############final accuracy of XGBoost algo is 0.9855################

############### PLOTS OF SOME XGBOOST MODELS #################

#plot\_tree(classifier)

library(data.tree)

install.packages(data.tree)

xgb.plot.tree(model = classifier,trees=0,show\_node\_id=TRUE)

index=c(1:nrow(model))

model=data.frame(index,model)

ggplot(model,aes(x=index,y=c(y\_pred),col=test\_set$SepsisLabel)) +geom\_point()

plot( model$index, model$test\_set.SepsisLabel, type="p", col="red" )

par(new=TRUE)

plot( model$index,model$y\_pred, type="p", col="green" )

plot(dataset$SepsisLabel,dataset$WBC,type="p")

vals=c(table(y\_pred0)[1],table(dataset[1:80000,]$SepsisLabel)[2],table(y\_pred0)[2],table(dataset[1:80000,]$SepsisLabel)[1])

names(vals)[1]<- paste("Sepsis Present")

names(vals)[2]<- paste("Predicted Sepsis Present")

names(vals)[3]<- paste("Sepsis Not Present")

names(vals)[4]<- paste("Predicted Sepsis Not Present")

barplot(vals)

##################### Artificial Neutral Networks ###################

library(h2o)

h2o.init(nthreads=-1)

classifier=h2o.deeplearning(y= 'SepsisLabel',

training\_frame= as.h2o(training\_set),

activation='Rectifier',

hidden=c(6,6),

epochs=100,

train\_samples\_per\_iteration=-2)

ANN\_predict= predict(classifier,type='response', newdata = test\_set[,-41])

y\_pred\_ann=ifelse(ANN\_predict>0.5,1,0)

x\_Valzz<- y\_pred\_ann$C1

conf\_matrix\_ann<- table(as.vector(x\_Valzz),test\_set[,41])

total\_correct\_predicts=conf\_matrix\_ann[1,1]+conf\_matrix\_ann[2,2]

accuracy=total\_correct\_predicts/nrow(test\_set)

plot(classifier,timestep = "duration", metric = "rmse")

############final accuracy of ANN algo is 0.97905 ################

print("By far the best model with highest accuracy is XGBoost" )

find\_result<- function(){

training\_set=read.csv("compiled\_data.csv")

print(training\_set)

training\_set=training\_set[,2:41]

test\_set=read.csv("compiled\_data2.csv")

test\_set<- test\_set[1:1000,]

print(test\_set)

classifier\_xgb=xgboost(data=as.matrix(training\_set[,-41]),label=training\_set$SepsisLabel,nrounds=300)

y\_pred=predict(classifier\_xgb,newdata=as.matrix(test\_set[,-41]))

y\_pred0 = (y\_pred <=0.3)

y\_pred1=(y\_pred>=0.7)

final\_pred<- ifelse(y\_pred<0.5,0,1)

final\_table=table(final\_pred,test\_set[,41])

total\_correct\_predicts=final\_table[1,1]+final\_table[2,2]

accuracy=total\_correct\_predicts/nrow(test\_set)

write.csv(final\_pred,"final\_output.csv")

}

**Sep2.R**

setwd('C:\\xampp\\htdocs\\PA')

library(xgboost)

training\_set=read.csv("compiled\_data.csv")

training\_set=training\_set

test\_set=read.csv("testing.csv")

test\_set<- test\_set

classifier\_xgb=xgboost(data=as.matrix(training\_set[,-41]),label=as.matrix(training\_set[,41]),nrounds=300)

y\_pred=predict(classifier\_xgb,newdata=as.matrix(test\_set[,-41]))

final\_pred<- ifelse(y\_pred<0.5,0,1)

final\_table=table(final\_pred,test\_set[,41])

total\_correct\_predicts=max(final\_pred)

write(total\_correct\_predicts,"final\_output.txt",append = FALSE)

**Show.php (Front end)**

<?php

exec("C:\Progra~1\R\R-3.6.2\bin\Rscript.exe C:\xampp\htdocs\PA\Sep2.R");

$file = fopen("final\_output.txt","r");

$the\_big\_array=fgets($file);

$f=0;

if($the\_big\_array==0)

{print("<h1>You wont have sepsis . The output is 0.</h1>");

$f=0;

}

else{

print("<h1>You will have sepsis . The output is 1.</h1>");

$f=1;

}

?>

<html>

<head> <title>Result</title>

</head>

<body>

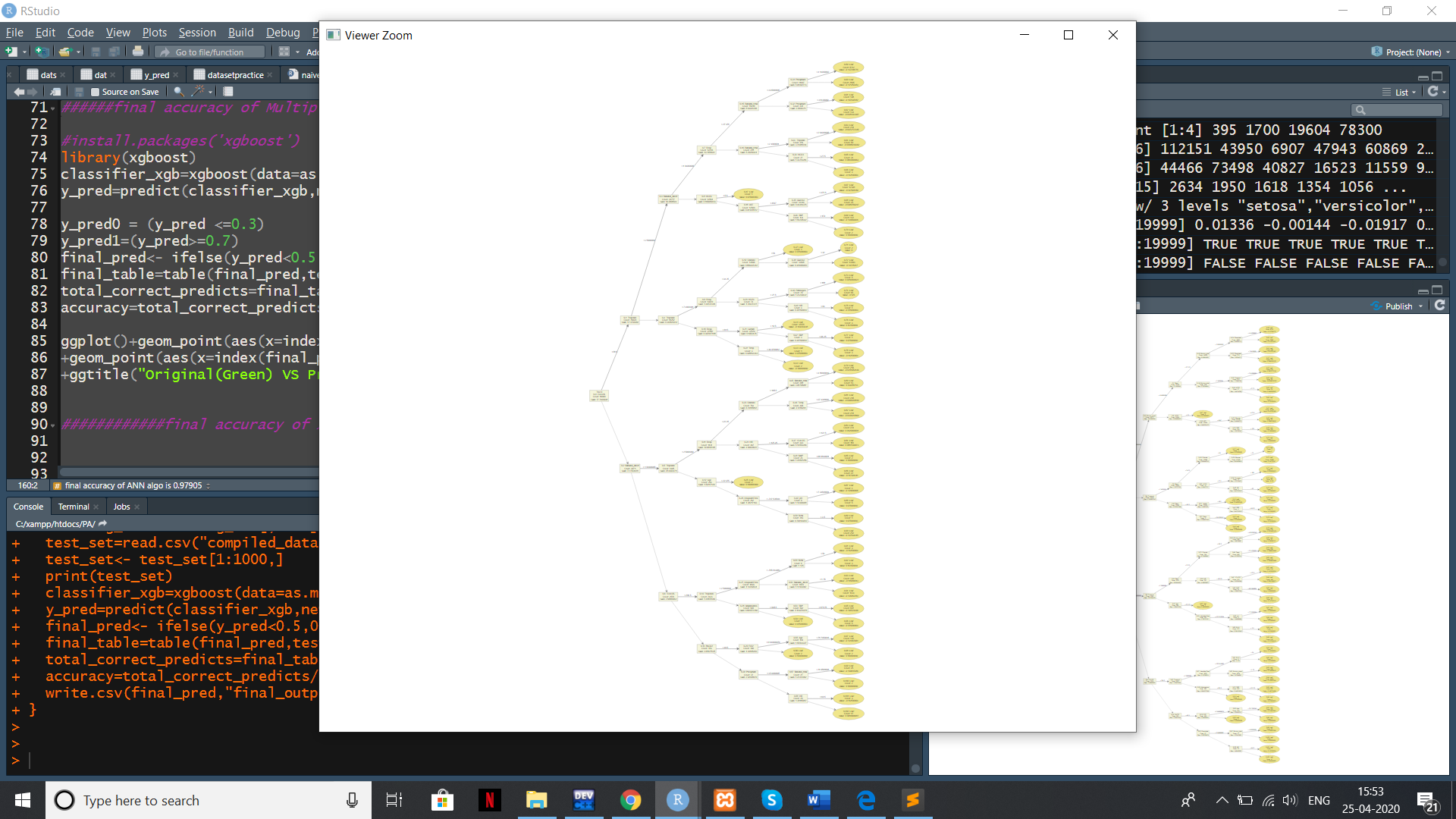
<img src="<?php if($f==1)print("Yes.png"); else print("No.png"); ?>">

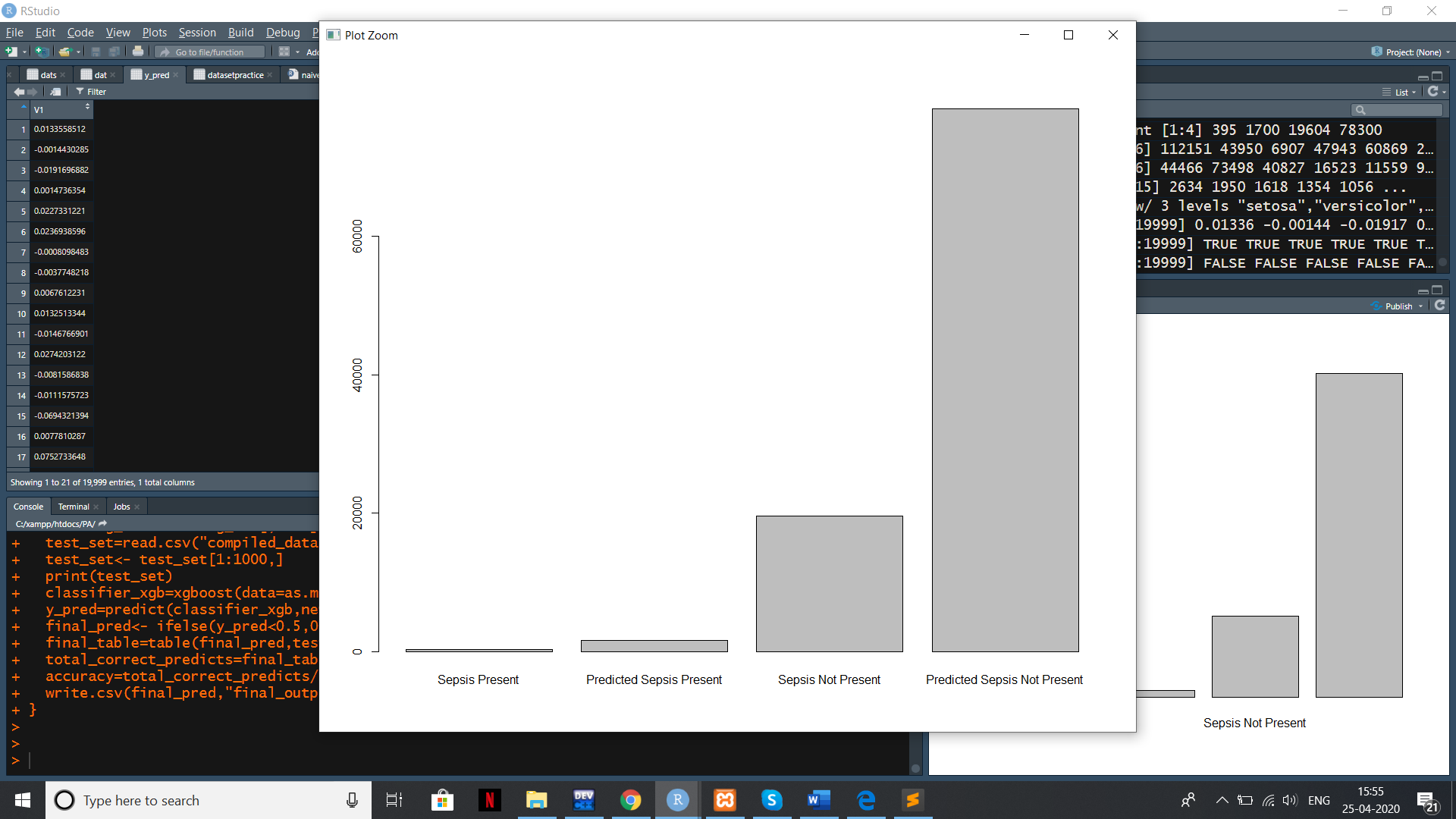
</body>

</html>

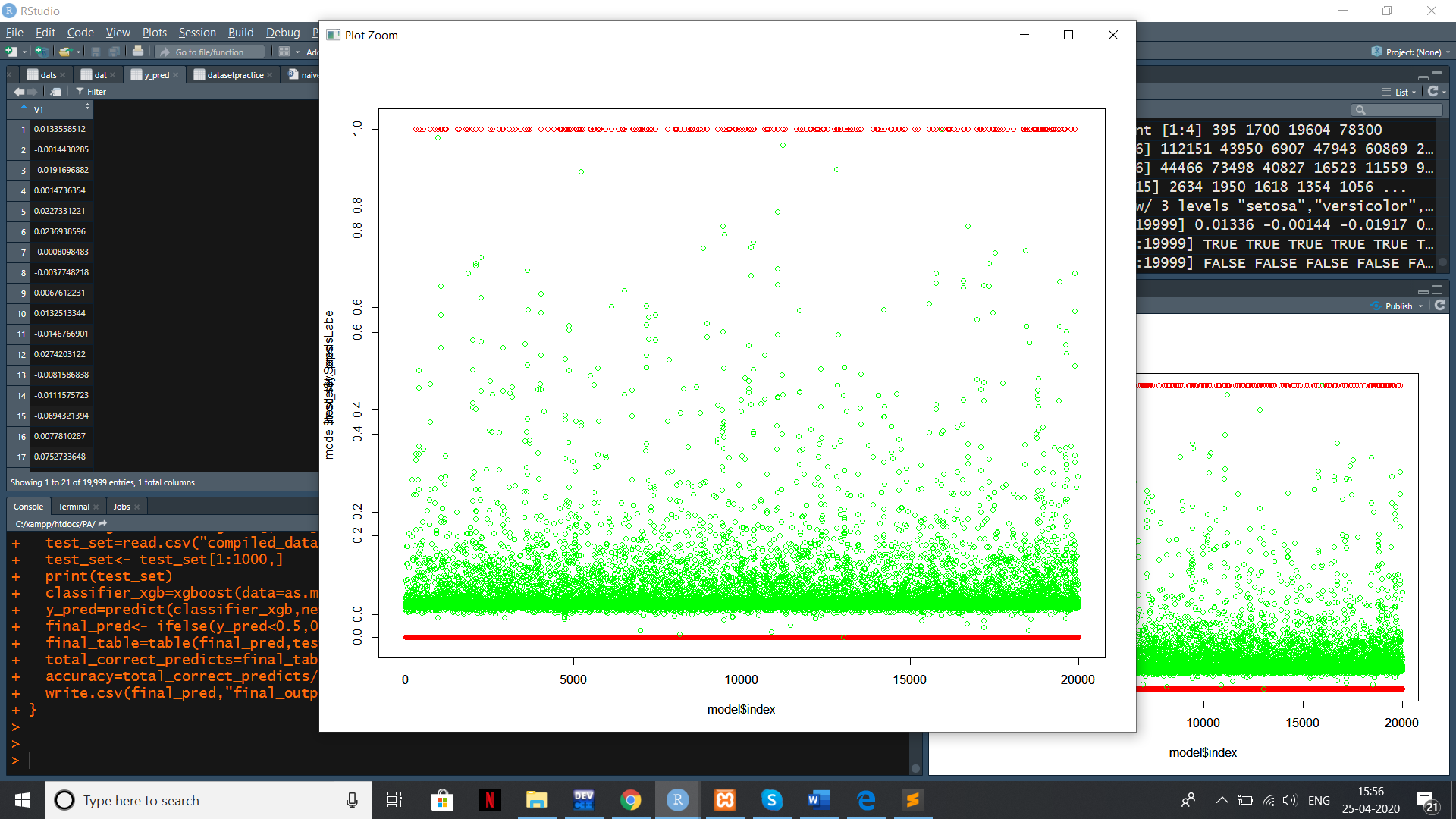
**OUTPUTS:-**

**Tree plot of the boosting algorithm:-**

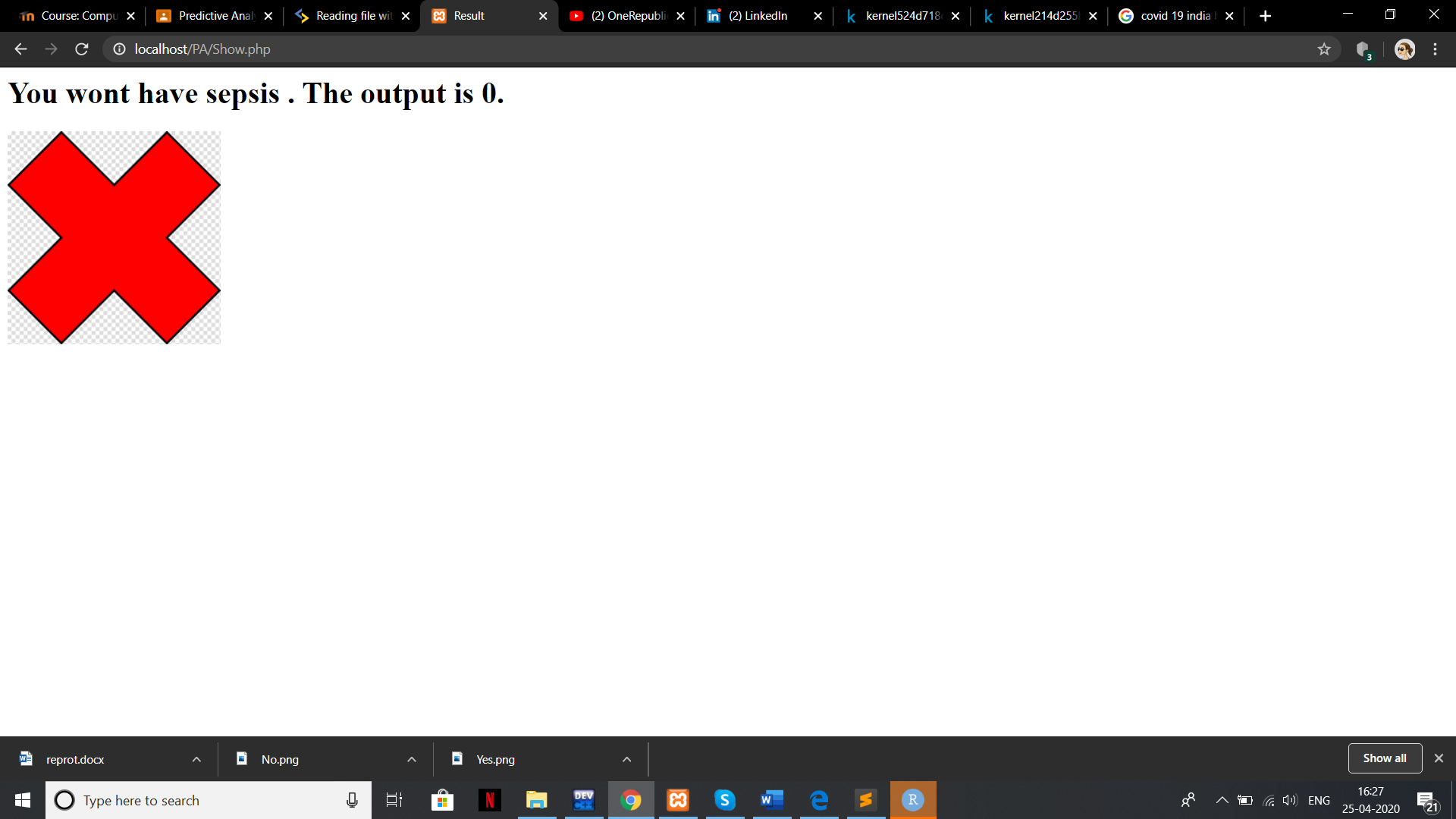




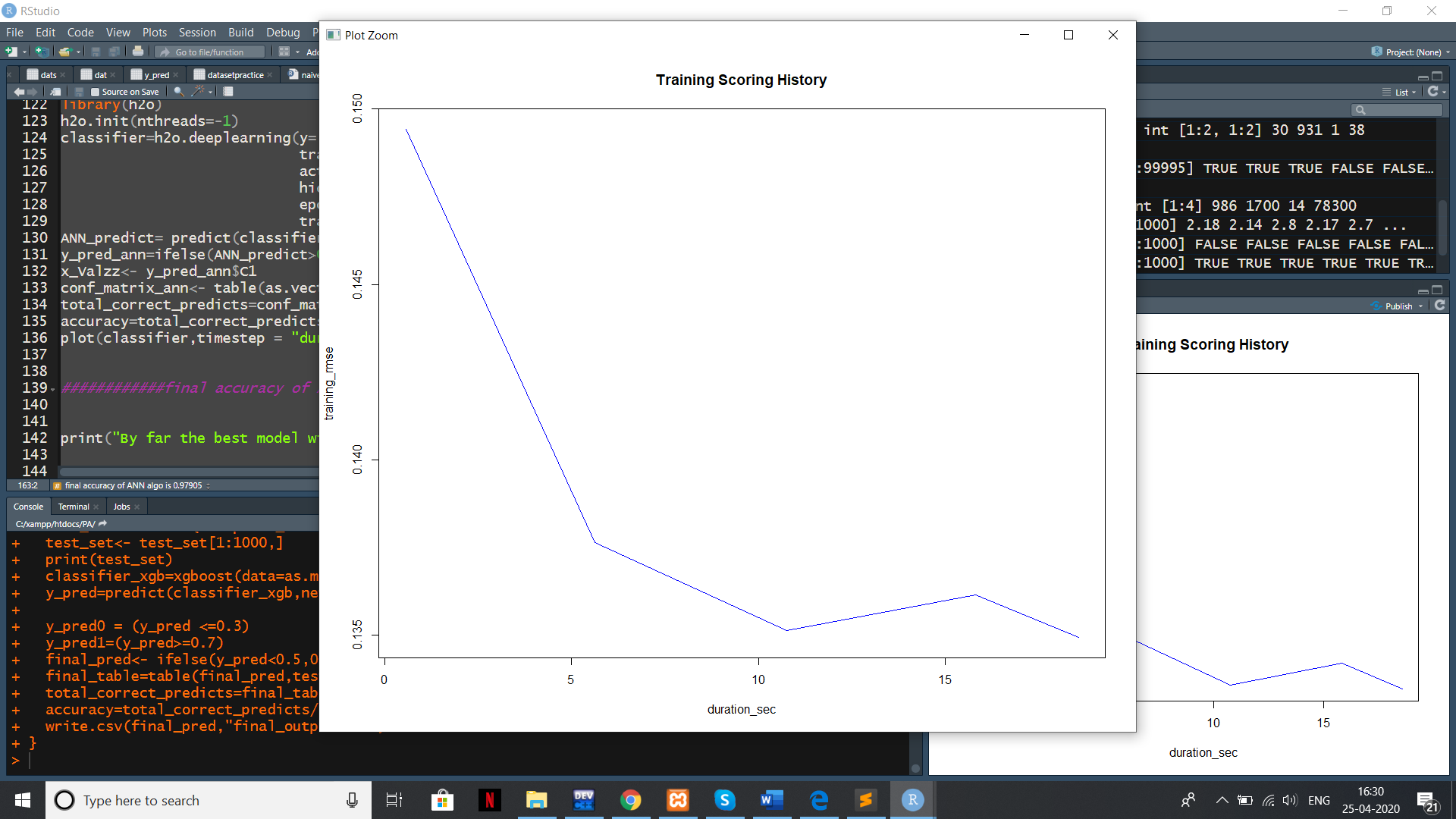
**Predictions of XGboost Algorithm :-**



**Output of front end:-**



**ANN model:-**



**UNIQUE FEATURES**

.

1) Graphs for better understanding of result.

2) Very accurate and fast prediction model that fits all the data perfectly that is XG boost.

3) No Data was wasted and Null values are properly predicted using predictive mean method.

4) Can be directly used for prediction once trained.

5) R is a very powerful tool for huge data.