#Load Data  
df3<-read.csv("df3.csv")  
str(df3)

## 'data.frame': 183045 obs. of 11 variables:  
## $ Outbreak.Associated: chr "Sporadic" "Sporadic" "Sporadic" "Sporadic" ...  
## $ Age.Group : chr "50 to 59 Years" "50 to 59 Years" "20 to 29 Years" "60 to 69 Years" ...  
## $ Source.of.Infection: chr "Travel" "Travel" "Travel" "Travel" ...  
## $ Client.Gender : chr "FEMALE" "MALE" "FEMALE" "FEMALE" ...  
## $ Ever.Hospitalized : chr "No" "Yes" "No" "No" ...  
## $ Ever.in.ICU : chr "No" "No" "No" "No" ...  
## $ Ever.Intubated : chr "No" "No" "No" "No" ...  
## $ Month : int 1 1 2 2 2 2 2 2 2 2 ...  
## $ Income : int 572155 572155 623453 443734 443734 260415 525507 272986 572155 202357 ...  
## $ Density : int 10087 10087 4691 23044 23044 5070 2830 5820 10087 6047 ...  
## $ Outcome : chr "RESOLVED" "RESOLVED" "RESOLVED" "RESOLVED" ...

#prepare data  
df6 <- df3  
  
df6$Income <- log(df3$Income)  
df6$Density <- log(df3$Density)  
col1 <- c("Outbreak.Associated","Age.Group","Source.of.Infection","Client.Gender", "Outcome","Ever.Hospitalized","Ever.in.ICU","Ever.Intubated","Month")  
df6[col1] <-lapply(df6[col1],factor)  
df6$Outcome <- ifelse(df6$Outcome == "RESOLVED", 1,0)  
  
write.csv(df6,"C:/KZ/S/CIND820\\df6.csv", row.names = FALSE)  
  
#Prepare test-train data  
str(df6)

## 'data.frame': 183045 obs. of 11 variables:  
## $ Outbreak.Associated: Factor w/ 2 levels "Outbreak Associated",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ Age.Group : Factor w/ 9 levels "19 and younger",..: 5 5 2 6 6 5 8 6 5 6 ...  
## $ Source.of.Infection: Factor w/ 7 levels "Close Contact",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ Client.Gender : Factor w/ 6 levels "FEMALE","MALE",..: 1 2 1 1 2 2 2 2 2 2 ...  
## $ Ever.Hospitalized : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 2 1 1 ...  
## $ Ever.in.ICU : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Ever.Intubated : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Month : Factor w/ 11 levels "1","2","3","4",..: 1 1 2 2 2 2 2 2 2 2 ...  
## $ Income : num 13.3 13.3 13.3 13 13 ...  
## $ Density : num 9.22 9.22 8.45 10.05 10.05 ...  
## $ Outcome : num 1 1 1 1 1 1 1 1 1 1 ...

set.seed(123)  
train\_index1 <- sample(1:nrow(df6), 0.7 \* nrow(df6))  
train1 <- df6[train\_index1,]  
test1 <- df6[-train\_index1,]

#Data imbalanced  
table(train1$Outcome)

##   
## 0 1   
## 2397 125734

#GLM model  
glm <- glm(Outcome~.,data=train1, family = "binomial")  
summary(glm)

##   
## Call:  
## glm(formula = Outcome ~ ., family = "binomial", data = train1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.5353 0.0163 0.0285 0.0679 2.4486   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) 5.60758 1.09873  
## Outbreak.AssociatedSporadic -0.15682 0.32234  
## Age.Group20 to 29 Years -0.58274 0.83993  
## Age.Group30 to 39 Years -1.26870 0.76771  
## Age.Group40 to 49 Years -1.93880 0.73765  
## Age.Group50 to 59 Years -2.90623 0.71856  
## Age.Group60 to 69 Years -3.96688 0.71394  
## Age.Group70 to 79 Years -4.95100 0.71264  
## Age.Group80 to 89 Years -5.76606 0.71189  
## Age.Group90 and older -6.24110 0.71239  
## Source.of.InfectionCommunity -0.03239 0.13364  
## Source.of.InfectionHousehold Contact 0.20964 0.15903  
## Source.of.InfectionOutbreaks, Congregate Settings -0.28950 0.41248  
## Source.of.InfectionOutbreaks, Healthcare Institutions -1.40299 0.33601  
## Source.of.InfectionOutbreaks, Other Settings -0.07959 0.26709  
## Source.of.InfectionTravel -0.05867 0.30518  
## Client.GenderMALE -0.47723 0.05094  
## Client.GenderNON-BINARY 7.09087 204.09306  
## Client.GenderTRANS MAN 6.91606 626.21349  
## Client.GenderTRANS WOMAN 7.81290 435.89272  
## Client.GenderTRANSGENDER 8.55909 367.75955  
## Ever.HospitalizedYes -2.25460 0.05711  
## Ever.in.ICUYes -1.55536 0.12785  
## Ever.IntubatedYes -1.46213 0.14689  
## Month2 -0.01885 0.13519  
## Month3 -0.30741 0.10476  
## Month4 -0.73284 0.07537  
## Month5 -0.34420 0.09286  
## Month6 0.03279 0.15462  
## Month7 1.00015 0.15639  
## Month8 1.02120 0.19574  
## Month9 0.37186 0.16033  
## Month11 -0.44283 0.11286  
## Month12 -0.62546 0.09261  
## Income 0.21663 0.04850  
## Density 0.20348 0.04202  
## z value Pr(>|z|)   
## (Intercept) 5.104 3.33e-07 \*\*\*  
## Outbreak.AssociatedSporadic -0.487 0.62661   
## Age.Group20 to 29 Years -0.694 0.48781   
## Age.Group30 to 39 Years -1.653 0.09842 .   
## Age.Group40 to 49 Years -2.628 0.00858 \*\*   
## Age.Group50 to 59 Years -4.045 5.24e-05 \*\*\*  
## Age.Group60 to 69 Years -5.556 2.76e-08 \*\*\*  
## Age.Group70 to 79 Years -6.947 3.72e-12 \*\*\*  
## Age.Group80 to 89 Years -8.100 5.51e-16 \*\*\*  
## Age.Group90 and older -8.761 < 2e-16 \*\*\*  
## Source.of.InfectionCommunity -0.242 0.80851   
## Source.of.InfectionHousehold Contact 1.318 0.18742   
## Source.of.InfectionOutbreaks, Congregate Settings -0.702 0.48276   
## Source.of.InfectionOutbreaks, Healthcare Institutions -4.175 2.97e-05 \*\*\*  
## Source.of.InfectionOutbreaks, Other Settings -0.298 0.76572   
## Source.of.InfectionTravel -0.192 0.84756   
## Client.GenderMALE -9.369 < 2e-16 \*\*\*  
## Client.GenderNON-BINARY 0.035 0.97228   
## Client.GenderTRANS MAN 0.011 0.99119   
## Client.GenderTRANS WOMAN 0.018 0.98570   
## Client.GenderTRANSGENDER 0.023 0.98143   
## Ever.HospitalizedYes -39.476 < 2e-16 \*\*\*  
## Ever.in.ICUYes -12.166 < 2e-16 \*\*\*  
## Ever.IntubatedYes -9.954 < 2e-16 \*\*\*  
## Month2 -0.139 0.88913   
## Month3 -2.934 0.00334 \*\*   
## Month4 -9.724 < 2e-16 \*\*\*  
## Month5 -3.707 0.00021 \*\*\*  
## Month6 0.212 0.83206   
## Month7 6.395 1.60e-10 \*\*\*  
## Month8 5.217 1.82e-07 \*\*\*  
## Month9 2.319 0.02038 \*   
## Month11 -3.924 8.71e-05 \*\*\*  
## Month12 -6.753 1.44e-11 \*\*\*  
## Income 4.467 7.94e-06 \*\*\*  
## Density 4.843 1.28e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 23823 on 128130 degrees of freedom  
## Residual deviance: 12114 on 128095 degrees of freedom  
## AIC: 12186  
##   
## Number of Fisher Scoring iterations: 14

predictglm <- predict(glm, test1, type="response")

# prediction for glm  
predictnum <- ifelse(predictglm >= 0.5,1,0)   
  
#Confusion matrix for glm  
library(caret)

## Warning: package 'caret' was built under R version 4.2.1

## Loading required package: ggplot2

## Loading required package: lattice

confusionMatrix(as.factor(test1$Outcome), as.factor(predictnum))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 218 870  
## 1 185 53641  
##   
## Accuracy : 0.9808   
## 95% CI : (0.9796, 0.9819)  
## No Information Rate : 0.9927   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.2848   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.540943   
## Specificity : 0.984040   
## Pos Pred Value : 0.200368   
## Neg Pred Value : 0.996563   
## Prevalence : 0.007339   
## Detection Rate : 0.003970   
## Detection Prevalence : 0.019813   
## Balanced Accuracy : 0.762491   
##   
## 'Positive' Class : 0   
##

#ROC curve  
library(pROC)

## Warning: package 'pROC' was built under R version 4.2.1

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

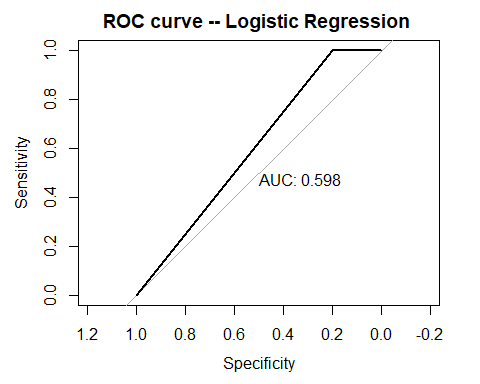
## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

lm=roc(test1$Outcome, predictnum) #AUC score

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

plot(lm ,main ="ROC curve -- Logistic Regression ",print.auc=TRUE)

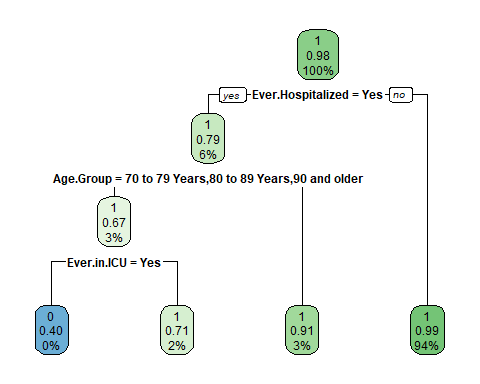


#Decision tree  
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.2.2

## Loading required package: rpart

tree <- rpart(Outcome~., data = train1, method = 'class')  
rpart.plot(tree)



#Prediction for decision tree  
treepred <-predict(tree, test1, type = 'class')

#Confusion matrix for decision tree  
confusionMatrix(as.factor(test1$Outcome), as.factor(treepred))

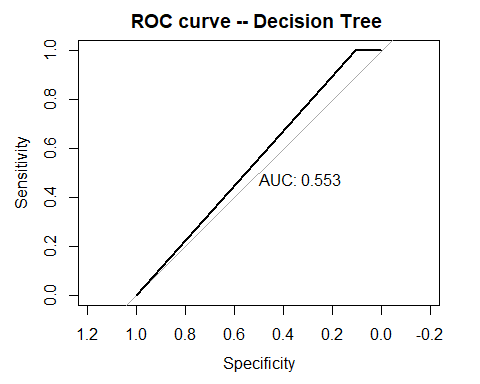
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 117 971  
## 1 84 53742  
##   
## Accuracy : 0.9808   
## 95% CI : (0.9796, 0.9819)  
## No Information Rate : 0.9963   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.1764   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.582090   
## Specificity : 0.982253   
## Pos Pred Value : 0.107537   
## Neg Pred Value : 0.998439   
## Prevalence : 0.003660   
## Detection Rate : 0.002131   
## Detection Prevalence : 0.019813   
## Balanced Accuracy : 0.782171   
##   
## 'Positive' Class : 0   
##

#ROC curve for decision tree  
roctree <-roc(test1$Outcome, as.numeric(treepred)) #AUC score

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

plot.roc(roctree,main ="ROC curve -- Decision Tree ",print.auc=TRUE)



#prepare for XGBoost  
library(fastDummies)

## Warning: package 'fastDummies' was built under R version 4.2.2

train1new<-dummy\_cols(train1)  
train1new <-subset(train1new[9:52])  
test1new <-dummy\_cols(test1)  
test1new <-subset(test1new[9:52])

#set up XGBoost  
params <- list(eta = 0.3,  
 set.seed = 123,   
 max\_depth = 6,  
 scale\_pos\_weight = 70,  
 eval\_metric = "auc",  
 objective = "binary:logistic")  
library(xgboost)

## Warning: package 'xgboost' was built under R version 4.2.2

x\_train = as.matrix(train1new[,-3])  
y\_train = train1new$Outcome  
x\_test = data.matrix(test1new[,-3])  
y\_test = test1new$Outcome  
xgboost\_train = xgb.DMatrix(data=x\_train,label=y\_train)  
xgboost\_test = xgb.DMatrix(data=x\_test,label=y\_test)

#Model XGBoost  
model <- xgboost(data = xgboost\_train,  
 nrounds = 20, max\_depth= 6, objective = "binary:logistic", eval\_metric = "auc", nthread = 4,booster="gbtree")

## [1] train-auc:0.957372   
## [2] train-auc:0.958583   
## [3] train-auc:0.965793   
## [4] train-auc:0.965948   
## [5] train-auc:0.966339   
## [6] train-auc:0.966555   
## [7] train-auc:0.967284   
## [8] train-auc:0.967834   
## [9] train-auc:0.968254   
## [10] train-auc:0.971109   
## [11] train-auc:0.971302   
## [12] train-auc:0.971567   
## [13] train-auc:0.972049   
## [14] train-auc:0.972335   
## [15] train-auc:0.972633   
## [16] train-auc:0.972864   
## [17] train-auc:0.973077   
## [18] train-auc:0.975147   
## [19] train-auc:0.975516   
## [20] train-auc:0.975709

#Prediction for XGBoost  
xgpred <- predict(model, xgboost\_test)  
xgnum <- ifelse(xgpred >= 0.5,1,0)

#Confusion matrix for XGBoost  
confusionMatrix(as.factor(test1new$Outcome),as.factor(xgnum))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 156 932  
## 1 113 53713  
##   
## Accuracy : 0.981   
## 95% CI : (0.9798, 0.9821)  
## No Information Rate : 0.9951   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.2238   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.579926   
## Specificity : 0.982944   
## Pos Pred Value : 0.143382   
## Neg Pred Value : 0.997901   
## Prevalence : 0.004899   
## Detection Rate : 0.002841   
## Detection Prevalence : 0.019813   
## Balanced Accuracy : 0.781435   
##   
## 'Positive' Class : 0   
##

#ROC curve for XGBoost  
rocxg=roc(y\_test , xgnum) #AUC score

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

plot(rocxg ,main ="ROC curve -- XGBoost ",print.auc=TRUE)

