datathon

krishna

22 May 2018

finding missing values in the train dataset

```
colSums(is.na(tr))
```

```
##                        id                program_id 
##                         0                         0 
##              program_type          program_duration 
##                         0                         0 
##                   test_id                 test_type 
##                         0                         0 
##          difficulty_level                trainee_id 
##                         0                         0 
##                    gender                 education 
##                         0                         0 
##                 city_tier                       age 
##                         0                     27729 
##   total_programs_enrolled            is_handicapped 
##                         0                         0 
## trainee_engagement_rating                   is_pass 
##                        77                         0
```

finding missing values in the test data set

```
colSums(is.na(te1))
```

```
##                        id                program_id 
##                         0                         0 
##              program_type          program_duration 
##                         0                         0 
##                   test_id                 test_type 
##                         0                         0 
##          difficulty_level                trainee_id 
##                         0                         0 
##                    gender                 education 
##                         0                         0 
##                 city_tier                       age 
##                         0                     11791 
##   total_programs_enrolled            is_handicapped 
##                         0                         0 
## trainee_engagement_rating 
##                        31
```

missing value imputation in train and test data set

```
te=te1



a=tr %>% group_by(education,difficulty_level,trainee_engagement_rating) %>% summarise(t=n())

z=a %>%
  arrange_(~ desc(t)) %>%
  group_by_(~ education) %>%
  top_n(n =5)
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.3
```

```
## Selecting by t
```

```
##### imputing missing age value with the mean of age ######

tr$age=ifelse(is.na(tr$age),36,tr$age)
te$age=ifelse(is.na(te$age),36,te$age)
```

```
########### Feature Engineering #########

tr$trainee_engagement_rating=ifelse(is.na(tr$trainee_engagement_rating)&tr$education=="High School Diploma",1,tr$trainee_engagement_rating)

tr$trainee_engagement_rating=ifelse(is.na(tr$trainee_engagement_rating)&tr$education=="Matriculation",1,tr$trainee_engagement_rating)

tr$trainee_engagement_rating=ifelse(is.na(tr$trainee_engagement_rating)&tr$education=="Masters",3,tr$trainee_engagement_rating)


tr$trainee_engagement_rating=ifelse(is.na(tr$trainee_engagement_rating)&tr$education=="No Qualification",1,tr$trainee_engagement_rating)

tr$trainee_engagement_rating=ifelse(is.na(tr$trainee_engagement_rating)&tr$education=="Bachelors",1,tr$trainee_engagement_rating)



te$trainee_engagement_rating=ifelse(is.na(te$trainee_engagement_rating)&te$education=="High School Diploma",1,te$trainee_engagement_rating)

te$trainee_engagement_rating=ifelse(is.na(te$trainee_engagement_rating)&te$education=="Matriculation",1,te$trainee_engagement_rating)

te$trainee_engagement_rating=ifelse(is.na(te$trainee_engagement_rating)&te$education=="Masters",3,te$trainee_engagement_rating)


te$trainee_engagement_rating=ifelse(is.na(te$trainee_engagement_rating)&te$education=="No Qualification",1,te$trainee_engagement_rating)

te$trainee_engagement_rating=ifelse(is.na(te$trainee_engagement_rating)&te$education=="Bachelors",1,te$trainee_engagement_rating)


tr$diff=ifelse(tr$difficulty_level=="intermediate",2,ifelse(tr$difficulty_level=="easy",1,ifelse(tr$difficulty_level=="hard",3,4)))
te$diff=ifelse(te$difficulty_level=="intermediate",2,ifelse(te$difficulty_level=="easy",1,ifelse(te$difficulty_level=="hard",3,4)))


tr$testtype=ifelse(tr$test_type=="online",1,0)
te$testtype=ifelse(te$test_type=="online",1,0)


tr=tr %>% select(-id)
te=te %>% select(-id)

trn=tr %>% select(-program_id,-test_type,-program_type,-difficulty_level,-education,-is_handicapped)
tst=te %>% select(-program_id,-test_type,-program_type,-difficulty_level,-education,-is_handicapped)

trn$is_pass=as.factor(trn$is_pass)
trn=trn %>% select(-gender)
```

```
####### Splitting into Train and Test Dataset #######

train=trn[sample(1:nrow(trn),0.7*nrow(trn)),]

test=trn[sample(1:nrow(trn),0.3*nrow(trn)),]
```

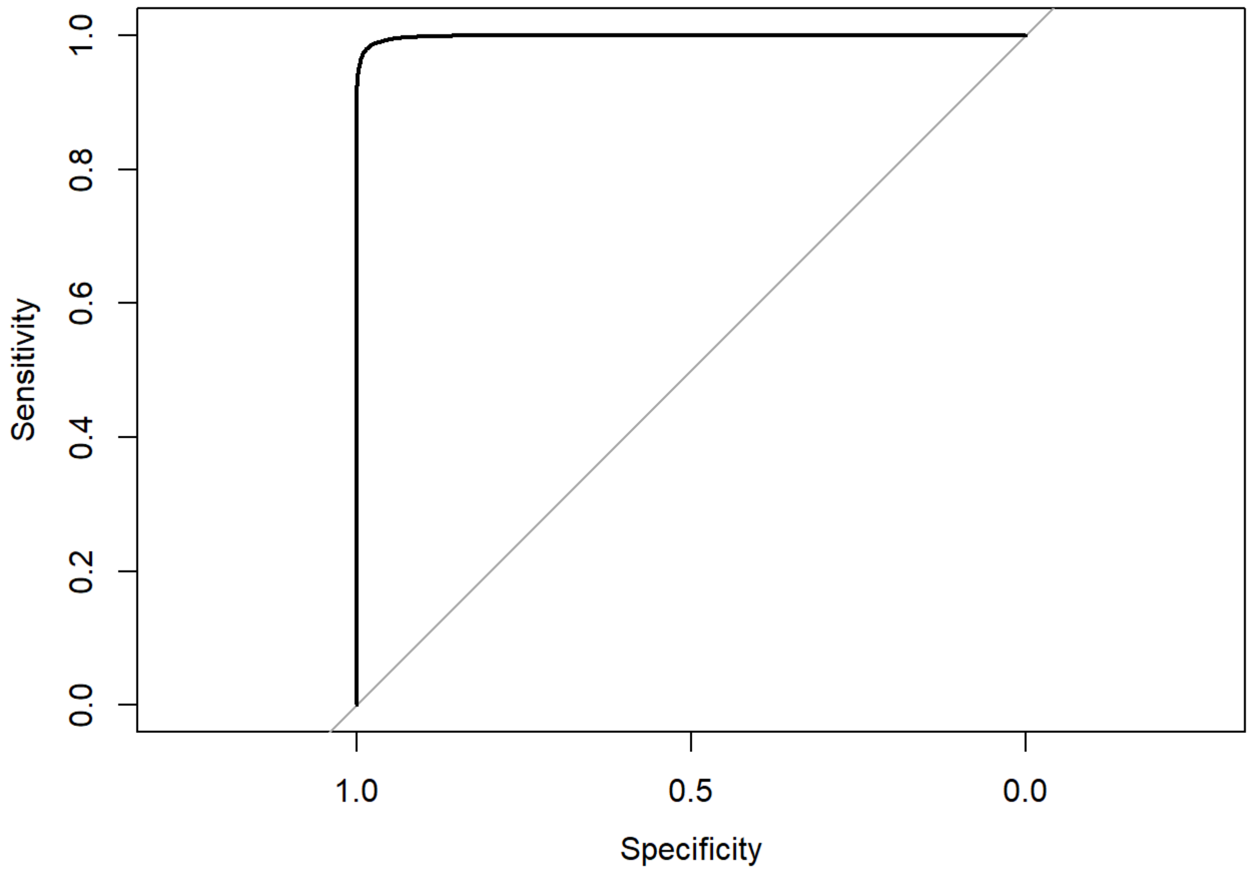
```
######### Applying Random Forest on the Dataset #########

model=randomForest(is_pass~. , data = trn,ntree=260, mtry = 4)

pred=predict(model,test,type = "prob")
View(pred)
pred1=data.frame(pred)
mean(pred1$X0)
```

```
## [1] 0.2883906
```

```
pred1$v3=as.factor(ifelse(pred1$X0>0.31,0,1))


######### Model Evaluation using Confusion Matrix on the splitted train dataset provided ####
#####

confusionMatrix(test$is_pass,pred1$v3,positive = "1")
```

```
## Confusion Matrix and Statistics
## 
##             Reference
## Prediction     0     1
##          0  6714    32
##          1   696 14502
##                                           
##                Accuracy : 0.9668          
##                  95% CI : (0.9644, 0.9692)
##     No Information Rate : 0.6623          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.9242          
##  Mcnemar's Test P-Value : < 2.2e-16       
##                                           
##             Sensitivity : 0.9978          
##             Specificity : 0.9061          
##          Pos Pred Value : 0.9542          
##          Neg Pred Value : 0.9953          
##              Prevalence : 0.6623          
##          Detection Rate : 0.6609          
##    Detection Prevalence : 0.6926          
##       Balanced Accuracy : 0.9519          
##                                           
##        'Positive' Class : 1               
## 
```

```
######### Model Evaluation using ROC-AUC curve on the splitted train dataset provided #######
#

x = roc(test$is_pass,pred[,2])
plot(x)
```



```
auc(x)
```

```
## Area under the curve: 0.9987
```

```
######## Applying Model on Test dataset ###########

pred=predict(model,tst,type = "prob")
pred1=data.frame(pred)
pred1$v3=as.factor(ifelse(pred1$X0>0.31,0,1))


output=data.frame(id=te1$id,is_pass=pred1$v3)

###### Writing csv for final submission #######
write.csv(output,file="C:\\Users\\Administrator\\Desktop\\hacka\\output.csv",row.names = F)
```