Predição de notas de filmes para inclusão no catálogo da Netflix

Thiago Silva

29/08/2021

DESAFIO

Vamos supor que a empresa Netflix deseja um modelo preditivo para prever as notas de filmes, para assim decidir se vale a pena ou não colocar esse filme no catálogo.

Contextualização.

O banco de dados em questão possui 6234 observações e 12 atributos, sendo eles o ID do espetáculo, Tipo de Filme, Nome do filme, Diretor, Elenco, País, Data de adição, Ano de produção, Avaliação, Duração, Gênero e Descrição.
O objetivo do Desafio é criar um modelo para prever se um filme novo será um sucesso e decidir se este deve entrar no catálogo.

1.Leitura e visualização do banco de dados

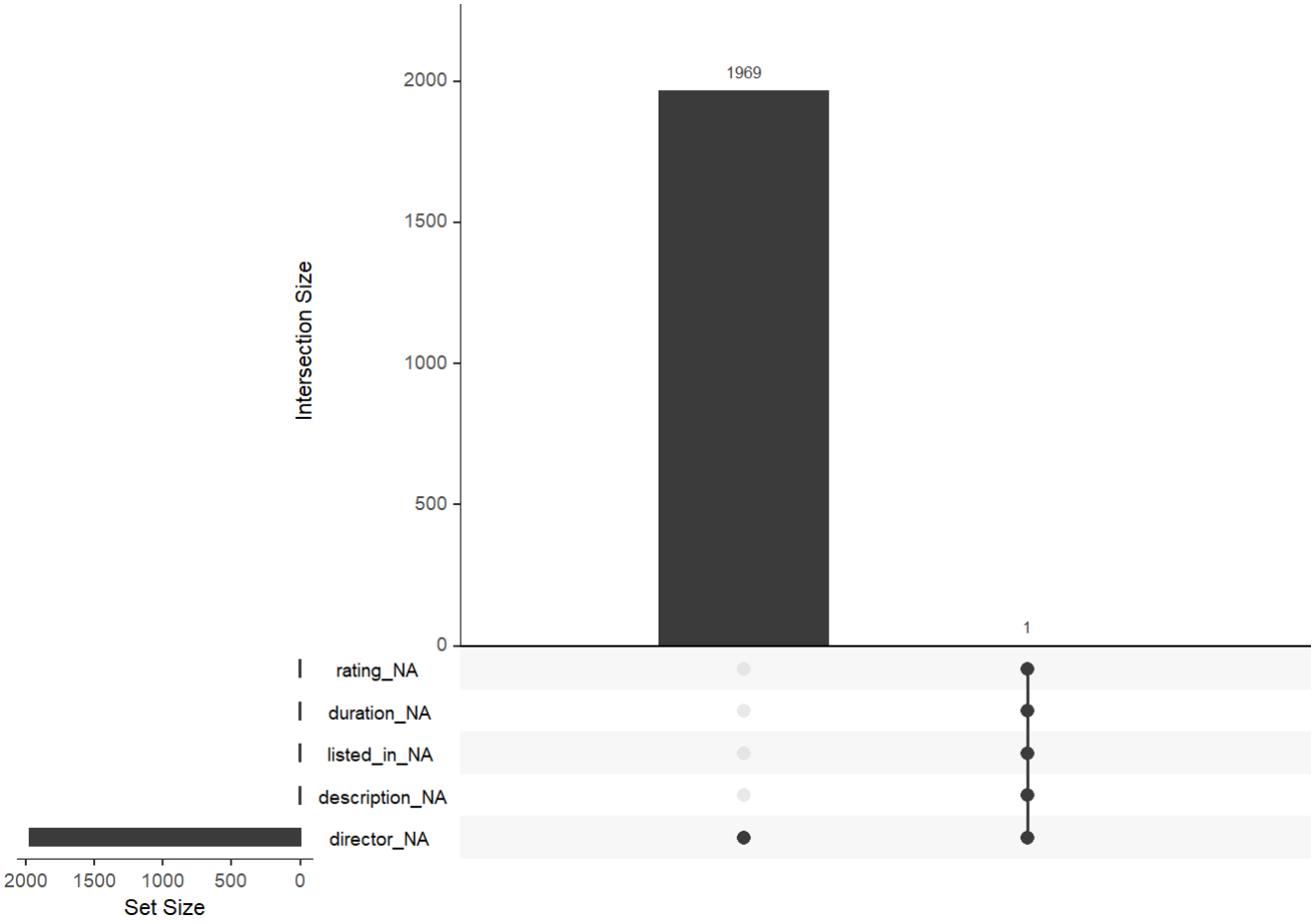
```
library(dplyr)
library(pacman)
BD <- read.csv2("C:/Users/Thiago Silva/Documents/DataScience/ProcessoSeletivo/EleflowBigData/
Netflix.csv", sep="," , header = TRUE , encoding="UTF-8", na.strings ="", stringsAsFactors =
FALSE)
head(BD)
```

```
##   X  show_id    type                                    title
## 1 1 81145628   Movie Norm of the North: King Sized Adventure
## 2 2 80117401   Movie              Jandino: Whatever it Takes
## 3 3 70234439 TV Show                     Transformers Prime
## 4 4 80058654 TV Show        Transformers: Robots in Disguise
## 5 5 80125979   Movie                             #realityhigh
## 6 6 80163890 TV Show                                  Apaches
##                   director
## 1 Richard Finn, Tim Maltby
## 2                     <NA>
## 3                     <NA>
## 4                     <NA>
## 5         Fernando Lebrija
## 6                     <NA>
##
cast
## 1                                      Alan Marriott, Andrew Toth, Brian Dobson, Cole Ho
ward, Jennifer Cameron, Jonathan Holmes, Lee Tockar, Lisa Durupt, Maya Kay, Michael Dobson
## 2
Jandino Asporaat
## 3 Peter Cullen, Sumalee Montano, Frank Welker, Jeffrey Combs, Kevin Michael Richardson, Ta
nia Gunadi, Josh Keaton, Steve Blum, Andy Pessoa, Ernie Hudson, Daran Norris, Will Friedle
## 4                                                          Will Friedle, Darren Criss, Co
nstance Zimmer, Khary Payton, Mitchell Whitfield, Stuart Allan, Ted McGinley, Peter Cullen
## 5           Nesta Cooper, Kate Walsh, John Michael Higgins, Keith Powers, Alicia Sanz, Jak
e Borelli, Kid Ink, Yousef Erakat, Rebekah Graf, Anne Winters, Peter Gilroy, Patrick Davis
## 6
Alberto Ammann, Eloy Azorín, Verónica Echegui, Lucía Jiménez, Claudia Traisac
##                                  country date_added release_year rating
## 1 United States, India, South Korea, China 2019-09-09         2019     41
## 2                          United Kingdom 2016-09-09         2016     52
## 3                           United States 2018-09-08         2013     82
## 4                           United States 2018-09-08         2016     64
## 5                           United States 2017-09-08         2017     57
## 6                                   Spain 2017-09-08         2016     72
##   duration                                                    listed_in
## 1   90 min                              Children & Family Movies, Comedies
## 2   94 min                                                  Stand-Up Comedy
## 3 1 Season                                                      Kids' TV
## 4 1 Season                                                      Kids' TV
## 5   99 min                                                      Comedies
## 6 1 Season Crime TV Shows, International TV Shows, Spanish-Language TV Shows
##
description
## 1         Before planning an awesome wedding for his grandfather, a polar bear king must t
ake back a stolen artifact from an evil archaeologist first.
## 2    Jandino Asporaat riffs on the challenges of raising kids and serenades the audience w
ith a rousing rendition of "Sex on Fire" in his comedy show.
## 3         With the help of three human allies, the Autobots once again protect Earth from
the onslaught of the Decepticons and their leader, Megatron.
## 4                      When a prison ship crash unleashes hundreds of Decepticons on Eart
h, Bumblebee leads a new Autobot force to protect humankind.
## 5 When nerdy high schooler Dani finally attracts the interest of her longtime crush, she l
ands in the cross hairs of his ex, a social media celebrity.
## 6             A young journalist is forced into a life of crime to save his father and fam
ily in this series based on the novel by Miguel Sáez Carral.
```

```
#Formatando dados
#glimpse(BD)
#dim(BD)
#Convertendo rating para tipo inteiro
BD$rating <- as.integer(BD$rating)
#Removendo dados desnecessarios
BD$X <- NULL
BD$show_id <- NULL
```

2.Analisando dados faltantes.

```
#Checando dados faltantes
subs <- BD
#data.frame("variable"=c(colnames(subs)), "missing values count"=sapply(subs, function(x) sum
(is.na(x))), row.names=NULL)
library(naniar)
gg_miss_upset(BD)
```




```
#Verificando se ha algum valor NA no dataset
#anyNA(BD)
#Porcentagem de valores NAS em cada coluna
NAS <- round(colSums(is.na(BD))*100/nrow(BD), 2)
#VER TODOS NAS
NAS
```

```
##         type        title     director         cast      country   date_added
##         0.00         0.00        31.60         0.02         0.02         0.00
## release_year       rating     duration    listed_in  description
##         0.00         0.02         0.02         0.02         0.02
```

O banco de dados está danificado por dados faltantes, e como ilustrado acima a coluna de Diretores é a mais afetada com um pouco mais de 30% de “missing values”. Todos os demais atributos que apresentaram dados faltantes contém apenas 1 linha de dados faltantes. Como o banco de dados contém 6234 observações, eliminaremos os dados faltantes e trabalharemos com 4264 observações.

3.Removendo dados faltantes.

```
#Removendo dados faltantes
dim(BD)
```

```
## [1] 6234   11
```

```
BD<- na.omit(BD)
dim(BD)
```

```
## [1] 4264   11
```

4.Determinando atributos dos modelos.

Geralmente, grandes diretores repetem bons trabalhos, assim usaremos a avaliação média destes como um dos atributos para prever se o filme será um sucesso. Como a grande maioria dos filmes vencedores do Oscar geralmente são produzidos nos EUA, também usaremos um atributo para sabermos se a produção foi feita no país. Por fim, nosso último atributo será se o espetáculo em questão se trata de um Filme ou algum outro tipo de programa (TV Show). Com estes atributos criaremos 3 modelos de classificação com os algoritmos Regressão Logistica, Árvore de classificação e Floresta Aleatória para prever o sucesso de novos filmes e determinar se estes entrarão no catálogo ou não.

Para que o filme entre no catálogo de filmes ele deverá ter avaliação igual ou superior a 70. Caso contrário não será aceito no catálogo.

```
#Inserindo atributo de filmes produzidos nos Estados Unidos
BD$US <- grepl("United States", BD[,5])
BD$US <- as.character(BD$US)
BD$US <- gsub("TRUE" ,1, BD$US)
BD$US <- gsub("FALSE" ,0, BD$US)

#Categorizando filmes e TV Shows
BD$typemov <- grepl("Movie", BD[,1])
BD$typemov <- as.character(BD$typemov)
BD$typemov <- gsub("TRUE" ,1, BD$typemov)
BD$typemov <- gsub("FALSE" ,0, BD$typemov)

#Calculando e inserindo atributo da média dos diretores
MeanRating <- BD %>% group_by(director) %>% summarise(AvMedia= mean(rating))
BD <- BD %>%
  left_join(MeanRating, by = c('director'))

#SELECIONANDO ATRIBUTOS A SEREM UTILIZADOS
BCKP <- BD
BD<- BCKP[,c(8,12:14)]
```

5.Visualização do novo banco de dados.

```
head(BD)
```

```
##   rating US typemov AvMedia
## 1     41  1       1      41
## 2     57  1       1      57
## 3     61  1       1      61
## 4     49  0       1      49
## 5     56  1       1      56
## 6      0  0       1       0
```

6.Transformando o novo banco de dados.

Convertendo os tipos de dados e classificando os filmes com avaliação igual ou acima de 70 como aceitos e filmes com avaliação abaixo de 70 como recusados.

```
#Determinando avaliações de producoes aprovadas
BD$rating[BD$rating>=70]<- "SIM"
BD$rating[BD$rating<70]<- "NAO"
#table(BD$rating)
#prop.table(table(BD$rating))

#Formatando tipo de dados
BD$rating <- as.factor(BD$rating)
BD$US <- as.factor(BD$US)
BD$typemov <- as.factor(BD$typemov)
```

7.Divindo o banco de dados em treino e teste.

Dividiremos o banco de dados em 70% para treinar o modelo e 30% para teste.

```
library(caret)
#Criando matriz com linhas dos dados de treino - 70%
set.seed(1)
#Separando dados de treino e teste
filtro <- createDataPartition(y=BD$rating, p=0.7, list=FALSE)
treino <- BD[filtro,]
teste <- BD[-filtro,]
```

8.Construindo modelo de regressão logistica.

Este modelo de regressão logistica está com o metódo ‘glmnet’ e usaremos validação cruzada de 5 níveis, onde serão separados 5 grupos de dados para que possamos garantir uma eficácia mínima do modelo.

```
#Criando o modelo de regressao logistica
set.seed(1)
modelo <- train(rating ~ ., data=treino, method="glmnet", tuneLenght=4, trControl = trainCont
rol(method="cv", number = 5))
#Precisao no modelo de treino
mean(modelo$resample$Accuracy)
```

```
## [1] 0.9263238
```

O modelo de regressão logistica obteve uma boa perfomance nos dados de treino com uma acurácia média de 92%.
Assim, irei aplica-lo no conjunto de teste.

```
#Prevendo dados no modelo de teste
Prev <- predict(modelo, teste)
#head(data.frame(teste$rating, Prev))
library(gmodels)
#Visualizando matriz de confusao
#Verificando acuracia
confusionMatrix(Prev, teste$rating, dnn = c("Previsto", "Real"))
```

```
## Confusion Matrix and Statistics
##
##          Real
## Previsto NAO SIM
##      NAO 753  62
##      SIM  42 421
##
##                Accuracy : 0.9186
##                  95% CI : (0.9023, 0.933)
##     No Information Rate : 0.6221
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8255
##
##  Mcnemar's Test P-Value : 0.06245
##
##             Sensitivity : 0.9472
##             Specificity : 0.8716
##          Pos Pred Value : 0.9239
##          Neg Pred Value : 0.9093
##              Prevalence : 0.6221
##          Detection Rate : 0.5892
##    Detection Prevalence : 0.6377
##       Balanced Accuracy : 0.9094
##
##        'Positive' Class : NAO
##
```

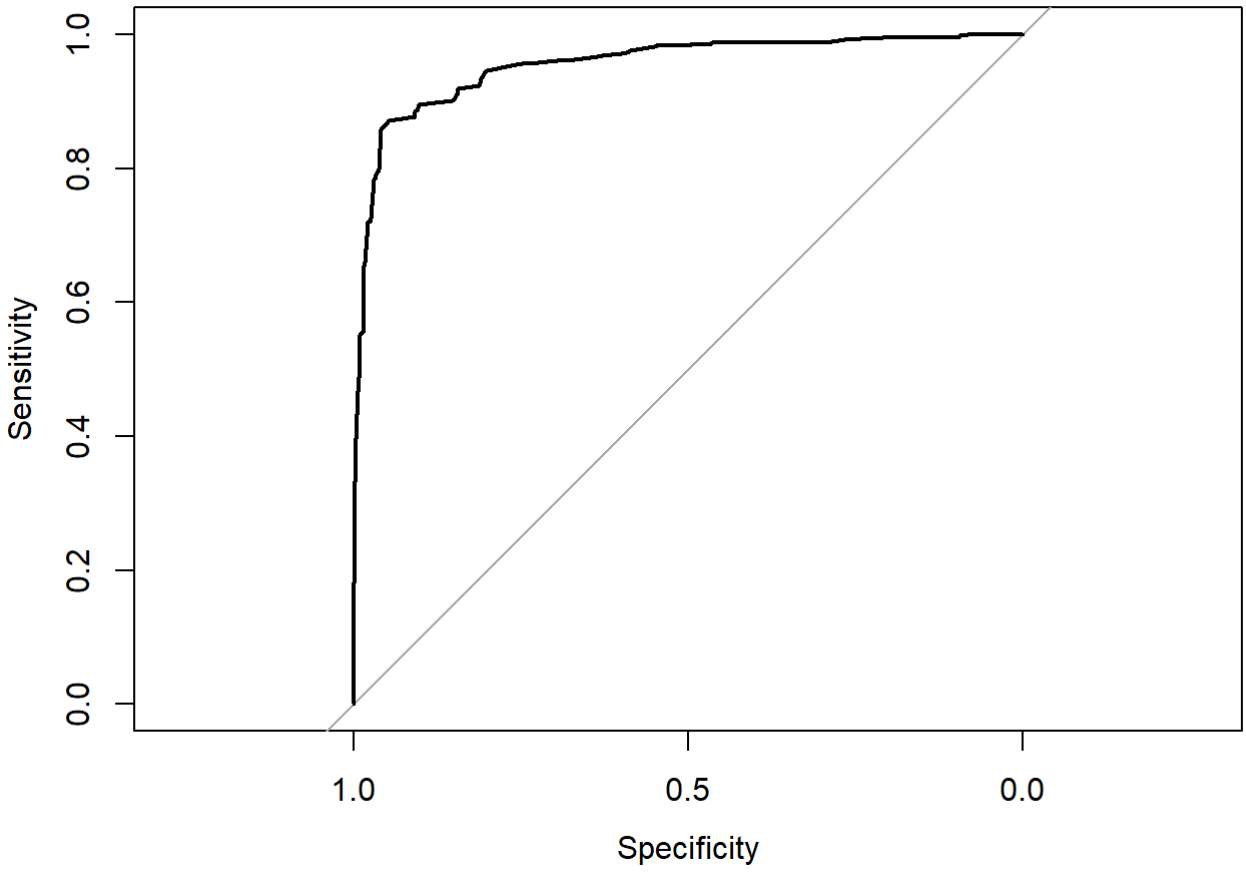
Conforme matriz de confusão acima, o modelo de regressão logistica obteve uma acurácia de **91,86%** no conjunto de teste, resultado satisfatório.

A seguir, será plotada a **curva ROC** para visualizarmos as melhores combinações de sensibilidade e especificidade que entregam a melhor acurácia do modelo.

```
#Calculando Probabilidades de ser aceito no catalogo ou nao
PrevProb <- predict(modelo, teste, type="prob")

#Visualizando probabilidades
#head(round(PrevProb, 2))
#Precisa-se armazenar apenas uma das colunas de vetores, neste caso as probabilidades dos bel
ignos
PrevProb <- PrevProb$SIM

library(pROC)
#Calculando os valores com base nos dados de teste em funÃ§Ã£o da probabilidade
ROC <- roc(teste$rating ~ PrevProb, levels= c("NAO", "SIM"))
plot(ROC)
```



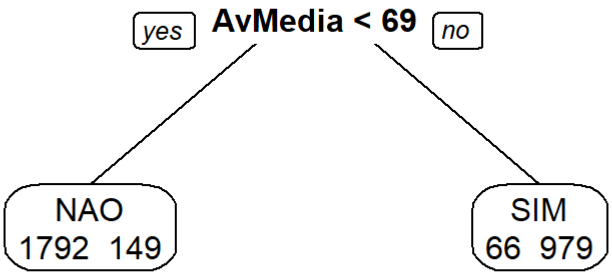

```
ROC$auc
```

```
## Area under the curve: 0.9551
```

Nota-se que a área abaixo da curva é de **0.9551** ou seja muito próxima de 1, o que torna as classificações do modelo aceitáveis.

9.Criando modelo de Árvore de Decisão.

```
#Criando o modelo de arvore de decisao
set.seed(1)
library(rpart)
modelo <- rpart(rating ~., data =treino)
library(rpart.plot)
prp(modelo, extra=1)
```



Aplicando modelo de árvore de decisão no conjunto de dados de teste.

```
confusionMatrix(predict(modelo, teste, type="class"), teste$rating)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction NAO SIM
##        NAO 755  65
##        SIM  40 418
##
##                Accuracy : 0.9178
##                  95% CI : (0.9014, 0.9323)
##     No Information Rate : 0.6221
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8235
##
##  Mcnemar's Test P-Value : 0.01917
##
##             Sensitivity : 0.9497
##             Specificity : 0.8654
##          Pos Pred Value : 0.9207
##          Neg Pred Value : 0.9127
##              Prevalence : 0.6221
##          Detection Rate : 0.5908
##    Detection Prevalence : 0.6416
##       Balanced Accuracy : 0.9076
##
##        'Positive' Class : NAO
##
```

```
confusionMatrix(predict(modelo, teste, type="class"), teste$rating)$overall["Accuracy"]
```

```
##  Accuracy
## 0.9178404
```

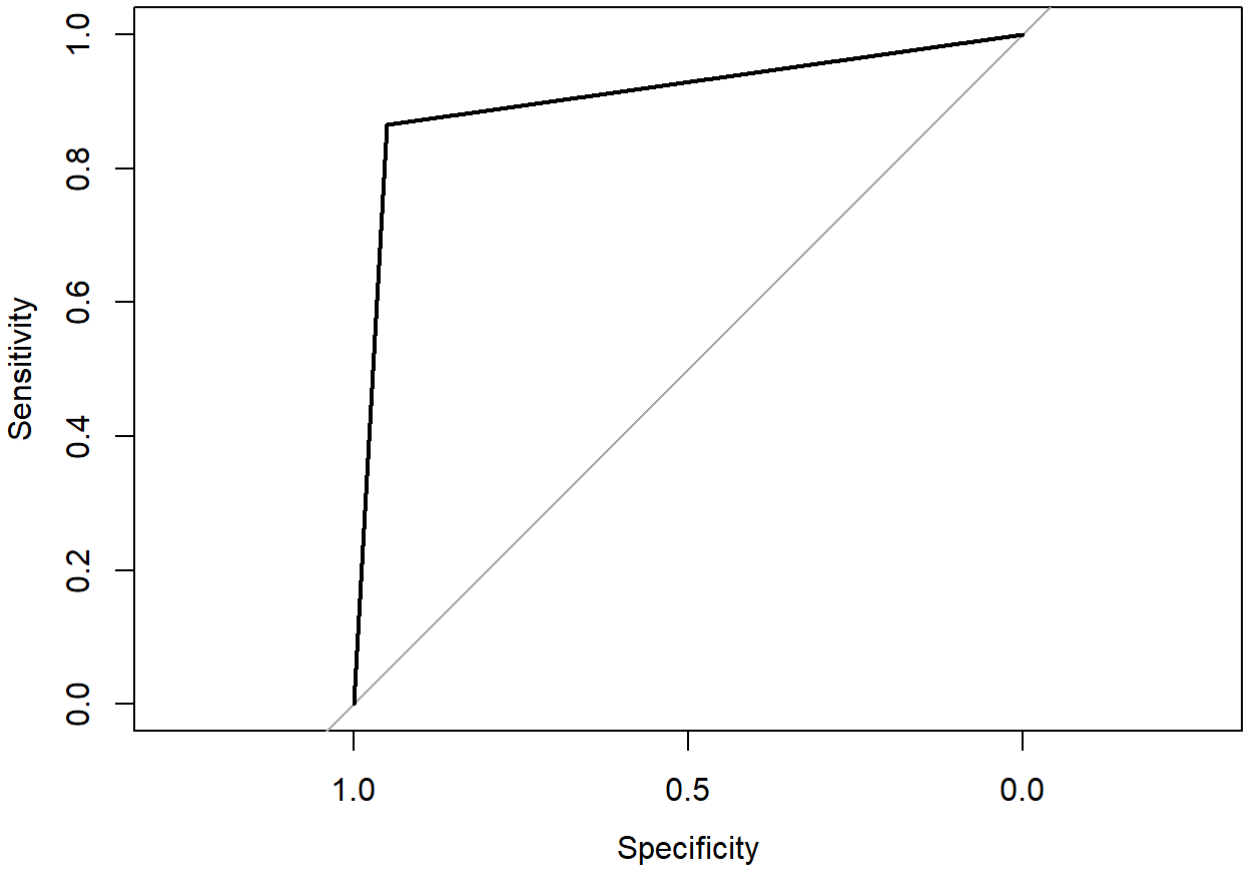
Conforme matriz de confusão acima, o modelo de Árvore de classificação obteve uma acurácia de **91,78%** no conjunto de teste.

Plotando **curva ROC** e identificando área abaixo da curva(**AUC**) para o modelo de Árvore de Classificação.

```
#Prevendo dados no modelo de teste
Prev <- predict(modelo, teste)
#head(data.frame(teste$rating, Prev))

#ROC/ AUC
#Calculando Probabilidades de ser aceito no catalogo ou nao
PrevProb <- as.data.frame(predict(modelo, teste, type="prob"))
#Visualizando probabilidades
#head(round(PrevProb, 2))
#Precisa-se armazenar apenas uma das colunas de vetores, neste caso as probabilidades de sim
PrevProb <- PrevProb$SIM

library(pROC)
#Curva ROC
#Calculando os valores com base nos dados de teste em funcao da probabilidade
ROC <- roc(teste$rating ~ PrevProb, levels= c("NAO", "SIM"))
plot(ROC)
```




```
ROC$auc
```

```
## Area under the curve: 0.9076
```

Nota-se que a área abaixo da curva do modelo de Árvore de classificação é de **0.9076%** o que tornam suas classificações aceitáveis.

10.Criando modelo de Floresta Aleatória.

```
####### RANDOM FOREST
modelo<- train(rating~., data=treino, method= "rf", ntree=100, trControl= trainControl(method
= "cv", number = 5))
```

```
## note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .
```

```
mean(modelo$resample$Accuracy)
```

```
## [1] 0.9253115
```

Aplicando modelo de Floresta Aleatória no conjunto de dados de teste.

```
confusionMatrix(predict(modelo, teste), teste$rating)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction NAO SIM
##        NAO 755  64
##        SIM  40 419
##
##                Accuracy : 0.9186
##                  95% CI : (0.9023, 0.933)
##     No Information Rate : 0.6221
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8252
##
##  Mcnemar's Test P-Value : 0.02411
##
##             Sensitivity : 0.9497
##             Specificity : 0.8675
##          Pos Pred Value : 0.9219
##          Neg Pred Value : 0.9129
##              Prevalence : 0.6221
##          Detection Rate : 0.5908
##    Detection Prevalence : 0.6408
##       Balanced Accuracy : 0.9086
##
##        'Positive' Class : NAO
##
```

```
confusionMatrix(predict(modelo, teste), teste$rating)$overall["Accuracy"]
```

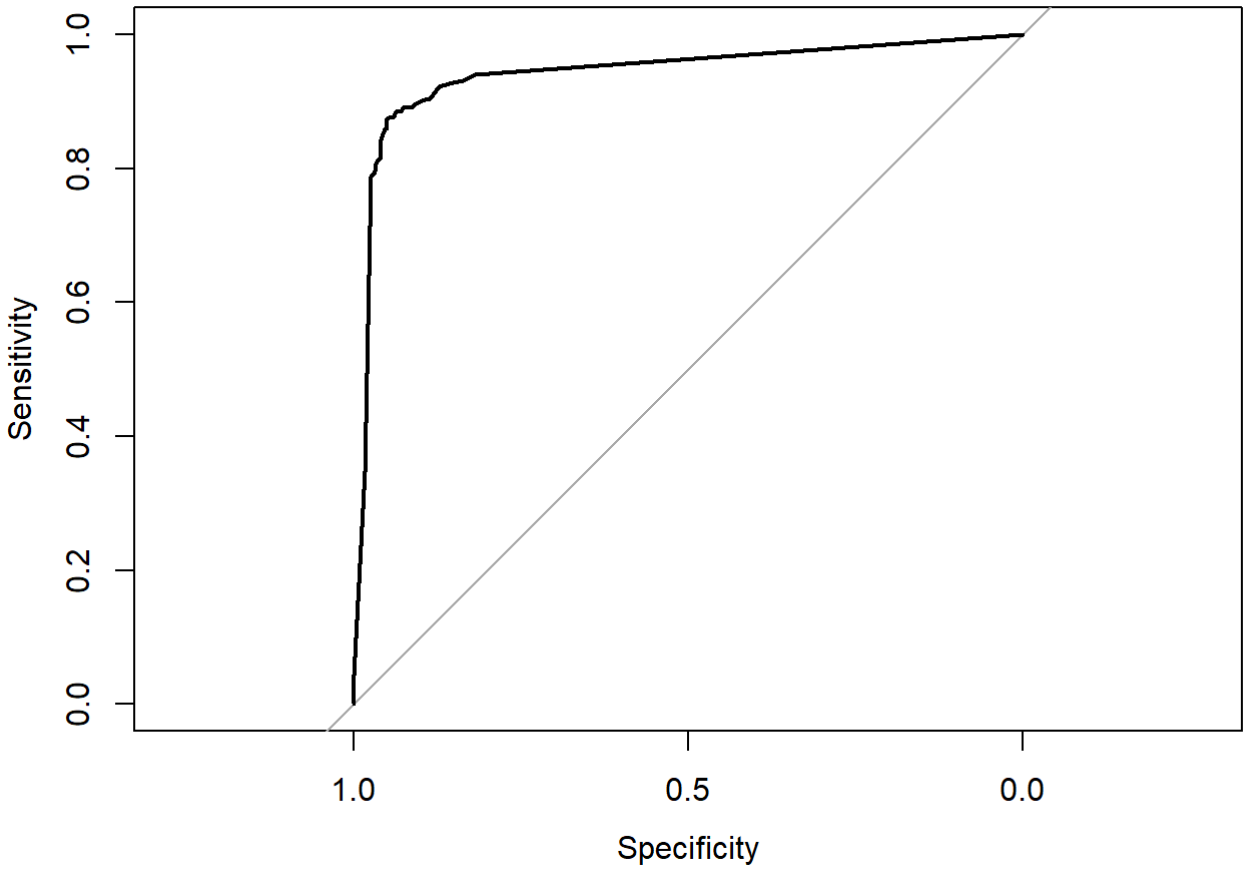
```
##  Accuracy
## 0.9186228
```

Conforme matriz de confusão acima, o modelo de Floresta Aleatória obteve uma acurácia de **91,86%** no conjunto de teste.
Plotando **curva ROC** e calculando área abaixo da curva (**AUC**) para o modelo de Floresta aleatória.

```
#Prevendo dados no modelo de teste
Prev <- predict(modelo, teste)
#head(data.frame(teste$rating, Prev))

#ROC/ AUC
#Calculando Probabilidades de ser aceito no catalogo ou nao
PrevProb <- as.data.frame(predict(modelo, teste, type="prob"))
#Visualizando probabilidades
#head(round(PrevProb, 2))
#Precisa-se armazenar apenas uma das colunas de vetores, neste caso as probabilidades dos bel
ignos
PrevProb <- PrevProb$SIM

library(pROC)
#Curva ROC
#Calculando os valores com base nos dados de teste em funÃ§Ã£o da probabilidade
ROC <- roc(teste$rating ~ PrevProb, levels= c("NAO", "SIM"))
plot(ROC)
```




```
ROC$auc
```

```
## Area under the curve: 0.9413
```

Nota-se que a área abaixo da curva para o modelo de Floresta Aleatória é de **0.9413** o também torna suas classificações aceitáveis.

11.Conclusão.

Todos os modelos apresentaram resultados satisfatórios, sendo os modelos de **Regressão logistica** e **Floresta Aleatória** ambos com **91,86%** de acurácia e o modelo de **Árvore de classificação com 91,78%** de acurácia. Portanto, qualquer um dos modelos será útil para prever o sucesso de um filme e consequentemente classificar se este deve entrar no catálogo ou não. Recomendo a utilização da regressão logistica por possuir maior parte dos dados com área abaixo da curva como pudemos observar acima e por exigir um menor custo computacional quando comparado com a Floresta Aleatória.

—

O que poderia ser feito para melhorar o modelo?

Em suma, o modelo pode ser melhorado com um tratamento de dados mais profundo e uma exploração maior dos atributos não utilizados, como por exemplo, realizar agrupamentos nos gêneros dos filmes e explorar as principais palavras chaves que geralmente aparecem na descrição dos filmes de sucessos.

–