Movie rating prediction for inclusion in Netflix catalog

Thiago Silva

08/29/2021

CHALLENGE

Let's assume that the company Netflix wants a predictive model to predict movie ratings, in order to decide whether or not it is worth putting that movie in the catalog.

Contextualization.

The database in question has 6234 observations and 12 attributes, namely the show ID, Movie Type, Movie Name, Director, Cast, Country, Date added, Year of production, Rating, Duration, Genre and Description.

The purpose of the Challenge is to create a model to predict whether a new movie will be a hit and decide if it should go into the catalogue.

1. Reading and viewing the database

```
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)
```

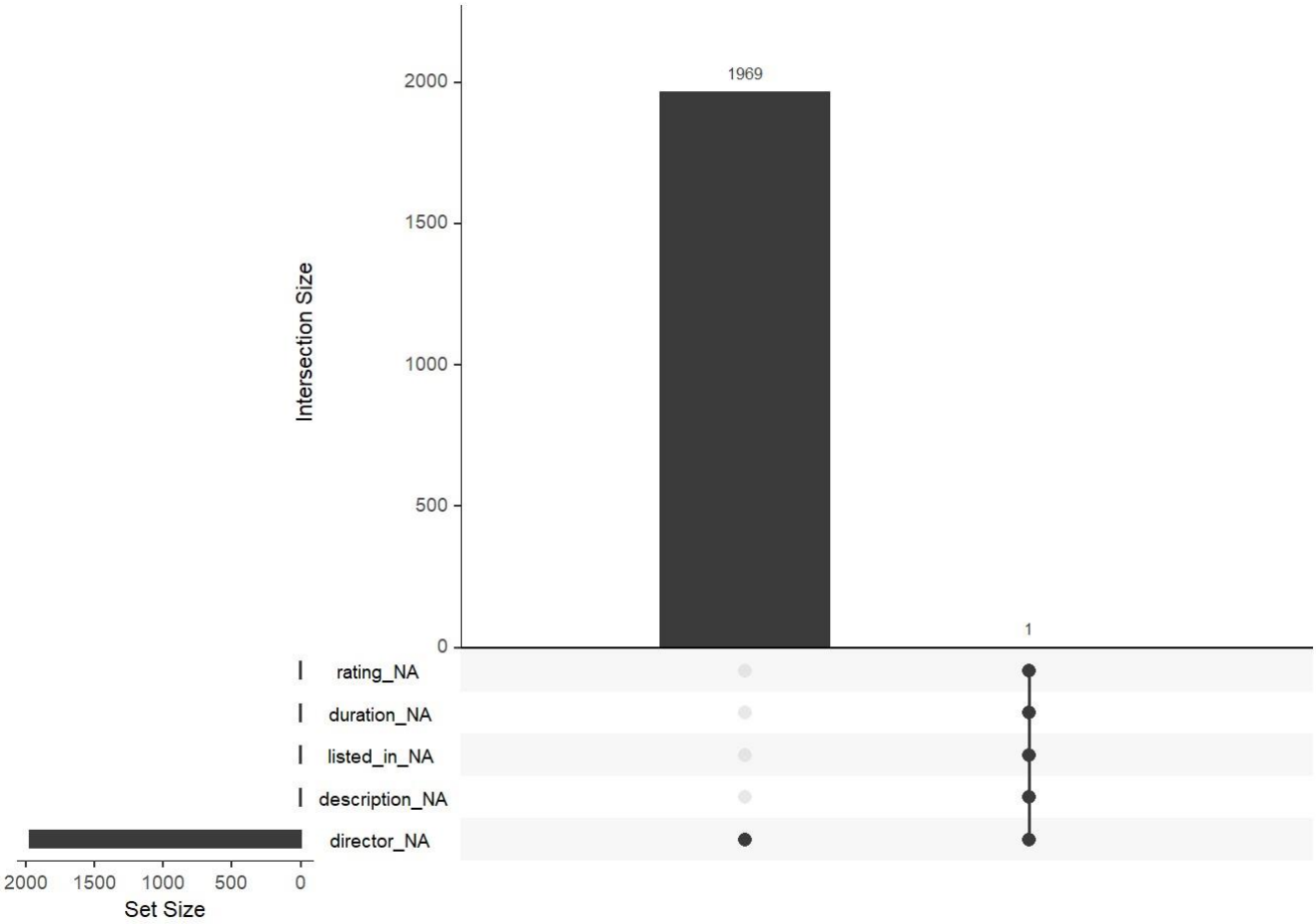
```
##   X  show_id    type                                    title
## 1 1 81145628   Movie Norm of the North: King Sized Adventure
## 2 2 80117401   Movie              Jandino: Whatever it Takes
## 3 3 70234439 TV Show                      Transformers Prime
## 4 4 80058654 TV Show        Transformers: 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. Analyzing missing data.

```
#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
```

The database is damaged by missing data, and as illustrated above the Directors column is the most affected with a little over 30% “missing values”. All other attributes that had missing data contain only 1 line of missing data. As the database contains 6234 observations, we will eliminate the missing data and work with 4264 observations.

3.Removing missing data.

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

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

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

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

4.Determining model attributes.

Generally, great directors repeat good works, so we will use the average rating of these as one of the attributes to predict whether the film will be a success. Since the vast majority of Oscar-winning films are usually produced in the US, we'll also use an attribute to know if the production was made in the US. Finally, our last attribute will be whether the show in question is a Movie or some other type of program (TV Show). With these attributes we will create 3 classification models with the Logistic Regression, Classification Tree and Random Forest algorithms to predict the success of new films and determine whether they will enter the catalog or not.

For the film to enter the film catalog it must have a rating equal to or greater than 70. Otherwise, it will not be accepted in the catalogue.

```
#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. Preview of the new database.

```
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. Transforming the new database.

Converting the data types and classifying movies with a rating of 70 or above as accepted and movies with a rating below 70 as rejected.

```
#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. Splitting the database into training and testing.

We will split the database into 70% for training the model and 30% for testing.

```
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. Building a logistic regression model.

This logistic regression model is using the 'glmnet' method and we will use 5-level cross-validation, where 5 groups of data will be separated so that we can guarantee a minimum efficiency of the model.

```
#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
```

The logistic regression model performed well on the training data with an average accuracy of 92%.
So, I will apply it to the test set.

```
#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"))
```

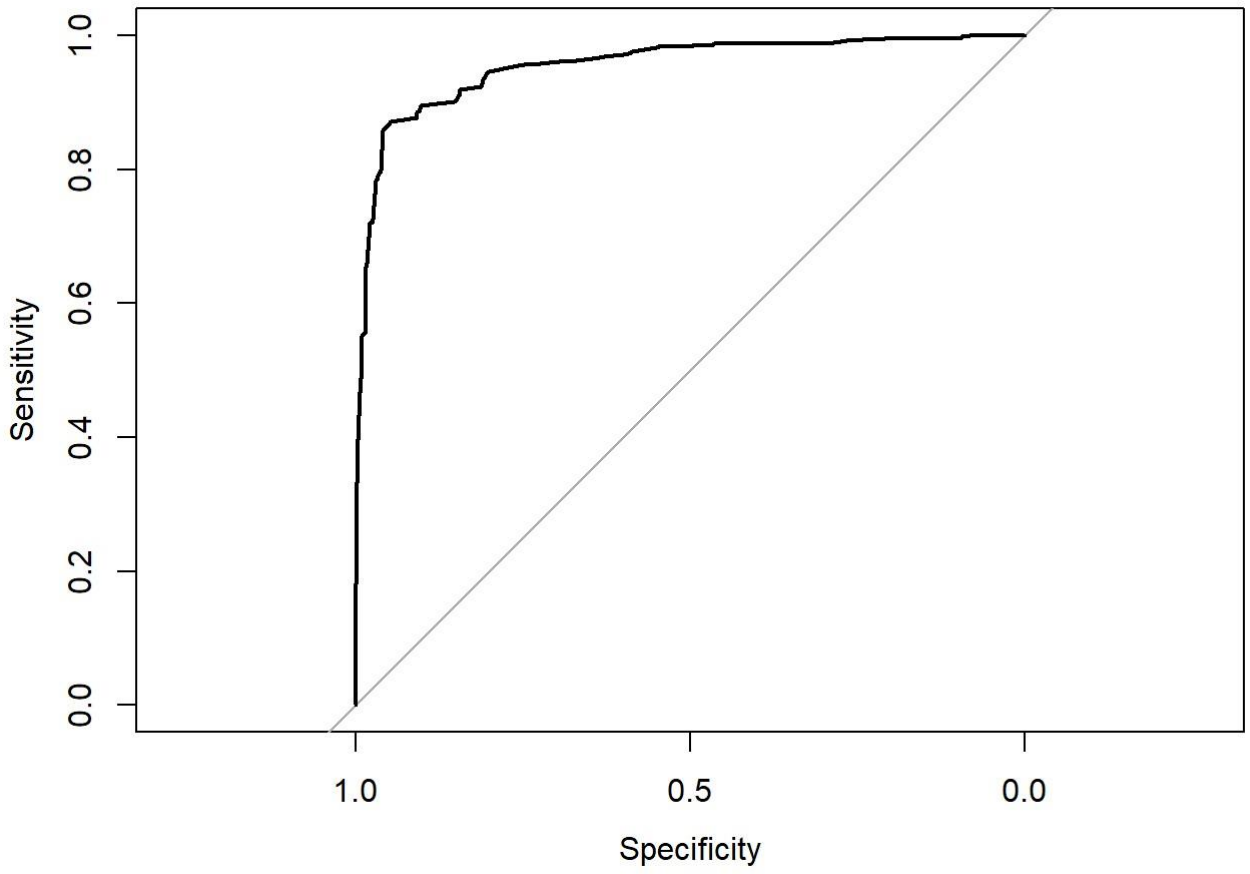
```
## Confusion Matrix and Statistics
##
##          Real
## Previsto NAO SIM
##      NAO 753  62
##      SIM  42 421
##
##                Accuracy : 0.9186
##                  95% CI : (0.9023, 0.933)
##     No Information Rate : 0.6221
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 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
##
```

According to the confusion matrix above, the logistic regression model obtained an accuracy of 91.86% in the test set, a satisfactory result.
Next, the ROC curve will be plotted to visualize the best combinations of sensitivity and specificity that deliver the best model accuracy.

```
#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)
```



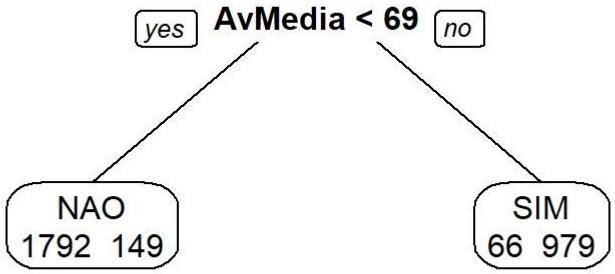

```
ROC$auc
```

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

Note that the area under the curve is 0.9551, which is very close to 1, which makes the model classifications acceptable.

9. Creating Decision Tree model.

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



Applying decision tree model to test dataset.

```
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
```

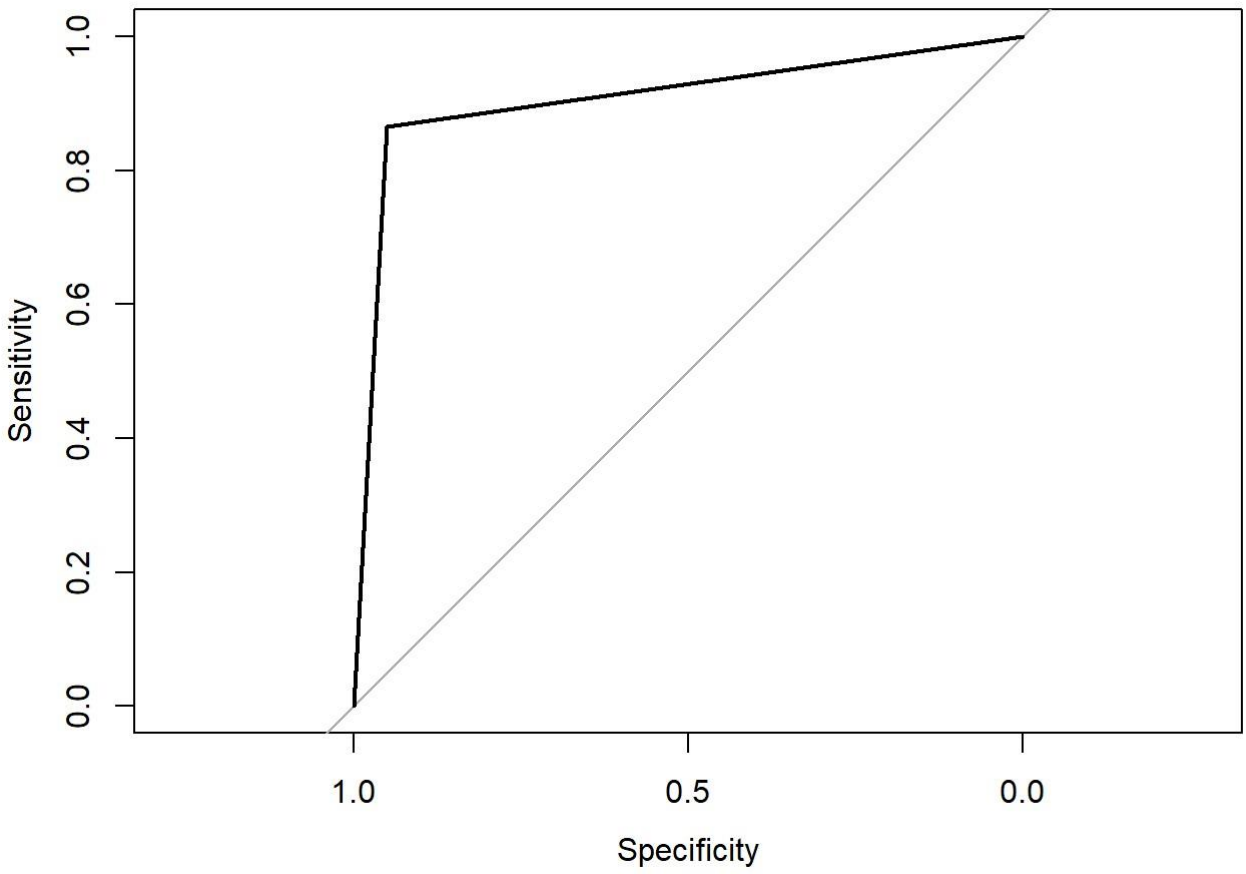
According to the confusion matrix above, the Classification Tree model obtained an accuracy of 91.78% in the test set.

Plotting ROC curve and identifying area under the curve (AUC) for the Classification Tree model.

```
#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
```

Note that the area under the curve of the Classification Tree model is 0.9076%, which makes its classifications acceptable.

10. Creating Random Forest model.

```
######## 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
```

Applying Random Forest model to test dataset.

```
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"]
```

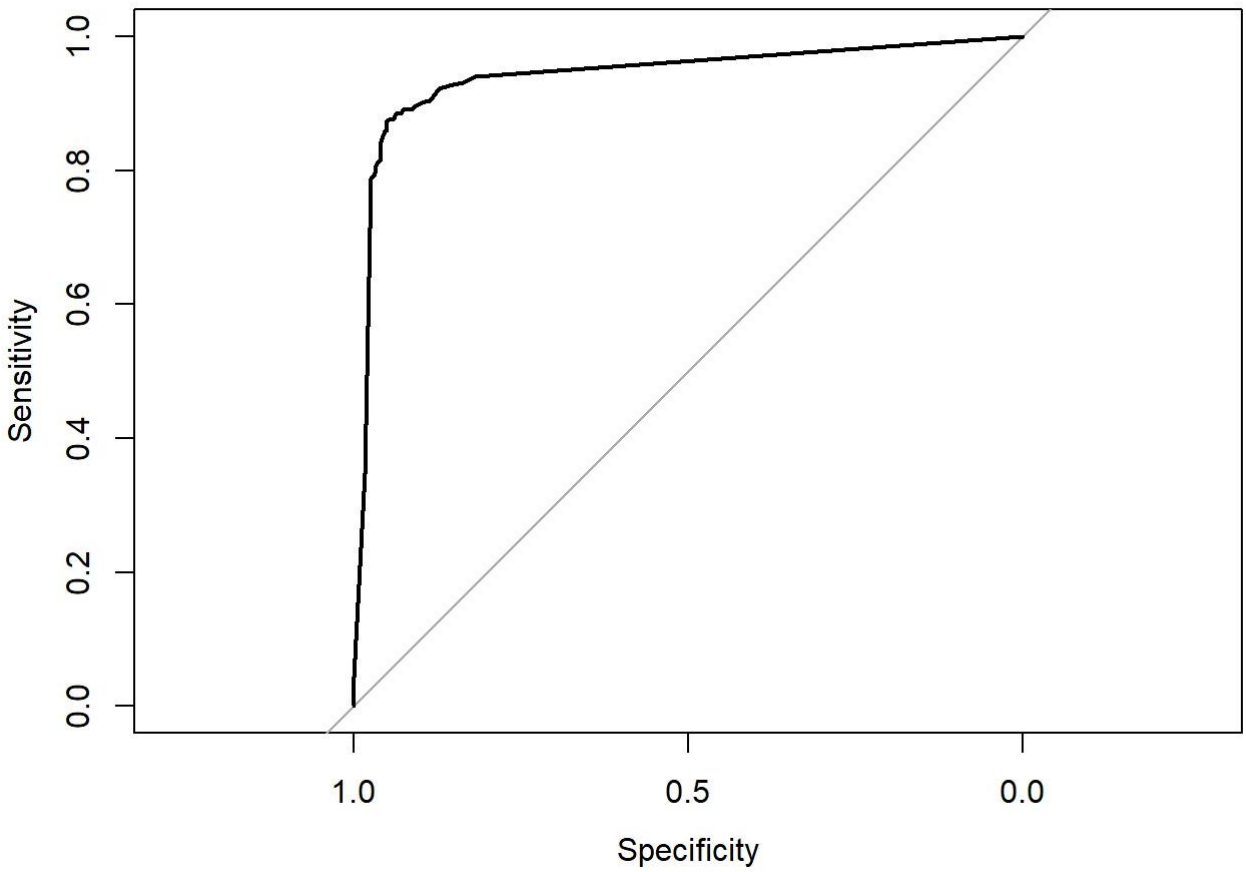
```
##  Accuracy
## 0.9186228
```

According to the confusion matrix above, the Random Forest model obtained an accuracy of 91.86% in the test set. Plotting ROC curve and calculating area under the curve (AUC) for Random Forest model.

```
#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
```

Note that the area under the curve for the Random Forest model is 0.9413 o also makes its classifications acceptable.

11. Conclusion.

All models showed satisfactory results, being the Logistic Regression and Random Forest models both with 91.86% accuracy and the Classification Tree model with 91.78% accuracy. Therefore, any of the models will be useful to predict the success of a movie and

consequently classify whether it should enter the catalog or not. I recommend the use of logistic regression for having most of the data with area under the curve as we could see above and for requiring a lower computational cost when compared to the Random Forest.

—

What could be done to improve the model?

In short, the model can be improved with deeper data processing and a greater exploration of unused attributes, such as, for example, grouping the film genres and exploring the main keywords that usually appear in the description of successful films.