# Análise - Machine Learning

Antonio Mendes M. Jr

27/02/2021

## Análise Machine Learning

Neste documento é descrita a manipulação de um conjunto de dados, o desenvolvimento de um classificador e o cálculo de métricas de avaliação.

### Pacotes utilizados

Primeiramente foram carregados os pacotes utilizados. O pacote `readxl` para leitura de arquivos xls, `dplyr` para operações com os dataframes e afins, `ggplot2` para trabalhar com gráficos, `pROC` para curvas ROC e `caret` para treinar modelos e calcular métricas de avaliação.

```
library(readxl)
library(ggplot2)
library(dplyr)
library(pROC)
library(caret)
```

### Carregamento dos dados

### Dataset

O dataset é composto por quatro colunas, onde se encontram classe predita (Pred_class), probabilidade (probabilidade), status (status) e classe verdadeira (True_class). Há dados faltantes em classe verdadeira e foram completados, conforme orientação, com os dados de classe predita.

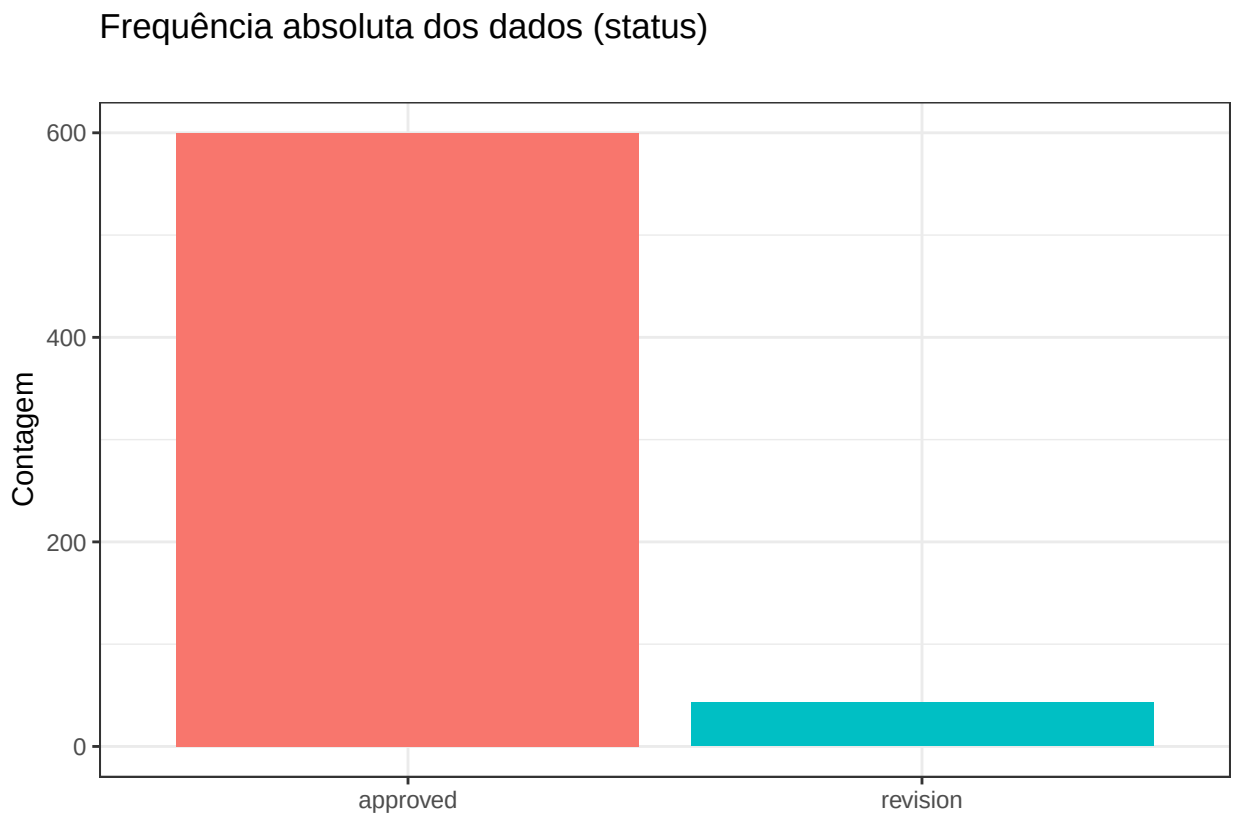
```
dados <- read_excel("teste_smarkio_lbs.xls", sheet=1, col_names=TRUE)
dados$True_class <- ifelse(is.na(dados$True_class),
                           dados$Pred_class, dados$True_class)
dados$probabilidade <- as.numeric(dados$probabilidade)
dados$True_class <- as.numeric(dados$True_class)
head(dados, 6)
```

```
## # A tibble: 6 x 4
##   Pred_class probabilidade status   True_class
##        <dbl>         <dbl> <chr>         <dbl>
## 1          2        0.0799 approved          0
## 2          2        0.379  approved         74
## 3          2        0.379  approved         74
## 4          2        0.421  approved         74
## 5          2        0.607  approved          2
## 6          2        0.691  approved          2
```

## 1) Análise exploratória

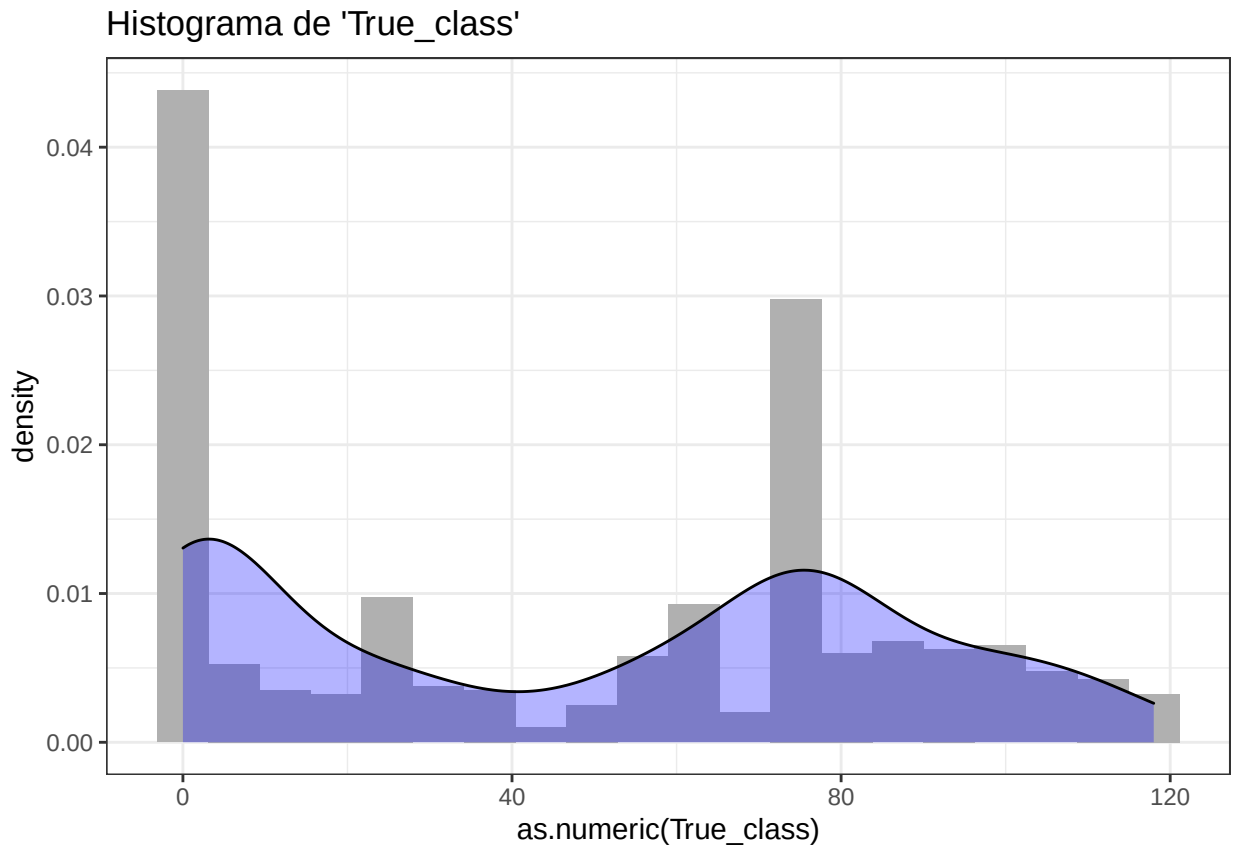
A fim de conhecer melhor a distribuição dos dados, foram plotados alguns gráficos. O primeiro gráfico é referente à variável status. Esta e uma variável qualitativa e divide os dados entre duas classes: approved e revision.

```
ggplot(dados, aes(x = status,y = ..count.., fill=status)) +
  geom_bar(alpha=1)+
  theme_bw()+
  labs(
    title = "Frequência absoluta dos dados (status)",
    subtitle = " ",
    x = "",
    y = "Contagem") +
  theme(legend.position="none")
```

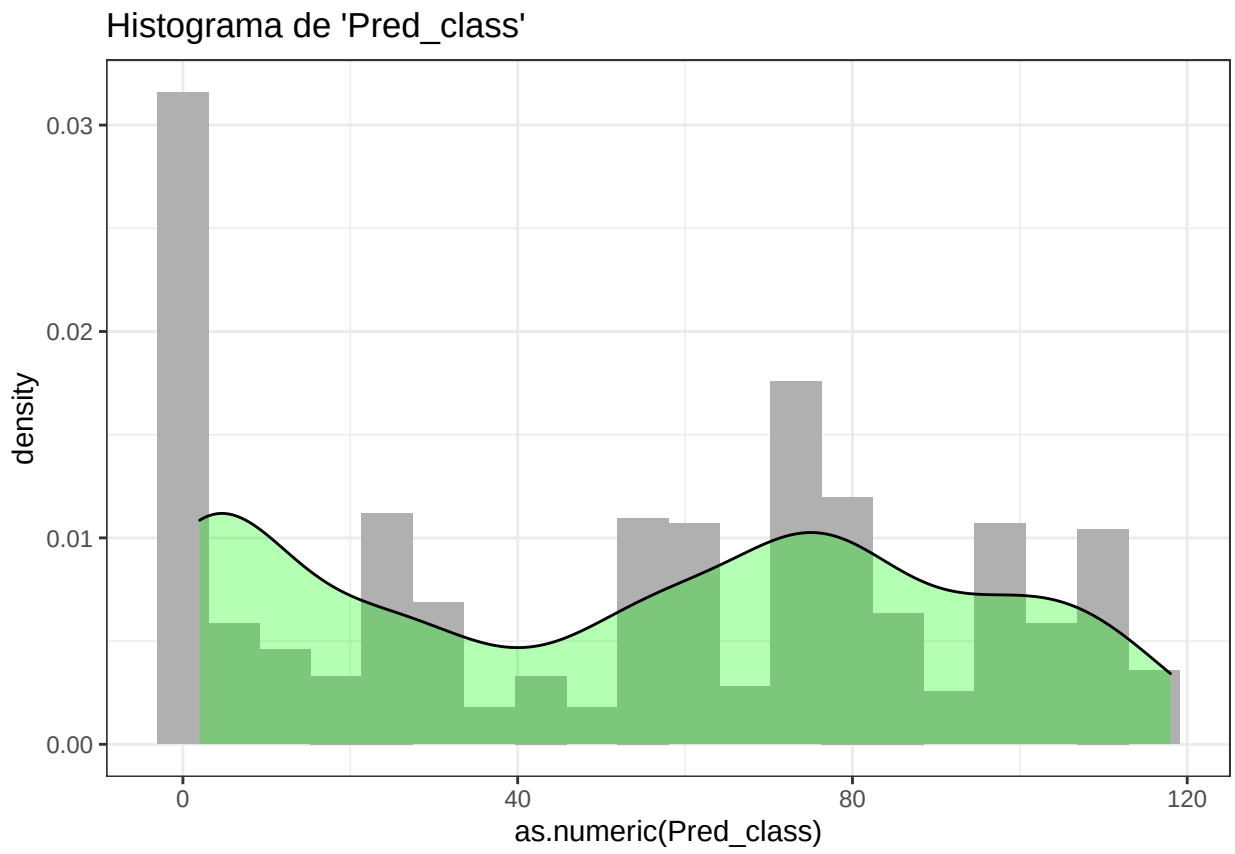



A seguir são plotados os histogramas das variáveis True_class, Pred_class e probabilidade. Por meio dos histogramas é possível visualizar importantes medidas de posição, como a média e moda.

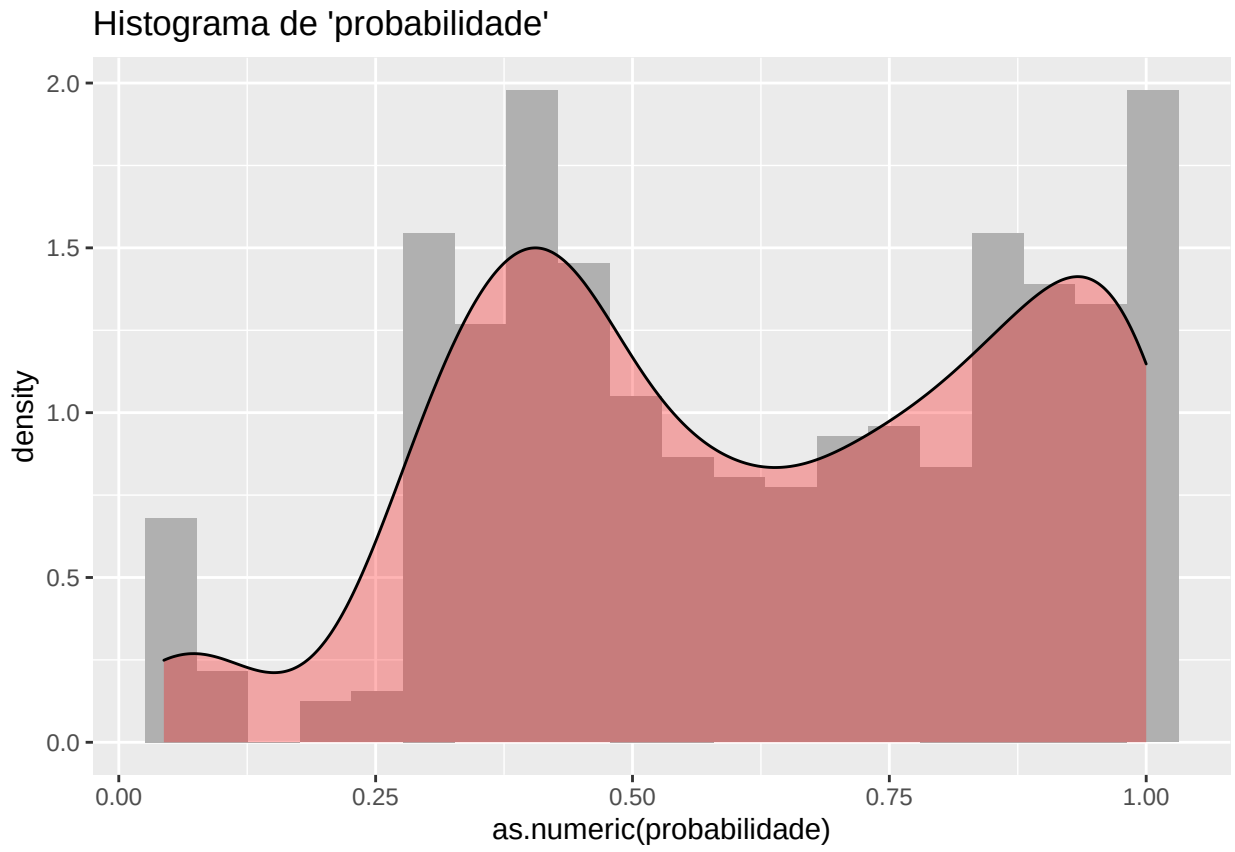
```
# Histograma de True_class
ggplot(dados, aes(x = as.numeric(True_class),y =..density..)) +
  geom_histogram(bins = 20,fill="gray69")+
  theme_bw()+
  labs(
    title = "Histograma de 'True_class'")+
  geom_density(fill = "blue", alpha = 0.3)
```




```
# Histograma de Pred_class
ggplot(dados, aes(x = as.numeric(Pred_class),y =..density..)) +
  geom_histogram(bins = 20,fill="gray69")+
  theme_bw()+
   labs(
    title = "Histograma de 'Pred_class'")+
  geom_density(fill = "green", alpha = 0.3)
```




```
# Histograma de probabilidade
ggplot(dados, aes(x = as.numeric(probabilidade),y =..density..)) +
  geom_histogram(bins = 20,fill="gray69")+
  labs(
    title = "Histograma de 'probabilidade'")+
  geom_density(fill = "red", alpha = 0.3)
```




## Desempenho do modelo

O conjunto de dados traz consigo a predição realizada por um dado modelo. Nesta seção são calculados algumas métricas para avaliação deste modelo.

Foram detectadas todas as classes possíveis a partir da união das classificações em Pred_class e True_class. Em seguida foi calculada a Matriz de Confusão do modelo e construídas curvas ROC multiclasses, dessa forma foi possível extrair métricas como acurácia, índice Kappa e AUC.

```
true_c <- as.factor(as.numeric(dados$True_class))
pred_c <- as.factor(dados$Pred_class)
u <- union(levels(true_c ),levels(pred_c))
t <- table(factor(pred_c, u), factor(true_c, u))

roc <- pROC::multiclass.roc(dados$True_class ~ dados$Pred_class, plot=F,
                            print.auc=F, legacy.axes=T)

metrics <- caret::confusionMatrix(t)
cm <- metrics$table
(acc <- metrics$overall[1])
```

```
## Accuracy
## 0.718507
```

```
(kappa <- metrics$overall[2])
```

```
##     Kappa
## 0.7063119
```

```
(auc <- roc$auc)
```

```
## Multi-class area under the curve: 0.8819
```

## Construção de um classificador

Nesta etapa foi construído um classificador do tipo Floresta Aleatória. Os dados da classe `approved` foram utilizados como dados de treinamento, enquanto os da clase `revision` foram utilizados como dados de teste. Novamente foram calculadas acurácia, índice Kappa e AUC.

```
dados <- read_excel("teste_smarkio_lbs.xls", sheet=1, col_names=TRUE)

dados$True_class <- ifelse(is.na(dados$True_class),
                           dados$Pred_class, dados$True_class)
dados$probabilidade <- as.numeric(dados$probabilidade)
dados$Pred_class <- as.numeric(dados$Pred_class)

dados_approved <- filter(dados, status=='approved')
dados_revision <- filter(dados, status== 'revision')

dados_approved$True_class <- as.factor(as.numeric(dados_approved$True_class))
dados_revision$True_class <- as.factor(as.numeric(dados_revision$True_class))
```

Foram utilizados como atributos as variávels `Pred_class` e `probabilidade`. Desta forma, a coluna com a variável `status` foi removida.

```
data_train <- dados_approved[-3]
data_test <- dados_revision[-3]
```

O próximo passo foi criar o modelo e treiná-lo. O treinamento se deu utilizando K-fold com repetição. Foram definidas 10 fols e 3 repetições.

```
train_control <- trainControl(method= 'repeatedcv', number=10, repeats=3)
model_rf <- train(True_class ~ ., data = data_train, method = 'rf',
                  trControl=train_control)
```

```
## note: only 1 unique complexity parameters in default grid. Truncating the grid to 1 .
```

Em seguida o modelo foi utilizado para fazer predição acerca dos dados da classe `revision`.

```
pred_rf <- predict(model_rf, data_test)
```

A curva ROC para o caso multiclasse foi calculada. Neste são calculadas N curvas, utilizando a técnica OvA (One vs All), em que uma classe é comparada contra todas as demais. A AUC final é a área média abaixo das curvas geradas.

```
data_test_code <- as.numeric(data_test$True_class)
resp_test_code <- as.numeric(pred_rf)
roc <- pROC::multiclass.roc(data_test_code ~ resp_test_code, plot=F)
```

Por fim, foi calculada a Matriz de Confusão e as métricas de avaliação.

```
levels_all <- union(levels(as.numeric(pred_rf)),
                    levels(dados_revision$True_class))

true_c <- factor(dados_revision$True_class, levels_all)
pred_c <- factor(pred_rf, levels_all)
```

```r
t <- table(pred_c, true_c)

metrics <- caret::confusionMatrix(t)
cm <- metrics$table
(acc <- metrics$overall[1])
```

```
##  Accuracy
## 0.8148148
```

```r
(kappa <- metrics$overall[2])
```

```
##     Kappa
## 0.7932619
```

```r
(auc <- roc$auc)
```

```
## Multi-class area under the curve: 0.9247
```

## Comparação das métricas

- Acurácia: é a métrica mais básica; é extraída da matriz de confusão, dada pela contagem dos itens na diagonal principal dividido pelo total de itens; é a soma dos verdadeiros positivos e verdadeiros negativos divididos pela soma dos verdadeiros positivos e negativos e falsos negativos e positivos.
- Índice Kappa: este índice mede a concordância entre avaliadores; neste caso, mede a concordância entre a predição realizada pelo modelo e a predição real.
- ROC: as curvas ROC representam o trade-off entre verdadeiros positivos e falsos positivos; a fim de facilitar sua interpretação, geralmente se reduz sua avaliação à AUC, que é a área abaixo da curva; AUC = 1 representa um classificador perfeito.