PECE Programa de Educação Continuada
Escola Politécnica da USP

BIG-008
Análise Preditiva na Prática

Profa.: Rosangela de Fátima Pereira Marquesone

Big Data: Inteligência na Gestão dos Dados

PECE Programa de Educação Continuada
Escola Politécnica da USP

Tópicos

- Hands-on: classificação de mensagens

PECE Programa de Educação Continuada
Escola Politécnica da USP

Classificação

Passos para a construção de um classificador:

1. Coleta de dados históricos
2. Limpeza dos dados
3. Definição de variáveis
4. Definição de variáveis explanatórias
5. Seleção de um algoritmo
6. Construção de um modelo com base de treinamento
7. Avaliação do modelo com base de teste
8. Ajuste de variáveis explanatórias (caso necessário)
9. Executar novamente com a base de teste (caso necessário)

Desafio

- Calcular a probabilidade de um SMS (*Short Message Service*) ser ou não um spam
- Uma das técnicas mais populares utiliza filtro de conteúdo baseado em **Classificação Bayesiana (Naive Bayes)**

- Esse classificador atribui uma **probabilidade de que uma nova amostra está em uma classe** (spam) ou em outra (ham).

- A partir das palavras que estão e das que não estão na mensagem, essa técnica calcula a probabilidade de spam ou não-spam.

- É baseado na **regra de Bayes** e da **análise de frequência de ocorrências de palavras**.

Base de dados

Arquivo contendo mensagens SMS do tipo spam e ham

```
type,text
ham,Hope you are having a good week. Just checking in
ham,K..give back my thanks.
ham,Am also doing in cbe only. But have to pay.
spam,"complimentary 4 STAR Ibiza Holiday or £10,000 cash needs your
URGENT collection.
spam,okmail: Dear Dave this is your final notice to collect your 4*
Tenerife Holiday or #5000 CASH award! Call 09061743806 from landline.
TCs SAE Box326 CW25WX 150ppm
ham,Aiya we discuss later lar... Pick u up at 4 is it?
```

5.559 mensagens

O que utilizaremos:

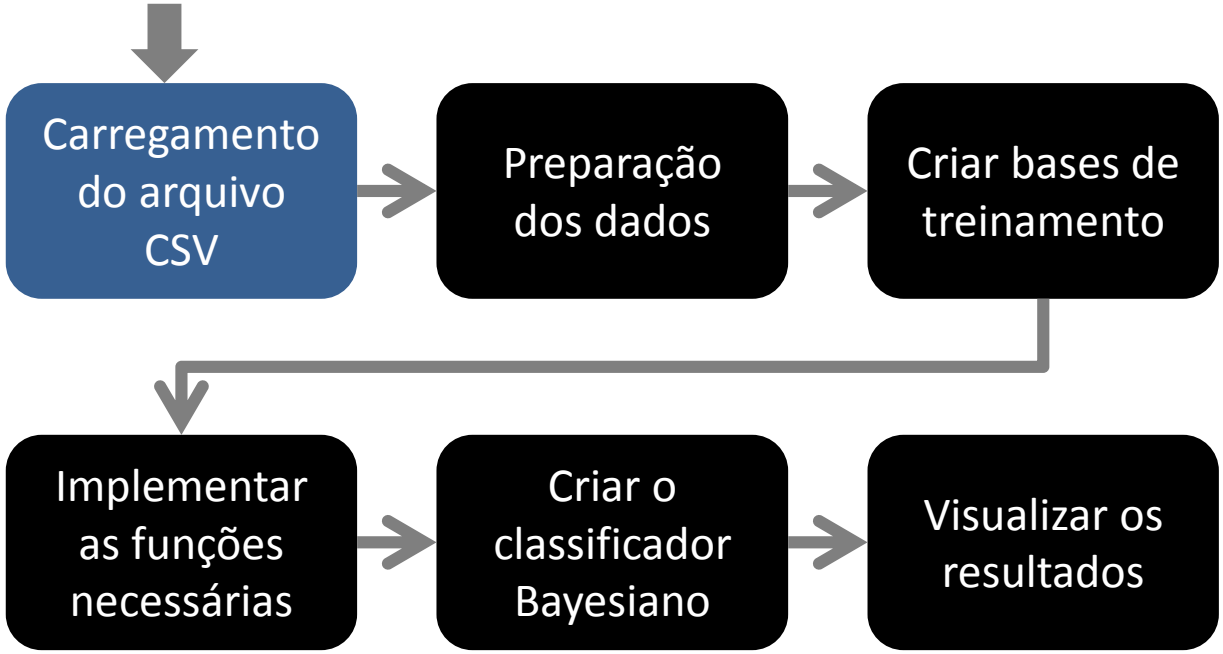
- R versão 3.4.1 - https://cran.r-project.org/bin/windows/base/

- Bases de dados:
  - sms_spam.csv - https://goo.gl/uTT2pG

Configurar o diretório de trabalho (*work directory*) do R

Console

```
> setwd("C:/Users/Aluno/Documents/Rnapratica")

> getwd()
[1] "C:/Users/Aluno/Documents/Rnapratica"
```


Carregamento do arquivo CSV
Preparação dos dados
Criar bases de treinamento
Implementar as funções necessárias
Criar o classificador Bayesiano
Visualizar os resultados

Pacotes utilizados

```
> install.packages("tm")
> install.packages("SnowballC")
> install.packages("wordcloud")
> install.packages("e1071")
> install.packages("gmodels")
> library(tm)
> library(SnowballC)
> library(wordcloud)
> library(e1071)
> library(gmodels)
```

Carregando a base de dados para o R

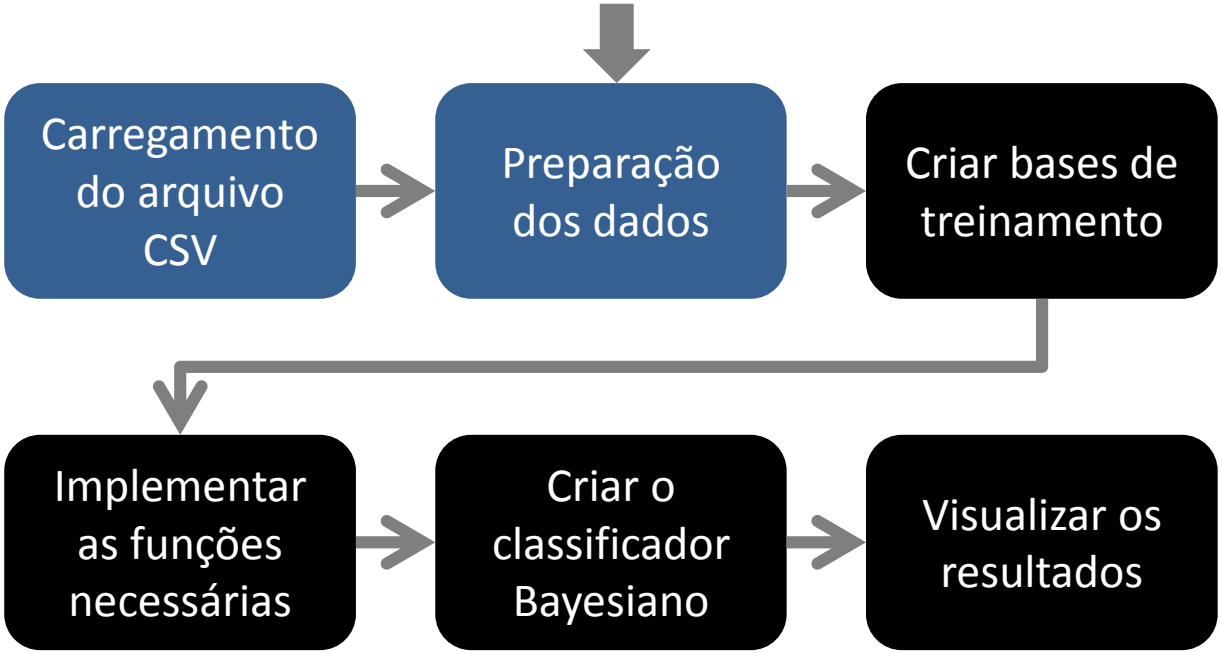
Console

```
> sms_df <- read.csv("sms_spam.csv", stringsAsFactors = FALSE)

#criando a variável categórica “type” em um fator no R
> sms_df$type <- factor(sms_df$type)
```

Verificando a proporção de dados na base

Console

```
> table(sms_df$type)
##
##  ham spam
## 4812  747
```


Carregamento do arquivo CSV
Preparação dos dados
Criar bases de treinamento
Implementar as funções necessárias
Criar o classificador Bayesiano
Visualizar os resultados

Criando uma coleção de dados (Corpus) para mineração da base

Console

```
> sms_corpus <- VCorpus(VectorSource(sms_df$text))
```

Verificando o conteúdo do Corpus

Console

```
> print(sms_corpus)
<<VCorpus>>
Metadata:  corpus specific: 0, document level
(indexed): 0
Content:  documents: 5559
```

Aplicando métodos para “limpeza” dos dados

Console

```
> sms_corpus_clean <- tm_map(sms_corpus,
content_transformer(tolower))
```

Aplicando métodos para “limpeza” dos dados

Remove números encontrados na base de dados

Console

```
> sms_corpus_clean <- tm_map(sms_corpus_clean,
removeNumbers)
```

Aplicando métodos para “limpeza” dos dados

Visualizar ***stop words***

Console

```
> stopwords()
  [1] "i" "me" "my" [4] "myself" "we" "our"
  [7] "ours" "ourselves" "you"
 [10] "your" "yours" "yourself"
 [13] "yourselves" "he" "him"
 [16] "his" "himself" "she"
```

Aplicando métodos para “limpeza” dos dados

Remover *stop words*

Console

```
> sms_corpus_clean <- tm_map(sms_corpus_clean,
removeWords, stopwords())
```

Aplicando métodos para “limpeza” dos dados

Remove pontuações da base de dados

Console

```
> sms_corpus_clean <- tm_map(sms_corpus_clean,
removePunctuation)
```

Aplicando métodos para “limpeza” dos dados

Remove espaços extra em branco da base de dados

Console

```
> sms_corpus_clean <- tm_map(sms_corpus_clean,
stripWhitespace)
```

Aplicando métodos para “limpeza” dos dados

Remove sufixo de palavras

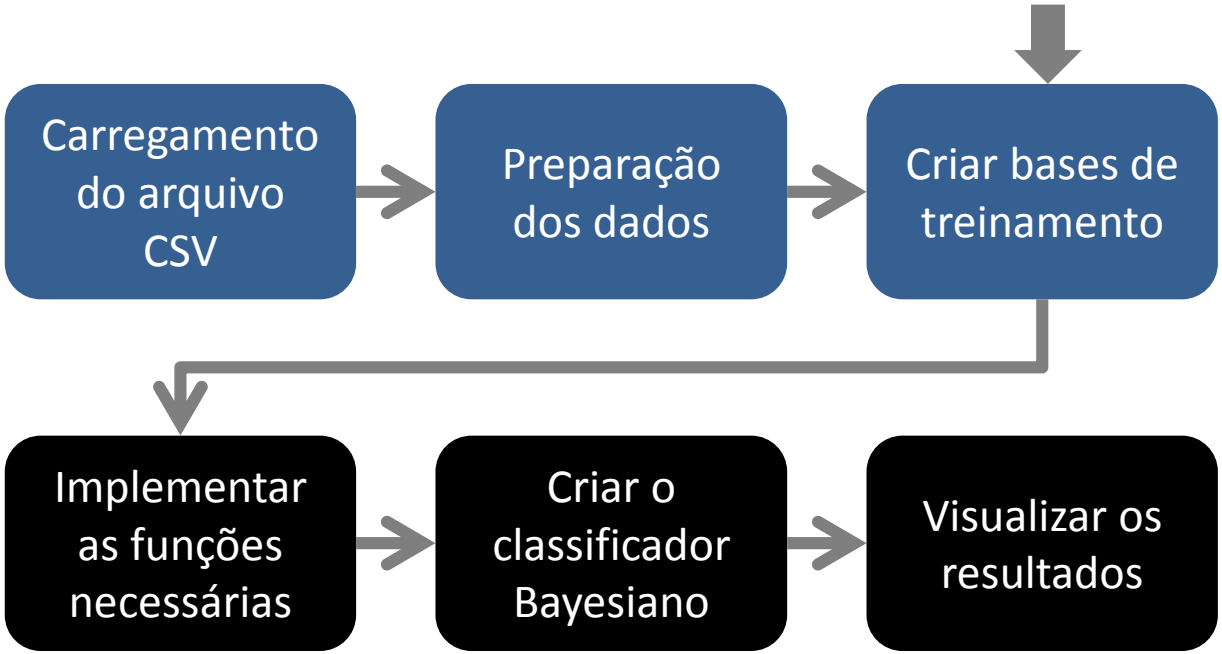
Console

```
> wordStem(c("learn", "learned", "learning", "learns"))
## [1] "learn" "learn" "learn" "learn"

> sms_corpus_clean  <- tm_map(sms_corpus_clean,
stemDocument)
```

Comparando a base antes e após a limpeza

Console

```
> lapply(sms_corpus[1:3], as.character)
$`1`
[1] "Hope you are having a good week. Just checking in"
$`2`
[1] "K..give back my thanks."
$`3`
[1] "Am also doing in cbe only. But have to pay.“

> lapply(sms_corpus_clean[1:3], as.character)
$`1`
[1] "hope good week just check"
$`2`
[1] "kgive back thank"
$`3`
[1] "also cbe pay"
```

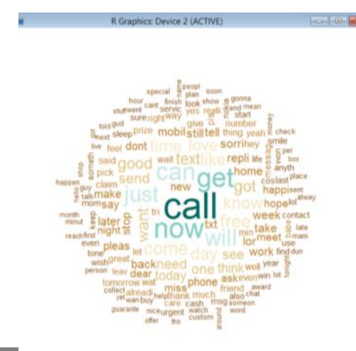

Carregamento do arquivo CSV
Preparação dos dados
Criar bases de treinamento
Implementar as funções necessárias
Criar o classificador Bayesiano
Visualizar os resultados

Visualização das palavras encontradas na base

Chamada da biblioteca wordcloud

Console

```
> wordcloud(sms_corpus_clean, min.freq = 50, random.order = FALSE, colors=brewer.pal(8, "BrBG"))
```



Criação de índices das mensagens spam e ham

Console

```
> spam <- subset(sms_df, type == "spam")

> ham  <- subset(sms_df, type == "ham")
```

Visualização das palavras encontradas em cada tipo de mensagem

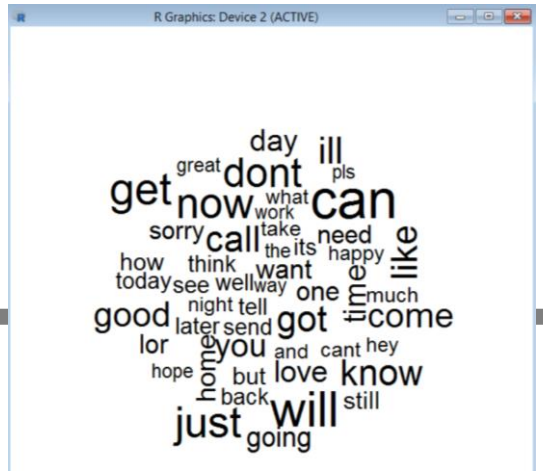
Console

```
> wordcloud(spam$text, max.words = 50)
```

Visualização das palavras encontradas em cada tipo de mensagem

Console

```
> wordcloud(ham$text, max.words = 50)
```



Criação da matriz de termos em documentos

Console

```
> sms_dtm <- DocumentTermMatrix(sms_corpus_clean)
```

Verificando o conteúdo da matriz

Console

```
> inspect(sms_dtm[1:4, 30:35])
<<DocumentTermMatrix (documents: 4, terms: 6)>>
Non-/sparse entries: 0/24
Sparsity           : 100%
Maximal term length: 8
Weighting          : term frequency (tf)

    Terms
Docs abstract abt abta aburo abuse abusers
   1        0   0    0     0     0       0
   2        0   0    0     0     0       0
   3        0   0    0     0     0       0
   4        0   0    0     0     0       0
```

Criar base de teste e base de treinamento

75% dos dados para treinamento e 25% para teste.

Console

```
> sms_dtm_train <- sms_dtm[1:4169, ]
> sms_dtm_test  <- sms_dtm[4170:5559, ]
> sms_train_labels <- sms_df[1:4169, ]$type
> sms_test_labels  <- sms_df[4170:5559, ]$type
```

Verificando se a proporção de spam e ham está similar nas duas bases

Console

```
> prop.table(table(sms_train_labels))
sms_train_labels
      ham      spam
0.8647158 0.1352842
> prop.table(table(sms_test_labels))
sms_test_labels
      ham      spam
0.8683453 0.1316547
```

Eliminar palavras que aparecem poucas vezes

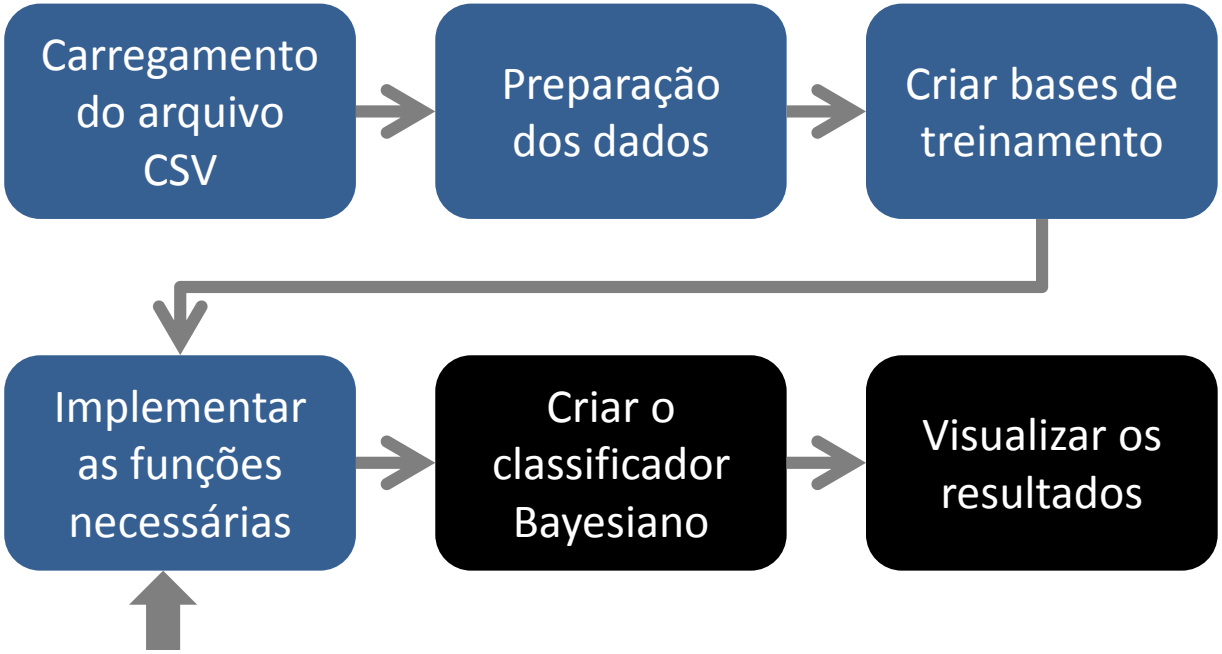
Console

```
> sms_dtm_freq_train <- removeSparseTerms(sms_dtm_train, 0.999)

> sms_freq_words <- findFreqTerms(sms_dtm_train, 5)

> str(sms_freq_words)
 chr [1:1139] "â,â€œ" "abiola" "abl" "abt" "accept" "access" "account" ...
```

Atualizar as matrizes

Console

```
> sms_dtm_freq_train <- sms_dtm_train[ , sms_freq_words]

> sms_dtm_freq_test <- sms_dtm_test[ , sms_freq_words]
```


Carregamento do arquivo CSV
Preparação dos dados
Criar bases de treinamento
Implementar as funções necessárias
Criar o classificador Bayesiano
Visualizar os resultados

Criar função para converter a matriz termo-documento em valores booleanos

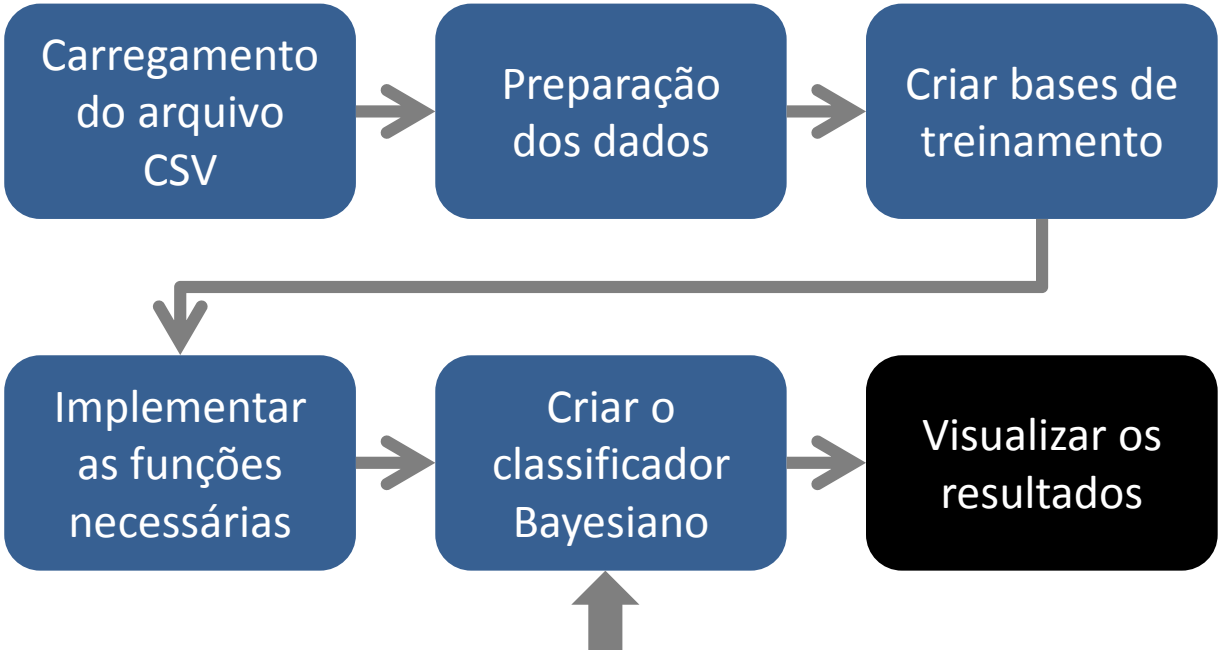
Console

```
> convert_counts <- function(x) {;
  x <- ifelse(x > 0, "Yes", "No");
}
```

Atualizar a matriz com a aplicação da função

Console

```
> sms_train <- apply(sms_dtm_freq_train, MARGIN = 2, convert_counts)

> sms_test  <- apply(sms_dtm_freq_test, MARGIN = 2, convert_counts)
```


Carregamento do arquivo CSV
Preparação dos dados
Criar bases de treinamento
Implementar as funções necessárias
Criar o classificador Bayesiano
Visualizar os resultados

Criação de um classificador Naive Bayes

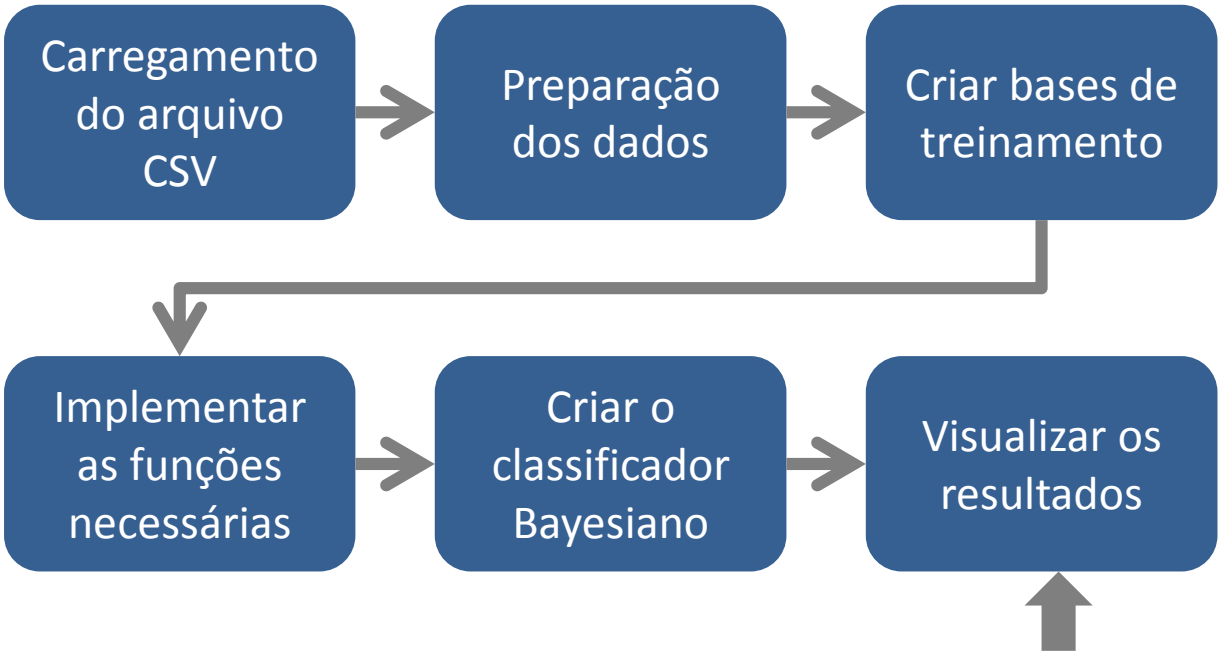
Console

```
> sms_classifier <- naiveBayes(sms_train, sms_train_labels)

> class(sms_classifier)
[1] "naiveBayes"
```

Execução do algoritmo

Console

```
> sms_test_pred <- predict(sms_classifier, sms_test)
```


Carregamento do arquivo CSV
Preparação dos dados
Criar bases de treinamento
Implementar as funções necessárias
Criar o classificador Bayesiano
Visualizar os resultados

Análise dos resultados

Console

```
> CrossTable(sms_test_pred, sms_test_labels,
prop.chisq = FALSE, prop.t = FALSE, prop.r = FALSE, dnn
= c('predicao', 'atual'))
```

```
Total Observations in Table:  1390


             | atual
    predicao |       ham |      spam | Row Total |
-------------|-----------|-----------|-----------|
         ham |      1201 |        30 |      1231 |
             |     0.995 |     0.164 |           |
-------------|-----------|-----------|-----------|
        spam |         6 |       153 |       159 |
             |     0.005 |     0.836 |           |
-------------|-----------|-----------|-----------|
Column Total |      1207 |       183 |      1390 |
             |     0.868 |     0.132 |           |
-------------|-----------|-----------|-----------|
```

Não foi isso que quis dizer quando falei que você precisava limpar os dados!!

www.iwaysoftware.com/go/dataquality

DÚVIDAS?

Perguntas

rpereira@larc.usp.br

Referências

- WHITE, Tom. Hadoop: The definitive guide. " O'Reilly Media, Inc.", 2014.
- OWENS, Jonathan R.; FEMIANO, Brian; LENTZ, Jon. **Hadoop Real World Solutions Cookbook**. Packt Publishing Ltd, 2013.
- ANIL, Robin; DUNNING, Ted; FRIEDMAN, Ellen. **Mahout in action**. Shelter Island: Manning, 2011.
- WITHANAWASAM, Jayani. **Apache Mahout Essentials**. Packt Publishing Ltd, 2015.
- WITTEN, Ian H.; FRANK, Eibe. **Data Mining: Practical machine learning tools and techniques**. Morgan Kaufmann, 2005.
- WU, Xindong et al. Top 10 algorithms in data mining. **Knowledge and Information Systems**, v. 14, n. 1, p. 1-37, 2008.