

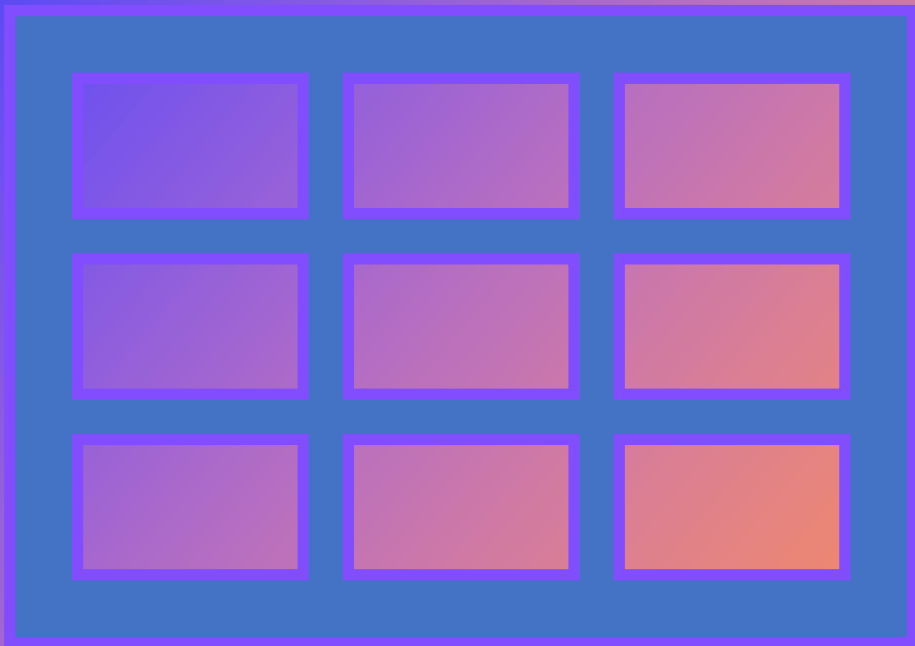
+
○

DESENVOLVIMENTO DE FILTRO ANTI- SPAM



Objetivo do Trabalho

- Classificar a coluna SMS do dataset `validation_data.csv` como “ok” ou “blocked”.



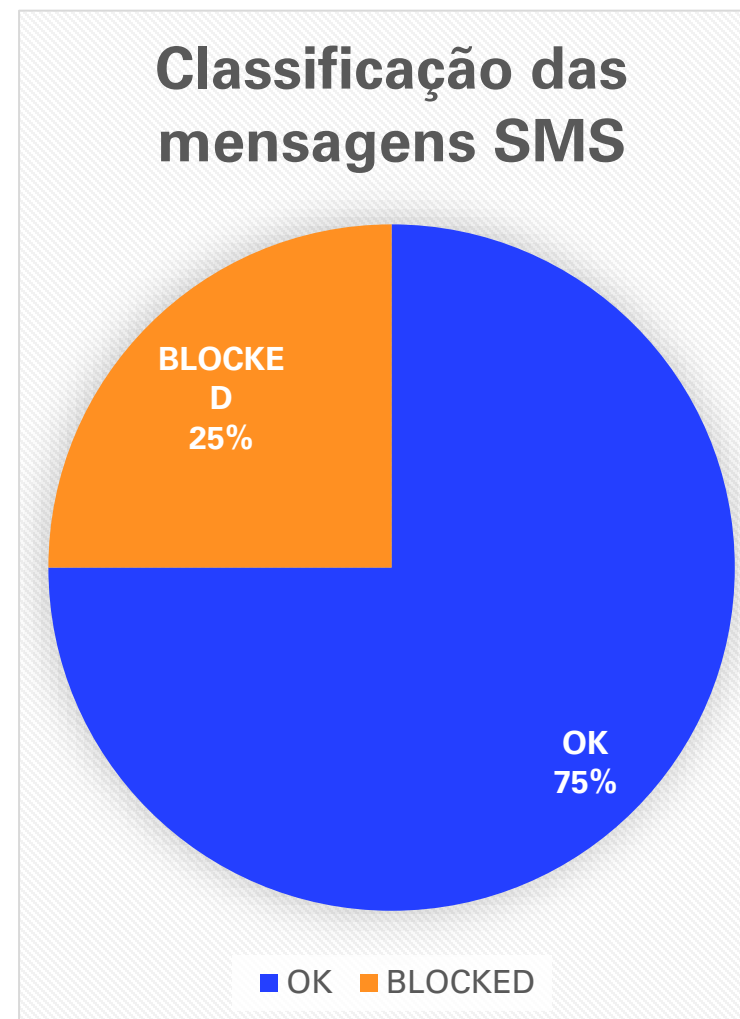
Contextualização do Problema

SMS	LABEL
ISO DO CONSÓRCIO PELO BB.COM VENCIMENTO PARA HOJE Ñ PODE HAVER QUEBRA NO ACORDO. BONATTO ADV 0800 606 3301.	blocked
050003DA0202 lcloud-apple-inc.com/?iphone=VtBqROY .	blocked
060804CB5F0303 ps: //bancodobrasil.seguranca.gq	blocked
ao ainda nao foi executado em sua conta. Evite bloqueios, atualize agora! Acesse: www.avisobbsms.com	blocked
0608042BE40303 ps://bancodobrasil.seguranca.cf/	blocked
...	...
Host : RB_Bicanga Ip: 170.244.231.14 nao esta respondendo ao ping - 2019-04-19 22:30:23	ok
Host : RB\$Bicanga Ip: 170.244.231.14 nao esta respondendo ao ping - 2019-07-05 05:05:17	ok
Host : RB\$Bicanga Ip: 170.244.231.14 nao esta respondendo ao ping - 2019-07-15 17:35:17	ok
Host : RB_Bicanga Ip: 170.244.231.14 nao esta respondendo ao ping - 2019-08-02 02:20:20	ok
050003730201 Faça login no aplicativo OlympTrad?: 170.244.24.150, BR, David Cana	ok

- Diariamente, recebemos mensagens SMS em nossos telefones. Com a proliferação de bancos de dados empresariais, vazamentos e vendas de banco de dados, nosso número de telefone pode frequentemente acabar em posse de empresas ou pessoas as quais nunca autorizamos, e assim, passamos a receber mensagens indesejadas de diversos tipos: desde aquelas que realizam publicidade até as mais danosas, que trazem links maliciosos para nossos aparelhos.
- Partindo de 2 datasets que contém apenas o conteúdo de mensagens SMS e sua classificação (bloqueadas ou não bloqueadas), o objetivo deste trabalho é criar um algoritmo de classificação automático utilizando técnicas de processamento de linguagem natural para barrar o máximo possível de mensagens indesejadas com base exclusivamente no conteúdo das mensagens.

Análise Exploratória de Dados – train_data

- 6000 observações
- 2 variáveis: SMS (texto da mensagem) e LABEL (classificação em OK ou BLOCKED)
- Há 1500 mensagens classificadas como BLOCKED
 - Aparentemente, nem todo BLOCKED possui link
- Há 4500 mensagens classificadas como OK
- Não há valores faltantes
- Há 5859 mensagens únicas e 141 mensagens repetidas
 - Há mensagens repetidas classificadas tanto como BLOCKED como OK.




Análise Exploratória de Dados – train_data

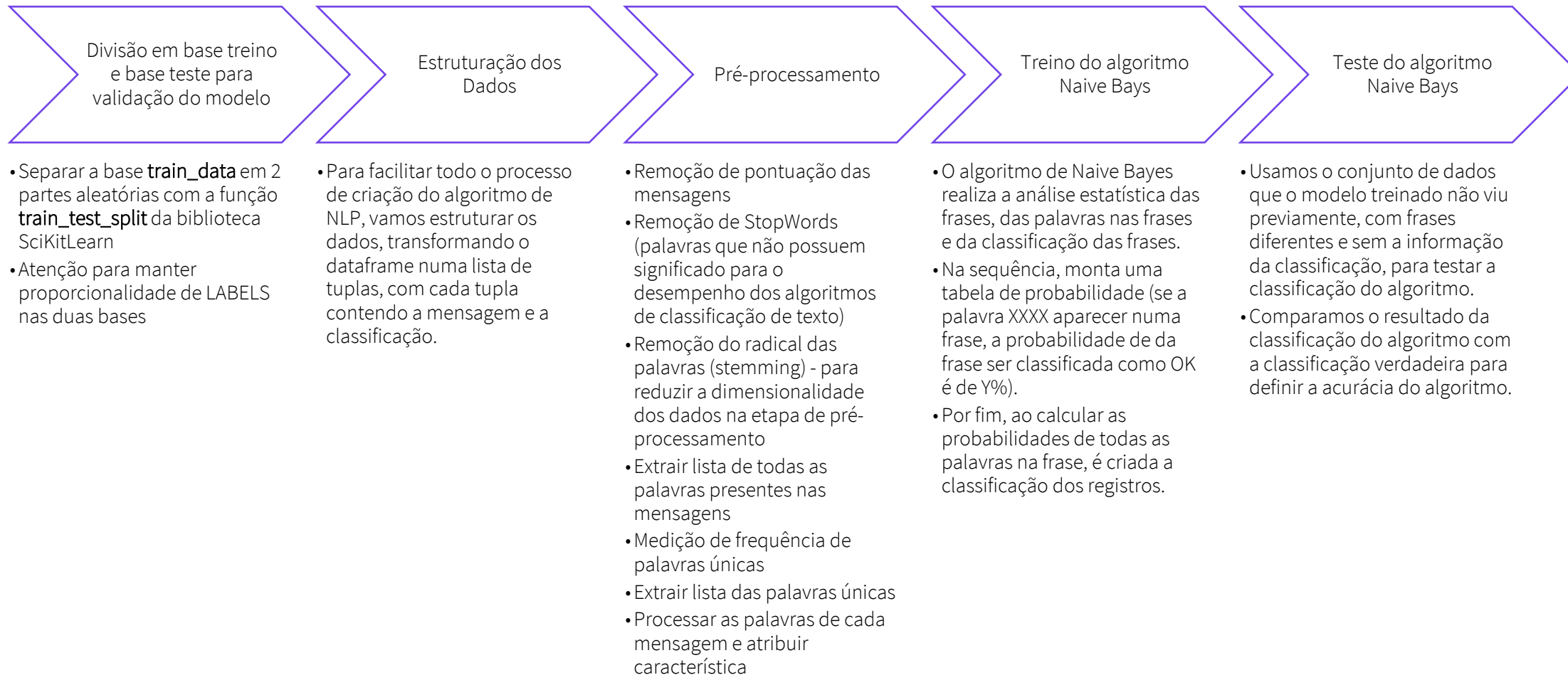
- Hipótese: a classificação é dada de acordo com a presença de URLs no SMS. Vamos tentar verificar a presença de URLs nas amostras de SMS com Status = OK e Status = Blocked. Como na visualização dos primeiros 20 e dos últimos 20 de cada grupo foi possível perceber que há URLs de diferentes formatos (com e sem "http://", com e sem "https://", com e sem ".com" etc.), vamos nos ater à sintaxe básica: caractere alfanumérico - . - caractere alfanumérico para identificar a maioria das URLs.
- Analisando as supostas URLs dos SMS com status = OK, percebemos que a maioria é numérica, o que pode indicar excertos de CPFs (quando o segundo elemento é composto de 3 dígitos) ou endereços tipo IP. Também verificamos algumas URLs verdadeiras, como bitnuvem.com, gmail.com, cemporcentoskate.uol.com.br, ocean.overseaweb.com.br e outros. Portanto, a simples presença de URL no SMS não classifica a mensagem automaticamente como status = blocked (ou há uma lista branca de URLs).
- Analisando as supostas URLs dos SMS com status = BLOCKED, percebemos que a maioria é de URLs verdadeiras, como www.Bbrasildesbloqueio, tinyurl.com, www.buscaiphone.com, app.biz, bitly.com, fotosms.xyz, santander.com e outros. A alta frequência de algumas URLs (ex.: www.Bbrasildesbloqueio) sugere que essas mensagens foram bloqueadas ativamente, gerando uma lista negra.



Modelagem

- Levando em conta o tipo de dados presentes nos datasets, a recomendação para desenvolver o algoritmo de classificação é utilizar técnicas de processamento de linguagem natural (NLP) como primeira abordagem e verificar os resultados obtidos: se a acurácia for satisfatória, empregamos o algoritmo; se não, buscamos outra abordagem.
 - Para isso, vamos utilizar a biblioteca **Natural Language Toolkit (NLTK)** para Python.
- 

Modelagem – Metodologia Aplicada



Modelagem - Resultados

- Acurácia da classificação com a base teste: 98,42%
 - **1. Análise de cenário:** acerto do algoritmo é muito bom, mas corre o risco de overfitting (o modelo aprendeu muito bem como classificar essa base, mas pode se comportar mal com uma base desconhecida).
 - **2. Análise do número de classes:** a probabilidade mínima aceitável para o algoritmo ser melhor do que usar a aleatoriedade é que a acurácia seja maior que 50%, ou seja, dividir 100% pela quantidade de classes.
 - **3. ZeroRules:** nessa análise, estamos comparando o resultado obtido pelo sistema, com o método de classificar uma frase de acordo com a classe que possui maior quantidade de frases na base de dados de treino e teste. Por exemplo, dividimos a classe com maior número de registros pelo total de registros na base de dados ($887/1200 = 73,92\%$). Desta forma, conclui-se que o sistema apresenta mais acertos do que classificar todas as novas frases nessa classe.

- **Matriz de Confusão:**

	BLOCKED (classe do algoritmo)	OK (classe do algoritmo)
BLOCKED (classe original)	301	12
OK (classe original)	7	880

Precisão: de todos os BLOCKED classificados pelo algoritmo, quantos o algoritmo acertou.

$$\text{Precisão} = 301 / (301 + 7) = 97,7\%$$

Recall: de todos os BLOCKED, quantos o algoritmo acertou.

$$\text{Recall} = 301 / (301 + 12) = 96,2\%$$

Modelagem - Resultados

10 atributos mais significativos para classificar uma frase:

Most Informative Features

bb = True
fot = True
br = True
j = True
debit = True
prezado = True
local = True
login = True
aplic = True
lig = True

blocke : ok	=	1778.2 : 1.0
blocke : ok	=	396.8 : 1.0
ok : blocke	=	314.7 : 1.0
blocke : ok	=	266.9 : 1.0
ok : blocke	=	208.2 : 1.0
blocke : ok	=	192.9 : 1.0
blocke : ok	=	166.9 : 1.0
ok : blocke	=	121.9 : 1.0
ok : blocke	=	104.2 : 1.0
ok : blocke	=	100.0 : 1.0

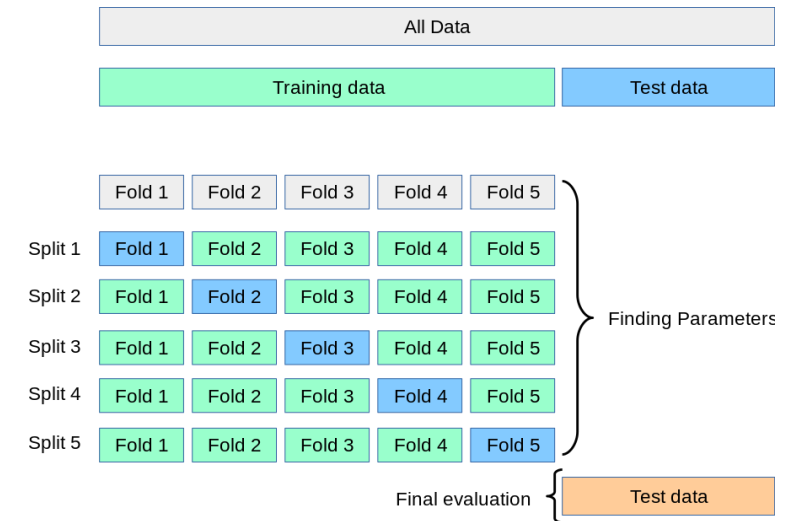
Exemplo:

bb = True blocke : ok =
1778.2:1.0

- Neste exemplo de saída a probabilidade de a frase ser classificada como **blocked** quando a palavra “bb” estiver presente na frase (True) é 1778 vezes maior do que ok.

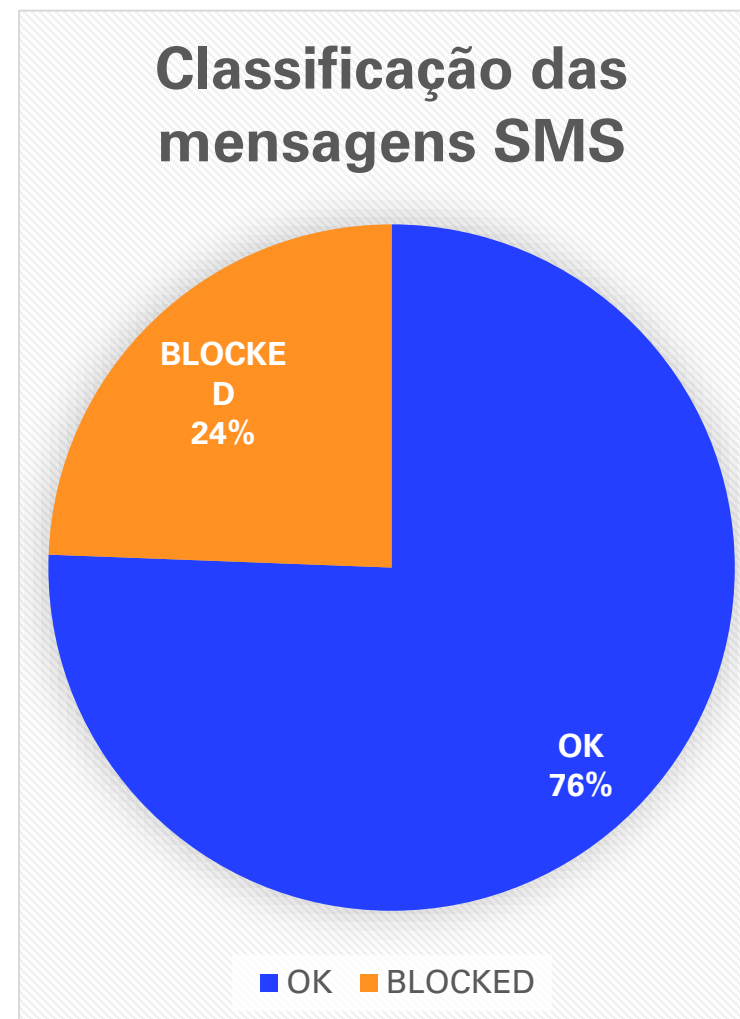
Modelagem - Resultados

- Como a acurácia foi bastante alta, precisamos revalidar o algoritmo de treino com validação cruzada para garantir que não há um viés específico da base de treino devido a separação aleatória realizada pela função `train_test_split`.
- Utilizamos validação cruzada tipo K-Fold, com 10 subconjuntos de dados.
- A acurácia média do teste com validação cruzada K-Fold ($k=10$) foi de **89,35%**, o que significa que não temos problemas de overfitting e podemos agora realizar o treino com a base completa.



Classificação da base validation_data

- Com o modelo treinado, realizamos a classificação das mensagens da base `validation_data`.
- Das 2000 mensagens não classificadas, o algoritmo classificou 488 como bloqueadas, com probabilidade média de 99% de serem bloqueadas.
 - A mensagem bloqueada com menor probabilidade foi de 61,5%.
- Das 2000 mensagens não classificadas, o algoritmo classificou 1512 como OK, com probabilidade média de 99,35% de serem OK.
 - A mensagem não bloqueada com menor probabilidade foi de 50%.





Conclusões

- O algoritmo de **Naive Bays** da biblioteca NLTK **é indicado para a solução do problema** (filtro anti-SPAM), pois processa a linguagem utilizada nas mensagens e indica a probabilidade da mensagem ser classificada como OK ou BLOCKED de acordo com seu conteúdo.
- A **acurácia geral** do algoritmo treinado com a base train_data e validação cruzada com 10 subconjuntos de dados foi de **89,35%**, o que indica que o algoritmo **não está sofrendo de overfitting** e está **acertando mais** do que se classificássemos as mensagens aleatoriamente ou se classificássemos todas as mensagens como OK.
- Para melhorar o algoritmo, recomenda-se monitorar as classificações erradas, levantar quais são as palavras mais frequentes nessas classificações e avaliar a inclusão de todas ou algumas dessas palavras como StopWords, para que sejam ignoradas pelo algoritmo.
- A acurácia do algoritmo deve ser acompanhada periodicamente. Quando a acurácia cair, recomenda-se retreinar o modelo.