Desafio-Movile

November 11, 2018

1 Desafio Movile

Neste desafio, estudamos o problema de classificação de spams em mensagens de SMS.
Foram fornecidas pela Wavy amostras de mensagens de diferentes operadoras.

```
In [12]: import numpy as np
         import pandas as pd
         from sklearn.model_selection import train_test_split
         pd.set_option('display.max_colwidth', -1)
```

```
In [2]: dataset = pd.read_csv('SPAM Data _ Akari - SPAM Data.csv')
```

```
In [20]: print(len(dataset))
         dataset.sample(frac=1).head(10)
```

```
997
```

```
Out[20]:        vendor  \
         619  difize
         950  mailserr
         118  zootude
         850  difize
         896  centigen
         279  centigen
         864  centigen
         225  centigen
         112  centigen
         595  difize


         619  Oi FRANCISCO! Abasteca e pontue com seu cartao fidelidade na REDE CAXUXA e ganhe c
         950  Traga seu numero para CLARO ganhe mais internet para navegar, Whastapp a vontade,
         118  Carlos, a CLARO ainda precisa de sua ligacao! Retorne ate o final do dia de hoje n
         850  REDE GMAXX: Oi JOSE! Preparamos uma oferta para voce! Abasteca na REDE GMAXX e gan
         896  Sebastiana,mantenha seu plano da TIM em dia! Efetue o pagamento da divida.Evite a
         279  Franciele,mantenha seu plano da TIM em dia! Efetue o pagamento da divida.Evite a s
         864  Renato,mantenha seu plano da TIM em dia! Efetue o pagamento da divida.Evite a sua
```

```
225  Eliene,mantenha seu plano da TIM em dia! Efetue o pagamento da divida.Evite a sua
112  Caio,mantenha seu plano da TIM em dia! Efetue o pagamento da divida.Evite a sua pe
595  Oi EMERSON! Abasteca e pontue com seu cartao fidelidade na REDE CAXUXA e ganhe o D

    destino   spam  total
619  NaN     False  239
950  CLARO   False  624
118  NaN     False  133
850  NaN     False  233
896  NaN     False  104
279  NaN     False  163
864  NaN     False  608
225  NaN     False  128
112  NaN     False  215
595  NaN     False  103
```

Na amostra acima, podemos ver que temos 5 features: * Vendor: Nome da empresa que enviou a mensagem * Mensagem: texto da mensagem * Destino: Operadora do destinatário da mensagem * Spam: se foi classificada como spam (true) ou não (false) * Total: Quantidade de cópias da mensagem enviadas

1.1 Abordagens

Existem inúmeras formas de explorar o problema. Primeiro, devemos olhar bem as características dos nossos dados.

```
In [4]: dataset['spam'].value_counts()

Out[4]: False    991
        True       6
        Name: spam, dtype: int64
```

Aqui temos um claro problema de *skewed classes*, em que uma classe (negativa) é muito mais predominante do que a outra (positiva). Casos assim não são triviais de serem solucionados. Vamos então primeiramente explorar os dados e analisar uma possível solução.

Podemos começar vendo o que temos nesses SMSs classificados como spam

```
In [14]: dataset.loc[dataset['spam'] == True]

Out[14]:        vendor  \
         772  mailserr
         783  quasiyo
         785  quasiyo
         786  quasiyo
         787  quasiyo
         788  quasiyo


         772  OI. Temos uma OFERTA especial de CELULAR ILIMITADO para todo BRASIL + 10GB de INTE
```

```
783  Ola, somos da TIM! Parabens! Seu chip esta ativado, realize uma ligacao de 30 segu
785  Ola, somos da TIM! Seu chip foi ativado e liberado para fazer ligacao. Utilize com
786  Ola, somos da TIM! Seu chip foi ativado e liberado para fazer ligacao. Utilize com
787  Ola, somos da TIM! Seu chip ja foi ativado e esta gerando fatura. Digite 1 se ja r
788  Ola, somos da TIM! Verificamos que voce ainda nao utilizou seu chip, e estamos ger

    destino  spam  total
772  NaN     True  2276
783  NaN     True  495
785  NaN     True  1041
786  NaN     True  458
787  NaN     True  283
788  NaN     True  193
```

A primeira coisa que nos chamou a atenção foi a grande semelhança entre a classe positiva e negativa. Notei apenas erros sutis de ortografia ou gramática. Além disso, 5 dos 6 SMS classificados como spams foram enviados pela "quasiyo". Vamos verificar se todos os SMS enviados pela quasiyo são spams.

```
In [21]: dataset.loc[dataset['vendor'] == 'quasiyo'].head(5)
```

```
Out[21]:       vendor  \
         238  quasiyo
         239  quasiyo
         240  quasiyo
         241  quasiyo
         242  quasiyo


         238  Esta difIcil lidar com as taxas de emprestimos? CLARO Q NAO! So RedeCifrao lhe apr
         239  Esta difIcil lidar com as taxas de emprestimos? CLARO Q NAO! So UBLA lhe apresenta
         240  Esta dificil lidar com as taxas de emprestimos? CLARO QUE NAO! So a REDE CIFRAO lh
         241  Esta dificil lidar com as taxas de emprestimos? CLARO QUE NAO! So a REDE CIFRAO lh
         242  Esta dificil lidar com as taxas de emprestimos? CLARO QUE NAO! So a RedeCifrao lhe

             destino   spam  total
         238  NaN     False  781
         239  NaN     False  1128
         240  NaN     False  1923
         241  NaN     False  186
         242  NaN     False  1067
```

Confirmamos que vários outros SMSs enviados pela quasiyo não foram classificados como spam. Será que existe algum padrão nas mensagens spams?

```
In [23]: dataset[dataset['mensagem'].str.contains("Ola, somos da TIM!")]
```

```
Out[23]:       vendor  \
         783  quasiyo
```

```
785  quasiyo
786  quasiyo
787  quasiyo
788  quasiyo


783  Ola, somos da TIM! Parabens! Seu chip esta ativado, realize uma ligacao de 30 segu
785  Ola, somos da TIM! Seu chip foi ativado e liberado para fazer ligacao. Utilize com
786  Ola, somos da TIM! Seu chip foi ativado e liberado para fazer ligacao. Utilize com
787  Ola, somos da TIM! Seu chip ja foi ativado e esta gerando fatura. Digite 1 se ja r
788  Ola, somos da TIM! Verificamos que voce ainda nao utilizou seu chip, e estamos ger

    destino  spam  total
783  NaN     True  495
785  NaN     True  1041
786  NaN     True  458
787  NaN     True  283
788  NaN     True  193
```

Como apenas mensagens de spams tem o texto "Ola, somos da TIM!", uma solução seria classificar todas as mensagens com esse texto como spam. Mas claramente não seria um bom classificador.

Com estes testes, concluímos que as mensagens spams e não spams são muito semelhantes. É muito complicado criar um modelo de aprendizado de máquina para fazer uma tarefa que nem mesmo nós, humanos, seríamos capazes de fazer.

O único padrão que pudemos encontrar nos spams foi erros de ortografia e gramática. Provavelmente as mensagens foram classificadas como spams por este motivo. Outras features, como o total de cópias da mensagem, também não apresentaram nenhum padrão. Poderíamos propor filtrar as mensagens através de um corretor ortográfico, mas aparentemente é um padrão nas mensagens de SMS não utilizar acentuação, o que nos geraria muitos falsos positivos.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Podemos primeiramente fazer uma regressão, observar os resultados e, se necessário, melhorar a solução. Para isso, precisamos separar o conjunto em treino, validação e teste.

```
In [14]: label = dataset['spam']
         data = dataset.drop('spam', axis=1)

         #Separar em treino e teste aleatoriamente
         X_train, X_test, y_train, y_test =  train_test_split(data, label, test_size=0.15, rando
```

```
        #Separar em treino e validação
        X_train, X_valid, y_train, y_valid =  train_test_split(X_train, y_train, test_size=0.3,
```

In [ ]: