# Detecção de SPAM no Twitter

**Cláudio Torres Júnior**[1], **Eloiza Rossetto dos Santos**[1]

[1]Departamento de Informática (Dinf) – Universidade Federal do Paraná (UFPR)
Caixa Postal 19.081 – 81531-980 – Curitiba – PR – Brasil

ctj17@inf.ufpr.br, ers17@inf.ufpr.br

## 1. Introdução

As redes sociais se tornaram inerentes ao ser humano. Ao pensarmos em fatos relevantes ou questões que estão sendo discutidas no momento, muitas vezes procuramos por palavras chaves na rede social que mais utilizamos para sanar a curiosidade, tendo como resultado pessoas que postaram algo relacionado ou portais de notícias comentando sobre o tema. Dentre as redes sociais mais populares atualmente e que facilitam a pesquisa de informações específicas, destaca-se o Twitter. Nele, é possível ficar por dentro de assuntos de diversas maneiras: hashtags, palavras chave, trending topics, dentre outros. Por ser um ambiente repleto de usuários (estima-se que a rede possui 396 milhões de usuários, sendo que 206 milhões a acessam diariamente [Dean ]), acaba sendo um alvo fácil de pessoas mal intencionadas. A maior parte dos posts, os tweets, são públicos (o que uma pessoa posta pode ser acessado por qualquer usuário da plataforma somente com a hashtag ou palavra chave do tweet), fazendo com que alguém possa se aproveitar disso, induzindo pessoas que estão acompanhando certa hashtag a clicarem em links que podem redirecionar a sites externos que realizam download de malware e de phishing, por exemplo [Chen et al. 2015]. Essa técnica utilizada para criar posts em massa com o intuito de atrair usuários para algo fora do escopo da plataforma é denominada spam.

Os usuários mal intencionados podem se aproveitar de posts que estão no trending topics da rede social (os assuntos do momento na plataforma, os mais comentados) para lançar a isca, induzindo cliques em links maliciosos e tentar fisgar novos seguidores ao curtir um tweet aleatório e/ou seguir o usuário que postou, com a finalidade de ter o ”seguir”retribuído, aumentando assim a sua área de contato, pois ao seguir alguém, os posts criados e compartilhados por essa pessoa ficarão em destaque e sempre irão aparecer na página de quem o seguiu. Com isso, o perfil de spam irá ter um alcance maior de pessoas, podendo ter seus posts compartilhados mais facilmente.

O próprio Twitter colocou em prática alguns detectores de spam para suspender contas com comportamento estranho, verificando a quantidade de tweets por medida de tempo, quantidade de usuários marcados no post e tweets que possuem somente o link de algum site sem mais nenhuma informação, como visto em [Chen et al. 2015]. Vários desses modelos levam em consideração características suspeitas dos usuários, como as citadas anteriormente para classificá-los como um spammer ou não. Mas isso pode não ser muito bem aproveitado, pois existem bots que conseguem burlar esse sistema, ao postar em intervalos de tempo maior, por exemplo e pessoas que ao invés de dar o retweet, apenas copiam o conteúdo e postam como sendo os autores. Para isso, é necessário um classificador que consiga extrair informação do próprio tweet, para que a classificação inicial seja feita a partir dele, para que, somente em seguida, passe a verificar o usuário e a rede de interação que esse tweet desencadeou.

**Tabela 1. Descrição e tipo dos atributos do dataset utilizado**

| Atributo | Descrição | Tipo |
|---|---|---|
| Tweet | Texto a mão livre do tweet coletado | Textual |
| Following | Total de contas seguidas pelo autor do tweet | Numérico |
| Followers | Total de contas que seguem o autor do tweet | Numérico |
| Actions | Total de favoritos, respostas e retweets do tweet | Numérico |
| $Is\_retweet$ | Valor binário que indica se o tweet é retweet | Numérico |
| Location | Texto a mão livre escrito pelo usuário | Textual |
| Type | Classe do tweet | Categórico |

## 2. Descrição dos Conjuntos de Dados

Para realizar o trabalho utilizamos o dataset da competição de detecção de SPAM no Twitter do grupo UTKML (University of Tennessee Machine Learning Student Organization) de 2018 [UTKML 2018]. Os dados estão disponíveis publicamente no site Kaggle.

O dataset é dividido em um conjunto de treinamento e um conjunto de teste. O conjunto de treinamento possui 14.899 amostras sendo 7.443 (49.96%) da classe SPAM e 7.454 (49.96%) da classe não SPAM. O conjunto de teste possui ao todo 785 amostras que não foram rotuladas. Segundo os organizadores foi considerado como SPAM tweets postados por contas ativas conhecidas por serem perfis falsos. Esses perfis possuem as características de serem politicamente motivados, postarem *click baits*, gerarem conteúdo automaticamente e sem sentido.

As amostras coletadas possuem sete atributos que são do tipo numérico ou textual. Dentre os atributos três são características relacionadas a conta que postou o tweet e o restante relacionado ao próprio tweet. A Tabela 1 apresenta os atributos do conjunto de dados bem como a descrição e o tipo. A partir do atributo *Tweet* foi possível extrair novas características numéricas baseadas na referência de [Benevenuto et al. 2010]. A Tabela 2 apresenta as características geradas a partir da coluna *Tweet*.

**Tabela 2. Características geradas a partir do texto do *tweet***

| Atributo | Descrição | Tipo |
|---|---|---|
| Words | Total de palavras no texto do tweet | Numérico |
| Hashtags | Total de hashtags no texto tweet | Numérico |
| Hashtag Ratio | Razão entre o número de hashtags pelo total de palavras | Numérico |
| URLs | Total de URLs no texto do tweet | Numérico |
| Numbers | Total de caracteres numéricos | Numérico |
| Mentions | Total de menções feitas no texto do tweet | Numérico |

Ao final do processo de extração de características iniciamos a análise e limpeza dos dados restantes. Nessa fase optamos por remover o campo de localização. Esse atributo muitas vezes tinha sido preenchido com localizações fictícias ou com caracteres especiais como emojis e outros símbolos. Durante essa análise preliminar encontramos duas amostras pertencentes a classe ”South Dakota”, classe não descrita no problema.

Provavelmente houve um erro durante a leitura do arquivo *csv* ou durante a escrita. De qualquer forma, optamos por remover essas duas amostras do conjunto de dados. Havia também uma coluna de característica chamada ”Unamed Column: 7”que provavelmente resultou de um erro durante a leitura do arquivo. Essa coluna extra também foi removida do conjunto de dados.

Ao analisar os atributos numéricos originais foram encontradas entradas do tipo Nan. Para o atributo actions fora encontrados 3436 Nans, para following 158, para followers 17 e $is\_retweet$ apenas 1. Os atributos restantes não possuem Nans. Para lidar com os dados faltantes, substituímos os valores pela mediana da coluna. A escolha da mediana se deu pelo desvio padrão dos atributos ser grande assim impossibilitando o uso da média.

Ao final de todo o processo de limpeza e extração de características obtivemos um dataset com dez características. Sendo todas as características apresentadas nas Tabelas 1 e 2 menos os atributos *location*, *tweet* e *type*. Além disso o dataset final conta com 14.887 amostras sendo a mesma distribuição entre as classes. Isso se deve por termos removido duas amostras de uma terceira classe.

Com isso, temos a seguinte disposição dos dados de treino:

| | spam | non-spam |
|---|---|---|
| **Tweets identificados** | 7443 (49.96%) | 7454 (50.04%) |
| **Tweets únicos (sem repetição)** | 7424 | 7256 |
| **Tweets únicos/identificados** | 99.74% | 97.34% |
| **Total de urls** | 4554 | 3904 |
| **urls/identificados** | 61.19% | 52.37% |
| **mais de 1 url** | 1581 | 616 |
| **mais de 1 url/total de urls** | 34.72% | 15.78% |

**Figura 1.**

Iremos discutir alguns gráficos que representam cada característica da figura 1 de forma mais detalhada, além de outras que também pareceram promissoras para a detecção de spam. A seguir temos dois gráficos representando a divisão dos *tweets* em spam e não spam. No da esquerda vemos todos os *tweets*, enquanto o da direita representa somente os *tweets* únicos (sem repetição de texto). Esse segundo foi realizado com a verificação completa do corpo do *tweet*, portanto ele não leva em consideração semelhanças de textos e nem diferenças sutis como a mudança de somente uma *hashtag* ou menção (para o gráfico em questão, tudo isso é semelhante). Percebe-se que há um certo equilíbrio no *dataset* de treino, indicando que a grande maioria dos *tweets* são diferentes entre si.

Uma das características de um spam, é fazer com que as pessoas comecem a com-

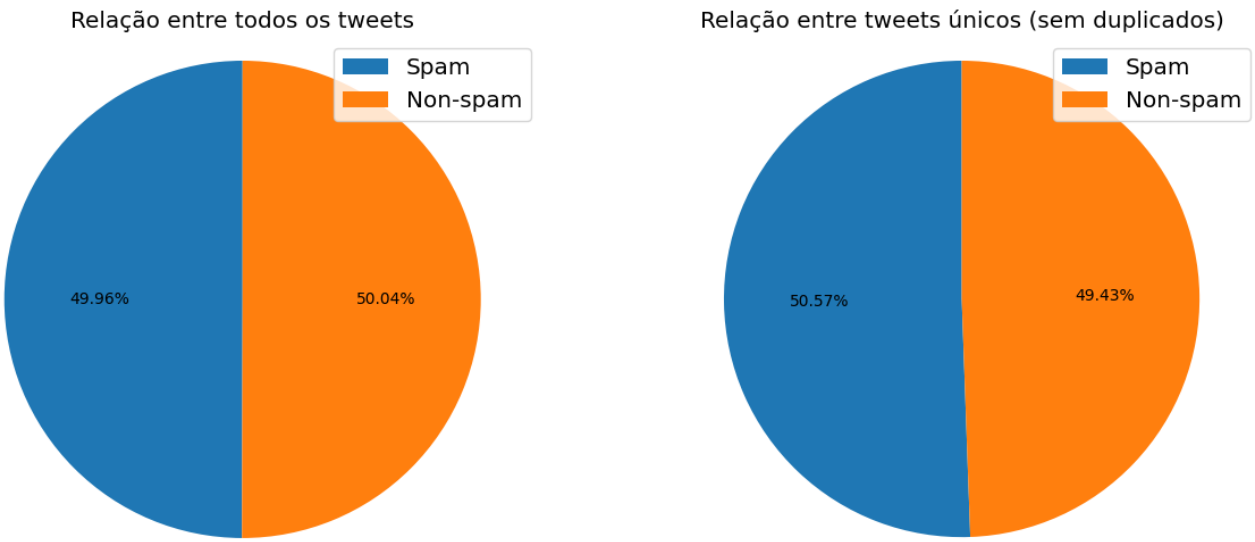


**Figura 2.**

partilhá-lo para que fique mais visível e receba ainda mais compartilhamentos. No Twitter, é possível compartilhar (retweetar) um *tweet* de duas maneiras: O texto completamente igual ou escrever uma publicação tendo o *tweet* que será retweetado no corpo da mensagem. Os gráficos a seguir mostram a relação de *retweets* entre os dois grupos que temos:

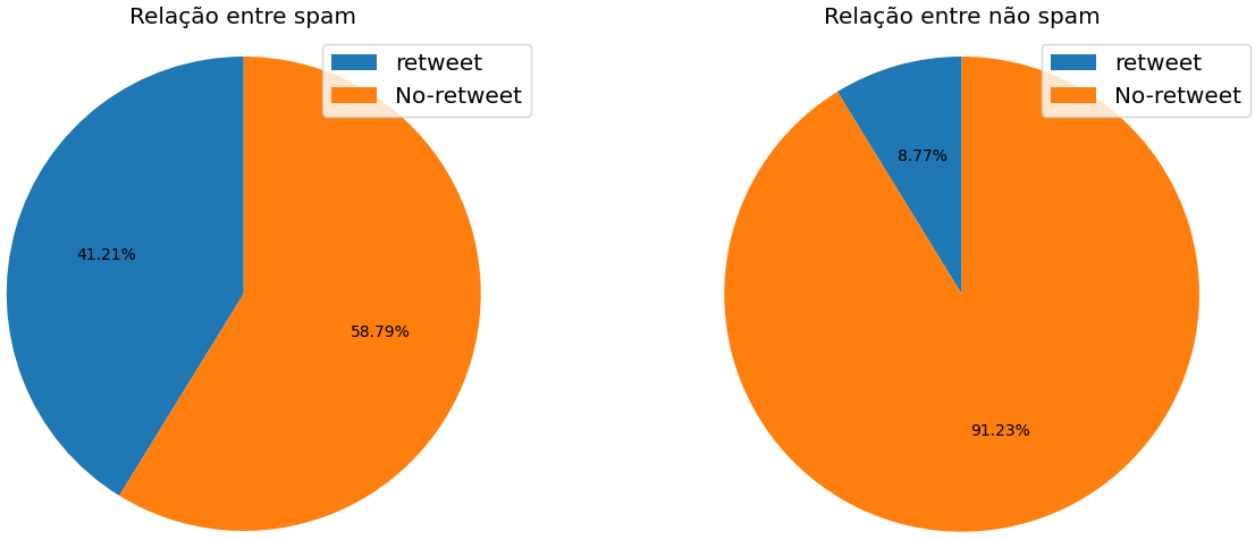


**Figura 3.**

É possível ver que, mesmo sendo um *dataset* equilibrado em relação a proporção de spam e não spam, o primeiro grupo possui uma taxa de *retweet* muito maior que o segundo. Como nosso modelo identifica *tweets* parecidos como sendo um texto completamente idêntico, a figura 2 nos permite inferir que o tipo predominante de *retweet* existente é o segundo que foi explicado anteriormente.

Uma característica que não foi calculada inicialmente na tabela da figura 1, foi a quantidade de *hashtags*. O gráfico da Figura 4 mostra que a grande maioria dos spams possuem um alto número de *hashtags* em seu texto, sendo maior que os *tweets* "normais". Vários artigos e sites utilizados como base para entendermos como os spams se compor-

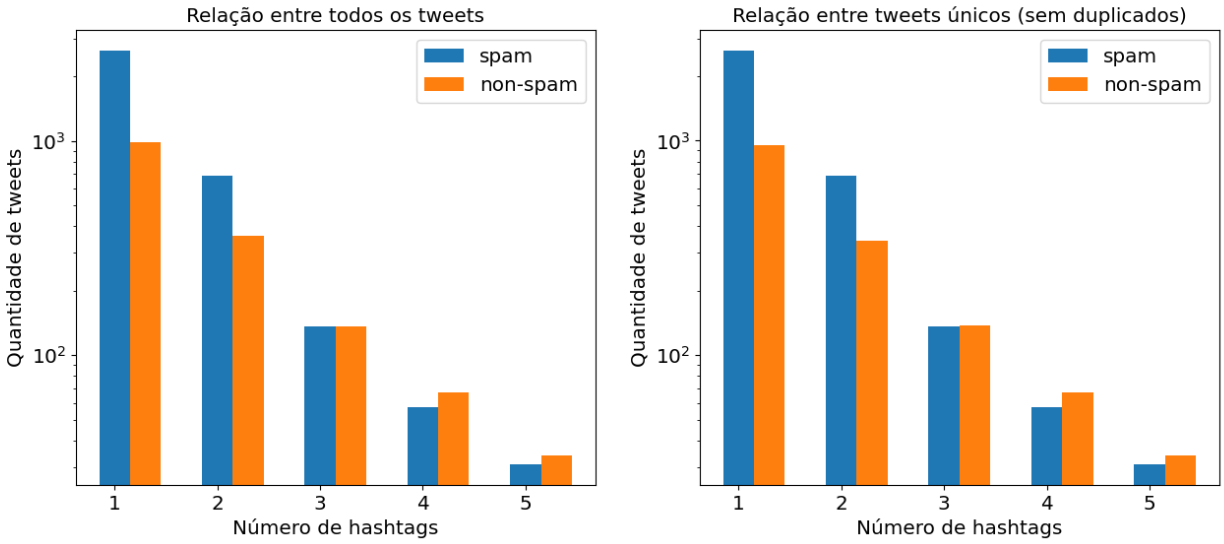


**Figura 4.**

tam mostraram que esses *tweets* costumam ter bem mais que uma *hashtgag* em seu texto para conseguir atingir diversas pessoas a partir de temas diferentes. Mas esse gráfico mostrou o contrário. Apesar de ainda ter bastante *hashtgag* nos *tweets* classificados como spam, ao elevar o número desses elementos, a proporção em relação aos *tweets* normais vai diminuindo (e até se invertem em dado ponto). Isso indica que a distribuição das classes do *dataset* em questão quebra a ideia sobre a quantidade de *hashtags* (lembrando que somente para o *dataset* em questão. Não é possível afirmar isso em relação ao universo real dos *tweets*).

Agora ao ver a distribuição dessas *hashtags*, percebemos que a distribuição delas começam a ser desproporcionais. A Figura 5 mostra as Top 5 *hashtags* entre spam e não spam. Percebemos que o volume pode até ser parecido, mas a quantidade de *hashtags* diferentes acaba sendo completamente desproporcional.

Além dessa métrica, foi verificado também o número de menções (Figura 6). Como a característica explicada anteriormente, quanto mais menções, mais propenso de ser *retweetado* e clicado por aqueles que estão mencionados (ou integrantes de suas redes). Mais uma vez, olhando essa característica sozinha, não parece ser um bom indicador de spam, pois os valores se equilibram conforme o número de menções vai aumentando. O que pode variar é a distribuição de menções diferentes (assim como aconteceu com as

| | Hashtags | Quantidade | Hashtags | Quantidade |
|---|---|---|---|---|
| 0 | #news | 475 | #HAPPYBAEBAEDAYpic | 21 |
| 1 | #sports | 196 | #HAPPYTAEYANGDAY | 17 |
| 2 | #politics | 150 | #WhyMediaHidesFacts | 11 |
| 3 | #local | 100 | #aqabiology | 11 |
| 4 | #world | 97 | #ALDUBLoversInITALYpic | 10 |

**Figura 5. Top 5 *hashtags* entre spam e não spam**

*hashtags).*

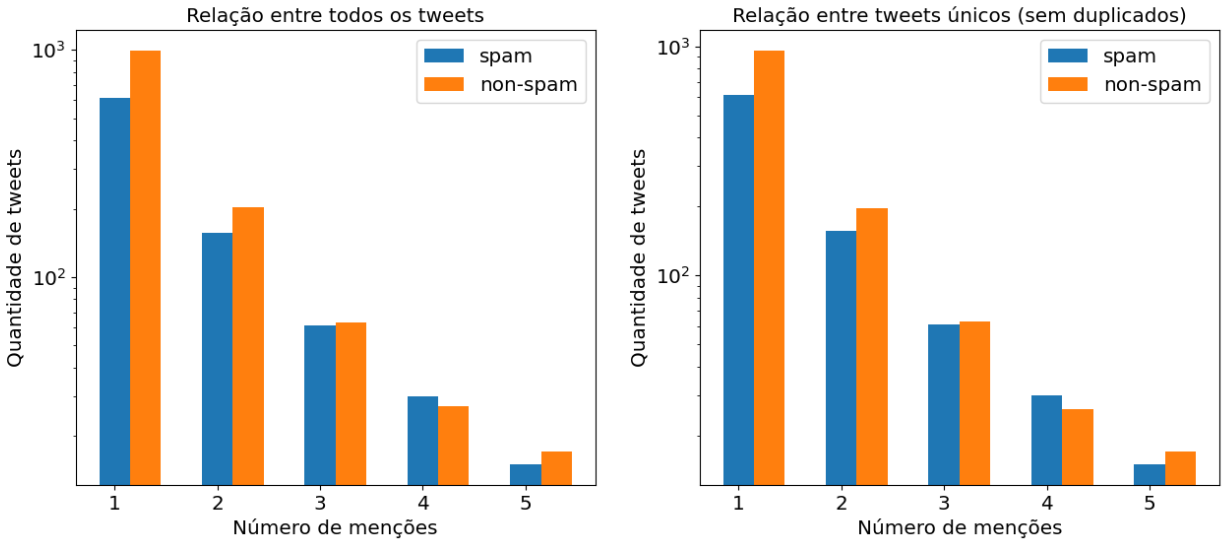


**Figura 6.**

Finalizando, verificamos a quantidade de URLs presentes nos *tweets*. Muitos spams pretendem que os usuários cliquem em anúncios e/ou links para que sejam redirecionados à sites de *phishing*, por exemplo. A ideia, portanto, é que o número desses elementos em spam sejam superiores aos não spam. A figura Mas como as outras características citadas anteriormente, o gráfico nos apresenta valores semelhantes. Estudando mais afundo o que acontece com as URLs nos *tweets*, descobrimos que, ao ser *retweetado* do segundo modo (criando uma nova publicação com o corpo do *retweet* inserido no novo texto), o corpo do *tweet* (exatamente no final) possui o link para o *tweet* original (isso ao extrair pela API do Twitter. Vendo a publicação real na rede social, iremos ver

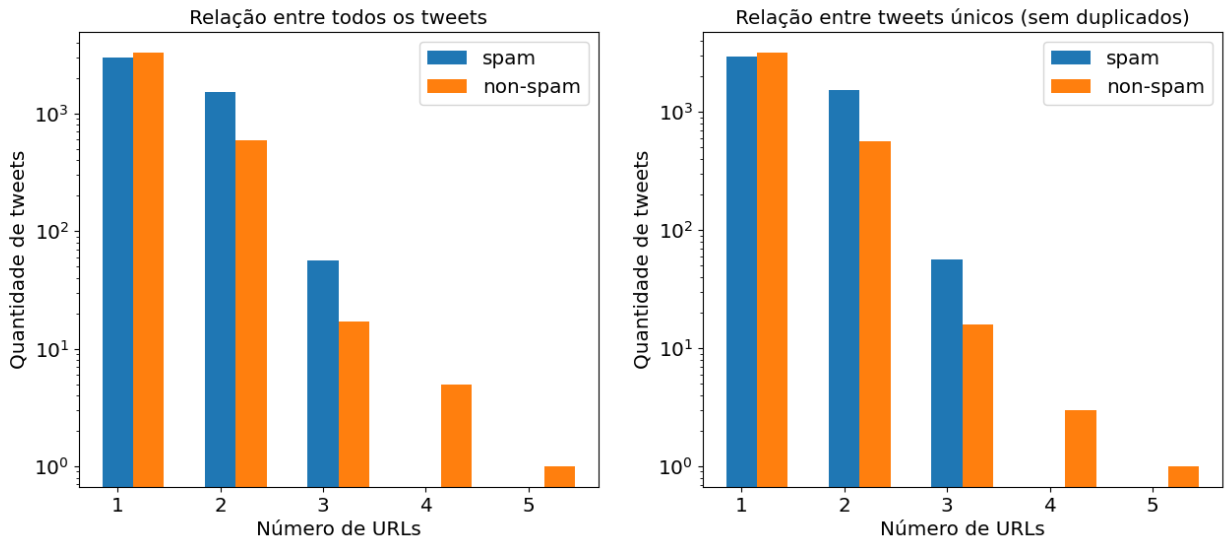


**Figura 7.**

uma representação do *tweet* original logo na sequência do texto do usuário, sem a URL explícita para ele). Portanto acreditamos que o número de URLs estão inflados por redirecionamentos à outros *tweets*. O que poderia ser feito é uma análise do domínio de cada link, verificando se o redirecionamento é um link externo ou para um outro *tweet*, para podermos eliminar aquelas de *retweet*. Resolvemos não seguir por essa abordagem pois acreditamos que essa informação em conjunto com as outras (especialmente com a *is_retweet*) consiga abstrair essa relação de URLs e os *retweets*. Na figura 8 vemos o Top

| | URLs | Quantidade | URLs | Quantidade |
|---|---|---|---|---|
| **0** | https://1063.mobi/ | 12 | https://yg-life.com/?lang=ko | 12 |
| **1** | https://twibble.io/ | 8 | http://HAPPYBAEBAEDAYpic.twitter.com/Lib0qxwsyr | 12 |
| **2** | https://twibble.io/ | 8 | http://mpic.twitter.com/C9lC69C4WD | 8 |
| **3** | https://covfefe.bz/ | 7 | https://twitter.com/skynews/status/73256156416... | 7 |
| **4** | https://twibble.io/ | 5 | http://www.gomplaces.com/ | 7 |

**Figura 8. Top 5 *URLs* entre spam e não spam**

5 das URLs. Percebe-se qe nas URLs de spam, o mesmo link aparece 3 vezes. O Twitter possui um encurtador de links e nosso modelo extrai o verdadeiro na hora de mostrar no *ranking*. Portanto URLs encurtadas com valores diferentes, podem redirecionar ao mesmo site.

## 3. Metodologia

Para cumprir o objetivo deste trabalho, identificar tweets SPAM, utilizamos o dataset rotulado disponibilizado pela UTKML para treinar os algoritmos: Random Forest, K-Nearest Neighbors (KNN) e Multi Layer Perceptron (MLP). Durante a preparação dos classificadores foram utilizadas técnicas para a seleção de parâmetros e técnicas de validação cruzada. As seguintes subseções tem como objetivo apresentar a arquitetura e os parâmetros escolhidos para a configuração dos algoritmos de classificação. Todos os classificadores se baseiam nos implementados na biblioteca sklearn.

### 3.1. Random Forest

Os parâmetros escolhidos para a busca foram: $min\_samples\_split$, $n\_estimators$, $max\_depth$ e $max\_features$. A Tabela 3 apresenta os valores testados para cada parâmetro e uma breve descrição de sua função. Ao final da execução do Grid Search os melhores valores para os parâmetros testados foram: $min\_samples\_split$: 3; $n\_estimators$: 100; $max\_depth$: 15; $max\_features$: 3; A partir destes valores de parâmetros encontrados pelo Grid Search utilizamos essa configuração durante os experimentos seguintes.

**Tabela 3. Valores utilizados no Grid Search para Random Forest**

| Parâmetro | Valores | Descrição |
|---|---|---|
| $min\_samples\_split$ | 3, 5, 10 | Número mínimo de amostras para considerar uma separação |
| $n\_estimators$ | 100, 300 | Total de árvores na floresta |
| $max\_depth$ | 15, 25 | Profundidade máxima de cada árvore |
| $max\_features$ | 3, 6, 11 | Características ao considerar a melhor separação |

### 3.2. K-Nearest Neighbors (KNN)

Para o classificador KNN os parâmetros escolhidos para a busca foram: $n\_neighbors$, algorithm e metric. A Tabela 4 apresenta os valores testados para cada parâmetro bem como uma breve descrição sobre sua função. Os melhores valores encontrados pelo algoritmo de Grid Search foram: $n\_neighbors$: 10; algorithm: auto; metric: manhattan. Neste caso o melhor valor para os parâmetros metric e $n\_neighbors$ foi diferente da configuração padrão oferecida pela biblioteca. No modo padrão o valor de metric seria auto e o valor de vizinhos seria 5. Neste caso o algoritmo de Grid Search resultou em parâmetros maiores que os padrão, tornando o algoritmo levemente mais complexo para a sua execução.

**Tabela 4. Valores utilizados no Grid Search para KNN**

| Parâmetro | Valores | Descrição |
|---|---|---|
| $n\_neighbors$ | 3, 5, 7, 10 | Total de vizinhos a ser considerado |
| algorithm | auto, $ball\_tree$, $kd\_tree$ | Algoritmo para o calculo de vizinhos |
| metric | euclidean, manhattan, minkowski | Métrica para o cálculo da distância |

### 3.3. Multi Layer Perceptron (MLP)

Para a MLP os parâmetros escolhidos para a busca foram: $hidden\_layer\_sizes$, $learning\_rate\_init$, $max\_iter$. A Tabela 5 apresenta os valores testados para cada parâmetro selecionado e uma breve descrição sobre sua função. Os melhores parâmetros encontrados pelo algoritmo Grid Search foram: $hidden\_later\_sizes$: 80; $learning\_rate\_init$: 0.001; $max\_iter$: 150. Além destes parâmetros também utilizamos como função de ativação RELU, total de camadas escondidas 88, solver adam, $early\_stop$ como True e random state 13.

**Tabela 5. Valores utilizados no Grid Search para MLP**

| Parâmetro | Valores | Descrição |
|---|---|---|
| $hidden\_later\_sizes$ | 80, 100, 130 | Número de neurônios nas camadas escondidas |
| $learning\_rate\_init$ | 0.001, 0.0001 | Taxa de aprendizado inicial e de atualização |
| $max\_iter$ | 100, 150, 200 | Máximo de iterações |

## 4. Resultados Obtidos

Para a validação dos classificadores foi utilizado o algoritmo de validação cruzada k-fold estratificado com k igual a 5. Desta forma o conjunto de dados foi dividido em cinco porções de igual proporção entre as classes, ou seja, independente da porção analisada a mesma proporção entre classes da base original foi seguida. Para cada porção gerada pelo algoritmo se respeitou o uso de $80\%$ de amostras usadas para treinamento e $20\%$ para teste. Dessa forma cada fold possui 14897 amostras no total, sendo 11918 destinadas ao treinamento e 2979 destinadas ao teste. Durante o treinamento e o teste de todas as folds calculamos as métricas: acurácia, precisão, recall, F1 Score, matriz de confusão e curva ROC para cada classificador em cada uma das Folds.

Os valores de acurácia, precisão, recall e F1 Score calculados durante a etapa de teste podem ser vistos na Tabela 6. Nela é possível identificar que todos os classificadores testados tiveram um bom desempenho com a base testada. O melhor deles foi o algoritmo de Random Forest que em todas as métricas esteve próximo a 1.00 independente da fold utilizada. No entanto a fold que o algoritmo obteve o melhor resultado foi a de número 0 e 1. Os demais classificadores tiveram também bons resultados na validação cruzada. Para a MLP a fold de melhor desempenho foi a de número 0 em que todas a métricas ficaram iguais ou muito próximas a 0.99. Para o KNN a melhor fold foi a de número 3 em que todos as métricas calculadas ficaram próximas a 0.98.

No geral os algoritmos testados tiveram uma performance constante e muito próxima entre as folds testadas. Desta forma mostrando a robustez dos algoritmos ao serem submetidos a diferentes porções da base de dados testada. Mesmo ao se determinar o algoritmo de Random Forest como o melhor nesta etapa os demais ainda assim poderiam ser utilizados para o seu propósito.

Outra métrica utilizada para avaliar os classificadores foi a curva ROC. As Figuras 9 a 11 apresentam as curvas ROCs geradas para o cada classificador durante a etapa de teste. No eixo x está a taxa de falsos positivos e no eixo y está a taxa de verdadeiros positivos.

**Tabela 6. Avaliação dos classificadores para cada porção de teste**

| | | Fold 0 | Fold 1 | Fold 2 | Fold 3 | Fold 4 |
|---|---|---|---|---|---|---|
| Random Forest | | | | | | |
| | Acurácia | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 |
| | Precisão | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | Recall | 1.00 | 1.00 | 0.99 | 0.99 | 0.99 |
| | F1 Score | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 |
| KNN | | | | | | |
| | Acurácia | 0.99 | 0.99 | 0.98 | 0.99 | 0.98 |
| | Precisão | 0.99 | 0.99 | 0.99 | 1.00 | 0.99 |
| | Recall | 0.98 | 0.98 | 0.97 | 0.98 | 0.97 |
| | F1 Score | 0.99 | 0.99 | 0.98 | 0.99 | 0.98 |
| MLP | | | | | | |
| | Acurácia | 0.99 | 0.98 | 0.98 | 0.98 | 0.98 |
| | Precisão | 0.99 | 0.99 | 0.98 | 0.98 | 0.99 |
| | Recall | 0.98 | 0.97 | 0.98 | 0.98 | 0.96 |
| | F1 Score | 0.99 | 0.98 | 0.98 | 0.98 | 0.98 |

Ao analisarmos as curvas obtidas, todas se mantiveram muito próximas entre os classificadores e entre as folds. Houve pouca variação nas taxas calculadas. Em alguns casos com o que se vê para a fold 4 e 0 do Random Forest temos um classificador perfeito, ou seja, as taxas calculadas são próximas a 1.00.

Ao avaliar o que foi obtido ao visualizar as curvas ROCS é poss;ivel identificar que houve pouca variação de resultados entre as folds testadas. Tais resultados reforçam o que foi visto anteriormente na Tabela 6 que apresentava as métricas de acurácia, precisáo, recall e F1 Score.

Durante o processo de validação cruzada, os resultados obtidos também foram avaliados por meio de matrizes de confusão. Dessa forma uma matriz de confusão foi gerada para cada algoritmo de classificação em cada uma das folds de teste. Todas as matrizes foram geradas a partir das predições feitas na porção de teste de cada fold do algoritmo de validação cruzada. As Figuras 12 a 14 apresentam as melhores matrizes de confusão geradas para cada classificador durante esse processo.

Ao avaliar as matrizes de confusão é possível visualizar de maneira intuitiva o erro e o acerto entre as classes de nosso problema. O algoritmo de Random Forest apresentou 100% de acerto para as amostras da classe não-SPAM durante a avaliação de sua melhor fold (fold 1). Houve uma pequena quantidade de erros ao classificar amostras do tipo SPAM, cerca de 0.0047, ou seja menos de 1% das amostras. Esse valor representa uma quantidade ínfima de amostras do conjunto de dados usado para o treinamento e não deveria ser considerado algo critico para a utilização do algoritmo em situações reais.

Ao avaliar a matriz de confusão do classificador KNN é possível identificar que há menor erro ao identificar amostras do tipo não-SPAM. Assim como ocorreu com o Random Forest. O erro ao se classificar amostras não-SPAM foi de 0.0047. Representando menos de 1% de erro do total de amostras da base classificadas. Enquanto o erro para a

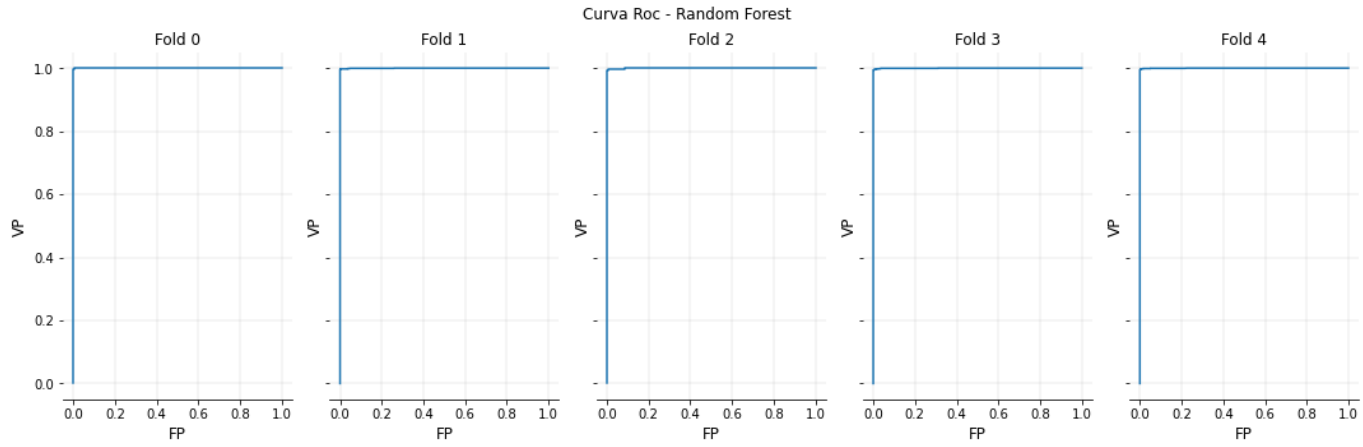


**Figura 9. Curvas ROC para Random Forest**

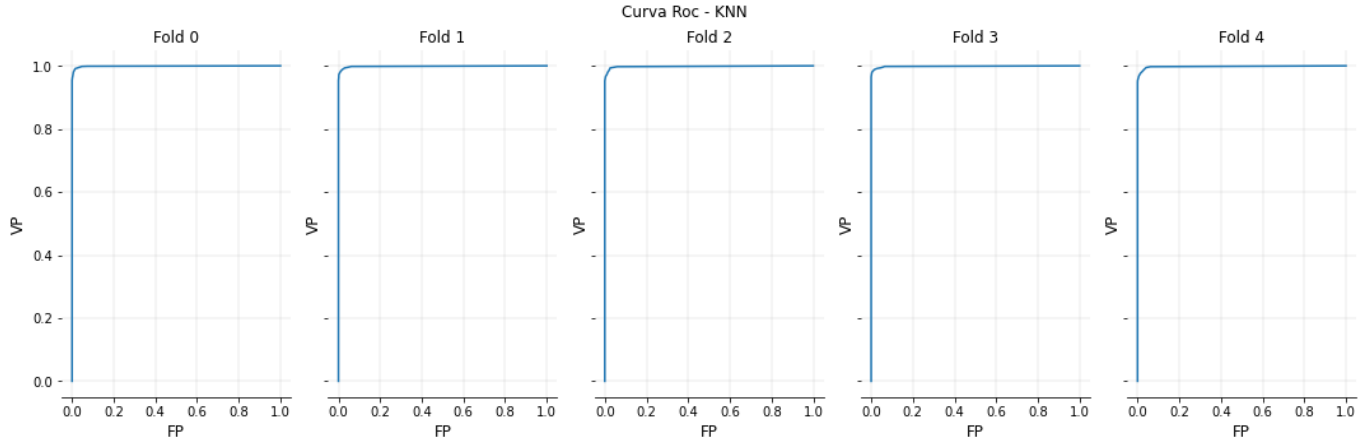


**Figura 10. Curvas ROC para KNN**

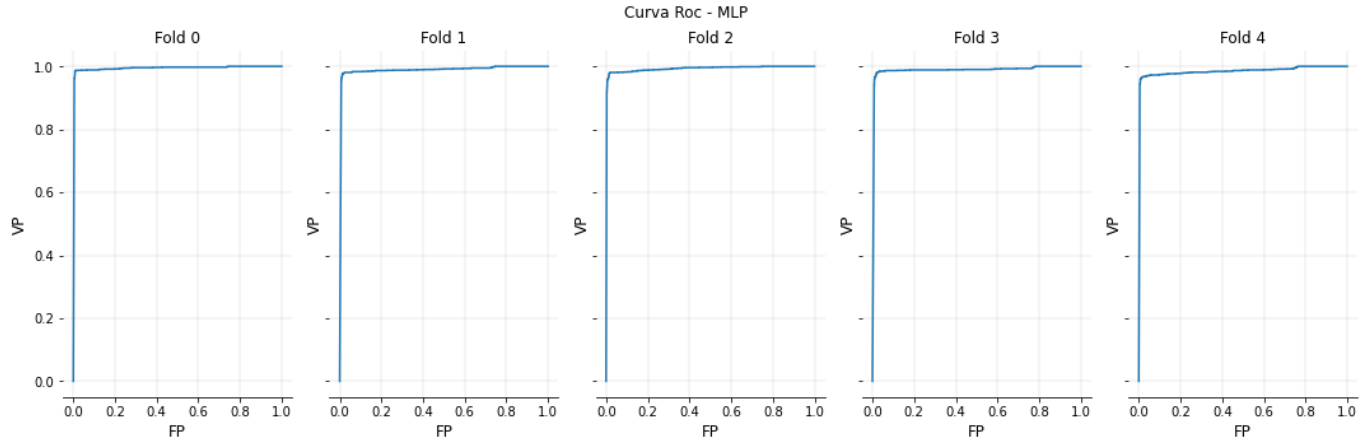


**Figura 11. Curvas ROC para MLP**

classificação de amostras do tipo SPAM foi um pouco maior, chegando a 0.0195, próximo as 2% de erro das amostras de teste. Tais resultados mostram a exatidão do classificador ao ser utilizado com dados novos.

Para o classificador MLP, o erro se concentrou também em identificar tweets do tipo SPAM. Tendo um erro de 0.0161, cerca de 1% do total de amostras. Enquanto que para a outra classe o erro foi de apenas 0.0080, menos de 1% da base apresentada.

De maneira geral ao avaliar as matrizes de confusão de todos os classificadores é possível concluir que a taxa de erro ficou concentrada na hora de identificar tweets do tipo SPAM. Essa dificuldade pode se dar pelas características extraídas ou até mesmo pelas características inerentes do problema. Outro ponto a se destacar da análise de performance dos classificadores é que todos tiveram bons desempenhos na base. Isso se dá provavelmente pelo uso do algoritmo de GridSearch. O algoritmo foi capaz de encontrar os melhores parâmetros e configurações para o nosso problema. Alinhando uma boa arquitetura de classificadores, parâmetros alinhados com o nosso objetivo de classificação e um bom dataset possibilitou os resultados excelentes obtidos por todos os classificadores testados.

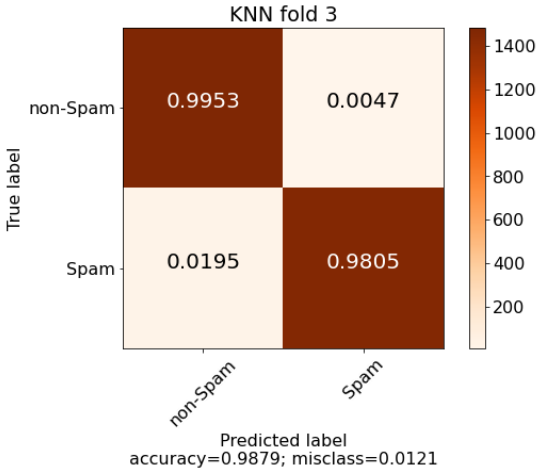


**Figura 12. Melhor matriz de confusão para KNN**

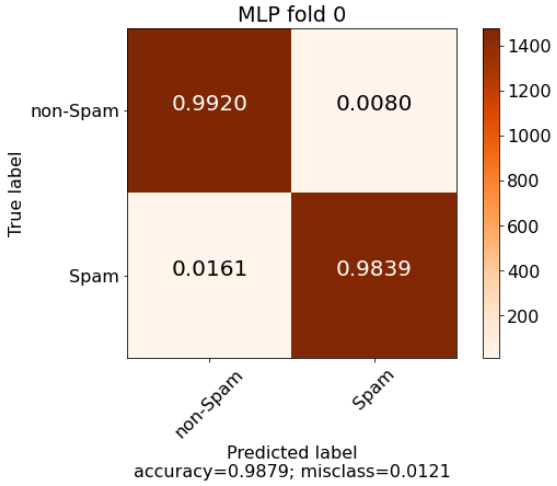


**Figura 13. Melhor matriz de confusão para MLP**

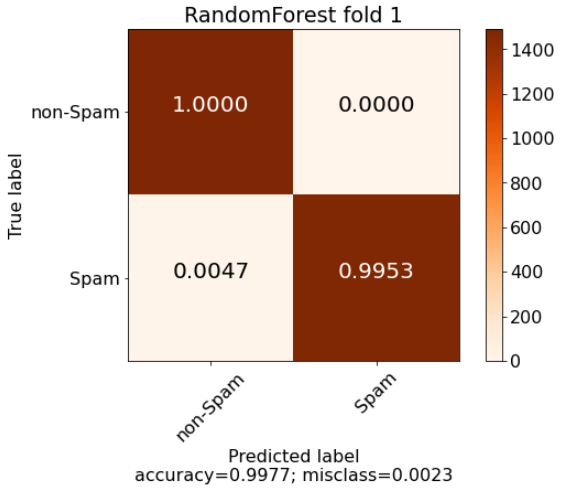


**Figura 14. Melhor matriz de confusão para Random Forest**

## 5. Experimentos

Vários experimentos foram realizados em *tweets* retirados com a API do Twitter. Todos os campos presentes no dataset utilizado para o treino foram calculados por completo, menos o *actions*. Esse campo utiliza a quantidade de *retweet*, favoritos e respostas que o *tweet* teve. Mas como utilizamos a API gratuita, a contagem de respostas em cada *tweet* não era disponibilizada, nos fazendo utilizar somente as outras métricas. O modelo utilizou o algoritmo de Random Forest para todos os experimentos, pois foi o que se saiu melhor nas validações.

### 5.1. Política - Bolsonaro

Política é um dos assuntos que há muita disseminação de notícias falsas. No Twitter diversos políticos e sua base de apoio mantém uma rede de interação para apoiar em algumas situações e também para atacar os adversários. Frequentemente, os ataques começam como uma notícia falsa que pode possuir um link redirecionando a algum site com mais informações sobre o assunto (sites que podem ser maliciosos e roubar informações de quem acessa). O teste foi realizado com 2500 *tweets* (os mais recentes) relacionados ao presidente Bolsonaro.

Na Figura 15 temos algumas informações importantes sobre os *tweets* que foram pegos. Percebe-se que os *tweets* que não foram considerados spam, não possuem muitas duplicatas, diferente dos spams. Isso indica que possivelmente os spams presentes no *dataset* possuem duplicações provindas de *retweet*. Os outros campos serão melhor explicados em seus gráficos a seguir.

A Figura 16 mostra a proporção de spam e não spam em todos os *tweets* pegos e nos que são únicos (sem duplicação). A diferença entre eles é devido o grande número de *tweets* parecidos (provavelmente *retweet*) que se encontram no grupo de spam. O próximo gráfico (Figura 17) mostra exatamente a diferença de compartilhamentos presentes em cada classe. Os spams poossuem um alto valor de *retweets* em comparação ao não spam, indicando que o *dataset* em questão foi inflado por post parecidos. A quantidade de *hashtags* e menções presentes no *tweet* são indícios de busca por visibilidade, como explicado ao estudar o dataset de treino anteriormente. Na Figura 18 vemos que

| | spam | non-spam |
|---|---|---|
| **Tweets identificados** | 1697 (68.37%) | 785 (31.63%) |
| **Tweets únicos (sem repetição)** | 616 | 775 |
| **Tweets únicos/identificados** | 36.30% | 98.73% |
| **Total de urls** | 990 | 241 |
| **urls/identificados** | 58.34% | 30.70% |
| **mais de 1 url** | 62 | 11 |
| **mais de 1 url/total de urls** | 6.26% | 4.56% |

**Figura 15. Características gerais do dataset**

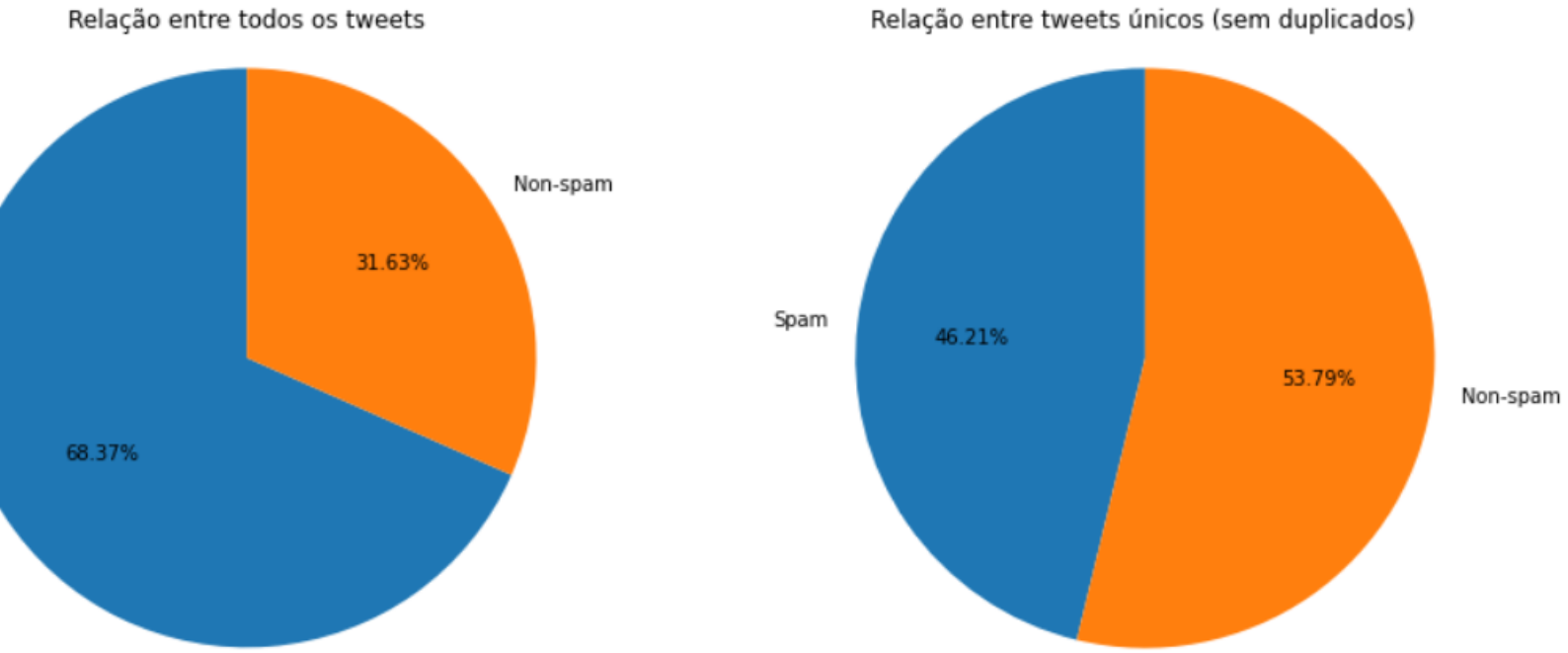


**Figura 16. Relação entre spam e não spam em todos os *tweets* e nos *tweets únicos***

os spams têm preferência em postar até duas *hashtags*, mesmo nos *tweets* únicos. Mas apesar disso, os não spam ainda continuam postando com mais de duas *hashtags*, o que pode indicar que por se tratar de política, as pessoas da rede que se comunicam sobre certo político estão querendo visibilidade pro seu *tweet*, comentando *hashtags* comuns que eles viram em outros posts, acreditando que aquilo faz parte do texto.

Na Figura 19, temos o Top 5 *hashtags* levando em consideração todos os *tweets*. Percebemos que existem *hashtags* relacionadas a política em geral e com o texto que foi

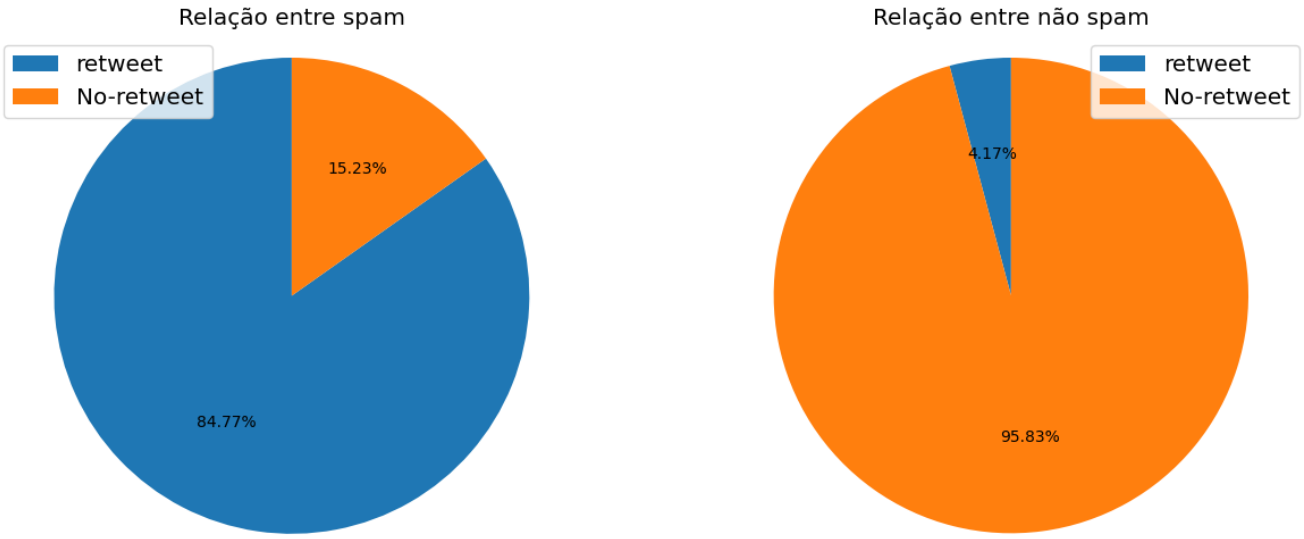


**Figura 17. Relação entre retweet em *tweets* spam e não spam**

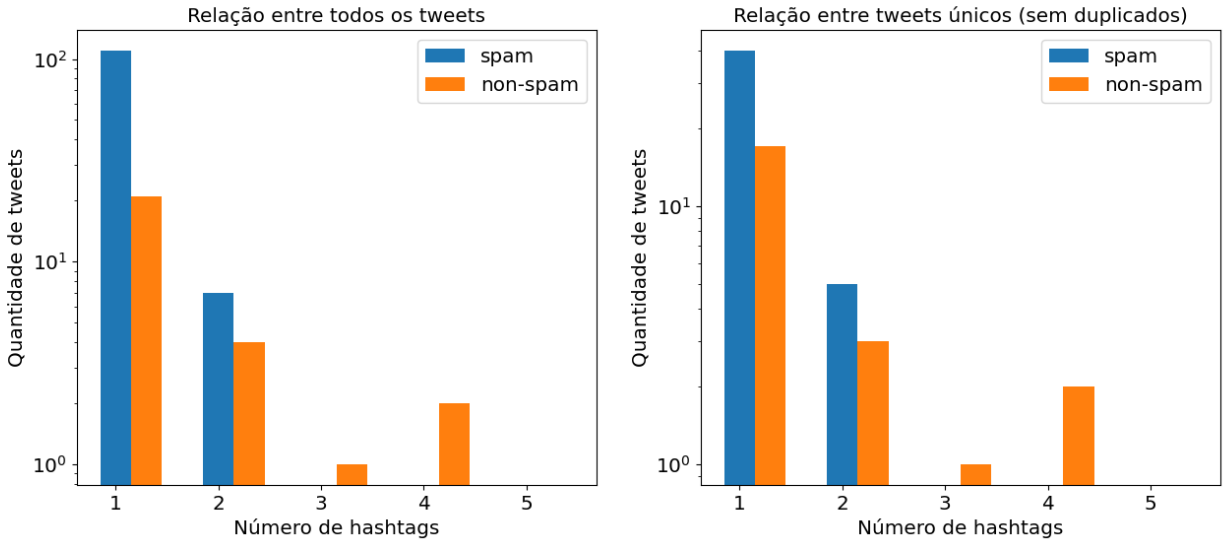


**Figura 18. Relação entre hashtags em todos os *tweets* e nos *tweets únicos***

**Figura 19.**

buscado (Bolsonaro).

Já na figura 20, vemos a distribuição entre os *tweets* não duplicados. A quantidade de *hashtags* passa a ser mais proporcionais. Levando em consideração aos dois Top 5 citados, vemos que há uma grande variabilidade de *hashtags* dentro do *dataset*. Isso pode ter acontecido pois diferenças gramaticais e letras maiúsculas e minúsculas não foram levadas em conta para entender semelhança de textos.

| | Hashtags | Quantidade | Hashtags | Quantidade |
|---|---|---|---|---|
| 0 | #MissGrandInternational | 45 | #ForaBolsonaro | 4 |
| 1 | #BolsonaroNuncaMais | 7 | #PTNuncaMais | 2 |
| 2 | #LulaEoPTMulheresComDireitos | 7 | #GrandLorena | 2 |
| 3 | #ForaBolsonaro | 6 | #STF | 2 |
| 4 | #tacladuran | 5 | #LulaLadrao | 2 |

**Figura 20. Top 5 *hashtags* entre spam e não spam (todos os *tweets*)**

| | Hashtags | Quantidade | Hashtags | Quantidade |
|---|---|---|---|---|
| 0 | #MissGrandInternational | 6 | #ForaBolsonaro | 3 |
| 1 | #ForaBolsonaro | 4 | #PTNuncaMais | 2 |
| 2 | #BolsonaroNuncaMais | 3 | #STF | 2 |
| 3 | #LulaEoPTMulheresComDireitos | 3 | #LulaLadrao | 2 |
| 4 | #LulaNoPodpah | 2 | #MoroTraidor | 1 |

**Figura 21. Top 5 *hashtags* entre spam e não spam (*tweets* únicos)**

Outro fator que pode ser levado em conta para detecção de spam, é a quantidade de menções presente no *tweet* (Figura 21). Do mesmo modo que nas *hashtags*, por ser política, é esperado que as menções sejam proporcionais.

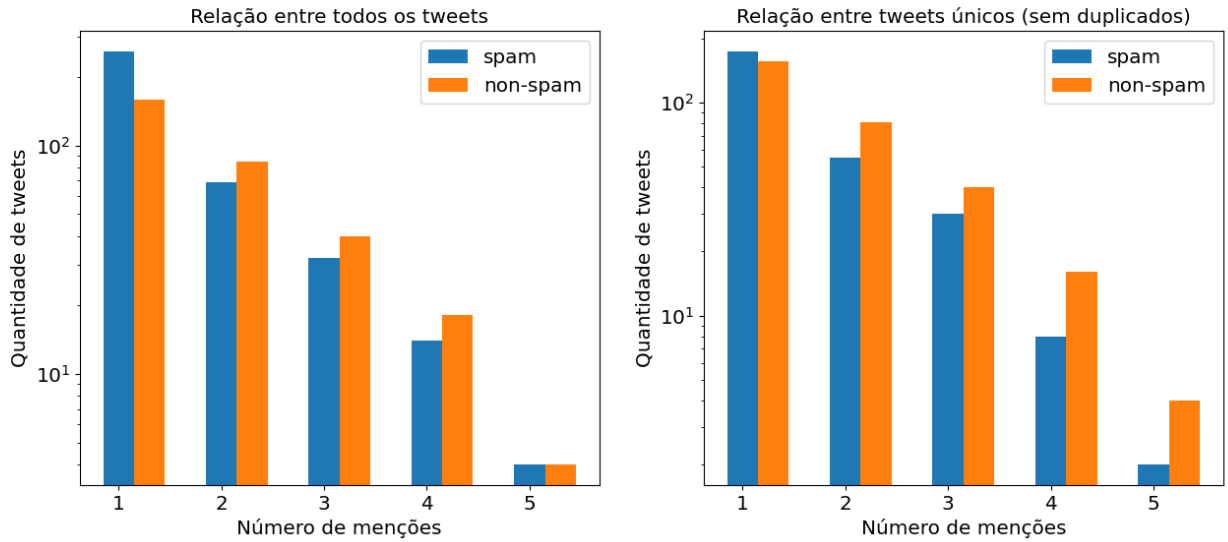


**Figura 22. Relação entre menções em todos os *tweets* e nos *tweets únicos***

Nossa última métrica é vista na Figura 22. Aqui, vemos claramente que em relação à todos os *tweets*, as URLs são em bem mais quantidade, pois a quantidade de *retweet* é grande (fazendo com que os *retweet* do segundo modo explicado tenha pelo menos uma URL). A partir de duas, já tiramos o peso dos *retweets* e percebemos que a proporção ainda é bem maior que os não spam.

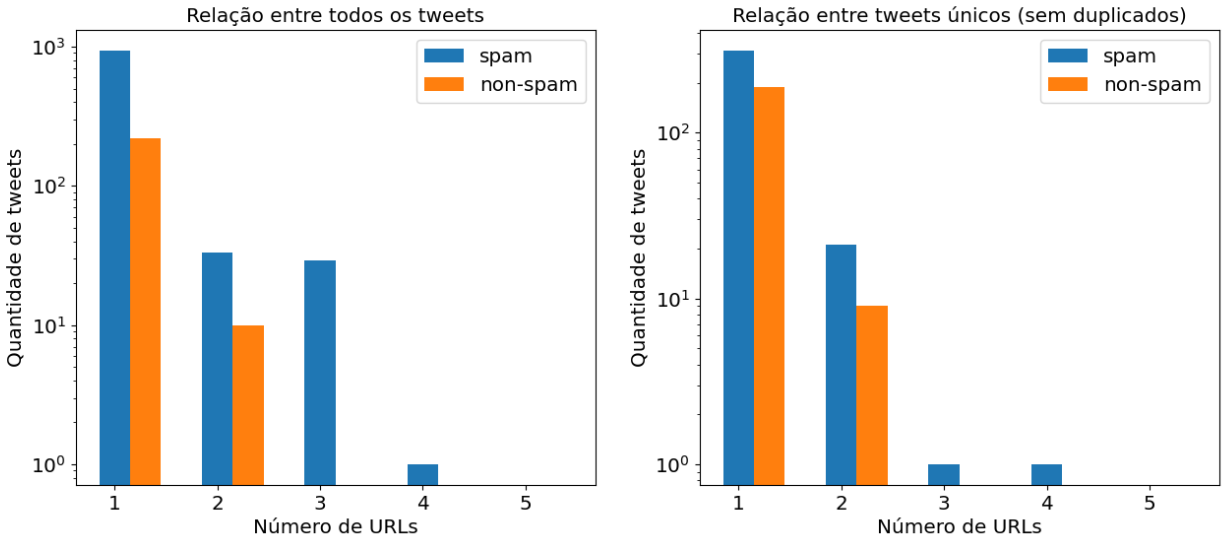


**Figura 23. Relação entre URLs em todos os *tweets* e nos *tweets únicos***

Nas Figuras 23 e 24, temos o Top 5 para todos os *tweets* e os não duplicados, respectivamente. Algumas URLs se manteram nos dois gráficos e a grande maioria são URLs que levam à posts que foram *retweetados*. No próximo experimento, entraremos em mais detalhes.

| | URLs | Quantidade | URLs | Quantidade |
|---|---|---|---|---|
| 0 | https://twitter.com/Jouberth19/status/14671085... | 48 | https://www.youtube.com/channel/UCGfNDjbkEK89D... | 16 |
| 1 | https://twitter.com/PATRlOTAS/status/146714514... | 48 | https://twitter.com/Simonycorredora/status/146... | 4 |
| 2 | https://uca97b68da57b4d006d7f74f789b.dl.dropbo... | 28 | https://twitter.com/phvitoria1/status/14553239... | 3 |
| 3 | https://t.me/desmentindobozo | 28 | https://pleno.news/brasil/eleicoes-2022/partid... | 2 |
| 4 | https://twitter.com/desmentindobozo/status/146... | 28 | https://twitter.com/Simonycorredora/status/146... | 2 |

**Figura 24. Top 5 *URLs* entre spam e não spam (*tweets* únicos)**

| | URLs | Quantidade | URLs | Quantidade |
|---|---|---|---|---|
| 0 | https://twitter.com/Jouberth19/status/14671085... | 1 | https://www.youtube.com/channel/UCGfNDjbkEK89D... | 16 |
| 1 | https://www.brasil247.com/regionais/sudeste/ra... | 1 | https://twitter.com/phvitoria1/status/14553239... | 2 |
| 2 | https://revistaoeste.com/politica/partido-de-b... | 1 | https://twitter.com/Marcia06513945/status/1467... | 1 |
| 3 | https://revistaoeste.com/politica/bolsonaro-af... | 1 | https://www1.folha.uol.com.br/equilibrioesaude... | 1 |
| 4 | https://terrabrasilnoticias.com/2021/12/michel... | 1 | https://www.youtube.com/watch?v=7wNCmZuw0aw&fe... | 1 |

**Figura 25. Top 5 *URLs* entre spam e não spam (*tweets* únicos)**

## 5.2. Hora do sorteio

Um dos temas que podem atrair várias pessoas são os sorteios realizados no Twitter. Vários perfis utilizam dessa prática para aumentar o número de seguidores: ”Me siga, compartilhe o post e participe do sorteio de um IPhone”. Com isso, o número de postagens semelhantes falando sobre o sorteio em questão cresce e acaba chamando a atenção de outras pessoas que acabam criando um ciclo de *retweet* desse post. Pessoas maliciosas aproveitam dessa estratégia para disponibilizar links de *phishing* como sendo de lojas que estão realizando os sorteios. O *dataset* utilizado possui 2500 *tweets* (os mais recentes) que possuem a palavra *giveaway*.

| | spam | non-spam |
|---|---|---|
| **Tweets identificados** | 2324 (93.03%) | 174 (6.97%) |
| **Tweets únicos (sem repetição)** | 260 | 88 |
| **Tweets únicos/identificados** | 11.19% | 50.57% |
| **Total de urls** | 2192 | 130 |
| **urls/identificados** | 94.32% | 74.71% |
| **mais de 1 url** | 296 | 47 |
| **mais de 1 url/total de urls** | 13.50% | 36.15% |

**Figura 26. Características gerais do dataset**

Vendo a Figura 25, a proporção de spam detectado fica gritante (os *tweets* que tiveram campos NaN não entraram nessa contagem. Por isso há uma diferença pequena entre os valores dessa tabela e os gráficos). 93% de todos os 2500 *tweets* foram classificados como spam. Olhando com mais calma as outras métricas calculadas, percebemos que essa *tag* está inflada de *tweets* idênticos (somente 11% dos *tweets* spam são únicos). Quanto mais próximo de 0, mais *tweets* repetidos existem no *dataset*. A seguir, veremos mais detalhadamente esses valores em gráficos. Percebe-se na Figura 26 que mais de

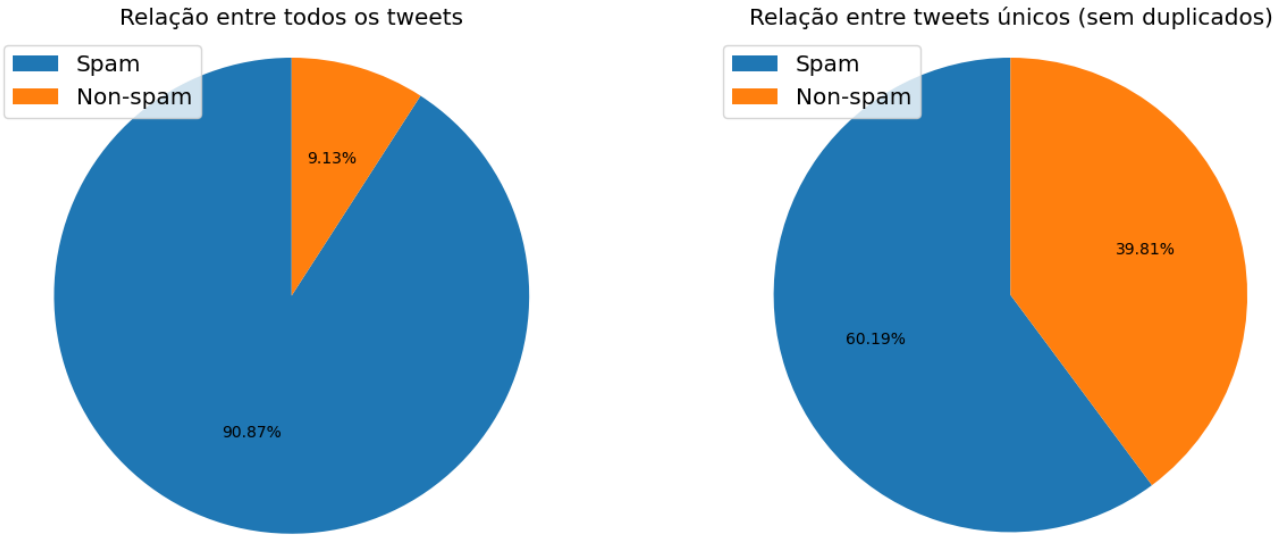


**Figura 27. Relação entre spam e não spam em todos os *tweets* e nos *tweets únicos***

**Figura 28.**

90% de todos os *tweets* são spam. Vendo o segundo gráfico, que representa o *dataset* sem repetição, verificamos que a proporção entre as classes diminui, pois os *tweets* spam inflaram o tema *giveaway* (exatamente o que um spam deseja. Ficar mais visível).

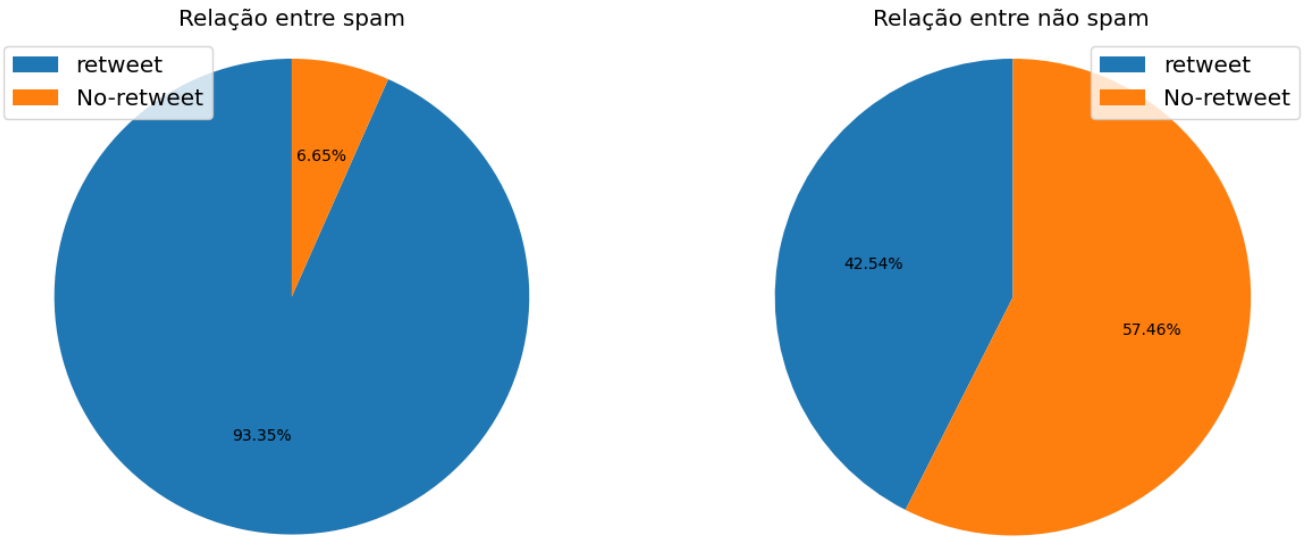


**Figura 29. Relação entre retweet em *tweets* spam e não spam**

O primeiro gráfico da figura 27 mostra que quase todos os *tweets* spam foram *retweets*, o que comprova a diferença entre os dois gráficos da Figura 26. Grande parte desses *retweets* são cópias idênticas do *tweet* original, indicando que o que foi explicado no inicio deste tópico (”compartilhe e concorra”) está acontecendo.

Como foi dito ao apresentar o dataset de treino, *hashtags* e menções são indícios de que o *tweet* está buscando visibilidade. No experimento em questão isso fica muito mais visível devido aos *tweets* não terem sido pré selecionados (a API devolve a listagem dos mais recentes, podendo ter vários repetidos). Nesse experimento percebe-se que o número desse elemento não segue o mesmo padrão. Por se tratar de um sorteio, quanto mais pessoas clicarem no post (ou link presente nele), mais benéfico será para o usuário que o fez.

Com isso, quanto mais *hashtags* forem colocadas, mais chances de uma pessoa procurando um sorteio ou promoção específica, se depare com um destes presente no *dataset* em questão. A figura 28 nos mostra que, levando em consideração os *tweets* duplicados, o numero de *hashtags* presentes são bem superiores aos *tweets* normais. Realizando o processo para não duplicados, pequenas quantidade de *hashtags* se mantém quase que proporcionais. Mas conforme q quantidade aumenta, os spams ganham mais espaço.

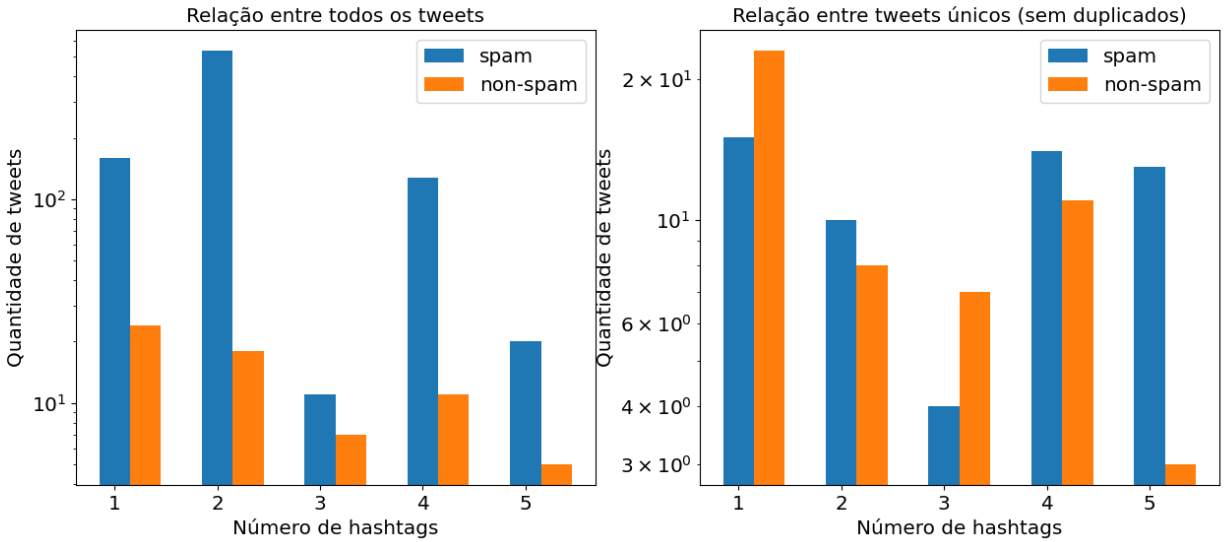


**Figura 30. Relação entre hashtags em todos os *tweets* e nos *tweets únicos***

**Figura 31.**

O Top 5 da figura 29 mostra que a mesma *hashtag* está inflando o *dataset*, portanto há uma chance muito grande de alguém que procure por uma delas, caia em algum *tweet* que foi classificado. O Top 5 da figura 30 já mostra a relação de *hashtags* retirando os *tweets* duplicados. É possível perceber que a maioria dos que estão presentes no top geral, não se encontram aqui, indicando novamente que os 2500 *tweets* pegos estão inflados com praticamente as mesmas *hashtags*.

| | Hashtags | Quantidade | Hashtags | Quantidade |
|---|---|---|---|---|
| **0** | #giveaway | 582 | #Giveaway | 36 |
| **1** | #Hearthstone | 511 | #TWICE | 17 |
| **2** | #OMENFTW | 131 | #giveaway | 16 |
| **3** | #parceria | 131 | #Airdrop | 10 |
| **4** | #ftwesports | 119 | #SCIENTIST | 10 |

**Figura 32. Top 5 *hashtags* entre spam e não spam (todos os *tweets*)**

| | Hashtags | Quantidade | Hashtags | Quantidade |
|---|---|---|---|---|
| **0** | #Giveaway | 47 | #Giveaway | 35 |
| **1** | #NFTGiveaway | 20 | #TWICE | 16 |
| **2** | #giveaway | 19 | #giveaway | 15 |
| **3** | #NFTs | 17 | #Airdrop | 10 |
| **4** | #NFTdrop | 14 | #SCIENTIST | 10 |

**Figura 33. Top 5 *hashtags* entre spam e não spam (*tweets* únicos)**

As menções seguem a mesma ideia. Para participar do sorteio, muitos perfis pedem que marquem alguém no post para estar apto a concorrer. Com isso, percebe-se pela Figura 31 que spams possuem uma quantidade de menção superior aos *tweets* normais.

Falando sobre as URLs, os spams estão inflados com links (Figura 32), pois como visto inicialmente, a grande maioria são *retweets* e por isso possuem em seu texto um link para o *tweet* original (o que infla *tweets* com um link). Pegando como base a partir de duas URLs, vemos que a relação entre spam e não spam fica cada vez maior, indicando que spams compartilham mais links do que os não spam.

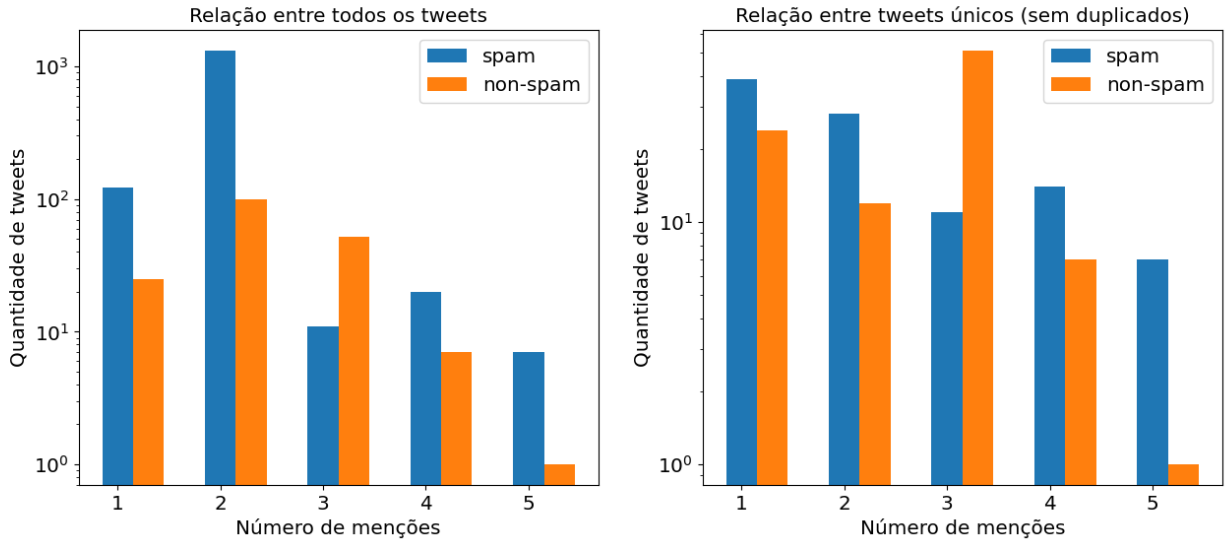


**Figura 34. Relação entre menções em todos os *tweets* e nos *tweets únicos***

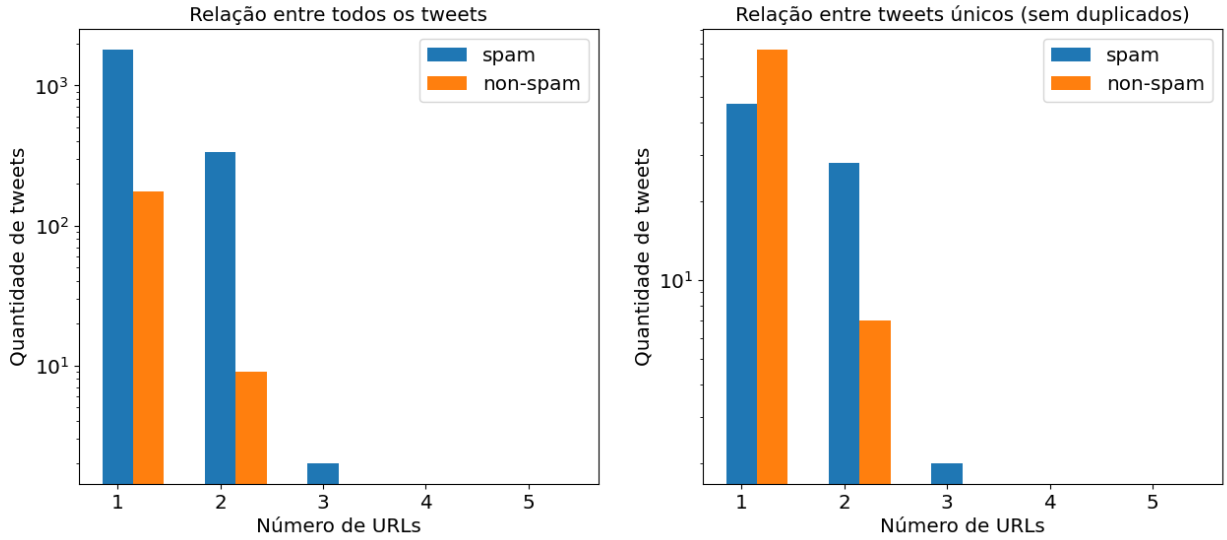


**Figura 35. Relação entre URLs em todos os *tweets* e nos *tweets únicos***

O Top 5 URLs da Figura 33 em relação ao da Figura 34 mostra como o *retweet* presente modifica a extração dos dados. A Figura 33 possui um link para um *tweet* específico (provavelmente quem iniciou o sorteio) e ele não aparece no Top URLs de *tweets* únicos. A discrepância gigante entre o top 1 das duas figuras, mostra mais uma vez o que chamamos de ”*tag* inflada por spam”.

| | URLs | Quantidade | URLs | Quantidade |
|---|---|---|---|---|
| 0 | https://twitter.com/JOliveira10_/status/146673... | 567 | https://twitter.com/SuperFakeHS/status/1466011... | 86 |
| 1 | https://twitter.com/KingVenomStream/status/146... | 511 | https://botisimo.com/u/tiorapadura/shop | 2 |
| 2 | https://twitter.com/SuperFakeHS/status/1466011... | 494 | https://twitter.com/mihyunvibe/status/14671390... | 1 |
| 3 | https://gleam.io/hZOOZ/logitech-g-x-astro-2021... | 122 | https://twitter.com/mihyunvibe/status/14671389... | 1 |
| 4 | https://twitter.com/ASTROGamingBR/status/14664... | 122 | https://twitter.com/itsnrjonce/status/14671366... | 1 |

**Figura 36. Top 5 *URLs* entre spam e não spam (*tweets* únicos)**

| | URLs | Quantidade | URLs | Quantidade |
|---|---|---|---|---|
| 0 | https://gleam.io/3yLbY-RTjF1TwvaY | 2 | https://botisimo.com/u/tiorapadura/shop | 2 |
| 1 | https://botisimo.com/u/tiorapadura/shop | 2 | https://twitter.com/mihyunvibe/status/14671390... | 1 |
| 2 | https://twitter.com/SuperFakeHS/status/1466011... | 1 | https://twitter.com/mihyunvibe/status/14671389... | 1 |
| 3 | https://gleam.io/cjdEt/ftw-omen-squad-giveaway... | 1 | https://twitter.com/itsnrjonce/status/14671366... | 1 |
| 4 | https://twitter.com/MoraisGaming/status/146613... | 1 | https://twitter.com/ilikeSweetberry/status/146... | 1 |

**Figura 37. Top 5 *URLs* entre spam e não spam (*tweets* únicos)**

## 6. Reproducibilidade

O projeto pode ser encontrado no GitLab.

## 7. Conclusão

Spams são elementos perigosos que podem atrair pessoas desatentas para clicarem em seus posts e com isso terem seus dados roubados. O Twitter, como dito no início, possui uma forma de compartilhamento de informações de forma pública e instantânea. Dessa forma qualquer usuário pode ter acesso a qualquer tipo de informação que não foram confirmadas sobre qualquer assunto desejado. Durante o trabalho desenvolvemos três classificadores diferentes para determinar se um tweet é SPAM ou não-SPAM. Um dos focos de nosso trabalho foi gerar características de fácil extração para que o tempo de processamento seja o mais rápido possível. Para construção dos algoritmos utilizamos o Grid Search para encontrar a melhor configuração para nosso problema. Além disso utilizamos também a validação cruzada com K-Fold estratificado para validar os classificadores.

Dentre os classificadores testados: Random Forest, MLP e KNN, aquele de melhor desempenho foi o Random Forest. Este em duas folds conseguiu obter resultados próximos de 100% em todas as métricas calculadas (acurácia, recall, precisão, F1 Score). Os demais classificadores também tiveram um desempenho ótimo e que se mostrou promissor para a utilização em situações do dia-a-dia.

Durante a extração de características e exploração das bases de dados, em muitos momentos as características selecionadas se mostram muito semelhantes entre as classes. Tal característica pode prejudicar o desempenho do modelo. Visto que muito se muda na forma de escrever um *tweet* de uma pessoa para a outra. Esse tipo de coisa se torna evidente ao trabalhar com *tweets* relacionados a temas políticos.

Em temas políticos, as pessoas estão acostumadas a copiarem o texto de um *tweet* (ou alguma outra informação) e colocar *hashtags* sem ter muita ideia do que está acontecendo, devido a toda rede que essa pessoa faz parte realizar a mesma coisa. Ao analisar

tweets de sorteios os resultados obtidos foram mais próximos do que visto no dataset de treinamento. A maioria dos sorteios pedem que mencionem pessoas no *reetweet* e por isso quanto mais posts desse modo, mais fácil será o cálculo das interações das características extraídas. Outro fator interessante que conseguimos descobrir, é como as URLs são dispostas no texto. Muitos *tweets* que são *retweet* possuem ao final do texto um link que redireciona ao *tweet* original. Isso pode atrapalhar o modelo na hora de decidir se é spam ou não levando em consideração somente a presença de URLs. Por isso há a presença da relação de quantidade de URLs por número de palavras e também uma flag indicando se é um *retweet* ou não. Além disso, a grande maioria estão encurtadas e possuem o domínio *t.co*. Um processamento superficial pode deixar esses links de lado, não indicando que haja um possível redirecionamento à algum site malicioso.

Também conseguimos entender um pouco do funcionamento dos spams. Eles buscam visibilidade e por isso tentam atrair o máximo de pessoas possíveis. Suas principais armas é o retweet na esperança de aumentar sua rede de interação. Usar várias *hashtags* diferentes para tentar ”pescar”pessoas que estão buscando algo relacionado a elas. Mencionar ou forçar que os usuários mencionem várias outras contas no post para obter mais interações a partir das pessoas mencionadas.

## Referências


Benevenuto, F., Magno, G., Rodrigues, T., and Almeida, V. (2010). Detecting spammers on twitter. In *In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS*.

Chen, C., Zhang, J., Xie, Y., Xiang, Y., Zhou, W., Hassan, M. M., AlElaiwi, A., and Alrubaian, M. (2015). A performance evaluation of machine learning-based streaming spam tweets detection. *IEEE Transactions on Computational Social Systems*, 2(3):65–76.

Dean, B. How many people use twitter in 2021? [new twitter stats]. `https://backlinko.com/twitter-users`.

UTKML (2018). Utkml's twitter spam detection competition. `https://www.kaggle.com/c/utkmls-twitter-spam-detection-competition/data?select=sample_submission.csv`.