Análise Exploratória de Atributos Textuais em Bases de Dados para Identificação de Campos Sensíveis

Bruno H. Labres, André Grégio, Fabiano Silva

[1]Departamento de Informática – Universidade Federal do Paraná (UFPR)
Curitiba – PR – Brasil

{bhl16,gregio,fabiano}@inf.ufpr.br

Abstract. *The imminent implantation of the Brazilian General Law for the Protection of Personal Data urges the creation of automated techniques in database anonymization. The existing tools are dependent of a specialist to manually adjust the data. In this work, we propose to apply classification algorithms to attributes commonly found in databases. We hope to improve the automated classification of database attributes, where it can be used in the development of new softwares or as a component used before the anonymization process. The experimental evaluation of the proposed digram frequency representation shows that it is possible to obtain simple machine learning models, such as random forest and neural network, capable of classifying people's names, addresses and textual descriptions reaching 97% of accuracy and using 676 features.*

Resumo. *A iminente implantação da Lei Geral de Proteção de Dados Pessoais brasileira torna urgente a criação de técnicas automatizadas para anonimização de bases de dados. As ferramentas existentes são dependentes de que um especialista manualmente escolha os campos de interesse. Neste trabalho, propõe-se o uso de algoritmos de aprendizado de máquina em atributos textuais a fim de explorar como identificar nomes e outras informações sensíveis. Com isso, espera-se fomentar aplicações capazes de classificar automaticamente campos sensíveis em bancos de dados em um passo anterior à anonimização. A avaliação experimental da representação por frequência de digramas proposta, mostra que é possível obter modelos simples de aprendizado de máquina, como florestas aleatórias e redes neurais, capazes de classificar nomes de pessoas, endereços e descrições textuais com alta taxa de acurácia (97%) usando 676 características.*

1. Introdução

Na atualidade, o cuidado com o gerenciamento de dados é essencial. Isso se deve à grande quantidade de dados gerados anualmente e, por consequência, os riscos de vazamentos e violações à privacidade. Para a contenção destas crises, governos estão se mobilizando com legislações, como a Lei Geral de Proteção de Dados Pessoais (LGPD) brasileira, que cita a anonimização de dados como uma forma de aumentar a segurança da informação. Porém, métodos de anonimização de dados exigem conhecimento técnico que é de difícil aplicação por um usuário médio [El Emam and Dankar, 2008].

Neste trabalho, propõe-se o uso de algoritmos de aprendizado de máquina em atributos textuais, em português brasileiro, a fim de explorar como identificar nomes e

outras informações sensíveis. Com isso, espera-se fomentar aplicações capazes de classificar automaticamente campos sensíveis em bancos de dados, que sirva de passo anterior à anonimização. Foram conduzidos experimentos com dados de nomes de pessoas, endereços e descrições em texto para treinamento e teste de um modelo com alta taxa de acurácia e baixo número de características necessárias. Como algoritmos de classificação foram utilizadas florestas aleatórias e redes neurais, e, como características, as medidas *Term Frequency - Inverse Document Frequency* (TF-IDF) de digramas no conjunto de amostras. O modelo de rede neural alcançou 97% de acurácia com 676 características.

2. Definição do Problema

De acordo com a LGPD e a Declaração Universal dos Direitos Humanos, o respeito à privacidade constitui a base de sociedades democráticas e dos direitos individuais. Na atualidade, o vazamento de dados de indivíduos é uma ocorrência cada vez mais frequente. Isso se deve à informatização de dados, onde estes estão cada vez mais relacionados com o mundo digital, e, por consequência, suscetível às vulnerabilidades deste. Entre os exemplos de vazamentos mais recentes, podemos citar o do DATASUS, que expôs os dados de cerca de 243 milhões de pessoas, contendo informações como nome completo, endereço, telefone e CPF. Uma alternativa para conter estes riscos seria suprimir ou cifrar informações em determinados bancos de dados. Por exemplo, uma instituição pode fornecer aos seus membros diferentes versões do banco de dados, com variações entre a quantidade de dados em estado bruto e anonimizado de acordo com os níveis internos de privilégios. Dessa forma, um vazamento da versão anonimizada seria menos danoso aos indivíduos afetados. Isso pode ser feito com o uso de técnicas de anonimização de bancos de dados.

Com o aumento da globalização, o acesso a informações privadas de determinado indivíduo é facilitado. Isso se deve ao fato de que diversas entidades e organizações possuem esses dados, que podem ter sido cedidos pelo indivíduo como forma de facilitar ou obter acesso a recursos que facilitem seu dia-a-dia, como atendimento médico, controle bancário, lazer, etc. Porém, com isso podem ocorrer vazamentos maliciosos ou trâmite de informações, e esses dados podem ser disponibilizados publicamente ou para entidades maliciosas. Dessa forma, para preservar a privacidade do indivíduo, é necessário que suas informações sensíveis estejam devidamente anonimizadas. Assim, a extração destas informações será dificultada.

Em suma, entre os maiores desafios para a anonimização de dados estão:

- Encontrar o conjunto de dados a ser anonimizado em cada banco.
- Definir a melhor abordagem de anonimização para cada dado, como supressão ou cifragem de dados, por exemplo.
- A complexidade para o usuário médio anonimizar seu banco sem ter conhecimento profundo em técnicas de anonimização.

3. Trabalhos Relacionados

A automatização do processo de anonimização de bancos de dados é uma área que ainda está em desenvolvimento. Um exemplo é o trabalho de [Malle et al., 2017] cujo uso de aprendizado de máquina interativo apresentam resultados superiores aos métodos mais rígidos.

Para auxiliar nos métodos de anonimização, nosso trabalho foca em estágios iniciais do processo, ou seja, na identificação de dados sensíveis, realizando experimentações com Term Frequency – Inverse Document Frequency (TF–IDF) e digramas como características. O uso dessas características combinadas em algoritmos de aprendizado de máquina tem sido testado em diversas áreas, como mostraremos a seguir.

O trabalho *Gender Identification in Twitter using N-grams and LSA* [Daneshvar and Inkpen, 2018] utiliza unigramas e digramas para identificar gêneros de autores no Twitter. Foi alcançada a acurácia de 82% com modelos de máquinas de vetores de suporte em bases de dados em espanhol. Outro trabalho relacionado é o de [Çöltekin and Rama, 2018], que utiliza máquina de vetores de suporte para identificar tweets que indicam o uso de drogas.

4. Proposta e metodologia

Neste trabalho, propõe-se o uso de algoritmos de aprendizado de máquina em atributos textuais a fim de identificar nomes de pessoas. Foram construídos classificadores e uma avaliação experimental foi conduzida sobre um conjunto de dados textuais em português brasileiro envolvendo nomes de pessoas, endereços e descrições. A arquitetura do método proposto é apresentada na figura 1.

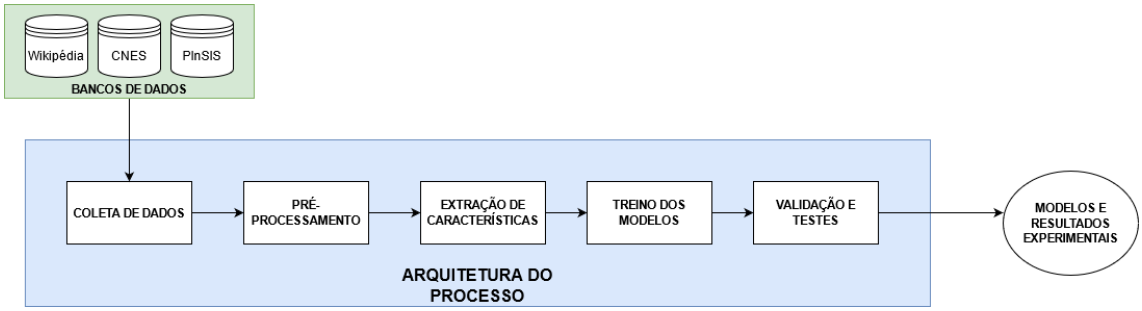

Figura 1. Arquitetura do processo proposto para classificação de atributos.

A partir disso, analisaremos cada etapa do processo.

4.1. Coleta de dados

Para realizar estes experimentos, obtivemos dados de três bancos de dados distintos:

- **PInSIS**: O Projeto para Inovação de Sistemas de Informação e Saúde (PInSIS) [Grégio, 2018] é um projeto firmado entre o Centro de Computação Científica e Software Livre da Universidade Federal do Paraná (C3SL) e o Ministério da Saúde. O objetivo do projeto é o monitoramento de equipamentos médicos com o intuito de auxiliar no bom uso do recurso público aplicado pelo Ministério da Saúde na compra de equipamentos médicos. Para isso, é preciso monitorar se o equipamento entregue é instalado, está em operação e, principalmente, se está atendendo pacientes pelo Sistema Único de Saúde (SUS). Neste trabalho foi utilizado o bancos de dados do PInSIS para a realização de avaliações experimentais iniciais.
- **Cadastro Nacional de Estabelecimentos de Saúde** (CNES): O Cadastro Nacional de Estabelecimentos de Saúde (CNES) é um documento público e sistema de

informação oficial de cadastramento de informações acerca de todos os estabelecimentos de saúde do país. É utilizado pelo Ministério da Saúde para a verificação das instalações e mão-de-obra dos estabelecimentos de saúde do país, independentemente de sua natureza jurídica ou integração com o Sistema Único de Saúde (SUS). Esses dados estão disponíveis no Portal CNES, uma plataforma que compila os dados dos estabelecimentos nacionais de maneira transparente e aberta. Este experimento utiliza bancos de dados do PInSIS, para a realização de diversos experimentos. Os atributos utilizados para o experimento correspondem aos nomes dos profissionais e aos endereços dos estabelecimentos de saúde.

- **Wikipedia**: A Wikipedia é um projeto de enciclopédia colaborativa, universal e multilíngue estabelecido na internet sob o princípio wiki. Tem como propósito fornecer um conteúdo livre, objetivo e verificável, que todos possam editar e melhorar. O projeto é definido pelos princípios fundadores. O conteúdo é disponibilizado sob a licença Creative Commons BY-SA e pode ser copiado e reutilizado sob a mesma licença — mesmo para fins comerciais — desde que respeitando os termos e condições de uso. Os dados da Wikipedia em português foram extraídos através de um espelho mantido pelo C3SL. As amostras de descrições textuais foram extraídas desta plataforma.

  No fim da coleta, obtemos os dados brutos para a etapa de pré-processamento.

4.2. Pré-processamento

O pré-processamento consiste nas seguintes etapas:

- **Extração de atributos**: A biblioteca Pandas é utilizada para a extração dos nomes e endereços referentes aos dados do PInSIS e CNES. Para a extração das descrições textuais da Wikipedia, os dados obtidos no espelho estão formatados com tags em HTML. Por isso, é utilizada a ferramenta HTML2TEXT que realiza a conversão dos dados brutos, com tags em HTML retirados da Wikipedia, para apenas seus textos limpos. Essa ferramenta foi escolhida devido a sua praticidade para executar a tarefa.
- **Mapeamento de caracteres**: A partir dos dados extraídos, os caracteres são mapeados para simplificar as operações. Ou seja, todos os caracteres são minúsculos e os caracteres com acento são mapeados para suas versões sem acento. Essa etapa auxilia na simplificação e visualização do uso de características no trabalho.
- **Escolha das amostras:** Devido ao grande número de amostras disponíveis nas bases de dados, o experimento foi limitado a utilizar 50.000 amostras de cada grupo (nomes, endereços e descrições textuais) totalizando 150.000 amostras. Isso se deve às limitações de processamento e tempo perante à grande quantidade de amostras. As 50.000 amostras de cada grupo foram escolhidas aleatoriamente dentro de seu conjunto. Dessas amostras, 80% são usadas para treino, e, também, obtenção da acurácia de validação cruzada. Os outros 20% são usados para a obtenção da acurácia de testes.

  Essa etapa resulta nos dados tratados para a etapa de extração de características.

4.3. Extração de características

As características são extraídas com base no valor ***Term Frequency – Inverse Document Frequency*** **(TF–IDF)** [Leskovec et al., 2014] de digramas. O valor TF–IDF, ou

frequência do termo–inverso da frequência nos documentos, é uma medida estatística que mede a importância que um termo possui em uma coleção de documentos [Robertson, 2004].

O valor da **frequência de termos (TF)** representa a relação entre a presença de um determinado termo em um documento. É calculado pela fórmula:

$$TF(t) = \frac{\text{Número de vezes em que o termo t aparece em um documento}}{\text{Número total de termos em um documento}}$$

O valor do **Inverso da Frequência do Documento (IDF)** representa o quão importante um termo é. Termos que aparecem muitas vezes em vários documentos diferentes provavelmente não são discriminantes importantes.

$$IDF(t) = \log_e \frac{\text{Número total de documentos}}{\text{Número total de documentos com o termo t}}$$

Dessa forma, o valor de TF-IDF é:

$$TF - IDF(t) = TF(t) \times IDF(t).$$

Os digramas usados neste experimento correspondem à presença de dois caracteres alfabéticos consecutivos. Ou seja, considere o conjunto:

$$A = (x \text{ ; } x \text{ é uma letra do alfabeto}).$$

O conjunto A corresponde às letras do alfabeto. Com isso, considere o conjunto:

$$D = (A \times A).$$

O conjunto D corresponde ao produto cartesiano do conjunto A com ele mesmo, ou seja, seus elementos são todas as combinações possíveis entre duas letras do alfabeto, como, por exemplo $\{$aa, ab, ac, ..., ba, bb, bc, ..., zz$\}$. Dessa forma, $|A| = 26$ e $|D| = 26.26 = 676$.

O vetor de características é uma matriz onde cada linha representa um documento (amostra de nome, endereço ou descrição textual) e cada coluna uma característica (digrama). No experimento proposto, isso resultaria em uma matriz de dimensão 150.000×676. Cada célula desta matriz contém o valor TD-IDF de determinado digrama em um documento em relação ao conjunto de documentos.

Para identificarmos quais são os valores mais relevantes de TF-IDF, precisamos aplicar uma transformação de dados na matriz de valores TF-IDF. Isso se deve às dimensões da matriz serem de 150.000×676. Uma representação ilustrativa está na tabela 1.

Devido a essa grande quantidade de elementos na matriz, não seria viável encontrar as características mais relevantes através dos dados brutos, pois devemos obter um

Tabela 1. Representação ilustrativa da matriz de tamanho 150.000×676 com os valores TF-IDF de acordo com cada amostra e característica.

Digramas / Amostras	'aa'	'ab'	'ac'	...	'zz'
Amostra 1	0,1	0	0	...	0,1
Amostra 2	0,1	0	0,2	...	0,5
Amostra 3	0,4	0,3	0	...	0,2
...	...	...	...	...	...
Amostra 150.000	0,5	0,2	0	...	0

único valor para cada uma das 676 características, a fim de selecionar as mais relevantes. Dessa forma, devemos aplicar uma transformação para obter um único valor para cada característica a partir destes 150.000 documentos. Para isso, aplicaremos a média dos valores TF-IDF de cada característica em todos os documentos. Isso equivale a escolher uma coluna (característica), efetuar o cálculo da média dessa coluna em todas as linhas (documentos) e inserir este valor em um novo vetor de características. Esse vetor resultante é representado pela tabela 2 terá até 676 características.

Tabela 2. Representação ilustrativa do vetor resultante da transformação de média que será realizada no experimento.

Digramas / Amostras	'aa'	'ab'	'ac'	...	'zz'
Valores transformados	0,52	0,78	0,11	...	0,20

Os valores de TF-IDF foram calculados utilizando a biblioteca *sklearn*. Essa biblioteca foi escolhida devido a sua praticidade e suporte continuo dado pelos desenvolvedores, além de satisfazer as necessidades deste experimento.

No final dessa etapa, temos os exemplos para serem utilizados no processo de classificação.

4.4. Treino dos modelos

Com a extração de características, diferentes números de características utilizadas são testados, a fim de treinar um modelo de alta precisão e baixo custo. O número de características de cada modelo treinado é reduzido aproximadamente pela metade para o treino de um novo modelo. Apenas as características com os maiores valores TF-IDF são mantidas a cada novo modelo. Ou seja, são treinados e testados modelos com as 5, 10, 20, 40, 85, 169, 338 e 676 características mais relevantes. O objetivo é comparar a acurácia de validação cruzada de cada algoritmo de aprendizado de máquina e com diferentes tamanhos de vetores de características. Nessa etapa são utilizadas 80% das amostras do conjunto de dados. Após essa etapa, submetemos os modelos aos testes com um outro subconjunto, equivalente a 20% da amostras do conjunto de dados. Com isso, discutiremos a viabilidade dos modelos.

Com as características selecionadas e os dados normalizados, os modelos são treinados e testados utilizando dois algoritmos diferentes. Os algoritmos utilizados

para classificação são as implementações da biblioteca *sklearn* em Python para florestas aleatórias e redes neurais.

A floresta aleatória utilizada possui 100 estimadores ou árvores. A rede neural utilizada é do tipo *feedforward* e possui uma camada de entrada, uma camada oculta e uma camada de saída. A camada oculta possui 100 neurônios para computação. A função de ativação utilizada é a ReLU. O número máximo de iterações até a convergência do algoritmo de treino é de 200 iterações.

Para o ajuste de pesos no treinamento utilizando *backpropagation*, utilizamos o algoritmo Adam [Kingma and Ba, 2017], que se trata de um algoritmo para otimização baseada em gradiente de primeira ordem de funções objetivas estocásticas, com uma taxa de aprendizagem de 0,001.

Os parâmetros utilizados foram escolhidos para gerarem redes ou florestas simples, com o intuito de refinar os parâmetros para os classificadores com melhores resultados em experimentos futuros.

4.5. Validação e testes

Na última etapa é realizado o processo de validação cruzada e testes a serem realizados com estes modelos.

O conjunto de dados total possui 150.000 amostras, onde, balanceadamente entre os grupos, o conjunto foi dividido em subconjuntos com 80% e 20% do total de amostras.

Com o subconjunto de 80% do total de amostras, foi realizado com o método de **validação cruzada**. Nesse método, foram utilizadas cinco partições para avaliar treino e teste. A validação cruzada possui cinco iterações, sendo usado em cada iteração um conjunto de teste diferente com 24.000 amostras, e, um de treino, com 96.000. O número de partições foi escolhido arbitrariamente.

Após a etapa de validação cruzada, os modelos treinados serão submetidos aos testes, onde será obtida a acurácia de teste dos modelos utilizando um conjunto de 30.000 amostras.

5. Resultados

Após a realização do experimentos, obtemos o vetor de características, onde os digramas estão ordenados em ordem decrescente, onde os maiores valores contém os digramas mais relevantes para a classificação. Os dez digramas mais relevantes corresponde ao conjunto {‘de’, ‘ra’, ‘an’, ‘ar’, ‘os’, ‘es’, ‘ua’, ‘ru’, ‘do’, ‘da’}. Os cinco digramas menos relevantes correspondem ao conjunto {‘zj’, ‘qj’, ‘qh’, ‘xg’, ‘xq’}.

Com os experimentos de classificação, obtivemos a acurácia de validação cruzada e o desvio padrão das acurácias dos modelos treinados. Os modelos são catalogados de acordo com o número de características ou digramas utilizado, e, também, de acordo com o algoritmo de aprendizado de máquina escolhido. Com isso, obtivemos as acurácia de validação cruzada dos modelos. Em seguida, testamos os módulos com um conjunto de testes independente, com isso obtivemos as acurácias de testes e suas matrizes de confusão.

A tabela 3 mostra os resultados de acurácia de validação cruzada obtidos. A tabela 4 mostra os valores de desvio-padrão obtidos. A tabela mostra os valores de acurácia de

Tabela 3. Valores da acurácia de validação cruzada de acordo com os algoritmos de aprendizado de máquina estudados e o número de características escolhido.

Algoritmo utilizado / Número de características	Floresta Aleatória	Rede Neural
5	**0.6559**	0.6473
10	**0.8015**	0.7994
20	**0.8728**	0.8673
40	**0.9146**	0.9098
85	0.9467	**0.9503**
169	0.9545	**0.9619**
338	0.9537	**0.9681**
676	0.9528	**0.9675**

Tabela 4. Valores de desvio-padrão da validação cruzada de acordo com os algoritmos de aprendizado de máquina estudados e o número de características escolhidas.

Algoritmo utilizado / Número de características	Floresta Aleatória	Rede Neural
676	$6,60 \times 10^{-}4$	$6,68 \times 10^{-}4$
338	$1,35 \times 10^{-}3$	$6,29 \times 10^{-}4$
169	$1,08 \times 10^{-}3$	$1,36 \times 10^{-}3$
85	$1,38 \times 10^{-}3$	$7,02 \times 10^{-}4$
40	$5,69 \times 10^{-}4$	$2,29 \times 10^{-}3$
20	$2,13 \times 10^{-}3$	$2,21 \times 10^{-}3$
10	$3,10 \times 10^{-}3$	$2,53 \times 10^{-}3$
5	$3,03 \times 10^{-}3$	$3,63 \times 10^{-}3$

teste. As figuras 2 e 3 apresentam as matrizes de confusão dos modelos que apresentaram maior acurácia de teste para cada algoritmo. As medidas são exibidas variando de acordo com o número de características mais relevantes utilizadas e o algoritmo de aprendizado escolhido.

6. Discussão

Analisando os resultados obtidos nos experimentos realizados podemos concluir que, considerando o tamanho do conjunto de dados inicial, os resultados foram satisfatórios.

Um fator interessante a ser analisado são as características ou digramas escolhidos como mais ou menos relevantes para a classificação. Observando os mais relevantes, podemos notar digramas que estão fortemente presentes em uma das classes estudadas. Por exemplo, o digrama ‘de’ aparece com frequência em descrições textuais; ‘ru’ aparece em endereços, devido à palavra ‘rua’; e ‘ar’ aparece em nomes comuns, como ‘Maria’. Além

Tabela 5. Valores da acurácia de de teste de acordo com os algoritmos de aprendizado de máquina estudados e o número de características escolhido.

Algoritmo utilizado / Número de características	Floresta Aleatória	Rede Neural
5	**0.6585**	0.6523
10	**0.8089**	0.8066
20	**0.8783**	0.8707
40	**0.9208**	0.9150
85	0.9518	**0.9532**
169	.0.9579	**0.9654**
338	0.9574	**0.9711**
676	0.9567	**0.9715**

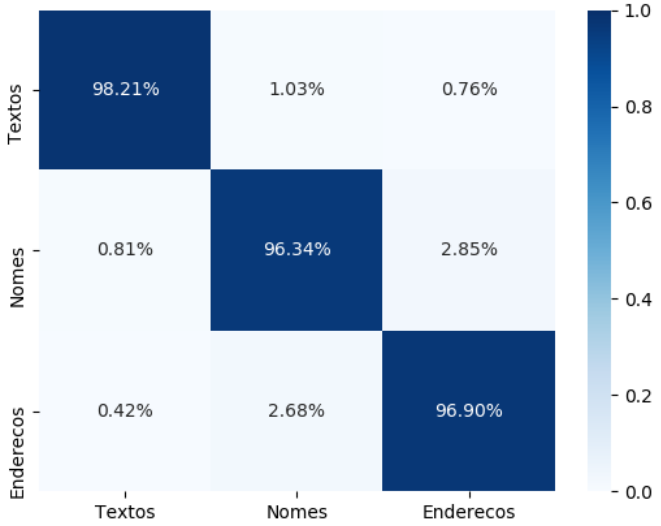

Figura 2. Matriz de confusão de validação cruzada (onde o eixo X indica as classes preditas e o eixo Y as classes reais) para o modelo de rede neural treinado com 676 características.

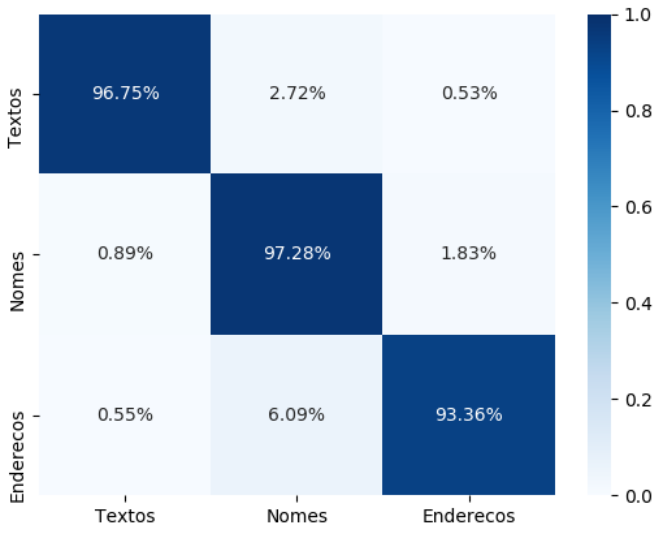

Figura 3. Matriz de confusão de validação cruzada (onde o eixo X indica as classes preditas e o eixo Y as classes reais) para o modelo de floresta aleatória treinado com 169 características.

disso, os digramas menos relevantes são os que pouco aparecem na língua portuguesa, como ‘xg’ ou ‘zj’.

A maior parte dos modelos obtidos com os dois algoritmos obtiveram bom desempenho quanto a acurácia, principalmente com maiores números de características. Os baixos valores de desvio-padrão validam a pontuação de acurácia dos modelos.

A pior taxa é quando são utilizadas apenas cinco características, onde a acurácia de validação cruzada é de 65% e 64%, para a floresta aleatória e rede neural, respectivamente. Já a de teste é de 65% em ambas. Com dez características, todas as taxas de acurácia são de cerca de 70%. Com 40, todas as taxas de acurácia estão acima de 90%. Nos testes, com 676 características, a rede neural tem a maior acurácia de teste, chegando a 97,15%. Isso significa que entre os dois algoritmos, a rede neural com 676 características é a que obteve maior taxa de acurácia. Porém, nos modelos com o número de características entre 5 e 40, a floresta aleatória possui maior acurácia.

Com a análise de erros registrados na classificação dos exemplos e na matriz de confusão, podemos perceber que, nos modelos de maior acurácia de cada algoritmo, as maiores taxas de erro se referem à distinção entre nomes e endereços. A taxa de nomes classificados como endereço variam entre 1,83% e 3,59%. Já a de endereços classificados como nomes varia entre 2,68% e 6,63%, contendo a maior taxa de erro das matrizes em questão. Isso se deve às similaridades entre os digramas dos dois tipos de atributo. Por exemplo, diversos endereços, como ruas e avenidas, possuem nomes ou sobrenomes próprios de pessoas. Por esse fator, os exemplos de texto possuem digramas mais distintos, e, consequentemente, com maior taxa de acertos e menor taxa de erros no geral. Entre os três modelos, a rede neural foi a que teve a maior taxa de acertos com textos (98,21%) e endereços (96,90%). A floresta aleatória teve a maior taxa de acertos com

nomes (97,28%).

Um fator a ser considerado é que, nas aplicações reais, o número de amostras que pode ser enviado é o número de elementos que determinada coluna possui no banco de dados. Com isso, todos esses elementos pertencem ao mesmo tipo de atributo, ou seja, à mesma classe. Isso faz com que a taxa de erros presente nos experimentos influencie de maneira mínima na classificação das tabelas para anonimização, já que existem diversas amostras disponíveis para serem classificadas por coluna. Ou seja, caso uma amostra seja classificada incorretamente, existem diversas outras pertencentes à mesma classe que serão classificadas corretamente (de acordo com a acurácia calculada no experimento).

Com isso, o melhor método para treinamento nesse experimento são as florestas aleatórias. Isso se deve à alta acurácia mesmo sem o uso de todas as características disponíveis, já que com 85 características esse modelo apresenta resultados satisfatórios. Além disso, é um método computacionalmente mais barato para treino do que as redes neurais.

7. Conclusão

Este trabalho teve como objetivo estudar o uso de algoritmos de classificação aplicados a atributos de bancos de dados a fim de se identificar automaticamente atributos sensíveis. Foram feitos experimentos com dados de nomes de pessoas, endereços e descrições em texto para treinamento e teste de um modelo com alta taxa de acurácia e baixo número de características necessárias. Os resultados podem ser considerados promissores já que foi obtida uma acurácia de 95% utilizando 85 características.

Além disso, foram validados diversos modelos através de experimentos e apresentamos uma análise de seus desempenhos. Foram alcançados resultados satisfatórios quanto ao estado da arte [Tveit et al., 2004], o que abre espaço para a continuidade da linha de pesquisa na área. Além disso, obtivemos uma lista de digramas mais relevantes na língua portuguesa utilizando a pontuação TF-IDF.

O campo de anonimização de dados associado com aprendizado de máquina ainda oferece muitas possibilidades de estudo, portanto, esse trabalho pode ser complementado e aprimorado de diversas formas. O próximo passo é expandir o número de classes dos experimentos, adicionando outros atributos comuns aos bancos de dados que geralmente precisam ser anonimizados em aplicações reais, como de um sistema de saúde ou bancário. Com isso poderemos estudar se a taxa de acurácia se mantém alta com um conjunto maior de classes. Por fim, o modelo de classificação deve ser acoplado com um anonimizador de dados. Dessa forma construiremos uma plataforma de anonimização eficiente e acessível ao usuário médio.

8. Referências Bibliográficas

Ç. Çöltekin and T. Rama. Drug-use identification from tweets with word and character n-grams. In *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task*, pages 52–53, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5914. URL https://aclanthology.org/W18-5914.

S. Daneshvar and D. Inkpen. Gender identification in twitter using n-grams and lsa: Notebook for pan at clef 2018. In *CLEF*, 2018.

K. El Emam and F. K. Dankar. Protecting privacy using k-anonymity. *Journal of the American Medical Informatics Association*, 15(5):627–637, 2008.

D. . C. L. . M. B. . V. L. . M. E. . S. M. Grégio, A. R. A. ; Aleo. *Monitoramento Remoto e Georreferenciamento de Tecnologias para Saúde. In: Fotini Santos Toscas; Maria Helenice de Castro. (Org.). Avanços, Desafios e Oportunidades no Complexo Industrial da Saúde em Serviços Tecnológicos.* MS, 2018.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

J. Leskovec, A. Rajaraman, and J. D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, USA, 2nd edition, 2014. ISBN 1107077230.

B. Malle, P. Kieseberg, and A. Holzinger. Interactive anonymization for privacy aware machine learning. 11 2017.

S. Robertson. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation - J DOC*, 60:503–520, 10 2004. doi: 10.1108/00220410410560582.

A. Tveit, O. Edsberg, T. Røst, A. Faxvaag, Nytrø, T. Nordgård, M. Ranang, and A. Grimsmo. Anonymization of general practioner medical records. 01 2004.