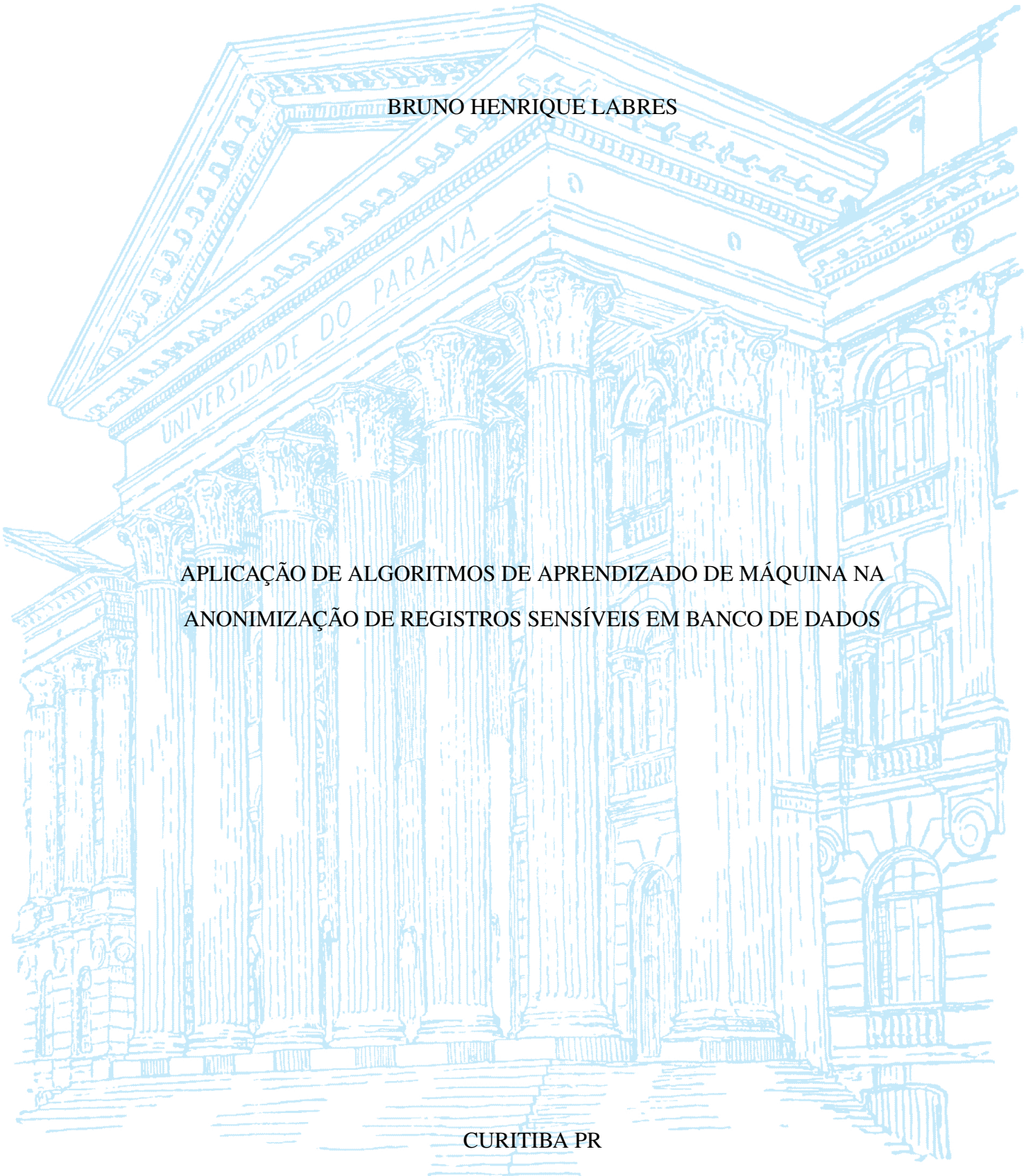
UNIVERSIDADE FEDERAL DO PARANÁ

BRUNO HENRIQUE LABRES



APLICAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA NA ANONIMIZAÇÃO DE REGISTROS SENSÍVEIS EM BANCO DE DADOS

CURITIBA PR

2021

BRUNO HENRIQUE LABRES

APLICAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA NA ANONIMIZAÇÃO DE REGISTROS SENSÍVEIS EM BANCO DE DADOS

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná..

Área de concentração: *Ciência da Computação*.

Orientador: Fabiano Silva.

Coorientador: André Gŕegio.

CURITIBA PR

2021

Aos meus avós...

RESUMO

O risco de vazamento de dados sensíveis de indivíduos, principalmente de usuários de sistemas de informação de saúde, é um problema que se agrava com a interação de diversos equipamentos, bancos de dados, aplicações e a Internet. Para evitar violações à privacidade dos usuários desse tipo de sistema, pode-se utilizar diversas ferramentas de anonimização de bancos de dados. Estas ferramentas protegem informações sensíveis, com o uso de métodos para suprimir ou cifrar identificadores que conectam um indivíduo ao dado armazenado. Entretanto, as ferramentas de anonimização atuais são de difícil uso e envolvem conhecimento aprofundado do modelo (tabelas, campos e suas relações). Neste trabalho, propõe-se o uso de algoritmos de classificação aplicados a atributos que podem ser encontrados em bancos de dados. Com isso, espera-se auxiliar na aplicação automatizada de métodos de anonimização de bancos de dados, tanto com novos softwares ou com modelos para servirem de entrada para softwares existentes. 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 algoritmos de aprendizado de máquina testados foram máquinas de vetores de suporte, floresta aleatória e redes neurais. Os resultados obtidos com a técnica de redes neurais resultaram em um modelo que alcançou 97% de acurácia com 676 características.

Palavras-chave: Anonimização de dados. Aprendizado de máquina. Banco de dados.

ABSTRACT

The risk of sensitive data leakage from individuals, mainly users of health information systems, is an issue aggravated by the interaction of several devices, databases, applications, and the Internet. Database anonymization tools may be used to prevent privacy violations of these systems' users. These tools help in the sensitive information protection, with the use of supression or criptography methods to cipher data linked to a person. However, current tools are difficult to use and rely on the deep knowledge about the database model—its tables, fields, and relationships). In this work, we propose to apply classification algorithms to attributes commonly found in databases. We hope to auxiliate in the automated anonymization of databases, where it can be used in the development of new softwares or as a component of a existing one. We conducted experiments to train and test effective models (high accuracy, few required features) using people names, addresses, and textual description as input data to obtain a model with high accuracy and low number of features. The machine learning algorithms applied in this experiment are the support vector machine, random forest and neural network. The obtained results were a model trained with neural networks with 97% of accuracy and using 676 features.

Keywords: Data anonymization. Machine Learning. Databases.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

CNES	Cadastro Nacional de Estabelecimentos de Saúde
CSV	*Comma-separated-values*
DINF	Departamento de Informática
HTML	Linguagem de Marcação de HiperTexto
PInSIS	Projeto para Inovação de Sistemas de Informação e Saúde
PPGINF	Programa de Pós-Graduação em Informática
SVC	*Support Vector Clustering*
SVM	*Support Vector Machine*
TA	Tabela anonimizada
TF-IDF	*Term Frequency–Inverse Document Frequency*
UFPR	Universidade Federal do Paraná

LISTA DE SÍMBOLOS

μ	média das amostras
σ	variância das amostras

SUMÁRIO

1 INTRODUÇÃO

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 [5], 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, de forma que um vazamento seja menos danoso ao indivíduos afetados. Isso pode ser feito com o uso de técnicas de anonimização de bancos de dados.

Este trabalho tem como objetivo estudar o uso de algoritmos de classificação aplicados a atributos de bancos de dados a fim de se identificar automaticamente as colunas que necessitam ser anonimizadas. 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.

Este Capítulo está dividido em cinco seções, onde a primeira aborda as motivações para este trabalho, a segunda trata dos desafios sobre o problema tratado, a terceira apresenta a solução proposta pelo trabalho, na quarta temos a exposição das contribuições para o estado da arte e na quinta tratamos da organização deste documento.

1.1 MOTIVAÇÃO

De acordo com a Lei Geral de Proteção de Dados Pessoais (LGPD) [6] e a Declaração Universal dos Direitos Humanos [7], o respeito à privacidade constitui a base de sociedades democráticas e dos direitos individuais.

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 individuo 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.

1.2 DESAFIO

O problema da anonimização de dados privados se baseia em utilizar uma variedade de técnicas para modificar os dados originais de maneira que as informações ou registros 'sensíveis' sejam mascarados [8]. Isso faz com que caso os dados sejam compartilhados com terceiros, estes não conseguirão extrair informações sensíveis dos indivíduos. Dessa forma, a anonimização de dados contribui para as questões éticas e legislativas da preservação de dados.

Existem diversos métodos e aplicações cujo objetivo é anonimizar os dados, como serão mostrados no Capítulo 3. Porém, os softwares de anonimização possuem alta complexidade para serem usados pelo usuário médio. Isso se deve a grande quantidade de informações complexas presentes nas aplicações, o que exige que o usuário possua um elevado nível de conhecimento quanto às metodologias da anonimização, mesmo que deseje realizar tarefas simples. Além disso,

as aplicações também exigem conhecimento do usuário quanto ao banco de dados, já que não são capazes de inferir qual é o conteúdo presente em cada atributo (se é um nome ou um endereço, por exemplo), e, também, pela complexidade em classificar manualmente o quão sigiloso é cada atributo, como forma de realizar o pré-processamento para um método de anonimização específico.

Portanto, 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.

1.3 OBJETIVOS

Levando em conta os desafios da área, esse trabalho tem como foco estudar diferentes técnicas para a classificação de registros sensíveis em banco de dados, ou seja, identificar qual tipo de informação está contido em cada atributo de uma tabela.

Derivando do objetivo geral, os seguintes objetivos específicos são visados:

- Avaliar algoritmos de classificação (SVM, floresta aleatória e rede neural) para classificar diferentes informações de um banco de dados.
- Verificar a viabilidade da classificação de tipos diferentes de informação, tais como nomes, endereços e descrições textuais.
- Extrair características adequadas para classificação, como os digramas das amostras, e testar a viabilidade da classificação com estas características.
- Utilizar os valores *term frequency–inverse document frequency* (TF-IDF) do conjunto de dados como características.

1.4 CONTRIBUIÇÃO

As principais contribuições deste trabalho são:

- A obtenção de diferentes modelos capazes de classificar amostras como nomes de pessoas, endereços e descrições em texto a fim de identificar automaticamente as colunas que necessitam ser anonimizadas. Os modelos foram validados por meio de experimentos e apresentamos uma análise de seus desempenhos, onde a acurácia dos modelos treinados com SVM, florestas aleatórias e redes neurais foram de 94%, 95% e 97%, respectivamente;
- A validação dos métodos de transformação de dados utilizando a análise *term frequency–inverse document frequency* (TF-IDF).

1.5 ORGANIZAÇÃO DO DOCUMENTO

Este trabalho é dividido em seis Capítulos: no Capítulo 2 são apresentados os fundamentos teóricos necessários para a compreensão do trabalho; no Capítulo 3 são expostos os trabalhos relacionados que apresentam o estado da arte das pesquisas nesta área em questão; no Capítulo 4 é um aprofundamento na definição do problema; o Capítulo 5 explica a metodologia proposta, onde tratamos de falar sobre a base de dados tratamentos sobre a mesma e o modelo de solução proposta; o Capítulo 6 aborda os experimentos e resultados e finalizamos no Capítulo 7 com a conclusão do trabalho, fazendo uma análise crítica sobre os resultados obtidos e contribuições para o estado da arte.

2 FUNDAMENTAÇÃO TEÓRICA

Este Capítulo apresenta fundamentos teóricos básicos para a composição deste trabalho. Estes fundamentos se dividem nas áreas de anonimização de dados e aprendizado de máquina, especificamente em máquinas de vetores de suporte (SVM), florestas aleatórias e redes neurais, que são utilizadas na implementação deste trabalho.

Este Capítulo está dividido em duas seções, onde a primeira aborda o que é anonimização dos dados, conceitos e métodos para sua aplicação. A segunda Seção aborda aprendizado de máquina, seus conceitos e algoritmos.

2.1 ANONIMIZAÇÃO DE DADOS

Essa Seção aborda temas relacionados ao campo de estudo da anonimização de dados.

2.1.1 Conceitos básicos

Nesta subseção são abordados alguns conceitos básicos que serão utilizados nesse trabalho.

Os **dados** trabalhados neste documento se referem a informações de determinados indivíduos contida em uma tabela com linhas e colunas, onde cada linha é chamada de tupla e cada coluna é um atributo. Cada atributo é único.

A **inferência** é o ato de descobrir um novo fato com base em uma informação prévia. Um **vazamento** será referenciado como o ato de tornar explicita ou inferível uma informação que não deveria ser divulgada [1].

Os **identificadores explícitos** são dados que entregam diretamente a identidade de um indivíduo, sem que seja necessário realizar um cruzamento de dados. Alguns exemplos são nomes, CPFs e endereços.

Uma **tabela anonimizada** (TA) é uma tabela cujos identificadores explícitos foram removidos ou cifrados [9].

Um **quasi-identificador** é um conjunto de atributos em uma tabela anonimizada em que, caso cruzado com informações externas, possibilita reidentificar a que essa informação se refere [10].

2.1.2 Reidentificação por inferência de dados

O problema da reidentificação por inferência de dados se refere a, a partir da união entre informações entre diferentes tabelas, obter a qual indivíduo determinadas informações se referem. O conceito de quasi-identificador está diretamente associado ao inferência entre dados, já que é a partir deles que podem ser realizados os cruzamentos.

Um exemplo da gravidade da reidentificação por inferência de dados foi demonstrado por Latanya Sweeney [1]. O autor comprou dados anonimizados da *Group Insurance Commission (GIC)*, uma organização responsável por comprar planos de saúde para empregados do estado de Massachusetts. Como os dados supostamente foram anonimizados, eles são vendidos para fins científicos [11] e assim foram obtidos pelo autor. Os dados vendidos correspondem a cerca de informações sobre 135.000 indivíduos, incluindo código postal, data de nascimento, gênero, etnia, datas de visitas, diagnóstico, procedimento realizado, medicação e dosagem. Essas informações correspondem ao círculo esquerdo da Figura 2.1.

Da mesma forma, o autor comprou os registros anonimizados da lista de eleitores da cidade de Cambridge em Massachusetts [12]. A lista incluia dados como nome, endereço, código postal, data de nascimento e gênero de cada eleitor. O círculo direito da Figura 2.1 corresponde a esses dados.

Combinando ambos os dados através dos quasi-identificadores de código postal, data de nascimento e sexo, é possível identificar informações médicas destes indivíduos. Sweeney demonstrou como através disso seria possível obter informações médicas do governador de Massachusetts da época, William Weld, utilizando a reindetificação por inferência de dados.

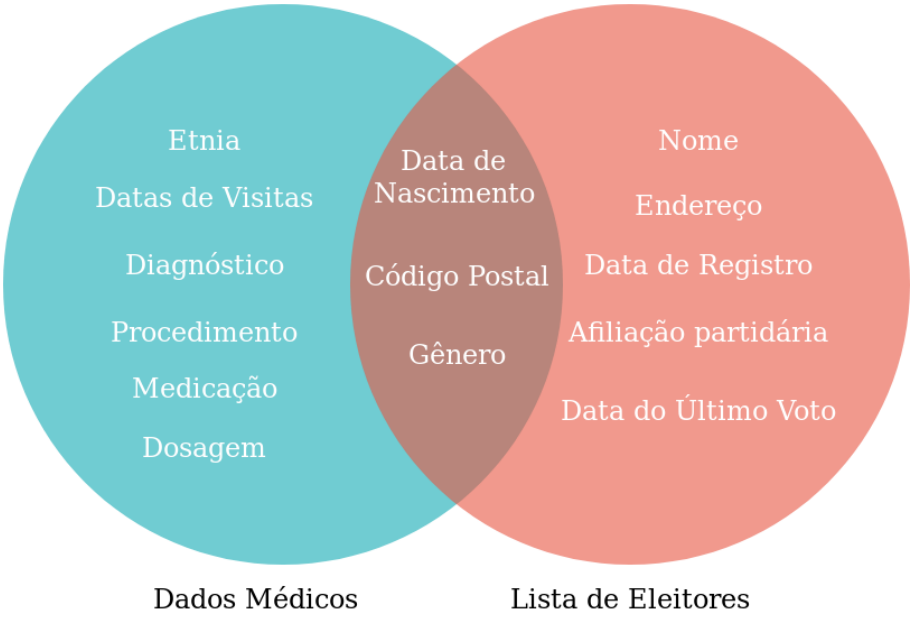


Figura 2.1: Diagrama de Venn apresentando o cruzamento entre os dados médicos e lista de eleitores da cidade de Cambridge em Massachusetts (adaptado de [1]).

2.1.3 Métodos de anonimização

Existem diversos métodos para anonimizar um banco de dados [10]. Eles podem ser usados individualmente ou em conjunto, de forma que um complemente as possíveis falhas dos outros. Entre eles temos a **supressão**, onde determinada informação é removida dos dados; a **generalização**, onde a informação é generalizada para um determinado intervalo e assim ser agrupada (por exemplo, transformar idades em intervalos de idade); a **perturbação**, em que são adicionados ruído aos dados; e a **permutação**, onde informações sensíveis são trocadas entre as entidades. As especificidades de cada método serão abordadas nesta Seção.

Os métodos de generalização e supressão, ao contrário dos outros métodos, preservam a integridade da informação. Isso ocorre já que não são adicionados ruídos ou trocas de valores entre indivíduos, o que resultaria em informações falsas, porém com o benefício da anonimidade, sendo necessário levantar o custo-benefício em que as metodologias utilizadas resultariam.

2.1.4 k-anonimidade

Uma metodologia-base da anonimização é a **k-anonimidade** [1].

Seja T uma tabela e QI_T o quasi-identificador referente a ela. T satisfaz a k-anonimidade se e somente se cada sequência de valores em $T[QI_T]$ aparece em pelo menos k ocorrências em $T[QI_T]$.

Tabela 2.1: Exemplos de domínios de um atributo de estado civil do indivíduo (adaptado de [1]).

	Nível Z_0	Nível Z_1	Nível Z_2
1	Casado	Esposo presente	*
2	Divorciado	Esposo não presente	*
3	Nunca casou	Esposo não presente	*
4	Separado	Esposo não presente	*
5	Viúvo	Esposo não presente	*

Tabela 2.2: Exemplo de k-anonimidade, onde $k = 2$ e QI = {Plano de saúde, Data de nascimento, Gênero, Código postal} (adaptado de [1]).

	Plano de saúde	Ano de nascimento	Gênero	Código postal	Problema
1	Público	1965	M	0214*	Respiração ofegante
2	Público	1965	M	0214*	Dor no peito
3	Público	1965	F	0213*	Hipertensão
4	Público	1965	F	0213*	Hipertensão
5	Público	1964	F	0213*	Obesidade
6	Público	1964	F	0213*	Dor no peito
7	Particular	1964	M	0213*	Dor no peito
8	Particular	1964	M	0213*	Obesidade
9	Particular	1964	M	0213*	Respiração ofegante
10	Particular	1967	M	0213*	Dor no peito
11	Particular	1967	M	0213*	Dor no peito

Por exemplo, a Tabela 2.2 satisfaz a condição de k-anonimidade com $k = 2$ e QI_T = {Plano de saúde, Ano de nascimento, Gênero, Código postal}. Então, para cada tupla da Tabela T, os valores dos atributos do quasi-identificador aparecem no mínimo duas vezes. Ou seja, para cada sequência de valores em $T[QI_T]$ existem pelo menos duas ocorrências desses valores em $T[QI_T]$. Na tabela dada como exemplo, onde $\{t1, t2, ..., t11\}$ representam as tuplas de T, $t1[QI_T] = t2[QI_T]$, $t3[QI_T] = t4[QI_T]$, temos que $t5[QI_T] = t6[QI_T]$, $t7[QI_T] = t8[QI_T] = t9[QI_T]$ e $t10[QI_T] = t11[QI_T]$.

Em uma tabela que satisfaz a propriedade de k-anonimidade, cada valor associado a um atributo de QI aparece pelo menos k vezes. No caso da Tabela 2.2, temos que $|T$[Plano de saúde = "Público"]$|$ = 6, $|T$[Plano de saúde = "Particular"]$|$ = 5, $|T$[Ano de nascimento = "1964"]$|$ = 5, $|T$[Ano de nascimento = "1965"]$|$ = 4, $|T$[Ano de nascimento = "1967"]$|$ = 2, $|T$[Gênero = "M"]$|$ = 6, $|T$[Gênero = "F"]$|$ = 5, $|T$[Código postal = "0213*"]$|$ = 9 e $|T$[Código postal = "0214*"]$|$ = 2.

Com o objetivo de divulgar a informação ao mesmo tempo que protege a privacidade do indivíduos, existem diversos processos para, a partir de uma tabela, encontrar sua versão que satisfaz a propriedade da k-anonimidade [13]. No sistema relacional de banco de dados, domínios são usados para descrever os valores que um atributo pode assumir. Alguns exemplos são os domínios numéricos, de string e código postal. Dizemos que o valor que o atributo possui na tabela original é seu valor de domínio base, Z_0. O objetivo de generalizar é transformar o valor

do atributo em algo menos significativo, ou seja, transformar algo do nível de domínio Z_0 em Z_1. Um exemplo seria transformar o último dígito de um código postal em 0, como $02139 \rightarrow 02130$. Outra mudança de domínio seria transformar um valor exato em um intervalo. Por exemplo, ao invés de divulgar a idade 26 anos, divulgar que o valor está no intervalo [20,30]. Ou seja, a transformação de Z_0 em Z_1 é $26 \rightarrow [20, 30]$. O ato de generalizar os dados facilita o pareamento de indivíduos para a obtenção de uma tabela k-anonimizada. Podem ser criados outros domínios ainda mais generalizados para cada valor, gerando assim uma hierarquia de domínios. A Tabela 2.1 contém exemplos de uma hierarquia de domínios envolvendo o estado civil dos indivíduos, onde conforme o valor nível do domínio aumenta, o valor do atributo está mais generalizado.

2.2 APRENDIZADO DE MÁQUINA

2.2.1 Conceitos básicos

De acordo com Mitchell (1997), um algoritmo de aprendizado de máquina é um algoritmo que é capaz de aprender a partir de dados. É dito que um algoritmo é capaz de aprender com uma experiência E a respeito de alguma classe de tarefas T com a performance medida por P se: a performance P de T melhora com a experiência E [14].

As **tarefas** descrevem como um sistema de aprendizado de máquina deveria processar um exemplo. Um exemplo é uma coleção de **características**, medidas quantitativamente, de algum objeto que queremos que o sistema de aprendizado de máquina processe. Um exemplo é matematicamente representado como um vetor $x \in \mathbb{R}$ onde cada entrada corresponde a uma característica diferente.

Para avaliar o algoritmo de aprendizado de máquina, devemos medir quantitativamente sua **performance**. Geralmente, essa medida de performance P é específica à tarefa T. Para tarefas como classificação, podemos medir, por exemplo, a acurácia do modelo. Acurácia é a proporção de exemplos para o qual o sistema produz a saída correta. Nós também podemos medir a taxa de erros, a proporção de exemplos para o qual o sistema produz a saída incorreta. Diferentes tarefas exigem diferentes medidas de performance. Geralmente estamos interessados em como o algoritmo vai se sair no mundo real, para isso utilizamos um conjunto de teste com dados separados dos presentes no conjunto de treino para avaliar a performance do algoritmo.

Algoritmo de aprendizado de máquina são categorizados como **supervisionados** ou **não supervisionados**, de acordo com as experiências que eles podem ter durante o processo de aprendizado.

Algoritmos de aprendizado não supervisionados experienciam um conjunto de dados e aprendem propriedades a partir disso, como, por exemplo, sua distribuição de probabilidade. Outra metodologia é agrupar os dados semelhantes em agrupamentos, onde os exemplos de cada agrupamento possuem propriedades em comum.

Algoritmos de aprendizado supervisionados experienciam um conjunto de dados contendo características, mas cada exemplo está associado a uma classe resultante. Por exemplo, um conjunto de dados de plantas poderia possuir a espécie anotada de cada planta.

Ou seja, algoritmos não supervisionados envolvem observar vários exemplos de um vetor e tentar aprender a distribuição de probabilidades $p(x)$ ou propriedades dessa distribuição; enquanto algoritmos de aprendizado supervisionados envolvem associar vários exemplos de um vetor x com seus respectivos alvos em y e criar estimativas $p(x|y)$. Com isso, nós podemos transformar problemas supervisionados em não supervisionados e vice-versa.

Uma forma de descrever um conjunto de dados é com uma matriz, onde cada exemplo é descrito em uma linha e cada coluna corresponde a uma característica diferente. Um problema dessa representação é quando exemplos possuem tamanhos diferentes.

Exemplos de algoritmos de aprendizado de máquina são floresta aleatória, redes neurais e máquina de vetores de suporte.

2.2.2 Padronização

Padronização é um método estatístico para transformar um conjunto de dados de forma que eles possam ser comparados uns aos outros. A padronização de uma amostra x é calculada com a fórmula:

$$z = \frac{x_i - \mu}{\sigma}$$

Nessa equação, μ corresponde a média das amostras e σ à variância.

A padronização de um conjunto de dados é um requisito comum para vários algoritmos de aprendizado de máquina [15]. Isso se deve a alguns algoritmos não alcançarem uma boa performance quando as características não estão em uma distribuição normal. Devido a esse fator, é comum padronizarmos as características antes de utilizarmos um algoritmo de aprendizado de máquina. Um exemplo disso são alguns tipos de *kernel* de máquinas de vetores suporte, que assumem que as características estão centralizadas em torno de 0 e a que a variância está na mesma ordem. Se uma característica tem variância em uma ordem maior que a magnitude das outras, o estimador pode não conseguir aprender a tarefa corretamente.

2.2.3 Validação cruzada

Validação cruzada é um tipo de técnica baseada em repetir as etapas de treino e teste computacional em diferentes subconjuntos de dados escolhidos aleatoriamente ou como partições do conjunto de dados original. O objetivo de seu uso é estimar como a acurácia de um modelo de predição seria na prática. A validação cruzada também diminui a incerteza sobre a taxa de erro do teste [14].

O tipo de validação cruzada mais comum é a k**-fold**, onde o conjunto de dados é dividido em k partições distintas e os dados são treinados e testados k vezes. Com isso, a cada iteração, $k - 1$ partições são usadas para treino e uma para teste. Cada iteração desse processo gera uma acurácia diferente. A acurácia média dessas iterações é chamada de acurácia da validação cruzada.

2.2.4 TF-IDF

O valor ***Term Frequency – Inverse Document Frequency (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 [16].

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:

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

Com isso, é criada uma tabela que associa a medida TF-IDF de cada documento em relação a cada termo, como no exemplo da tabela 4.1.

Tabela 2.3: Exemplo de tabela com os valores TF-IDF de acordo com cada documento e termo (autoria própria).

Documentos	**"abacaxi"**	**"banana"**	**"uva"**
Documento 1	0,1	0	0,15
Documento 2	0	0	1,0
Documento 3	0,5	0,3	0

Valores TF-IDF podem ser utilizados como características em algoritmos de aprendizado de máquina.

2.2.5 Máquina de vetores de suporte

As **máquinas de vetores de suporte** (SVM - *Support Vector Machine*), são um algoritmo de aprendizado de máquina para a classificação de amostras [17]. Nesse método, vetores de entrada que representam dados são mapeados em um espaço multidimensional de vetores de características.

O objetivo no espaço de características é traçar um hiperplano que separa os diferentes vetores de características. Esse hiperplano é chamado de **vetor de suporte**. A partir da posição de um vetor de características em relação a esse hiperplano separador, é possível identificar à qual classe o dado relacionado a aquele vetor pertence.

A identificação de classes em espaços diferentes se deve aos princípios das operações escalares nos vetores do espaço de características poder ser transportado para operações no espaço de dados.

Uma variação do SVM é o **Agrupamento por Vetores de Suporte** (SVC - *Support Vector Clustering*). O objetivo do método é encontrar a região onde estão agrupados os vetores de características de uma determinada classe [18].

2.2.6 Floresta aleatória

Bagging [19] é um método proposto por Breiman em 1996, que consiste em gerar um conjunto de modelos utilizando um algoritmo de aprendizagem que realiza a classificação por combinação de votos, ou seja, vários modelos juntos realizam a classificação, seja pela média de seus resultados ou votação da maioria destes. *Bootstrapping* é uma técnica de reamostragem que consiste em obter um novo conjunto de dados, a a partir da reamostragem dos dados originais, a fim de avaliar a variabilidade das quantidades de interesse [20].

Floresta aleatória [21] é um modelo de aprendizado supervisionado. Ele é uma variação substancial do método de *bagging* com elementos do *bootstrapping* em que se contrói um conjunto de árvores de decisão e utiliza a média de suas saídas ou classe resultante da maioria das árvores para obter um resultado final [22].

Algoritmo 1 Criação de floresta aleatória para classificação

1: **for** $b = 1$ até B **do**
2: Obtenha uma amostra *bootstrap* de tamanho N do conjunto de treino.
3: Crie uma árvore T_b da amostra *boostrap*, repetindo recursivamente os seguintes passos em cada nó terminal da árvore, até que o número mínimo de nós n_min seja alcançado:
4: 1. Selecione m variáveis aleatoriamente das p variáveis.
5: 2. Selecione a melhor variável para realizar a divisão no nó n.
6: 3. Divida o nó em dois nós filhos.
7: **end for**
8: Retorna o conjunto $T = T_b$ onde $b = 1..B$

Árvores são ótimas ferramentas para classificar os dados com os quais elas foram criadas. Elas possuem altas taxas de variância da função de predição e baixo valor de bias. Portanto, essas árvores seriam beneficiadas ao serem combinadas com elementos do *bagging*. Isso se deve à característica do *bagging* de, a partir de sua filosofia da obtenção de obter maior eficácia com a coletividade de modelos, reduzir essa taxa de variância. Florestas aleatórias combinam a simplicidade das árvores com a flexibilidade do *bagging*, assim gerando uma melhoria na acurácia. A variedade de árvores é o que concede às florestas aleatórias maior eficácia se comparada às árvores de decisão individuais.

A partir do algoritmo de criação de floresta aleatória mostrado nessa Seção, podemos utilizar esse método para realizar a predição em uma nova amostra x. Para isso, a amostra é inserida como entrada nas múltiplas árvores e a classe resultante retornada pela maioria destas árvores é tida como resultado final da floresta aleatória.

Floresta aleatória é um método simples para se treinar e realizar o ajuste de parâmetros, o que influencia em sua popularidade e implementação em diversos pacotes.

2.2.7 Redes neurais

Redes neurais são um modelo matemático cujo método de computar elementos é originário de um modelo biológico: o cérebro humano. Dessa forma, a computação de elementos através da aritmética corresponde aos neurônios do cérebro processando informações e enviando sinais uns aos outros, trabalhando em conjunto, assim formando uma rede neural [23].

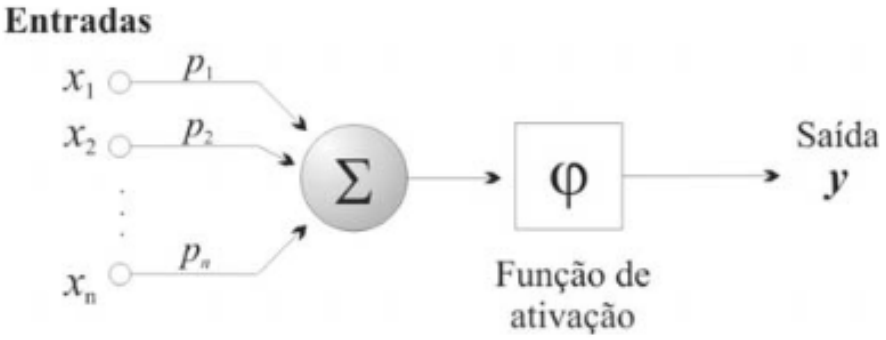


Figura 2.2: Modelo matemático de um neurônio [2].

A Figura 2.2 mostra uma versão simplificada de um neurônio artificial [2]. Ele é composto por:

- Um conjunto de n conexões de entrada $(x_1, x_2, ..., x_n)$, com um conjunto correspondente de pesos $(w_1, w_2, ..., w_n)$;

- Um somador ($\sum$) para agregar os sinais de entrada;
- Uma função de ativação (ϕ) que limita o intervalo permissível de amplitude do sinal de saída (y) a um valor fixo.

Cada unidade recebe o vetor x como entrada e realiza operações com seu vetor de pesos w conforme e equação:

$$f(x, w) = \sum_{i=1}^{n} x_i.w_i + b$$

Depois disso, o resultado passa por uma função de ativação. Ela define como deve ser a saída da unidade de acordo com uma função pré-definida. Sua fórmula pode ser definida como:

$$y = \phi(f(x, w)) = \phi(\sum_{i=1}^{n} x_i.w_i + b)$$

Existem várias de funções de ativação diferentes, cada uma sendo mais efetiva para um tipo de dado ou classificador específico. Um exemplo é a **ReLU**, ou unidade linear rectificada, que pode ser definida pela seguinte fórmula:

$$\phi(x) = max(0, x)$$

A ReLU é uma função de ativação muito utilizada no projeto de redes neurais. Isso se deve ao fato de ela não ativar todos os neurônios ao mesmo tempo, já que valores negativos serão convertidos a zero e, portanto, não serão ativados. Essa fator torna a rede esparsa, eficiente e de fácil computação [24].

Uma rede neural é composta por neurônios artificiais (também chamados de nós ou unidades) interconectados. Elas podem ser utilizadas tanto para aprendizado supervisionado quanto para não supervisionado. Um fator que diferencia as redes neurais é sua arquitetura, ou seja, a quantidade de neurônios presentes e a forma como estão conectados. Um exemplo disso é quantas camadas, conjuntos de valores emitidos a partir da conexão de entrada e seus pesos, que a rede possui e como estão divididas. A Figura 2.3 mostra um exemplo de arquitetura de rede.

Uma rede neural possui três tipos de camada: a **camada de entrada**, no qual os dados são inseridos, é representada pela camada onde o vetor x está diretamente conectado na Figura 2.3; a **camada de saída**, que é a última camada da rede e a que retorna o resultado das operações realizadas; e as **camadas ocultas**, que são camadas intermediárias, entre a camada de entrada e a de saída, e cuja sua quantidade pode variar dependendo da definição da arquitetura. Além disso, as conexões entre nós podem ser do tipo *feedback*, onde camadas possuem laços retroalimentando camadas anteriores, ou *feedforward*, onde esses laços não ocorrem.

O método tradicional para treinamento de redes neurais *feedforward* é chamado de **backpropagation**. Seu intuito é modificar os pesos dos neurônios da rede neural com o objetivo de minimizar os erros das saídas em relação às saídas esperadas correspondentes. A partir deste algoritmo de aprendizado supervisionado a rede pode corrigir os erros cometidos na classificação através de uma metodologia que utiliza conceitos de cálculo diferencial integral.

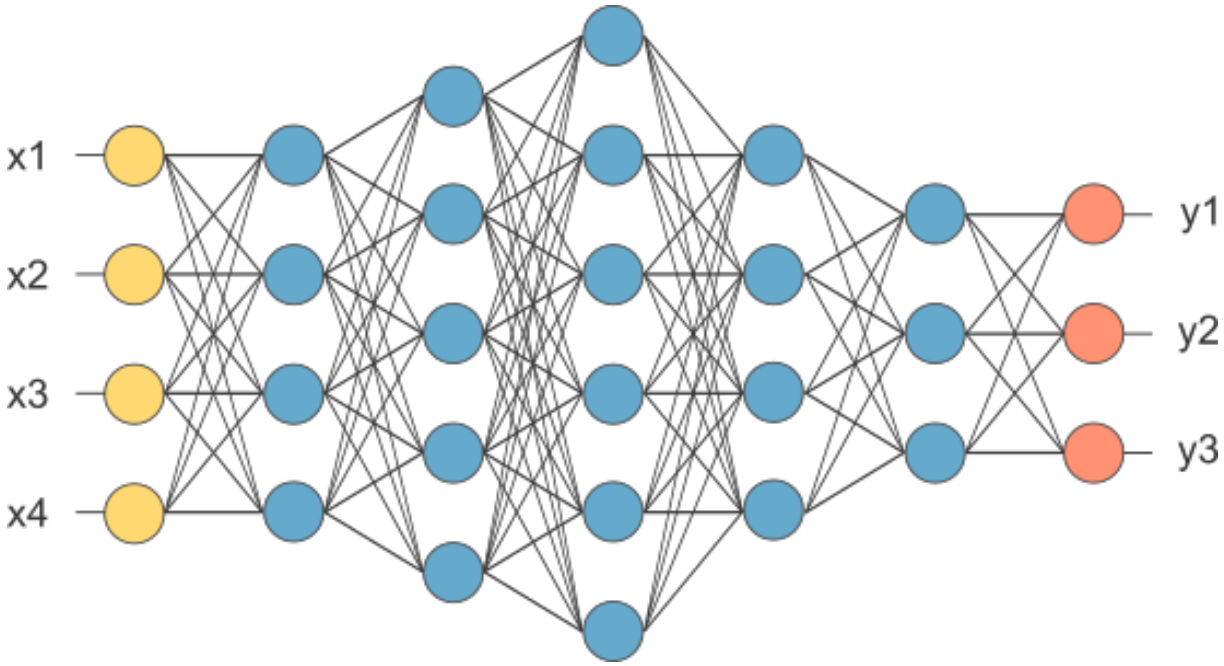


Figura 2.3: Exemplo de rede neural [3].

3 TRABALHOS RELACIONADOS

Apesar do uso de aprendizado de máquina para anonimização de dados ser uma área pouco explorada [25], existem alguns estudos que abordam este ponto. Este Capítulo apresenta trabalhos relacionados, que representam o estado da arte em que se encontram métodos que auxiliam na automatização da anonimização de bancos de dados. O primeiro se trata de um artigo com a proposta semelhante a este trabalho: identificar com aprendizado de máquina atributos relacionados a dados médicos. O segundo e terceiro trabalho são softwares relevantes para o estado da arte da anonimização de dados.

3.1 ANONYMIZATION OF SENSITIVE INFORMATION IN MEDICAL HEALTH RECORDS

Devido à privacidade dos pacientes, as informações sensíveis relacionadas à saúde não podem ser diretamente compartilhadas. Exemplos de dados são nome, sobrenome, endereços, hospitais, profissões, cobranças e telefones. Foi realizado um experimento com o intuito de identificar esses dados em um conjunto. Os dados estão em espanhol. O conjunto de dados do experimento é constituído de:

Tabela 3.1: Tamanhos dos conjuntos de teste, treino e validação do experimento, onde os valores correspondem ao número de pacientes e seus dados (adaptado de [4]).

Conjunto	**Treino**	**Teste**	**Validação**
Tamanho	8300	3231	4048

Cada amostra contém os atributos correspondentes a um paciente, como nome, endereço e telefone. No total foram 44 tipos de atributos. Para classificar esses dados, foram treinados dois modelos: um com máquina de vetores de suporte (*Support Vector Machines*, SVM) e outro com redes neurais recorrentes (RNR).

A extração de características é executada através de janelas, onde cada caracter é analisada a janela [-1, 0, 1, 2], e, a partir, disso, criado um vetor de características.

Os modelos alcançaram as seguintes pontuações de acurácia:

Tabela 3.2: Valores de acurácia do experimento, para os modelos treinados com SVM e RNR (adaptado de [4]).

Modelos	**Acurácia de testes**	**Acurácia de validação**
SVM	0,891	0,886
RNR	0,957	0,960

Apesar de ambos os modelos apresentarem bons resultados, o de RNR alcançou maior taxa de acurácia.

3.2 PSYNDB: ACCURATE AND ACCESSIBLE PRIVATE DATA GENERATION

Através de múltiplos domínios de aplicações, as entidades que coletam informações sensíveis precisam de mecanismos para disseminá-las ao público. Uma abordagem para tal seria gerar dados sintéticos, ou anonimizados, onde obteríamos um conjunto de dados similar ao

original, ao mesmo tempo que mantemos suas propriedades estatísticas, sem revelar informações sensíveis de indivíduos presentes nos dados. [26]

Com isso, os autores desenvolveram o PSynDB, uma plataforma web de geração de tabelas de dados sintéticos. Essas tabelas devem satisfazer garantias formais de privacidade diferencial e prover alta acurácia para as análises que o indivíduo deseja realizar com os dados. A plataforma permite que o usuário visualize as taxas de erro esperadas para assim tomar decisões preventivas, como ajustar a perda de taxa de privacidade adequada.

Para configurar a criação da TA do PSynDB, o usuário preenche na ferramenta qual é a definição do esquema a ser usado. O usuário pode definir o esquema da tabela manualmente ou por realizar o upload do arquivo de dados (em formato *csv*). Isso permite que os dados sejam visualizados e modificados no próximo passo.

O usuário também pode realizar a definição de domínios. Neste passo o usuário define os domínios que deverão estar disponíveis na saída deste processo. Por exemplo, se a saída desejada é uma relação dos ganhos financeiros de acordo com cada combinação de idade, raça e gênero, o usuário deve definir os domínios de atributos de {Idade, Ganho financeiro, Raça, Gênero}. Atributos numéricos serão discretizados em casos onde o usuário define um piso e teto para esses valores.

A plataforma possui ferramentas para tabela anonimizada criada como saída seja diretamente inserida em um ambiente seguro, como é recomendado pelos autores.

3.3 ARX - DATA ANONYMIZATION TOOL

ARX é uma ferramenta de código aberto para anonimizar dados sensíveis. Ela tem suporte para uma grande variedade de métodos de privacidade, métodos de transformação de dados e análise dos dados de saída. Ela é dividida em duas vertentes: uma plataforma gráfica que suporta a importação e curadoria de dados, é intuitiva para ajustar a anonimização de dados e visualizar os resultados; e uma API em Java para fornecer essas funcionalidades para um programa do usuário. A plataforma tem sido usada em vários contextos incluindo uso comercial, científico e clínico.

A plataforma permite definir diversos níveis de domínios de cada atributo, como por exemplo, generalização de idade, estado civil, etc. Também permite selecionar o valor de k na k-anonimização, quais atributos fazem parte do quasi-identificador, quais são sensíveis, análises de riscos e a saída anonimizada.

3.4 DISCUSSÃO

Apesar de poucos estudos relacionando anonimização de dados e aprendizado de máquina, o artigo Anonymization of Sensitive Information in Medical Health Records tem uma proposta semelhante a este trabalho. Entre as diferenças estão a língua em que os dados estão, o espanhol, os algoritmos usados para treinar o modelo e a quantidade de atributos estudados. Os resultados são bons devidas as proporções do experimento, principalmente com o uso de RNR.

As ferramentas PSynDB e ARX possuem métodos bastante completos para a anonimização de dados. Elas possuem algoritmos comparáveis ao estado da arte na área de estudo. Porém, ambas são bastante complexas para o usuário médio. A ferramenta PSynDB tem a proposta de ser mais acessível, e, comparando com a ARX, ela cumpre esse papel. Porém, o usuário ainda deve inserir todo o esquema do banco de dados manualmente e identificar o conjunto de quasi-identificadores. Esse ponto pode ser melhorado para tornar a ferramenta ainda mais acessível.

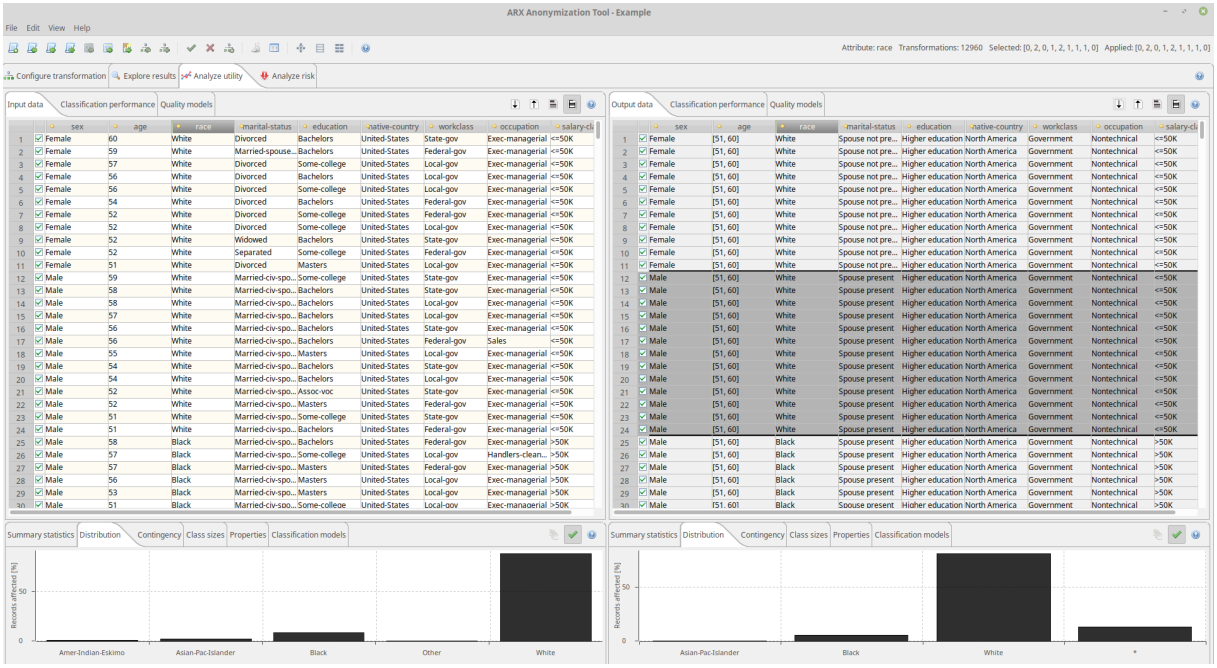


Figura 3.1: Captura de tela da ferramenta ARX, com os dados brutos ao lado esquerdo a k-anonimizados a direita (autoria própria).

4 PROPOSTA E METODOLOGIA

Existem diversas ferramentas de anonimização disponíveis no mercado. Porém, as mais completas também são complexas para o usuário médio. Com esse intuito, nesse Capítulo abordaremos uma proposta para minimizar essa complexidade.

4.1 DEFINIÇÃO DO PROBLEMA

Apesar das ferramentas de anonimização disponíveis no mercado possuírem uma grande variedade de recursos, como suporte à vários métodos de anonimização baseados na *k*-anonimidade, elas não são práticas para o usuário médio. Isso se deve à grande variedade de metodologias disponíveis para anonimizar e, também, pela complexidade em classificar manualmente o quão sigiloso é cada atributo, como forma de realizar o pré-processamento para um método de anonimização específico.

Um exemplo dessa dificuldade é que, caso o usuário insira seu banco de dados, terá que classificar manualmente como identificador ou *quasi*-identificador todos os atributos de seu banco. O que pode ser cada vez menos prático de acordo com o tamanho do banco e número de atributos. Além disso, dependendo da metodologia de anonimização, o usuário deve inserir a que cada atributo se refere. Por exemplo, anonimizar uma data pode ser diferente de anonimizar um nome, já que no primeiro caso pode ser mantido apenas o ano e no segundo apenas a primeira letra. Outra dificuldade é o cruzamento de dados, que pode ser realizado quando um indivíduo possui acesso a bancos distintos que possuem dados relacionados entre si. Por exemplo, se um banco contém nomes e datas de nascimento de cada pessoa, e outro banco possui datas de nascimento e diagnóstico médico de um subconjunto dos mesmos indivíduos, é possível realizar operações para tirar a anonimidade dos dados, e, assim, descobrir o diagnóstico dos pacientes. Para avaliar o quão anônimo é um banco, são analisados critérios relacionados à *k*-anonimidade e métricas relacionadas a esta.

Considerando esses fatores, é de grande utilidade a habilidade de classificar automaticamente qual é a informação contida no atributo em questão, e, também, seu grau de sigilo.

4.2 PROPOSTA

Levando em conta a definição do problema, foi criada uma arquitetura para utilizar diferentes técnicas na classificação de registros sensíveis em banco de dados, ou seja, identificar qual tipo de informação está contido em cada atributo de uma tabela. Essa arquitetura é apresentada na Figura 4.1.

A partir disso, analisaremos cada etapa do processo.

4.2.1 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) [27] é um projeto firmado entre o Centro de Computação Científica e Software Livre (C3SL) e o Ministério da Saúde. O objetivo do projeto é o monitoramento de equipamentos médicos. A ideia básica é auxiliar no bom uso do recurso público aplicado pelo

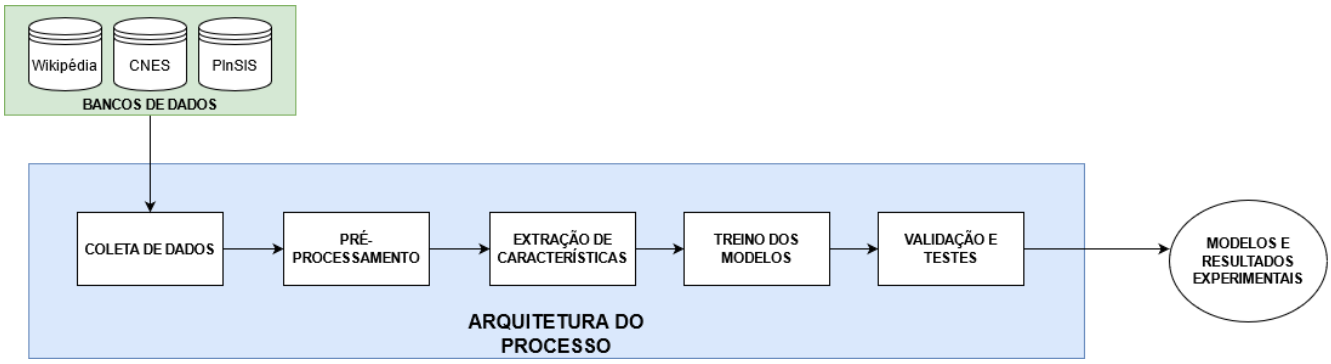


Figura 4.1: Arquitetura do processo proposto para identificação de registros sensíveis (autoria própria).

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). Devido a esse motivo, o PInSIS possui bancos de dados hospitalares brutos e anonimizados. Este experimento utiliza bancos de dados do PInSIS, para a realização de alguns experimentos iniciais.

- **Cadastro Nacional de Estabelecimentos de Saúde** (CNES): O Cadastro Nacional de Estabelecimentos de Saúde (CNES) [28] 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 principais atributos utilizados correspondem aos nomes dos profissionais e aos endereços dos estabelecimentos de saúde.

- **Wikipedia**: A Wikipedia [29] 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.

4.2.2 Pré-processamento

O pré-processamento consiste nas seguintes etapas:

- **Extração de atributos**: A partir dos arquivos CSV referentes aos bancos de dados do PInSIS e CNES, são utilizados comandos no terminal para a obtenção de apenas os dados referentes a nomes e endereços. 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 [30] 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.

- **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.

4.2.3 Extração de características

As características são extraídas com base no valor ***Term Frequency – Inverse Document Frequency (TF–IDF)*** de digramas.

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

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

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 4.1.

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 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 Figura 4.2 terá até 676 características.

Tabela 4.1: Representação ilustrativa da tabela de tamanho 150.000 × 676 com os valores TF-IDF de acordo com cada amostra e característica (autoria própria).

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

Tabela 4.2: Representação ilustrativa do vetor resultante da transformação de média que será realizada no experimento (autoria própria).

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

Os valores de TF-IDF foram calculados em Python 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, teremos os exemplos para serem utilizados no processo de classificação.

4.2.4 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 característica. Com isso, vamos discutir sobre qual dos modelos é mais viável para a aplicação desejada.

Com as características selecionadas e os dados normalizados, os modelos são treinados e testados utilizando três algoritmos diferentes. Os algoritmos utilizados para classificação são as implementações da biblioteca *sklearn* em Python para máquina de vetores de suporte (SVM), florestas aleatórias e redes neurais.

A máquina de vetores de suporte utilizada é do tipo SVC, o mais indicado para trabalhar com *clusters*. O *kernel* é linear. O SVC não possui número máximo de iterações, ou seja, para apenas na convergência do treino.

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 [31], 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 os definidos como padrão pelas bibliotecas. Isso foi tomado como estratégia com o intuito de realizar experimentos com medidas balanceadas e, em futuros experimentos, refinar os parâmetros para os classificadores com melhores resultados.

4.2.5 Validação e testes

Na última etapa é realizado o processo de validação de modelos treinados 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 utilizados 5 *folds* 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 *folds* foi escolhido arbitrariamente.

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

Essa questão é abordada em detalhes e discutida no próximo Capítulo.

4.2.6 Resultado do processo

Ao final do processo, obtemos os diferentes modelos treinados e seus resultados experimentais. Isso auxilia na busca pelo modelo mais eficiente para a resolução do problema apresentado. Esses fatores serão discutidos nos próximos Capítulos.

5 TESTES E RESULTADOS

Neste Capítulo serão apresentados os resultados experimentais sobre o problema de predição de classificação de registros sensíveis em bancos de dados. Este Capítulo está dividido em duas seções, onde a primeira se trada dos resultados experimentais do modelo de predição exposto na metodologia do trabalho. E na segunda temos uma discussão sobre os resultados obtidos.

5.1 RESULTADOS EXPERIMENTAIS

Nessa Seção apresentaremos os resultados experimentais do estudo.

5.1.1 Transformação de características

Para realizar as transformações de dados TF-IDF selecionamos a transformação por média. Com isso, a seguir está listado o vetor de características ordenado em ordem decrescente, com as mais relevantes no início e as menos relevantes no fim, utilizando a transformação proposta sobre a matriz TF-IDF:

```
['de' 'ra' 'an' 'ar' 'os' 'es' 'ua' 'ru' 'do' 'da' 'ma' 're' 'en' 'ri'
 'ca' 'er' 'ia' 'nt' 'co' 'as' 'al' 'to' 'na' 'ei' 'ro' 'or' 'in' 'te'
 'ta' 'ir' 'el' 'sa' 'on' 'li' 'ad' 'il' 've' 'se' 'st' 'is' 'va' 'io'
 'si' 'ci' 'me' 'ni' 'av' 'nd' 'no' 'la' 'em' 'ao' 'id' 'pa' 'ti' 'ne'
 'le' 'am' 'ic' 'lv' 'pe' 'so' 'lo' 'po' 'om' 'ue' 'ac' 'nc' 'qu' 'mo'
 'di' 'at' 'im' 'it' 'tr' 'ou' 'ol' 'ai' 'vi' 'ce' 'pr' 'jo' 'ba' 'dr'
 'br' 'ha' 'be' 'ss' 'um' 'rt' 'mi' 'gu' 'et' 'ec' 'go' 'un' 'ed' 'ul'
 'rr' 'iv' 'ui' 'ga' 'au' 'fe' 'ho' 'ge' 'ur' 'ch' 'za' 'ns' 'sc' 'fo'
 'us' 'lu' 'nh' 'oa' 'ie' 'lh' 'ab' 'he' 'od' 'su' 'ap' 'bo' 'ig' 'ag'
 'fi' 'tu' 'rn' 'pi' 'eu' 'uz' 'fr' 'mp' 'eg' 'oi' 'fa' 'rc' 'rd' 'oc'
 'ng' 'cr' 'ov' 'ea' 'iz' 'gr' 'gi' 'ut' 'mb' 'ja' 'ot' 'uc' 'ju' 'rg'
 'll' 'ev' 'vo' 'oe' 'rm' 'mu' 'bi' 'lm' 'du' 'ld' 'cu' 'ze' 'op' 'cl'
 'ib' 'ob' 'ez' 'ae' 'sp' 'rl' 'ip' 'az' 'hi' 'ex' 'og' 'ud' 'eo' 'lt'
 'ep' 'th' 'nu' 'ub' 'je' 'ix' 'rb' 'bu' 'pu' 'rs' 'rv' 'eb' 'ug' 'af'
 'zi' 'bl' 'aq' 'fl' 'tt' 'pl' 'of' 'if' 'ef' 'xa' 'xi' 'ka' 'sm' 'rq'
 'iu' 'fu' 'lb' 'uj' 'nn' 'up' 'ay' 'lc' 'ke' 'wa' 'ls' 'xe' 'sh' 'iq'
 'hu' 'nf' 'gl' 'km' 'nj' 'zo' 'aj' 'ct' 'oz' 'ly' 'xo' 'eq' 'lg' 'ya'
 'ck' 'nv' 'gn' 'ey' 'ki' 'ry' 'lf' 'wi' 'ii' 'nr' 'ej' 'sl' 'vn' 'we'
 'ax' 'nz' 'ah' 'ee' 'sq' 'oj' 'uv' 'xp' 'rp' 'nq' 'cc' 'ny' 'oo' 'ak'
 'bs' 'ko' 'uq' 'ts' 'xt' 'ox' 'ow' 'ph' 'sk' 'sd' 'ik' 'vr' 'ds' 'sb'
 'ff' 'ps' 'sf' 'pt' 'zz' 'dn' 'ys' 'nk' 'yo' 'dm' 'lz' 'oy' 'zu' 'rf'
 'dy' 'xv' 'pc' 'mm' 'hr' 'qd' 'oh' 'cn' 'gh' 'rk' 'yr' 'yn' 'uf' 'uo'
 'tl' 'pp' 'ku' 'cy' 'tz' 'hl' 'hn' 'ij' 'ye' 'kl' 'ty' 'tv' 'yl' 'wo'
 'ux' 'oq' 'ew' 'sv' 'sr' 'ok' 'gt' 'sg' 'aw' 'lp' 'my' 'xc' 'tc' 'aa'
 'bt' 'ht' 'ek' 'dv' 'rz' 'lq' 'ji' 'ks' 'sn' 'uk' 'kr' 'sy' 'sw' 'hm'
 'tm' 'bj' 'dt' 'cs' 'dj' 'ms' 'yc' 'gg' 'lk' 'yu' 'ln' 'hy' 'tn' 'nb'
 'eh' 'ft' 'cm' 'ws' 'bb' 'vu' 'by' 'dd' 'gs' 'ky' 'ym' 'nl' 'yd' 'dg'
 'gm' 'wn' 'kh' 'yt' 'rh' 'nm' 'rj' 'bd' 'uy' 'mc' 'jk' 'xu' 'cd' 'tw'
 'qn' 'dw' 'xx' 'dl' 'uh' 'wh' 'df' 'dh' 'cq' 'gd' 'cj' 'hs' 'kn' 'iy'
 'ih' 'mt' 'hc' 'vs' 'cz' 'bc' 'bm' 'db' 'yp' 'fm' 'jr' 'fc' 'hw' 'tp'
 'gy' 'wr' 'mn' 'tb' 'vy' 'bn' 'rw' 'mg' 'py' 'dc' 'dq' 'pm' 'gb' 'qi'
 'mf' 'mh' 'gc' 'vd' 'vl' 'xs' 'qs' 'lr' 'zy' 'fs' 'ml' 'wl' 'cb' 'iw'
 'md' 'hd' 'yg' 'yi' 'pf' 'lj' 'sz' 'nw' 'zm' 'kg' 'pn' 'ww' 'cp' 'wt'
 'bh' 'tk' 'np' 'kk' 'tf' 'kt' 'zh' 'yw' 'yb' 'mr' 'bg' 'wy' 'hb' 'wu'
```

```
'pd' 'tg' 'yk' 'yv' 'kw' 'bp' 'fg' 'nx' 'zl' 'yz' 'pk' 'bf' 'td' 'lw'
'gp' 'cg' 'uw' 'uu' 'zk' 'qa' 'mw' 'zb' 'pg' 'vp' 'dz' 'zw' 'wk' 'kf'
'cf' 'kb' 'pb' 'fb' 'vc' 'hz' 'jj' 'mv' 'jh' 'hk' 'bv' 'rx' 'cv' 'gw'
'gf' 'hf' 'mk' 'fy' 'wc' 'jd' 'fp' 'hv' 'zc' 'fn' 'mj' 'hp' 'vg' 'xy'
'xb' 'yf' 'qr' 'kd' 'sj' 'dk' 'tj' 'xf' 'vm' 'wb' 'bk' 'zq' 'cw' 'xm'
'xw' 'yh' 'hh' 'jm' 'zd' 'jn' 'zn' 'fh' 'fd' 'vh' 'zg' 'zt' 'bw' 'dp'
'vf' 'gk' 'kv' 'jp' 'jc' 'pq' 'ql' 'kp' 'vb' 'pz' 'js' 'pv' 'tx' 'kc'
'zs' 'zr' 'hq' 'qc' 'qe' 'vj' 'wm' 'jb' 'wf' 'qm' 'yy' 'fk' 'wd' 'jl'
'pw' 'xl' 'vt' 'mz' 'wp' 'gx' 'zv' 'jt' 'xd' 'yj' 'xh' 'gv' 'hg' 'fj'
'jg' 'kj' 'qq' 'gz' 'gj' 'qp' 'kz' 'yx' 'lx' 'jf' 'zf' 'bx' 'sx' 'vk'
'vz' 'cx' 'fx' 'pj' 'hj' 'zp' 'jv' 'wg' 'fw' 'fz' 'bz' 'mx' 'xr' 'dx'
'vv' 'fv' 'bq' 'wv' 'tq' 'qw' 'jy' 'mq' 'vw' 'qo' 'xk' 'qt' 'xn' 'jw'
'wz' 'hx' 'fq' 'qy' 'gq' 'qb' 'jq' 'px' 'yq' 'qg' 'qk' 'jz' 'qv' 'qf'
'xz' 'qx' 'vx' 'vq' 'zx' 'jx' 'kq' 'wj' 'wq' 'qz' 'xj' 'kx' 'wx' 'zj'
'qj' 'qh' 'xg' 'xq']
```

5.1.2 Classificação

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 modulos com um conjunto de testes independente, com isso obtivemos as acurácias de testes e suas matrizes de confusão. As matrizes de confusão são métricas para a visualização da proporção de erros e acertos de determinada classe.

A Tabela 5.1 mostra os resultados de acurácia de validação cruzada obtidos. A Tabela 5.2 mostra os valores de desvio-padrão obtidos. A Tabela mostra os valores de acurácia de teste. As Figuras 5.1, 5.2 e 5.3 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.

Tabela 5.1: Valores da acurácia de validação cruzada de acordo com os algoritmos de aprendizado de máquina estudados e o número de características escolhido (autoria própria).

Algoritmo utilizado / **Número de características**	**SVM**	**Floresta Aleatória**	**Rede Neural**
5	0.5760	**0.6559**	0.6473
10	0.7367	**0.8015**	0.7994
20	0.7947	**0.8728**	0.8673
40	0.8520	**0.9146**	0.9098
85	0.9185	0.9467	**0.9503**
169	0.9378	0.9545	**0.9619**
338	0.9466	0.9537	**0.9681**
676	0.9469	0.9528	**0.9675**

Tabela 5.2: Valores de desvio-padrão da validação cruzada de acordo com os algoritmos de aprendizado de máquina estudados e o número de características escolhidas (autoria própria).

Algoritmo utilizado / **Número de características**	**SVM**	**Floresta Aleatória**	**Rede Neural**
676	$2,21 \times 10^{-}3$	$6,60 \times 10^{-}4$	$6,68 \times 10^{-}4$
338	$2,25 \times 10^{-}3$	$1,35 \times 10^{-}3$	$6,29 \times 10^{-}4$
169	$2,40 \times 10^{-}3$	$1,08 \times 10^{-}3$	$1,36 \times 10^{-}3$
85	$2,84 \times 10^{-}3$	$1,38 \times 10^{-}3$	$7,02 \times 10^{-}4$
40	$1,90 \times 10^{-}3$	$5,69 \times 10^{-}4$	$2,29 \times 10^{-}3$
20	$9,08 \times 10^{-}4$	$2,13 \times 10^{-}3$	$2,21 \times 10^{-}3$
10	$3,27 \times 10^{-}3$	$3,10 \times 10^{-}3$	$2,53 \times 10^{-}3$
5	$1,68 \times 10^{-}3$	$3,03 \times 10^{-}3$	$3,63 \times 10^{-}3$

Tabela 5.3: Valores da acurácia de de teste de acordo com os algoritmos de aprendizado de máquina estudados e o número de características escolhido (autoria própria).

Algoritmo utilizado / **Número de características**	**SVM**	**Floresta Aleatória**	**Rede Neural**
5	0.5847	**0.6585**	0.6523
10	0.7464	**0.8089**	0.8066
20	0.8026	**0.8783**	0.8707
40	0.8601	**0.9208**	0.9150
85	0.9226	0.9518	**0.9532**
169	0.9417	.0.9579	**0.9654**
338	0.9495	0.9574	**0.9711**
676	0.9499	0.9567	**0.9715**

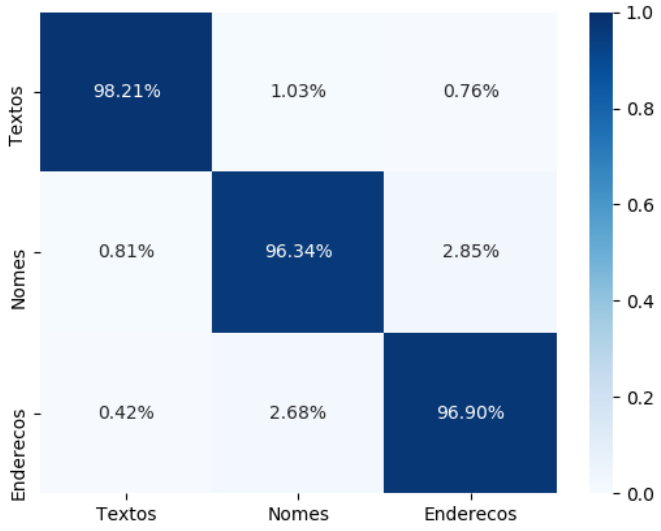


Figura 5.1: Matriz de confusão de validação cruzada (onde o eixo X indica as classes preditas e o eixo Y as classes reais) para o modelo de rede neural treinado com 676 características (autoria própria).

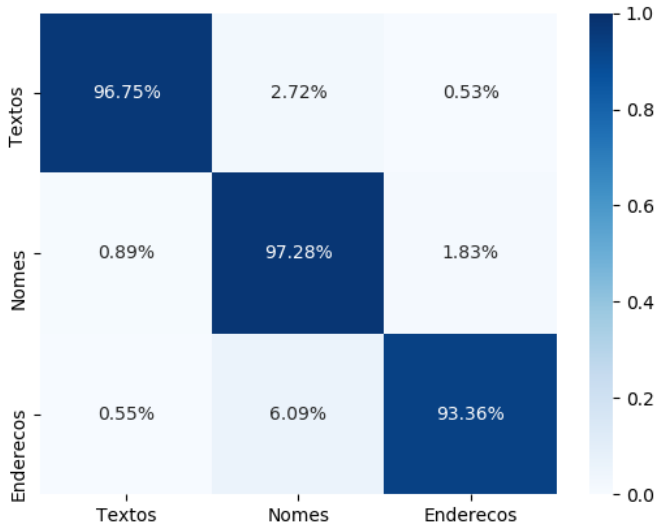


Figura 5.2: Matriz de confusão de validação cruzada (onde o eixo X indica as classes preditas e o eixo Y as classes reais) para o modelo de floresta aleatória treinado com 169 características (autoria própria).

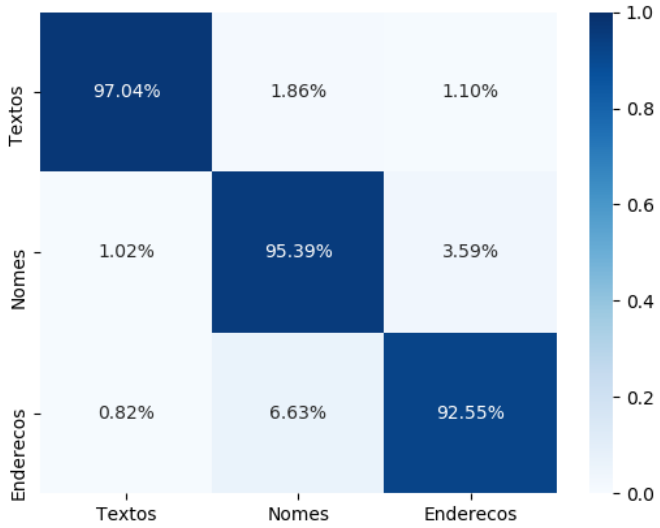


Figura 5.3: Matriz de confusão de validação cruzada (onde o eixo X indica as classes preditas e o eixo Y as classes reais) para o modelo de SVC treinado com 676 características (autoria própria).

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

A maior parte dos modelos obtidos dos três 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 está entre 57% e 65%, e, a de teste, entre 58% e 65%. Com dez características, todas as taxas de acurácia estão acima de 70%. Com 40, todas as taxas de acurácia estão acima de 85%. Porém, com 85 características, os três métodos ultrapassam 90% de acurácia. Com 676 características, metade do número total, a rede neural tem a maior acurácia de teste de todo o experimento, chegando a 97,15%. Isso significa que entre os três 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. Os modelos treinados com SVM obtiveram performance pior do que floresta aleatória e rede neural em todos os casos.

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 enviada 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. Além disso, é um método computacionalmente mais barato para treino do que as redes neurais, que também obtiveram resultados similares aos das redes neurais.

6 CONSIDERAÇÕES FINAIS

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 as colunas que necessitam ser anonimizadas. 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 96% utilizando metade das características disponíveis.

Na Seção 6.1 são analisadas as contribuições realizadas e em seguida em 6.2 é feito um balanço sobre os potenciais trabalhos futuros na área.

6.1 CONTRIBUIÇÕES ALCANÇADAS

Foram realizados 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. 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, o que abre espaço para uma nova variedade de pesquisas na área. Além disso, obtivemos uma lista de digramas mais relevantes na língua portuguesa utilizando a pontuação TF-IDF.

6.2 TRABALHOS FUTUROS

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.

REFERÊNCIAS

[1] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[2] Edberto Ferneda. Redes neurais e sua aplicações em sistemas de recuperação de informações. *Ci. Inf.*, 35:25 – 30, 04 2006.

[3] Fundamentos de redes neurais. http://www2.decom.ufop.br/imobilis/fundamentos-de-redes-neurais/. Acesso em 30 de janeiro de 2021.

[4] Amund Tveit, Ole Edsberg, Thomas Røst, Arild Faxvaag, Øystein Nytrø, Torbjørn Nordgård, Martin Ranang, and Anders Grimsmo. Anonymization of general practioner medical records. 01 2004.

[5] Ministério da saúde expõe dados de 243 milhões de pessoas. https://tecnoblog.net/390237/ministerio-saude-expoe-dados-243-milhoes-pessoas/. Acesso em 15 de março de 2021.

[6] Brasil. Lei nº 13.709, de 14 de agosto de 2018. http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm. Acessado em 01/03/2020.

[7] Nações Unidas. Declaração universal dos direitos humanos. https://nacoesunidas.org/wp-content/uploads/2018/10/DUDH.pdf, 1948. Acessado em 01/03/2020.

[8] G. Cormode and D. Srivastava. Anonymized data: generation, models, usage. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 1015–1018, 2009.

[9] Tore Dalenius. Finding a needle in a haystack or identifying anonymous census records. *Journal of official statistics*, 2(3):329, 1986.

[10] Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[11] Massachusetts. Group insurance commission testimony before the massachusetts health care committee, de 19 de março de 1997.

[12] Massachusetts. City of Cambridge. Cambridge voters list database., 1997.

[13] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 1998.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[15] Standardscaler. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html. Acesso em 30 de janeiro de 2021.

[16] Stephen Robertson. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation - J DOC*, 60:503–520, 10 2004.

[17] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[18] Asa Ben-Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. Support vector clustering. *J. Mach. Learn. Res.*, 2:125–137, March 2002.

[19] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, August 1996.

[20] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, Florida, USA, 1993.

[21] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[22] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[23] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition, 2010.

[24] Deep learning book. `http://deeplearningbook.com.br/funcao-de-ativacao/`. Acesso em 30 de janeiro de 2021.

[25] György Szarvas, Richárd Farkas, and Róbert Busa-Fekete. State-of-the-art Anonymization of Medical Records Using an Iterative Machine Learning Framework. *Journal of the American Medical Informatics Association*, 14(5):574–580, 09 2007.

[26] Zhiqi Huang, Ryan McKenna, George Bissias, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. Psyndb: accurate and accessible private data generation. *Proceedings of the VLDB Endowment*, 12(12):1918–1921, 2019.

[27] D. ; Chillemi L. ; Meyer B. ; Vasconcellos L. ; Maciel E. ; Sunye 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.

[28] Cadastro nacional de estabelecimentos de saúde. `http://cnes.datasus.gov.br/`, 2017. Acesso em 24 de agosto de 2020.

[29] Wikipédia. `http://www.wikipedia.com.br/`. Acesso em 24 de agosto de 2020.

[30] Html2text. `https://pypi.org/project/html2text/`. Acesso em 24 de agosto de 2020.

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.