Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Departamento de Engenharia Elétrica

Técnicas de Ciência de Dados com Base de Dados de Saúde

Gabriel Pereira Pinheiro

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador
Prof. Dr. Rafael Timóteo de Sousa Júnior

Coorientador
Iure Viera Brandão

Brasília
2019

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Departamento de Engenharia Elétrica

Técnicas de Ciência de Dados com Base de Dados de Saúde

Gabriel Pereira Pinheiro

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Prof. Dr. Rafael Timóteo de Sousa Júnior (Orientador)
ENE/UnB

Prof. Dra. Edna Dias Canedo
CIC/UnB

Bruno Justino Garcia Praciano
ENE/UnB

Prof. Dr. José Edil Guimarães de Medeiros
Coordenador do Curso de Engenharia da Computação

Brasília, 8 de julho de 2019

Dedicatória

Este projeto é dedicado a todas as pessoas que influenciaram minha conquista de concluir esse trabalho. Especialmente à minha família, que sempre me deram todo o suporte necessário e apoio incondicional.

Agradecimentos

Agradeço a minha família e amigos que me ajudaram nessa jornada final de conclusão de curso com apoio, motivação e dicas que foram essenciais para meu sucesso.

Ao professor Dr. Rafael Timóteo de Sousa Júnior pela orientação, apoio e confiança desse projeto que me fez descobrir uma nova paixão. E ao professor Dr.-Ing João Paulo Lustosa pela chance concedida a mim para ingressar nessa área.

Agradeço ao Ronaldo do Ministério da Saúde pelo suporte em minhas dúvidas e no auxílio na liberação da base de dados.

À Universidade de Brasília pela oportunidade de fazer o curso de Engenharia de Computação e a todos que diretamente ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

Resumo

Esse projeto visa analisar uma base de saúde utilizando técnicas de mineração de dados a fim de realizar um estudo a respeito de uma de umas das doenças que recentemente registraram um grande número de casos e foi bem destacada pela mídia. Podendo ser transmitida pelo mesmo mosquito da dengue, o Aedes aegypti, a *zika* e *chikungunya* se tornaram uma epidemia no Brasil em 2015.

Os dados foram solicitados por meio do sistema eletrônico do serviço de informação ao cidadão, eSic, do Ministério da Saúde por meio do protocolo 25820007257201805, sendo solicitados dados referentes aos resultados dos exames laboratoriais as doenças chikungunya e Zika que são armazenados no sistema Gerenciador de Ambiente Laboratorial (GAL) (gerenciado pela Coordenação Geral de Laboratórios-CGLAB - MS/SVS//DEVIT/CGLAB). A não identificação do paciente foi solicitada no momento da requisição dos dados para que não houvesse exposição. Com quase 600 mil registros, foram obtido dados de pacientes de 2014 a 2018.

O que é proposto para o problema é a utilização de técnicas de mineração de dados na base de dados recebida pelo Ministério da Saúde com a proposta de realizar um estudo e extrair conhecimento a partir da base.

Um dos objetivos é analisar como se deu a propagação dos vírus pelo pais a fim de exibir a epidemia e as áreas mais críticas, também a criação de um modelo de predição a partir de atributos selecionado da base de dado com o objetivo de prevê se o resultado o exame do paciente.

Foram utilizados três algoritmos de classificação para a criação de um modelo de predição dos resultados dos exames e nos dois cenários comparados o algoritmo *random forest* obteve os melhores resultados.

Palavras-chave: Mineração de Dados, Zika vírus, chikungunya, random forest

Abstract

This project aims to analyze a health database using data mining techniques to conduct a study of one of the diseases that have recently registered a large number of cases and was well highlighted by the media. Being transmitted by the same dengue mosquito, Aedes aegypti, *zika* and *chikungunya* became an epidemic in Brazil in 2015.

The data were requested through the electronic system of the information service to the citizen, eSic, of the Ministry of Health through the protocol 25820007257201805, being requested data referring to the results of the laboratory examinations the diseases chikungunya and Zika that are stored in the system Environment Manager Laboratory (GAL) (managed by the General Coordination of Laboratories - CGLAB - MS / SVS // DEVIT / CGLAB). Non-identification of the patient was requested at the time of requesting the data so that there was no exposure. With almost 600 thousand records, data were obtained from patients from 2014 to 2018.

What is proposed for the problem is the use of data mining techniques in the database received by the Ministry of Health with the proposal to conduct a study and extract knowledge from the base.

One of the objectives is to analyze how the virus spread through the country in order to show the epidemic and the most critical areas, also the creation of a prediction model from selected database attributes to predict whether the result of the patient's examination.

Keywords: Data mining, zika, chikungunya, random forest

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

CGLAB Coordenação Geral de Laboratórios.

csv Comma-separated values.

e-SIC sistema eletrônico do serviço de informação ao cidadão.

ESPIN Emergência em Saúde Pública de Importância Nacional.

GAL Gerenciador de Ambiente Laboratorial.

GB gigabytes.

KB Kilobytes.

KDD knowledge-discovery in databases.

SGB síndrome de Guillain–Barré.

SUS Sistema Único de Saúde.

Capítulo 1

Introdução

Em 2015, o Brasil se deparou com uma epidemia do vírus *zika* e *chikungunya* que se espalhou em pouco tempo por todo o país sendo declarado como Emergência em Saúde Pública de Importância Nacional (ESPIN) pelas autoridades sanitárias brasileiras [2]. A ligação do vírus *zika* com a microcefalia e com a síndrome de Guillain–Barré (SGB) foi um dos motivos pelo qual o surto se mostrou tão preocupante. Esse acontecimento mostrou a vulnerabilidade em que o país está exposto e, principalmente, a necessidade de estudar a causas e meios de combater o vírus.

O setor de saúde, historicamente, sempre gerou uma enorme quantidade de dados motivado pela manutenção de registros, requisitos de conformidade e regulatórios e atendimento ao paciente [3]. As informações eram anotadas em papéis, mas com o avanço da tecnologia, a tendência é que os dados migrem cada vez mais para sistemas de armazenamento digitais [4]. Isso afeta tanto a economia e modernização do sistema de saúde, quanto o potencial que essa enorme quantidade de dados pode oferecer, pois técnicas de mineração de dados de saúde tem uma grande capacidade de explorar padrões ocultos nos conjuntos de dados do domínio da saúde [5].

Dados brutos podem ser coletados de inúmeras fontes como imagens, entrevistas com o paciente, dados laboratoriais e observações e avaliações do médico [6].Tornando a atividade desafiadora por essa variedade de tipos e origem do dados, o que faz a atividade de procurar padrões um grande desafio mas com inúmeras oportunidades.

1.1 Motivação

O ministério da saúde possui em seu banco de dados os prontuários eletrônicos dos pacientes que realizaram exames tanto para o vírus *zika* quanto para a *chikungunya* e disponibiliza tais informações pelo portal sistema eletrônico do serviço de informação ao cidadão (e-SIC) [7].

Qualquer pessoa, física ou jurídica, pode encaminhar pedidos pelo portal solicitando acesso a informação, que torna prático e acessível a todos. Com a possibilidade da disponibilidade dos dados e uma epidemia de surto de casos ocorrendo em cenário nacional, surge a ideia de aprofundar e conhecer técnicas para analisar uma base de dados extensa e diversificada, com a possibilidade de encontrar padrões e conhecimentos ocultos.

1.2 Definição do Problema

Uma epidemia declarada como ESPIN se espalhou pelo país nos últimos anos e mostrou a vulnerabilidade em que o país está exposto. Possuindo base de dados dos prontuários dos pacientes que recorram ao sistema de saúde público para realizar exames de presença dos vírus, será possível analisar o cenário no qual o pais se encontra no combate a propagação do vírus. Inúmeras atividades podem ser desenvolvidas utilizando os dados desses prontuários como base de dados.

Entretanto, as atividades necessitam de atenção especial por ser tratarem de dados médicos, o que influencia em vários fatores no tratamento diferencial dos dados. A dificuldade por ser tratar uma base grande com dados não padronizados, grande dimensionalidade entre inúmeros outros desafios tornam a atividade complicada e com alto nível de complexidade.

Com uma base com grande quantidade de dados o conhecimento extraído sem processos de mineração dessa base é quase insignificante. Assim, organização, análise e a interpretação dos dados é de suma importância para que extração tangível de conhecimento venha se tornar possível [8]. Logo uma base de dados médicos antes de processamento e sem utilização técnicas de mineração de dados não apresenta informações claras e podem acabar escondendo conhecimento para tomadas de decisões e por ser tratar de base de dados médicos a respeito de uma doença que se tornou um surto recente no Brasil, ratifica ainda mais a importância do estudo e procura por conhecimento.

1.3 Objetivos

Por ser uma base de dados médicos, um dos principais objetivos é detalhar os etapas de técnicas utilizadas e os desafios encontrados nesse tipo de banco de dados e documenta-las.

A etapa de pré-processamento dos dados é essencial em qualquer analise de *Big data* e por ser uma base com grande variedade dados, essa será um dos focos do projeto detalhar e exibir os desafios e soluções aplicadas para limpar a base.

Na etapa de análise é esperado observar a forma na qual a epidemia se espalhou pelo país e mapear detalhadamente os resultados obtidos.

O objetivo do trabalho no geral é a aplicação de técnicas de mineração de dados na base de dados da *zika* e *chikungunya*. Possuindo os objetivos específicos de realizar um estudo e detalhar como se deu a propagação dos vírus, também realizar a análise a respeito da qualidade da base de dados, detalhar um perfil do paciente e por ultimo a criação de um modelo de predição do resultado dos exames dos pacientes.

1.4 Descrição dos capítulos

Essa monografia está estruturado em 5 capítulos e ao final as referencias que foram utilizadas para o embasamento.

O capitulo 1, é uma introdução do projeto executado. Contendo quatro subseções, que irão introduzir o projeto como um todo, explicar a motivação do projeto, definir o problema e ao final explicar o objetivo esperado desse trabalho.

O capítulo 2 irá apresenta a fundamentação teórica no qual todo o trabalho foi construído, isto é, o que é necessário para sua realização. Está divido em cinco seções principais, a primeira irá explicar brevemente a respeito de cada um dos vírus. A segunda seção é uma introdução a mineração de dados, na terceira será explicado o processo *knowledge-discovery in databases*. A quarta seção explica a respeito das bases de dados e a qualidade dos dados. E por último será apresentado a mineração de dados em base médicas, sua importância e desafios.

No capítulo 3, foi apresentado todo o estado da arte no qual esse trabalho seguiu para alcançar os resultados esperados, isto é, as etapas que foram realizadas a fim de obter o resultado final.

O capítulo 4, apresenta os resultados encontrados após toda a execução do trabalho, e foi divido em 3 áreas principais, os resultados encontrados a respeito da qualidade dos dados, do entendimento da base e por último o modelo de predição.

Por último, o capítulo 5 apresenta a conclusão do trabalho e o que será executado posteriormente como forma de melhoria.

Capítulo 2

Referencial Teórico

O capítulo 2 foi desenvolvido a fim de apresentar a fundamentação teórica no qual esse trabalho foi embasado. O capítulo está divido em 3 subseções, a subseção 2.1 será a uma breve explicação sobre os dois vírus para no qual os pacientes da base de dados foram submetidos a exames: *zika* e *chikungunya*.

A subseção 2.2 irá explicar conceitos e técnicas de mineração de dados que são essências para o entendimento do trabalho executado e na sequência a subseção 2.5 irá explicar uma área da mineração de dados que foi utilizada na realização do projeto, a mineração de dados em base de dados médicas, que é o foco desse projeto como um todo.

2.1 Os vírus

Zika e chukungunya são vírus transmitidos principalmente pela picada da fêmea do mosquito *Aedes aegypti*, infectado, sendo esse o mesmo mosquito transmissor da dengue, outro vírus que já é um desafio de saúde global [9].

Ambos ganharam forte importância da mídia nos últimos 5 anos devido a epidemia que se alastrou pelo Brasil em 2014.

Para explicar a respeito de cada um dos vírus foram criadas as subseções 2.1.1 e 2.1.2 que irão respectivamente detalhar a *Zika* e *Chikungunya*.

2.1.1 Zika

O vírus da zika foi detectado pela primeira vez na floresta zika na Uganda no ano de 1947 [10]. O vírus zika é transmitido pelos mosquitos Aedes, que também é o vetor de transmissão da dengue, chikungunya e febre amarela [11].

A taxa de morbidade da microcefalia em um bebê é muito alta se a gravidez foi infectada pelo vírus zika. Também a síndrome de Guillain-Barre coincide com as infecções causadas pelo vírus *zika* no Brasil[11].

Seu primeiro caso registrado no país foi em 2015 [12] e desde então o número de casos teve um grande aumento. Ainda não existe vacina ou medicamentos contra *zika*. Portanto, a única forma de prevenção é acabar com o mosquito.

Pessoas com doença do vírus zika geralmente possuem sintomas: febre moderada, erupções cutâneas, conjuntivite, e dor nas articulações, mal-estar ou dor de cabeça, que duram de 2 a 7 dias normalmente [11].

No geral, a evolução da doença é benigna e os sintomas desaparecem espontaneamente após 3 a 7 dias mas já aconteceram casos de óbitos [12].

2.1.2 Chikungunya

Chikungunya é uma doença viral transmitida aos seres humanos por mosquitos infectados, incluindo Aedes aegypti [13] , foi reconhecido pela primeira vez nos anos 1950 na África, e desde então, casos foram identificados. No Brasil, a circulação do vírus foi identificada pela primeira vez em 2014 [14].

Os sintomas de *chikungunya* são febre alta, dor nas articulações, dor de cabeça, inchaço das articulações, fadiga, erupção cutânea, náuseas, dores musculares [15]. O período de incubação do vírus é de 4 a 7 dias, e a doença, na maioria dos casos, é auto-limitante. A mortalidade em menores de um ano é de 0,4%, podendo ser mais elevada em indivíduos com patologias associadas [14].

Igualmente a *zika*, ainda não existe vacina ou medicamentos contra *chikungunya* [14].

2.2 Mineração de Dados

O rápido avanço na tecnologia de armazenamento de dados e a facilidade no acesso fez com que as organizações pudessem guardar uma enorme quantidade de dados de maneira mais barata e fácil, porem, extrair informação útil desse montante de dados se tornou um desafio [1]. E para isso surgiu a mineração de dados, que é uma tecnologia que combina métodos tradicionais de análise de dados com algoritmos sofisticados para processar grandes volumes de dados, essa quantidade volumosa de dados é chamadas pelos atuantes na área de *big data*, que será apresentado na subseção 2.2.1.

A mineração de dados abriu uma nova oportunidade na análise de dados, no quesito de uma interpretação e utilização desses dados. Inúmeras áreas começaram a utilizar esses dados que normalmente serviam apenas como backup ou tomadas de decisões simples, para agora retirar informações úteis desses dados.

Ela é uma etapa no processo de KDD, que será apresentado na subseção 2.3, que consiste em aplicar técnicas computacionais que, sob limitações de eficiência computacional aceitáveis, produzem uma enumeração particular de padrões (ou modelos) sobre os dados [16].

Pesquisadores de diferentes disciplinas começaram a se concentrar no desenvolvimento de mais ferramentas eficientes e escalonáveis que poderiam manipular diversos tipos de dados [1]

2.2.1 Big data

Originalmente *big data* significava um volume de dados que não poderia ser processado eficientemente por métodos e ferramentes de banco de dados [17].

Big data pode ser definido com uma quantidade de dados além da capacidade da tecnologia de armazenar, gerenciar e processar de formar eficiente essa grandeza, sendo essas limitações. Há poucos anos atrás, para armazenamento pessoal era esperado dezenas ou centenas de gigabytes (GB), mas atualmente, já se imagina a casa de dezenas a centenas de terabytes [17] o que mostra que o conceito vai se alterando com os avanço da tecnologia no tempo, o que poderia ser considerado uma grande quantidade de dados há anos atrás, hoje já é visto com uma quantidade normal.

2.3 Conhecimento descoberto em bancos de dados

Há uma necessidade crescente da análise do grande volume de dados digitais que são gerados todos os instantes. Para isso processos, ferramentas e teorias existem para auxiliar esse trabalho [18]. Técnicas de mineração de dados são usadas para executar esse trabalho, que é parte uma etapa do processo do knowledge-discovery in databases (KDD), que consiste no processo como um todo de converter dado bruto em informação útil [1].

Frequentemente, KDD e mineração de dados são usados como sinônimos de maneira equivocada. KDD se refere ao processo geral de descoberta de conhecimento útil a partir de dados, enquanto a mineração de dados se refere a uma etapa específica desse processo. [16]. Mineração de dados é a aplicação de algoritmos específicos para extrair padrões de dados. As etapas adicionais no processo KDD, como pré-processamento, pós processamento e informação, são essenciais para garantir que o conhecimento útil seja derivados dos dados.

Uma possível definição para todo o processo KDD: é o processo não trivial de identificar potenciais novos e válidos, padrões úteis e, finalmente, compreensíveis nos dados [16]. É um processo, porque é compostos por várias etapas e não é trivial por que não existe uma regra genérica, envolve estudo e adaptação para cada caso.

Conforme apresentado pela Figura 2.1, é possível observar que esse processo consiste em uma série de etapas a serem seguidas para no final ser possível obter informação útil. Cada etapa que aparece na Figura 2.1 será apresentada em uma subseção a seguir. O processo se inicia com a entrada dos dados bruto, que sera abordado na subseção 2.3.1. Na sequência esse montante de dados e enviado para o pré-processamento, que sera explicado na subseção 2.3.2 que em seguida é enviado para a parte de mineração de dados, subseção 2.3.3.

Após a mineração esses dados irão para o pós-processamento na subseção 2.3.4 e por último a para a etapa de validação da informação que será apresentado na subseção 2.3.5.

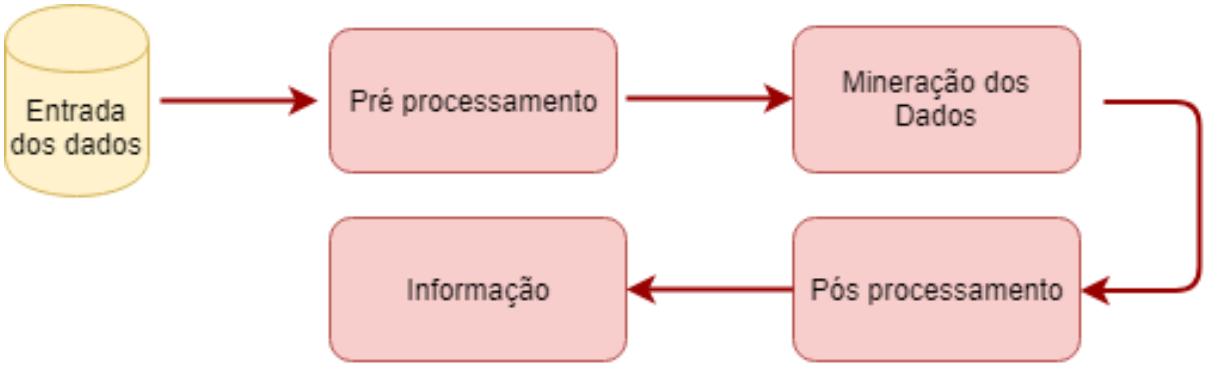

Figura 2.1: Etapas do processo KDD [1].

2.3.1 Entrada dos dados

Essa é a primeira etapa de todo o processo de descobrimento de informação. Nessa etapa é selecionado o conjunto de dados pertencente a um domínio contendo todos os possíveis atributos e registros que irão fazer parte da análise.

O processo de seleção é complexo, tendo em vista que a fonte dos dados podem ser diversas e é uma etapa muito importante no processo, porque é nessa fase que serão decididos quais os conjuntos de dados que serão relevantes para que sejam obtidos resultados com informações uteis.

2.3.2 Pré-processamento

Essa etapa representa o momento em que os dados são tratados para tornar a etapa de mineração de dados mais confortável [1]. Em muitos os casos os dados precisam ser processados para tonar a análise mais fácil, ou também para objetivos específicos da mineração os dados podem ser modificados nessa etapa. Como apresentado pela Figura2.1,

a entrada que essa etapa recebe é a base de dados bruta, que significa que ela não sofreu nenhum tratamento.

Por se tratar de uma base de dados até então sem nenhum processamento, é essencial que seja analisado qual técnica será necessária aplicar na base de modo que ajuda os algoritmos de mineração na etapa seguinte 2.3.3. Para cada tipo de erro existe pelo menos uma técnica possível para a correção.

Uma importante abordagem nessa etapa é a agregação. Essa ideia consiste na combinação de dois ou mais objetos em um único objeto. Por exemplo seria juntar informação de três colunas que informam o preço de um item e agregar em uma única coluna sendo calculado o preço médio do produto. Uma dos benefícios de se realizar essa técnica seria uma base dados menor que iria reduzir a quantidade de memória necessária para seu armazenamento e seu tempo de processamento. A utilização dessa base em algoritmo de mineração eles seriam processados mais rapidamente. Um segundo ponto importante seria a mudança de escopo ou de escala, já que de tal modo seria fornecido uma visão de alto nível dos dados, ao invés de uma visão de baixo nível. Um ponto negativo da técnica de agregação é uma perda potencial de detalhes do dados [1].

Uma outra técnica utilizada no pré-processamento dos dados é a amostragem, que é normalmente é utilizada para selecionar um subconjunto de dados para serem analisados. Para a mineração de dados esse técnica é muitas vezes utilizada devido ao alto custo de tempo ou dinheiro para processar todo os dados, fazendo que muitas vezes sejam utilizados algoritmos para amostrar os dados para que assim seja viável a utilização de algoritmos de mineração mais robustos. A dificuldade está em utilizar uma amostra que possua representatividade similar a original. Uma técnica de amostragem existente é a aleatória, na qual um percentual dos dados serão selecionado de forma aleatória [1].

Para ajudar no funcionamento dos algoritmos de mineração de dados, a técnica de redução de dimensionalidade é muito recomendada. Ela consiste em reduzir o número de atributos a serem processados. Um ponto favorável de sua utilização é que inúmeros algoritmos funcionam melhor com uma quantidade menor de atributos [1]. Com a redução é mais fácil a visualização dos dados já que são menos atributos serem interpretados. Essa técnica não necessariamente elimina dados fora de prejudicial, pode ser usada para retirar informação redundante ou irrelevante da base de dados.

Alguns algoritmos de mineração de dados, principalmente de classificação, necessitam de dados estejam na forma de atributos categóricos. Podem ser utilizada a técnica de discretização que transforma dados contínuos em categórico. E também existe a binarização que transforma dado tanto continuo ou discreto em dados binários [1].

Além disso, se um atributo categórico tiver um grande número de valores (categorias), ou alguns valores ocorrem com pouca frequência, então pode ser benéfico para certas

tarefas de mineração de dados para reduzir o número de categorias combinando alguns dos valores.

Em modo geral os itens são divididos em duas categorias: selecionar dados e atributos para se analisar ou alterar os atributos mas em ambos a finalidade é melhorar análise de dos dados em relação a tempo, custo e qualidade [1].

Todas essas técnicas podem ser utilizadas nesse etapa de pre-processamento de dados com a finalidade de melhorar o tempo de processamento, armazenamento e entre outras vantagens mencionadas anteriormente, uma vez que, os dados que são coletados de diferentes fontes podem ter dados sujos, a limpeza dos dados deve ser feita antes que os dados sejam processados, a fim de obter dados de qualidade [19].

2.3.3 Mineração dos dados

A mineração de dados é um processo de descobrir informações úteis em uma grande quantidade dados. As técnicas são geradas para vasculhar grande bancos de dados, com o objetivo de encontrar novos padrões úteis [1]. Esses padrões podem ser utilizados para prever o resultado de uma observação futura. As técnicas tradicionais de análise de dados frequentemente encontraram dificuldades práticas para enfrentar os desafios impostos pelos novos conjuntos de dados.

2.3.3.1 Classificação

A classificação que é uma tarefa de atribuir objetos a uma das várias categorias predefinidas, é um problema generalizado que engloba muitas aplicações diversas [1].

Classificação pode ser definido como a tarefa de aprender uma função f que mapeia cada conjunto de atributos x para uma das classes predefinidas y. Conforme pode ser apresentado na Figura2.2, onde um conjunto de atributos é a entrada do modelo, que irá analisar e falar para aqueles atributos qual será a classe dele.

Figura 2.2: Representação do fluxo da classificação.

Diferentemente da regressão, a classificação irá atribuir uma dado discreto para aquele registro, isto é, uma categoria pré-definida. Uma vez que regressão atribui um valor continuo.

Tabela 2.1: Métricas

		Predição	
		+	-
Atual	+	++ (TP)	+-(FN)
	-	-+ (FP)	– (TN)

As técnicas de classificação são mais aconselhadas para a predição ou descrição de um conjunto de dados binário ou categórico [1]. São menos eficientes para categorias ordinais como por exemplo classificar uma pessoa como alta,média ou baixa renda [1].

Uma técnica de classificação é uma abordagem sistemática de construir modelos de classificação a partir de uma base de dados de entrada. Existem vários classificadores como árvore de decisão, redes neurais entre outros. Cada técnica tem sua particularidade e utiliza um algoritmo de aprendizado para identificar um modelo que melhor se adapte ao relacionamento entre o conjunto de atributos e o rótulo de classe dos dados de entrada. O modelo que é gerado pelo algoritmo deve se ajustar aos dados de entrada do sistema e prever corretamente as classes dos registros novos. Sendo assim o objetivo principal do algoritmo construir um modelo com uma boa capacidade de generalização, ou seja, prever classes de registros desconhecidos [1].

A avaliação do desempenho de um modelo de classificação é baseada nas contagens de registros de teste, no qual, tenham sido corretamente e incorretamente previstas pelo modelo gerado. Essas contagens são tabuladas em uma Tabela conhecida como matriz de confusão [20].

Embora uma matriz de confusão forneça as informações necessárias para determinar o desempenho de um modelo de classificação, o resumo dessas informações com um único número tornaria mais conveniente e prático a comparação do desempenho de outros modelos e para isso podem ser utilizados métricas de desempenho que serão apresentado na sub subseção 2.3.3.2.

2.3.3.2 Métricas de avaliação

Conforme apresentado na sub subseção 2.3.3.1, existem métricas de desempenho que ajudam na avaliação de um ou mais modelos.

A avaliação da classificação é realizada usando as seguintes métricas: *precision*, *recall* e *f1-score* [21].

Precision, do português precisão, é a razão de observações positivas preditas corretamente para o total de observações positivas preditas, isto é, seguindo a Tabela 2.1 quando o algoritmo prevê uma categoria correta ele irá estar acertando um *True positive* (verdadeiro positivo) ou seja, acertou qual era a categoria daquele registro. Quando o algoritmo

prevê incorretamente a categoria do registro ele será um *false positive* (falso positivo), ou seja, o algoritmo falou que era uma categoria mas não era. Sua definição é dada então por :

$$Precision = \frac{TP}{TP + FP}$$

Já o *recall* são resultados corretos dividido pelo número de resultados que deveriam ter sido retornados. E bastante utilizado como métrica de modelos quando há um alto custo associado ao *false negative*. Uma vez que atribuir como positivo uma determinada categoria poderia acarretar em prejuízos. Por exemplo categorizar uma transação fraudulenta como não fraudulenta.

$$Recall = \frac{TP}{TP + FN}$$

O *f1-score* combina *precision* e *recall* de modo a trazer um número único que indique a qualidade geral do seu modelo e trabalha bem até com conjuntos de dados que possuem classes desproporcionais.

$$F1score = \frac{2.Precsion.Recall}{Precision + Recall}$$

2.3.3.3 Árvore de decisão

As árvores de decisão são ferramentas poderosas e populares para classificação e previsão [22]. Árvores de decisão são modelos estatísticos que utilizam o treinamento supervisionado para a classificação e previsão de dados. Em outras palavras, em sua construção é utilizado um conjunto de treinamento formado por entradas e saídas, que são as classes. [20]. Estes modelos utilizam a estratégia de dividir para conquistar, um problema complexo é decomposto em sub-problemas mais simples e recursivamente esta técnica é aplicada a cada sub-problema .

De um modo geral podemos definir como estruturas de dados formadas por um conjunto de elementos que armazenam informações chamadas nós, na Figura 2.3 os nós são apresentado pelos pontos onde estão as "condições"e as "classes".

Além disso, toda árvore possui um nó chamado raiz, que possui o maior nível hierárquico, que é o ponto de partida e faz a conexão com os outros nós, que são chamados de filhos. Esses filhos podem possuir seus próprios filhos que por sua vez também possuem os seus. O nó que não possui filho é conhecido como nó folha ou terminal que na Figura 2.3 é apresentado como C1, C2, C3, C4 e C5.

Logo, os nós podem ser classificados como [1] :

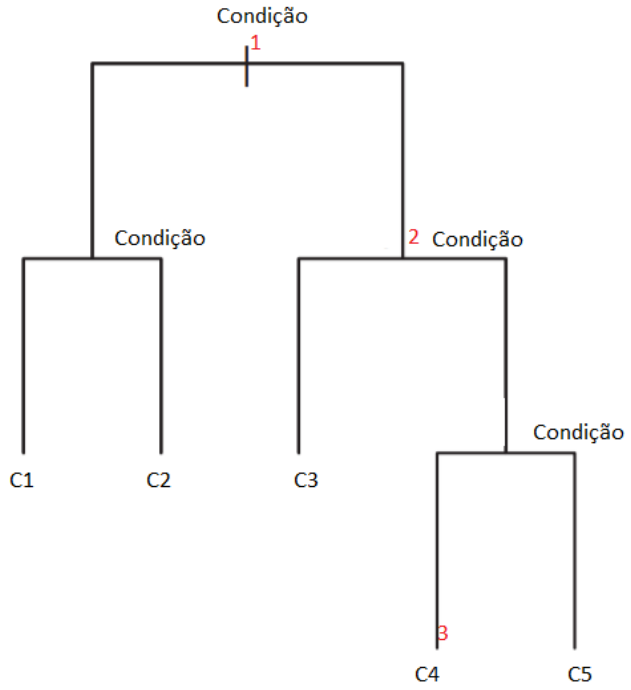

Figura 2.3: Exemplo de uma árvore de decisão.

- O nó raiz, que não tem arestas de entradas e possui nenhuma ou mais de uma saída. Que na Figura 2.3 é representado pelo nó 1.
- O nó interno, que contém exatamente uma aresta de entrada e duas de saída, é exemplificado na Figura 2.3 como o nó 2.
- O nó folha, que contém uma aresta de entrada e nenhuma de saída, que foi representado na Figura 2.3 como nó 3.

Tendo essas definições esclarecidas, uma árvore de decisão nada mais é que uma árvore que armazena regras em seus nós, e os nós folhas representam a decisão a ser tomada. Tendo como exemplo a Figura 2.3 é possível observar uma primeira condição que irá determinar no fluxo da classificação e ao final poderá ser definida entre 5 classes pré definidas.

Existem exponencialmente inúmeras árvores de decisão que podem ser geradas a partir de um conjunto de atributos, mesmo que algumas possam ser mais precisas do que outras, é computacionalmente inviável encontrar a árvore ideal. Entretanto algoritmos eficientes foram desenvolvidos para encontrar uma árvore de decisão razoavelmente precisa em um período de tempo razoável [1].

2.3.3.4 Random Forest

Random forest, do português, floresta aleatória é um algoritmo de aprendizagem supervisionada. Como o próprio nome sugere, o algoritmo cria de modo aleatório uma "floresta", que é uma combinação de árvores de decisão, na maioria dos casos treinados com o método de *baggaing*. A idéia principal do método de *bagging* é que a combinação dos modelos de aprendizado aumenta o resultado geral. A Figura 2.4 apresenta o fluxo de execução do algoritmo por uma visão de alto nível.

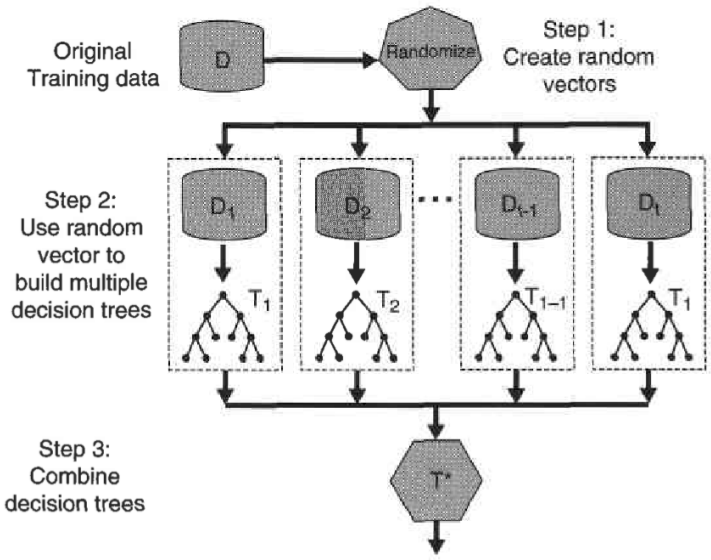

Figura 2.4: Exemplo do algoritmo *random forest* apresentado no livro [1].

De uma maneira geral, o algoritmo de florestas aleatórias cria várias árvores de decisão e faz uma combinação entre elas de modo a obter uma predição com maior acurácia.

2.3.3.5 Extra Trees

Extra Trees é como a *random forest*, onde um subconjunto aleatório de recursos candidatos é usado, mas em vez de procurar os limites mais discriminativos, os limites são desenhados aleatoriamente para cada recurso candidato e o melhor desses limites gerados aleatoriamente é selecionado como a regra de divisão [23].

As duas principais diferenças entre o *extra tree* e o *random forest*, são:

- O *extra tree* utiliza todas as amostras para construir cada árvore de decisão diferentemente do *random forest* que subamostra o conjunto,
- O *random forest* obtém a melhor bifurcação no subconjunto aleatório ja o *extra tree* seleciona aleatoriamente a bifurcação [24] .

2.3.4 Pós-processamento

Apos todo o processamento do dados e processamento por meio de algoritmos de mineração de dados, os padrões devem ser compreensíveis, se não imediatamente, depois de algum pós-processamento [25]. Para isso essa etapa é realizada a fim de tornar mais claros os resultados obtidos nas etapas anteriores. Nessa etapa pode ser gerado as matrizes de confusão apresentada na etapa anterior de maneira mais clara e fácil de entender o desempenho, pode ser realizada a comparação entre modelos a fim de identificar o melhor para o caso. Nesta fase é analisada as informações que são interpretadas e se transformam em conhecimento.

2.3.5 Informação

A última etapa é a informação, no qual consiste na interpretação e avaliação dos resultados para que o objetivo final desejado seja alcançado. Caso não seja alcançado o que era esperado o processo pode retornar a uma das etapas anteriores apresentada a fim de mudar de alguma mudança para ser possível um novo resultado.

A informação é o passo intermediário do dado bruto se tornar conhecimento,o resultado do processamento de dados geram informações que tem significado e podem contribuir no processo de tomadas decisões. Já o conhecimento vai além da informação, pois ele além de ter um significado tem uma aplicação. E nessa etapa é transformado a informação em um conhecimento.

2.4 Base de dados

A base de dados representa todo o conjunto de dados que são armazenados para salvar alguma informação. Ela é composta por seus atributos que representam a característica que aquele dados representa,as relações entre atributos e dados, e pelo próprio dado em si.

Essa seção irá detalhar a respeitos dados, será explicado sobre a qualidade deles na subseção 2.4.1 com seus desafios e soluções, os critérios de análise e por último o problema das classes desbalanceadas. Na sequência, a subseção 2.4.2 irá apresentar técnicas de visualização de dados mais comuns e seus exemplos.

2.4.1 Qualidade dos dados

Os dados que são utilizados na mineração de dados normalmente são coletados para outras finalidades ou aplicativos futuros, mas não especificamente para mineração. Por

esse motivo, normalmente, não é possível aproveitar os dados diretamente armazenamento, isto é, recupera-los de onde estão armazenados e usar sem a necessidade de nenhum tratamento. Como em muitos casos não é possível alterar a forma da origem do dados, um foco da mineração é a detecção e a correção da qualidade dos dados, conhecido como limpeza dos dados. Também é possível o uso de algoritmos que podem tolerar a má qualidade, entretanto, são mais robusto o que acarreta em um maior custo.

Não é esperado que os dados sejam perfeitos, devido a inúmeros problemas, como: erro humano, limitação de dispositivos de medição ou falha na coleta de dados [1]. O que torna a limpeza dos dados uma etapa já prevista para a mineração. Alguns problemas são mais comuns de serem vistos em base de dados, como:

- Erro de medição, quando um valor registrado que difere do verdeiro, sendo esse um erro de medição, por exemplo erros de digitação são comuns quando os dados são inseridos manualmente. Um erro de coleta de dados pode ser por exemplo omitir dados. Nesse caso entraria o erro de digitação humano, que por exemplo ao invés de escrever detectado em um formulário, escreve "detetado". Por ser um valor diferente de detectado seria considerado outra categoria, o que afetaria o resultado final.

- Também existe os dados conhecidos como *outliers*, que são dados que possuem características diferentes da grande maioria dos outros dados da base. A Figura 2.5 é um exemplo de *outlier*, é possível observar que todos os dados se concentram na lateral esquerda imagem, entretanto existe um pequeno percentual dos dados que está na extrema direita da imagem.

- Um caso que pode acontecer são dados faltantes, isto é, que não estão presentes na base de dados. Em alguns casos essa ausência do dado pode ter sido no momento da coleta, que não necessariamente é um erro, já que por exemplo, existem formulários que somente responde a aquele pergunta dependendo a resposta da anterior. Por simplicidade, esses valores faltantes são adicionados a base de dados e devem ser levados em conta na análise. Existem estratégias para lidar com esse tipo de dados como, a **eliminação de atributos ausentes**, sendo essa um método simples e eficaz de eliminar campos com valores ausentes, entretanto, dados parcialmente informados também guardam informações e sua eliminação acarreta em uma perda. E caso sejam eliminados muitos campos, a quantidade de dados para se trabalhar pode reduzir muito tornando difícil realizar uma análise confiável. Logo, essa técnica deve ser feita com cautela, já que o atributos eliminados podem ser os que são críticos para a análise [1].

  Uma outra técnica possível também é **estimação do valor faltante**. Em alguns casos dados que não estão presentes na base de dados podem ser estimados com

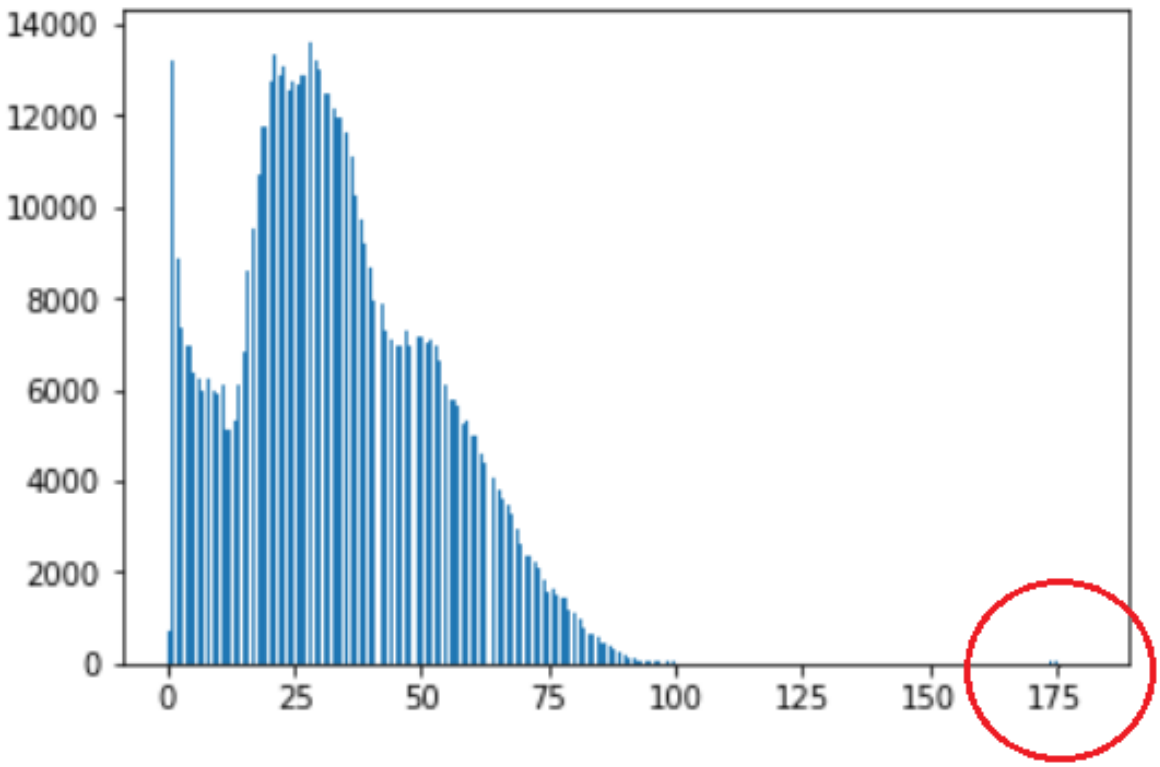

Figura 2.5: Exemplo de outliers.

confiabilidade. Caso o atributo seja contínuo, então o valor de atributo médio dos vizinhos mais próximos pode ser usado, e caso se o atributo for categórico, o mais frequente valor de atributo pode ser obtido. **Ignorar o valor na analise**, é uma alternativa por ser possível modificar o algoritmo para não considerar esses valores ausente. O que tornaria prático por não ser necessário o tratamento prévio do dado.

- Um outro tipo de problema que é possível estar presente na base são valores inconsistentes. O que torna sua detecção e tratamento de extrema importância já que possuem dados com valores fora do contexto. Por exemplo, o peso de um paciente ser um valor negativo [1].

- Dados duplicados, é quando um mesmo objeto está registrado mais de uma vez na mesa base dados. Em alguns casos existe alguma diferença entre os registros para que estejam listados duas vezes, como por exemplo, um erro de digitação no nome do cliente que logo possuirá dois registros. É necessário muita cautela no tratamento desses dados, pois casos os dados sejam apenas semelhantes e não iguais, pode haver perda de informação caso um deles seja eliminado. No caso em que os dados são duplicados por inteiro, não haveria problema.

Cada uma das estratégias deve ser avaliada para cada caso, já que possuem vantagens e desvantagens. Aplicações que depende de mediação e coletas, podem possuir problemas que devem ser lidados, como :

- O envelhecimento do dado, a partir do comento em que ele é coletado, já esta retratando uma informação passada, podendo ser um fator crucial de uma analise, já que dados desatualizados podem mostrar padrões antigos e não se aplicarem mais. Logo, é uma questão a ser planejada com cautela e vê se encaixa no objetivo.
- Os dados devem possuir relevância, isto é, conter informações necessárias para aplicação desejada, o que significa que devem possuir alguma ligação com o que deseja ser analisado.
- É necessário possuir conhecimento a respeito dos dados, ou seja, deve existir uma documentação que descreva-os, para que assim o tratamento dos dados por exemplo ocorra de maneira correta e eficiente. Sem saber o que o dado significa não é possível saber se é um valor condizente com o esperado, ou dentro do limite de aceitável.

Para analisar e poder argumentar a respeito sobre os dados, existem critérios. Seguindo como referencia o artigo [26], são definido cinco principais critérios para análise de qualidade e requisitos de dados:

- Integralidade: Representa a relação entre os valores dos dados esperados e aqueles que são efetivamente fornecidos.
- Pontualidade: Diferença de tempo do dado desde a sua coleta ou ultima atualização até o momento da análise.
- Unicidade: Cada objeto de dados do mundo real é representado no repositório de dados de uma forma única.
- Consistência: Definição e compreensão dos elementos de dados de maneira clara e também a integridade das estruturas de dados e as relações entre entidades e atributos.
- Precisão: É a capacidade de refletir corretamente os dados do mundo real ou pelo menos sendo aceito como verdadeiro.

Bancos de dados com uma distribuição desbalanceadas de classes são comuns em aplicações no mundo real [1]. Por exemplo imagine um sistema que detecta transações fraudulentas em uma bandeira de cartão de crédito, dentro de todo o escopo do banco de dados, isso deverá representar uma pequena parcela entre todas as transações. A Figura

Figura 2.6: Exemplo de classe desbalanceada.

2.6 ilustra o exemplo das fraude sendo apenas 5% das transações fraudulentas. Como é possível observar na figura, a classe dominante é a não fraude.

Mesmo com sua frequência baixa de acontecimentos, uma classificação correta dessa classe rara tem maior valor do que uma classificação da classe majoritária [1]. No entanto, como a distribuição de classes é desequilibrada, isso apresenta uma série de problemas para algoritmos de classificação. Por exemplo, se o algoritmo de predição retornar que todos as transações não foram fraudulentas estará com uma acurácia de 95%, entretanto para a outra classe estará com a acurácia de 0%.

Amostragem é uma possível alternativa para lidar com o problema das classes desbalanceadas. A ideia principal dessa técnica é modificar a distribuição das classes na base de dados[1].

Considerado ainda o exemplo da detecção de fraude, imagine que a base de dados conte com 10.000 registros. 500 deles registraram fraude e os outros 9.500 não fraude, a técnica de amostragem nesse caso seria formar um conjunto de dados de treino do algoritmo com 500 amostras de não fraude mais 500 amostras de fraude, resultado em uma base de treino com 1000 registros. A distribuição das classes ficaria conforme apresentado pela Figura 2.7, em que é possível observar o balanceamento das classes.

2.4.2 Visualização

A exibição dos dados em pode ser tanto por meios de gráficos ou por meio de tabelas. É necessário que os dados sejam convertido para uma forma visual para que suas características e suas relações entre si possam ser analisadas. O principal objetivo da visualização é a interpretação da informação visualizada e a formação de um modelo mental da informação de uma forma mais rápida e agradável . E o motivo primordial para se

Figura 2.7: Exemplo de classe desbalanceada.

utilizar a visualização dos dados é que seres humanos podem rapidamente absorver grande quantidade de informações visuais e encontrar padrões nelas [1].

As técnicas de visualização são normalmente especializada para o tipo do dado que está sendo analisado, ou seja, para cada tipo do dado pode haver uma ou mais maneiras de representa-lo visualmente, e essas maneiras estão sendo criadas continuamente, devido aos novos tipos de dados. Nas subseções seguinte, serão apresentadas algumas técnicas de visualização de dados.

2.4.2.1 Histograma

Conhecido também como distribuição de frequências, histograma é a representação em barras ou retângulos de um conjunto de dados. A base do retângulo significa a classe na qual o dado pertence, já a altura representa a quantidade ou frequência absoluta com que aquela classe ocorre no conjunto de dados utilizado. A utilização de histogramas tem caráter preliminar em qualquer estudo e é um importante indicador da distribuição de dados.

A Figura 2.8 e 2.9 são um exemplo de um gráfico histograma. É possível observar a frequência sendo representada pelas barras em azul. O eixo x está nesse caso representando uma data do evento, e o eixo y a quantidade que esse evento ocorreu. No caso das imagens apresentadas, o histograma informa a quantidade de exames realizados para *chikungunya* de acordo com cada data.

É possível observar a diferença dos gráficos pela espessura da barra no eixo X, isso deve-se pelo fato do gráfico ser formado pelas barras, ao pre determinar uma quantidade de barras, pode ser que seja sub amostrado os dados e o resultado não seja tão especifico quanto deveria. A imagem 2.8 tem aproximadamente 100 barras, já com os mesmo dados e apenas 10 barras, foi gerado o gráfico 2.9.

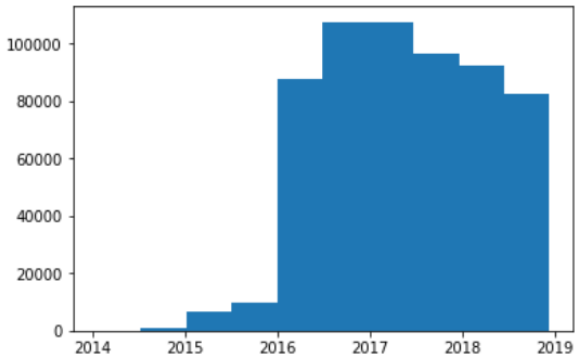

Figura 2.8: Exemplo do tipo de gráfico histograma com um número definido de 'bins'.

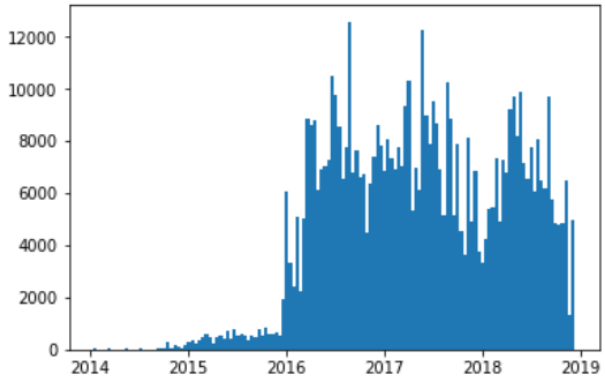

Figura 2.9: Exemplo do tipo de gráfico histograma com um número ajustável de 'bins'.

2.4.2.2 Pizza

O gráfico pizza, como o próprio nome já diz, possui um formato similar a de uma pizza redonda. Esse tipo de gráfico é um histograma, mas é normalmente usado com atributos categóricos que possuem um número relativamente pequeno de valores de classes.

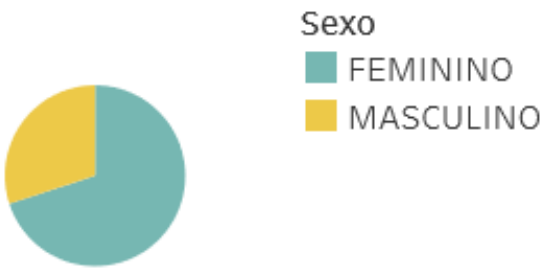

Figura 2.10: Exemplo do tipo de gráfico pizza.

Ele se diferencia do histograma por ao invés de exibir a frequência de uma classe com a altura da barra, o gráfico pizza usa a sua área relativa de um circulo para indicar a frequência relativa.

A imagem 2.10 apresenta um gráfico pizza separando duas classes, feminino e masculino. É possível observar que a classe feminino é predominante, ocorrendo mais na metade dos casos. Já a Figura 2.11, é um exemplo do motivo no qual esse tipo de gráfico é indicado somente quando há um pequeno número de classes a serem representados Com uma grande variedade a informação visual se torna mais difícil de absorver.

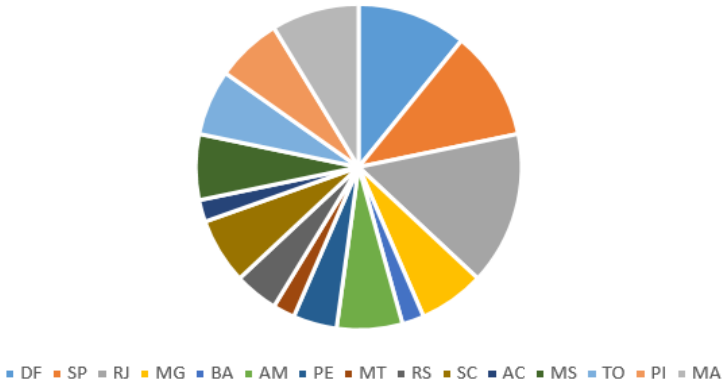

Figura 2.11: Exemplo do tipo de gráfico pizza com muitas classes.

2.4.2.3 Animação

Um outro tipo de abordagem é a animação, essa técnica ajuda a lista com "fatias"de dados que podem estar relacionadas com o tempo. A ideia principal e exibir sucessivas fatias bidimensionais dos dados. O sistema de visão humano é adequado para detectar mudanças visuais [1], o que ratifica o uso dessa técnica.

Um exemplo desse tipo de gráfico é apresentado pela Figura 2.12, no qual é apresentado três gráficos em ordem temporal da quantidade de exames para o vírus *chikungunya*.

2.5 Mineração de Dados de Saude

A mineração de dados como já apresentado na subseção acima 2.2 possui inúmeros desafios por ser tratar de uma grande quantidade de dados e técnicas de bancos de dados não possuírem eficácia com o volume de dados maiores. Os dados médicos além possuírem bases de dados volumosas possuem um desafio a mais pela diversidade de tipos de dados e a velocidade com que ele deve ser gerenciado. Sua diversidade se deve ao fato de suas

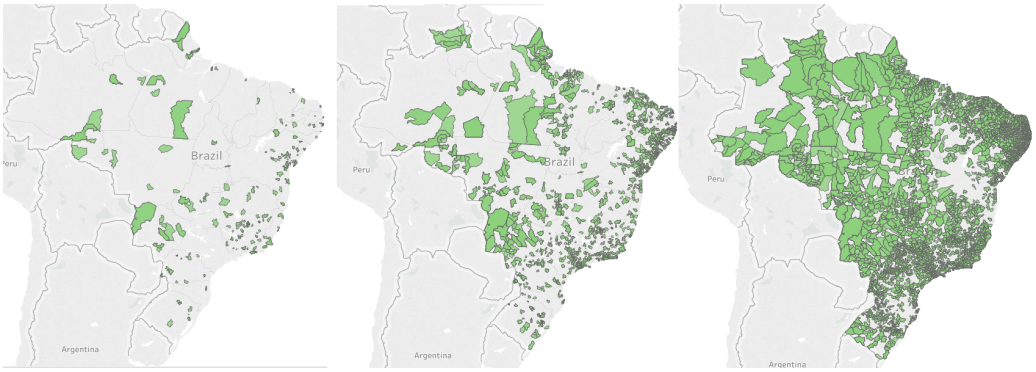

Figura 2.12: Exemplo de gráfico animação, exibindo dados em datas diferentes.

múltiplas origens como notas e prescrições médicas do médico, imagens médicas, dados laboratoriais, farmácia, dados administrativos, sensores [4].

Essa grande quantidade e variedade de dados torna oportuno a aplicação de técnicas de mineração para se descobri associações, compreender padrões nos dados. Sendo que ao descobrir alguma informações existe potencial de melhorar o atendimento seja no diagnostico ou nos tratamentos, a redução de custos operacionais ou de tratamentos e tudo isso pode resultar em mais vidas salvas pela medicina aliada a computação [3].

Um motivo que ratifica a necessidade da utilização das técnicas de mineração de dados é por exemplo, no área de saúde, é comum que especialista analisem periodicamente as tendencias e mudanças em dados de saúde com periodicidade [18]. Utilizando inúmeras bases dados para agregar mais e possuir uma maior abrangência do assunto desejado, a complexidade da tarefa irá aumentar fazendo com que técnicas convencionais de consultas ou de análise não seja tão eficientes para extrair informação. Utilizando mineração e algoritmos de analise, predição, classificação e entre outros, a informação seria obtida com maior facilidade, com possibilidade de automação e descoberta de padrões que até então não eram conhecidos.

Os desafios das bases médicas está primeiramente nas inúmeras formas na qual os dados podem ser armazenados, como imagens, sinais, dados clínicos entre outros [4]. Assim mostrando-se uma base pluriforme, no sentido de serem vários tipos de dados armazenados e de diferentes origens.

Dados clínicos oferecem o desafio da não padronização por meio do usuário final que irá inserir as informações, tornando suscetível ao erro e criando bases de dados sem regras de padrão.

As bases médicas estão em constante atualização, uma vez que, a todo momento acontecem novos atendimentos de pacientes e dados são inseridos nas bases, o que demostra ser um processo continua para sempre estar atualizando os conhecimentos obtidos [27].

Os dados médicos vão sempre referenciar um paciente, dos quais aqueles dados foram obtidos, o que implica em uma forte questão ética desde a privacidade do paciente ser preservada quanto para quais os fins do dados serão utilizados. Uma das principais questões da utilização desses dados é preservar a integridade moral do paciente nos quais seus aspectos de saúde estão sendo expostos.

O diagnóstico médico é considerado uma tarefa significativa, porém complexa, que precisa ser realizada com precisão e eficiência. Um método de automação de diagnostico seria de extrema importância e altamente benéfico uma vez que as decisões clínicas são normalmente tomadas com base na intuição e experiência do médico e não em banco de dados ricos em conhecimentos ocultos. O que torna um desafio criar modelos que possam auxiliar na tomada de decisões médicas com o objetivo de melhorar a qualidade das decisões clinicas.

Capítulo 3

Metodologia

3.1 Estado da arte

Este capítulo tem o objetivo de exibir o processo no qual esse trabalho foi baseado. Isto é, explica o passo a passo de como foi alcançado o objetivo final do projeto de analisar e utilizar técnicas de mineração na base de dados médica obtida.

Para ilustrar as etapas necessárias para chegar no resultado, foi gerado a figura 3.1 que exibe o fluxo do estado da arte. Na figura, é possível notar que foram necessárias 8 etapas principais, essas que serão explicadas individualmente nas seções seguintes.

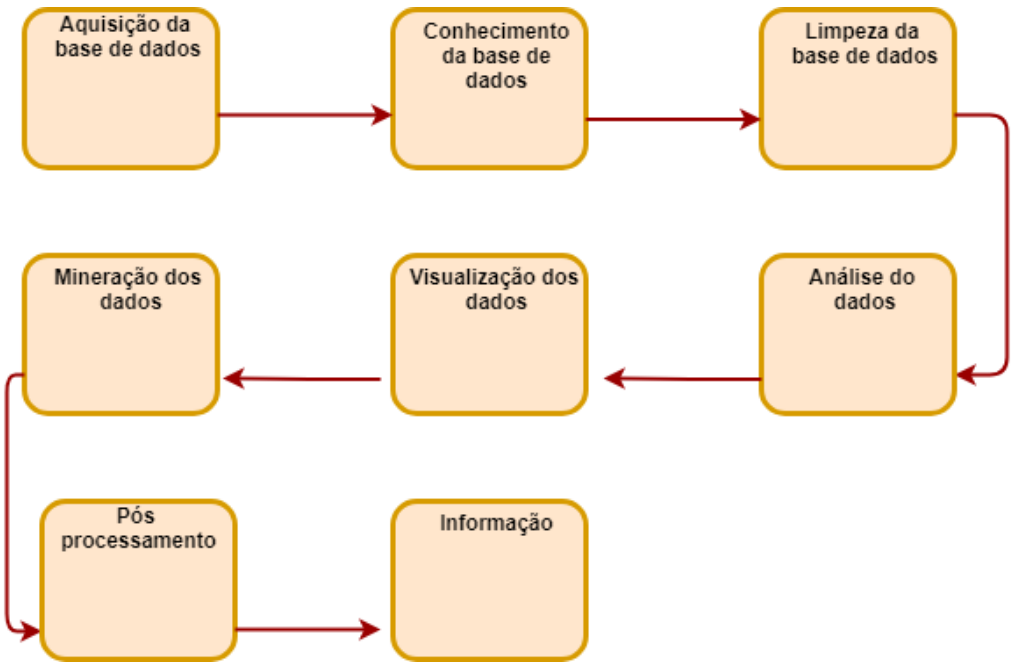

Figura 3.1: Fluxo realizado no desenvolvimento do trabalho proposto.

3.2 Aquisição da base de dados

O primeiro passo necessário para a realização do trabalho foi a aquisição da base de dados a ser manipulada. Os dados foram solicitados por meio do sistema eletrônico do serviço de informação ao cidadão (e-SIC), do Ministério da Saúde, por meio do protocolo 25820007257201805. O e-SIC permite que qualquer pessoa, física ou jurídica, encaminhe pedidos de acesso à informação, acompanhe o prazo e receba a resposta da solicitação realizada para órgãos e entidades do Executivo Federal [7].

O processo foi encaminhado para a ouvidoria do Sistema Único de Saúde (SUS) sob o protocolo 2862031 e reencaminhado para a coordenação geral de laboratórios em saúde pública -CGLAB/DEVIT/SVS/MS.

Foram solicitados dados referentes aos resultados dos exames laboratoriais as doenças *chikungunya* e *zika*, que são armazenados no sistema Gerenciador de Ambiente Laboratorial (GAL) (gerenciado pela Coordenação Geral de Laboratórios (CGLAB) - MS/SVS//DEVIT/CGLAB). A não identificação dos pacientes foi solicitada para que não houvesse exposição e todas as questões éticas fossem respeitadas. O período solicitado dos exames foi de 2010 a 2018 para que houvesse uma extensa base de dados para se trabalhar. A razão apresentada ao órgão para a liberação dos dados, foi a de realizar esse trabalho de análise de base de dados.

O prazo para a finalização de todo o processo foi de 16 dias corridos, prazo conforme estabelecido no art. 11, § 1º, da Lei nº 12.527/2011, que limita o prazo máximo de 20 dias podendo ser prorrogado por até 10 dias. O envio da base de dados foi por meio de um link enviado via e-mail, que havia sido cadastrado no portal e-SIC.

3.3 Conhecimento da base de dados

A base foi recebida no formato Comma-separated values (csv), que significa valores separados por vírgula, um arquivo regulamentado pela RFC 4180. Com a base de dados em mãos, a primeira etapa foi converter o arquivo de csv para XLSX que tornaria mais fácil a importação e tratamento, pois de tal modo já estaria familiarizado com as funções de importação.

Foi criado um módulo capaz de importar a base de dados para ser possível sua manipulação e análise. Para isso foi utilizado a linguagem de programação Python, na versão 3.7.3, com o auxílio da biblioteca *pandas* que é *open source*, que significa software aberto, e fornece estruturas de dados de alto desempenho e ferramentas de análise de dados para a linguagem de programação Python [28].

A base de dados recebida possuía um tamanho de 144.416 Kilobytes (KB), o que no primeiro uso do módulo de importação criado em python demorava mais de 10 minutos, na máquina de desenvolvimento, para criar um *data frame*. O que implicou na decisão de instalar e configurar na máquina de desenvolvimento com o software Jupyter [29] para auxiliar na programação do código que importaria e analisaria a base de dados.Pois com o software seria possível a programação modular e evitaria o retrabalho de a cada teste de código fosse necessário a importação de toda a base de dados novamente. Com o auxílio da interface Jupyter, a base de dados é carregada apenas uma vez e fica salva localmente em tempo de execução, o que economizaria tempo de desenvolvimento.

Com a primeira análise da base de dados, foi possível descobrir sua dimensão, que possui 591429 registros (linhas) de pacientes e 30 atributos (colunas), confirmando uma alta dimensionalidade e uma grande quantidade de dados.

Cada coluna foi analisada com o objetivo de uma tabela com informações a respeito de cada uma delas. Para isso foi criado um programa que importasse a base de dados, e após isso, utilizando funções da biblioteca *pandas*, fizesse a análise geral da base de dados e na sequência de cada uma das 30 colunas. Dessa análise de dados, foi gerada a tabela presente no apêndice A. Para detalhar a base de dados, foram utilizados os quatro atributos mostrados abaixo:

- Nome do atributo: Nome original da coluna da base de dados que foi recebida.
- Quantidade de variáveis: Foi contabilizado o número de valores distintos em cada coluna
- Tipo de dados: Foi exibido o tipo de cada variável (coluna). Para a confirmação, houve analise subjetiva, que consistiu em observar todos os dados de cada coluna.
- Descrição: Explicação sobre os dados de cada coluna, para que fosse possível saber seu significado. Para essa tarefa foi necessária contar com a ajuda de um funcionário do Ministério da Saúde e com a analise subjetiva dos dados.

Esses quatro itens citado acima são as colunas da tabela apresentada no apêndice A, que contem todos os atributos da base recebida. A coluna "quantidade"é uma abreviação para quantidade de variáveis.

Após a análise inicial da base de dados, foi possível concluir a respeito de sua composição. Foi gerada a tabela 3.1 com o tipo do dado e percentual que ele representa na base. É possível notar que o tipo string, um conjunto de caracteres, é dominante na base, o que pode se mostrar como um desafio devido ao fato de que dados de string podem ser inconsistentes causados pelos erros de digitação ou pelas diferenças nos formatos de dados [30].

Tabela 3.1: Composição da base de dados quanto ao tipo.

Tipo	Quantidade de colunas	Percentual
String	20	67%
Inteiro	4	13%
Data	6	20%

3.4 Limpeza da base de dados

A terceira etapa do processo foi realizar a limpeza da base de dados, pois com a primeira análise foi constatado inúmeras inconsistências nos dados, grande parte por a maioria ser do tipo *string*.

Essa etapa é primordial por causa da interferência que os dados não processados podem gerar nos algoritmos e nas análises futuras [1], logo toda a base de dados foi tratada a fim de evitar tais problemas.

Para realizar essa tarefa, foi utilizado a estrutura do módulo de análise de seção 3.3, e criado um módulo também na linguagem Python para realizar essa ação.

O fluxo de atividades apresentado na figura 3.2, ilustra o que foi seguido nessa etapa do projeto. Primeiramente foi utilizado a análise superficial de cada coluna para o entendimento do que cada dado significaria. Com esse conhecimento sobre o dado seria realizado a limpeza daquela coluna, a fim de de tornar o dado mais fácil de utilizar nos algoritmos de mineração ou na elaboração da visualização. Ao final do processamento de cada coluna, foi gerado uma nova base de dados a fim de salvar esse processo.

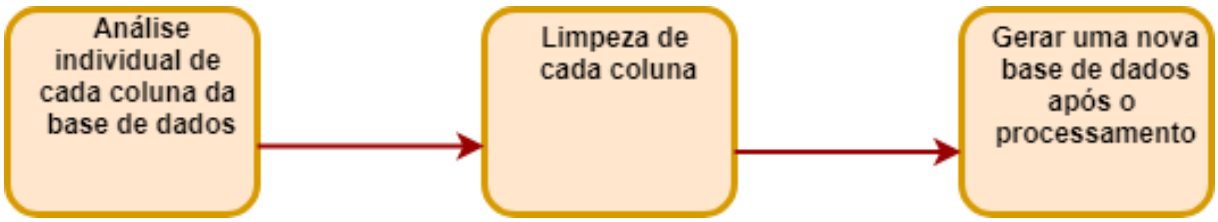

Figura 3.2: Fluxo seguido na etapa de limpar os dados.

Na grande maioria das colunas foi possível constatar que em inúmeras vezes o campo não havia sido preenchido, isto é, o dado aparecia apenas com um espaço. O que dificultaria na etapa de processamento com técnicas de mineração de dados e também na visualização por não saber o conteúdo daquele dado. Para isso foi realizado em cada coluna uma renomeação dos campos que eram nulos para "não informado". Colunas nais quais uma grande parte dos dados fossem "não informado"poderiam perder seu valor devido a falta de informação.

A parte trabalhosa seria analisar os dados duplicados com erros de ortografias entre si, isto é, verificar dados que possuíam a mesma informação, mas que foram inseridos na base com diferenças ortográficas. Em cada coluna foi realizado esse trabalho de verificar se havia dados com essa característica e assim seria necessário renomeá-lo para obter uma nomenclatura padronizada.

A tabela 3.2 exibe qual processo foi realizado em cada coluna. A coluna na qual o erro tiver sido tratado haverá um X, onde não houver significa que não houve tratamento daquele erro na coluna pela ausência dele, pois todos os casos detectados foram tratados.

Após a realização dos processos listados na tabela 3.2, foi realizado alteração do nome de algumas colunas, pois não seria possível manipular posteriormente caso fosse mantida a nomenclatura original. Os campos "1CampoResultado"e "2CampoResultado"foram renomeados para "PrimeiroCampoResultado"e "SegundoCampoResultado", respectivamente.

Com a análise dos campos de resultado, "PrimeiroCampoResultado"e "SegundoCampoResultado", foi possível notar que o "PrimeiroCampoResultado"informava o resultado final do exame, e caso não estivesse nessa coluna o resultado seria informado no campo "SegundoCampoResultado". Também foi constatado que em alguns casos os resultados não indicavam a presença do vírus e também não negavam, eles indicavam que era "inconclusivo".

Como existem inúmeras formas de mencionar que o resultado era positivo, negativo ou inconclusivo foi gerado um rotina no programa que iria categorizar os campos resultados para as três categorias existentes.

A tabela 3.3 exibe qual foi a categoria gerada para cada tipo de resultado existente na base. A classificação do resultado na categoria foi realizado de maneira subjetiva, isto é, foram analisados todos os tipos possíveis de resultados presentes na base e cada um foi adicionado em uma das três possíveis categorias.

Isso foi gerado para diminuir o tamanho da tabela visto que que seriam economizados bytes de dados de armazenamento e como não importava como era descrito o resultado apenas o sua categoria, a categorização acarretaria apenas em fatores positivos.

Para que fosse possível ter os dados de resultado em apenas uma coluna foi necessária analisar quando o resultado não estivesse na coluna "PrimeiroCampoResultado"fosse analisado a coluna "SegundoCampoResultado"e com isso o valor seria adicionado na coluna "PrimeiroCampoResultado". Para que ao final da função, fosse possível eliminar a coluna "SegundoCampoResultado", ajudando a diminuir a dimensionalidade da base. Os campos de resultado podiam variar dependendo do vírus estudado. Por exemplo para zika existiam os campos : "Resultado zika positivo"e "Zika Positivo"que foram categorizados como 1, seguindo a tabela 3.3. Para Chikungunya existiam, por exemplo, os campos "Resultado chikungunya inconclusivo"e "Resultado inconclusivo", o que foi considerado como

Tabela 3.2: Identificação dos tipos de erro em cada atributo.

Nome da coluna	**Campo Nulo**	**Digitação**
AgravodaRequisição	X	X
Amostra	X	
DatadaColeta		
DatadaLiberação		
DatadaSolicitação	X	
DatadeCadastro		
DatadeNascimento	X	
Datado1ºSintomas	X	
DescriçãoFinalidade	X	X
EstadodeResidência	X	
EstadoSolicitante		
Etnia	X	
Exame		
Finalidade	X	X
IBGEMunicípiodeResidência	X	
IBGEMunicípioSolicitante		
Idade	X	X
IdadeGestacional	X	X
MaterialBiológico	X	X
MaterialClínico	X	X
Metodologia	X	X
MunicipiodeResidência	X	
Nacionalidade	X	
PaisdeResidencia	X	
RaçaCor	X	X
Sexo	X	
TipoIdade		X
Zona		X
1CampoResultado		X
2CampoResultado	X	X

Tabela 3.3: Categoria criada para cada tipo de resultado.

Tipo de resultado	**Categoria**
Positivo	1
Negativo	0
Inconclusivo	2

categoria 2.

Ao final de todo os processos mencionados nessa seção, foi exportada a base de maneira a existir uma nova base processada, com uma redução de dimensionalidade e de tamanho. De modo que não apenas o que foi processado ajudasse a melhorar o desempenho do algoritmo, mas também sua nova dimensão. Essa nova base foi também gerada no formato xlsx e será utilizada como a base de dados tanto para a etapa de criar a visualização quanto para a aplicação de algoritmos de mineração de dados.

3.5 Análise dos dados

Essa etapa consiste em analisar os dados após o processo da seção 3.4. Para essa atividade foi desenvolvido um módulo em Python que iria analisaria os dados e gerararia as informações necessárias, e também seria utilizado software Tableau que será apresentado na subseção 3.5.1

A imagem 3.3 exibe o fluxo que a análise dos dados seguiu, recebendo como entrada os dados da limpeza, e seguindo duas principais análises, que são a utilização do software Tableau e o módulo python.

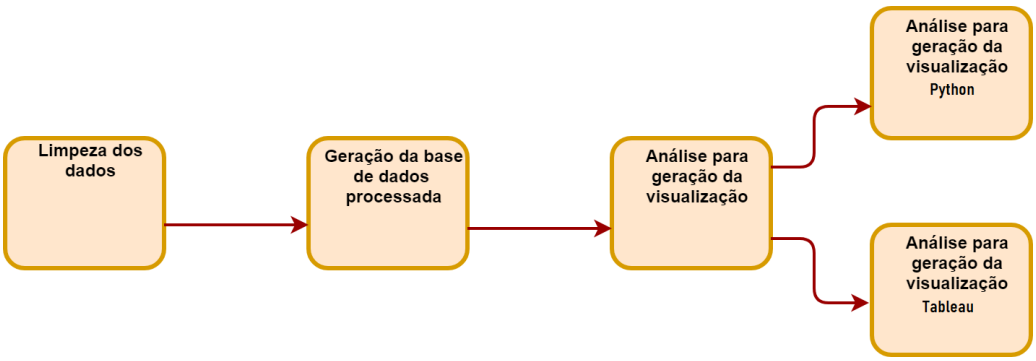

Figura 3.3: Fluxo da análise de dados.

Essas duas principais frentes de análise serão explicar nas subseções a seguir.

3.5.1 Tableau

Tableau Software é uma poderosa ferramenta iterativa de visualização de dados, no qual viabiliza a criação de gráficos e cruzamentos de informações de maneira clara [31] .

É uma ferramenta paga, entretanto, para estudantes e professores a companhia disponibiliza licença gratuita. Foi utilizado na realização do projeto o *Tableau Desktop*, que possibilita ter o produto instalado na máquina. Com ele foi possível o cruzamento de informações entre variáveis que se demonstraram importantes e essenciais para análise.

Tabela 3.4: Tabela divisão da base de dados por exames

Nome do exame	**Vírus**
Chikungunya, IgM	Chikungunya
Chikungunya, IgG	Chikungunya
Chikungunya, Biologia Molecular	Chikungunya
Chikungunya, Isolamento Viral	Chikungunya
Chikungunya, Teste Rápido IgM	Chikungunya
Pesquisa de Chikungunya , Isolamento Viral	Chikungunya
Zika,IgG	Zika
Zika,IgM	Zika
Zika, Biologia Molecular	Zika
Zika, Isolamento Viral	Zika
Zika, Teste Rápido IgG e IgM	Zika

Um dos grande benefícios do tableau foi a utilização do *tableau public*, uma comunidade no qual seria possível divulgar os resultados obtidos e deixar de maneira acessível e fácil gráfico iterativos.

Outra vantagem enorme que o tableau produziu foi a possibilidade do mapeamento dos vírus, isto é, a sua visualização gráfica sob o mapa geográfico do Brasil.

3.5.2 Módulo Python

Esse módulo foi desenvolvido com o objetivo de ter conhecimento sobre a base de dados, isto é, saber quantidade e percentual de erros, perfil da base de dados para algumas colunas. Também tem o objetivo de saber sobre a disposição dos dados para sua utilização nos algoritmos de mineração de dados.

Para isso, foram feitas três análises separadas:

- Análise da base completa, isto é, os 591.429 registros para que de tal forma fosse possível obter informações gerais;
- Uma análise separada somente para dados vinculados ao vírus da *zika*, para um melhor entendimento dele separado;
- E uma análise somente para dados do vírus da *chikungunya*.

Para realizar a distinção dos dados para cada vírus, foi utilizado como separador o campo Exame. Com esse campo foi possível verificar se o exame era realizado para o vírus da zika ou da chikungunya e a partir disso separar as bases de dados, conforme apresentado na tabela 3.4.

A primeira etapa foi a analise dos campos nulos da base e verificar se sua perda seria significativa para uso futuro. Uma vez que um atributo com muitos campos nulos

reduziria o escopo e abrangência da base de dados e poderia estar focando em apenas um tipo de informação, isto é, ao separar apenas o dado que aquele atributo possuiria valores a amostra que seria selecionada poderia induzir a apenas um resultado especifico.

O mapeamento do perfil dos pacientes foi realizado com objetivo de se conhecer os mais afetados para cada vírus o que tornaria mais fácil uma possível ação de combate ao vírus.

3.6 Mineração de dados

Após a análise e geração da visualização dos dados, é possível entender melhor a situação de que os vírus se encontram. A análise dos dados ajuda a entender também melhor a respeito da base e ter a noção dos atributos que poderiam influenciar na predição.

Como existiam campos que uma parte majoritária estava sem dados, isso ajudou o módulo de limpeza reduzir a dimensionalidade da base, uma vez que, foram retiradas da base limpa.

O primeiro passo foi a definição do objetivo da mineração de dados, e foi estipulado que seria desejado prever quando uma pessoa tivesse o vírus. Com isso também foi definido que para cada vírus seria feito uma predição separada, isto é, fazer programas distintos para cada vírus.

A divisão da base de dados para cada vírus está de acordo com o apresentado na figura 3.4. É possível observar uma grande quantidade de registros para cada vírus, o que é um grande ponto positivo na hora do treinar e testar os algoritmos. A base de dados foi separada para cada vírus a fim de trabalhar individualmente

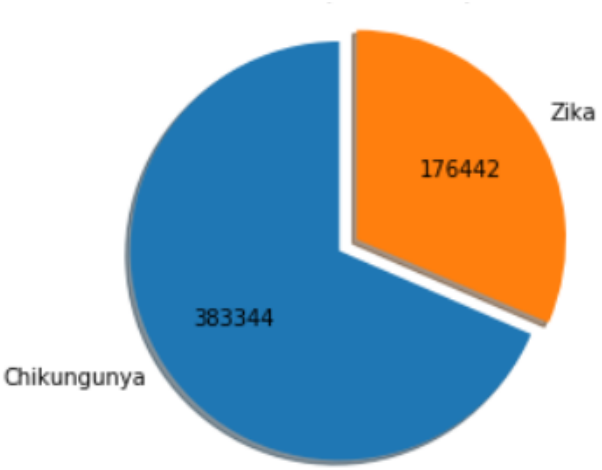

Figura 3.4: Composição da base de dados por vírus: em azul Chikungunya e em laranja Zika.

3.6.1 Base de dados desbalanceada

O primeiro desafio enfrentado foi o desbalanceamento da base de dados de cada vírus, como é apresentado na figura 3.5. Existem classes predominantes em ambas as bases, sendo que para *zika* a classe resultado inconclusivo (2) é predominante e a classe resultado positivo (1) muito menor que as demais. Para *chikungunya* também ocorre o desbalanceamento, a classe negativo (0) é predominante e a classe inconclusivo (2) a mais rara.

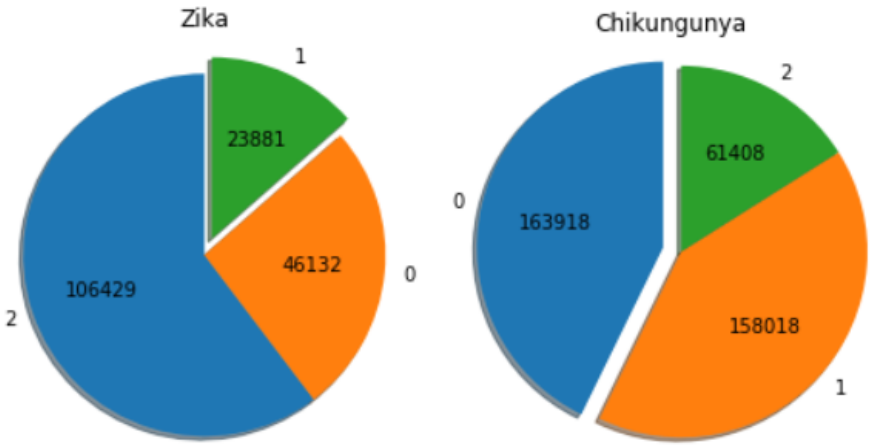

Figura 3.5: Composição da base de dados por vírus.

O fato de as classes serem muito desbalanceadas se torna um problema por não treinar adequadamente o algoritmo adequadamente para as classes mais raras, isto é, as que possuem menos quantidade de registros para cada base. Para isso foi necessário utilizar-se da técnica de amostragem. Foi selecionado de maneira aleatória, a quantidade de cada classe utilizando como referencia a quantidade da menor classe daquela base.

A figura 3.6 exibe o resultado da base balanceada para cada vírus. Para o vírus chikungunya é possível observar que cada classe contem 61.408 registros para cada tipo de resultado, e ao analisar a figura 3.5 é possível observar que a classe com menor quantidade é a classe inconclusivo (2) com exatamente 61.408 registros.

O mesmo acontece para o vírus zika, é possivel observar na figura 3.6 que cada classe possui 23.881 registros, o mesmo valor da menor classe apresentada na figura 3.5, na classe positivo (1).

Esse técnica serve para melhorar os resultados para as classes mais "raras". As bases balanceadas de cada vírus foram utilizadas tanto para treinar o algoritmo quanto para validar.

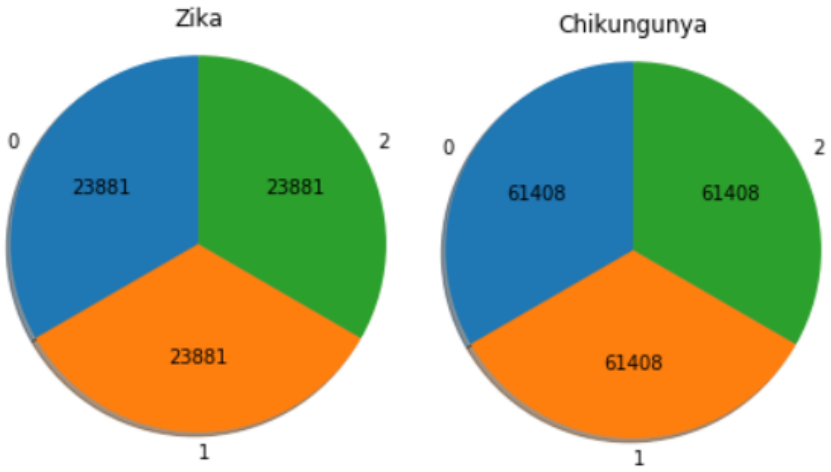

Figura 3.6: Distribuição por resultado das bases de dados após amostragem.

3.6.2 Algoritmos

Não existe um modelo ou algoritmo de mineração de dados que possa ser aplicado em qualquer lugar [32], a escolha dos algoritmos utilizados veio após leitura de artigos relacionados na área. Mesmo que não exista um consenso entre qual é o recomendado para área médica, por ser uma área vasta, alguns são sempre mencionados e testados. Em [5] foi possível observar a melhor acurácia para árvore de decisão. Por conseguir trabalhar com dados de grande dimensões e ser normalmente utilizado o primeiro algoritmo testado foi a árvore de decisão. É um algoritmo robusto e altamente utilizado, existindo sistemas já validado para predição de algumas doenças, por exemplo do coração [33]. Também já foi utilizado para prever diabete [34]. E também utilizado para a predição de diabete foi utilizado e obteve melhores resultados quando comparado a outro em [35]. A ideia é gerar um modelo capaz de auxiliar a tomada de decisões médicas.

O primeiro passo a ser realizado foi a divisão da base de dados para teste e para treino. A configuração utilizada foi 10% da base balanceada para teste e 90% também da base balanceada para treinar em ambos os casos, para cada base a quantidade foi a informada pela tabela 3.5. É possível notar que existe também a validação pela base dados completa para cada vírus, isto é, além de validar o modelo para a base de dados apresentada na figura 3.6 também foi testado o modelo na base de dados completa de cada base, mostrado pela figura 3.4.

Após o uso da árvore de decisão, por motivos de comparação foi utilizado o algoritmo que utiliza um conjunto de árvores de decisão, o que poderia melhorar o resultado e com isso também foi testado o *random forest*. Para ele foi utilizado a mesma configuração das bases de dados apresentada na tabela 3.5.

Tabela 3.5: Quantidade de registros dos dados por base.

Função	**Zika**	**Chikungunya**
Treino	64.551	165.801
Validação com a base balanceada	7.173	18.442
Validação com a base completa	17.6442	383.344

E também foi utilizado outra varição da árvore de decisão e um pouco similar ao *random forest*, o *Extra Tree*, que também seguiu a disposição dos dados da tabela 3.5.

Os testes entre os algoritmos ocorreram seguindo a distribuição da tabela 3.5. Com o objetivo de confirmar o problema das bases desbalanceadas foi testado a distribuição 10% de treino e 90% de teste utilizando a quantidade total de registros a fim de ratificar o uso das bases balanceadas.

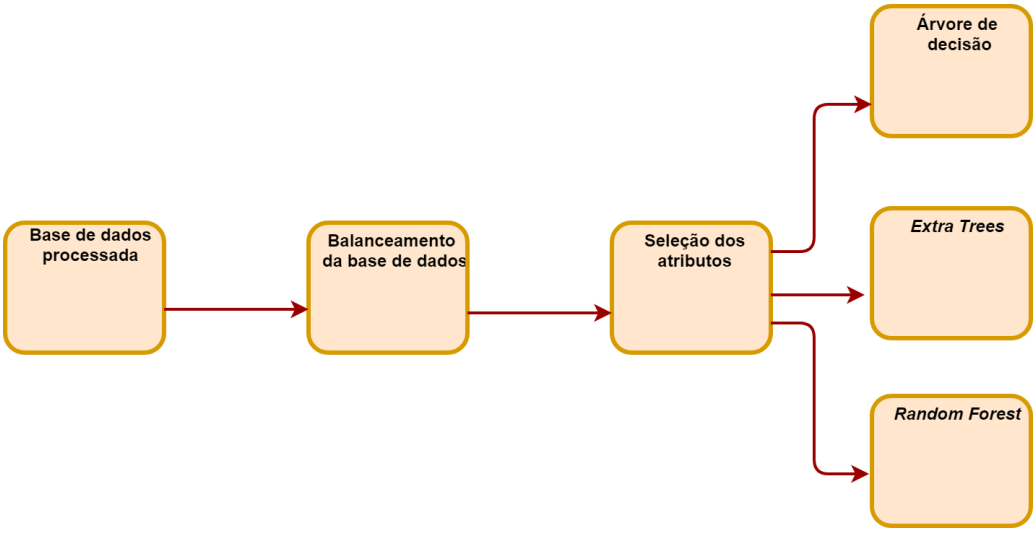

Figura 3.7: Composição da base de dados por vírus.

O fluxo do uso dos algoritmos está apresentado na figura 3.7, nela é possível observar que antes de utilizar os algoritmos para a predição do resultado é necessário a escolha dos atributos que serão utilizados para isso, logo, o modo utilizado para a seleção dos atributos foi verificando a importância de um conjunto de atributos e dentre eles escolhendo os que forem identificados com os melhores valores. As importâncias foram geradas por meio de um metodo da random forest.

A fim de obter os melhores atributos, foi utilizado a importância da *random forest* com alguns atributos que após a análise poderiam influenciar no resultado e desses atributos foram retirados os que seriam utilizados no modelo. Todo esse processo descrito foi realizado nos módulos python criado, um para cada vírus. O módulo utilizou-se da implementação do algoritmo *random forest* da biblioteca Scikit learn [36]. O resultado

da importância dos atributos para zika é apresentado na figura 3.8 e para chikungunya na figura 3.9, quanto mais próximo o valor de 1, mais relevância aquele campo possui no resultado final. Foram calculados a partir da importância gerada pelo algoritmo random forest dos atributos. A tabela 3.6 exibe os valores de cada importância para os atributos de zika e da chikungunya.

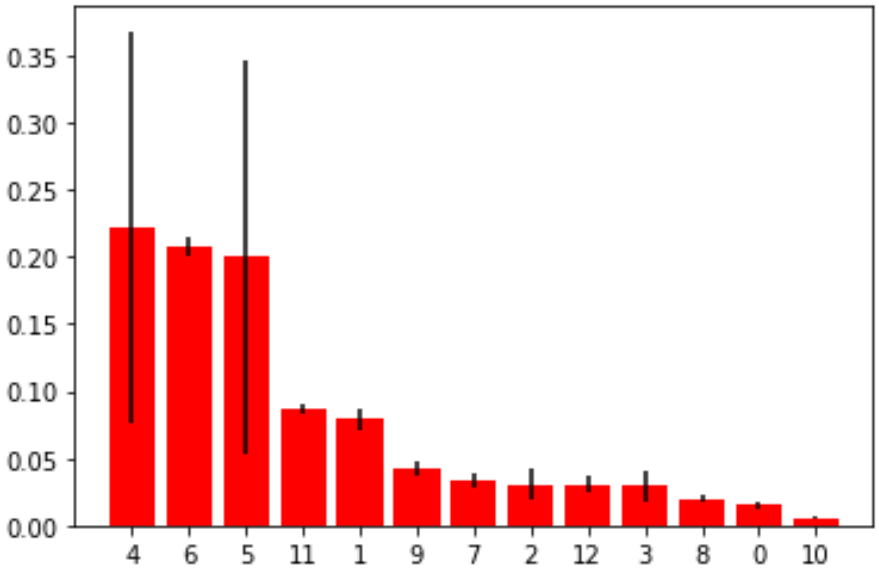

Figura 3.8: Importância dos atributos para *zika*.

Com a tabela 3.6 foi possível escolher quais seriam os atributos utilizados nas predições. Para cada vírus foram escolhidos atributos diferentes, tendo em vista que a importância varia para cada uma das bases de dados. Foram utilizado os 9 melhores atributos de cada na utilização dos três algoritmos.

3.7 Pós processamento

Com a análise dos dados gerados pelos módulos python, os resultados obtidos do software tableau e o uso dos algoritmos de mineração de dados é necessário um processamento posterior seja para a visualização final, no caso do algoritmos.

Para validar os algoritmos de mineração de dados, as tabelas 4.4 e 4.3 comparando os seus resultados e exibindo de maneira mais clara como foi o desempenho de cada um deles, para as métricas escolhidas.

Também foi gerada a matriz de confusão de cada um dos algoritmos. Foram geradas tanto de maneira bruta apenas exibindo o que foi preditos quanto uma imagem que mostra de maneira mais intuitiva os valores para cada classe.

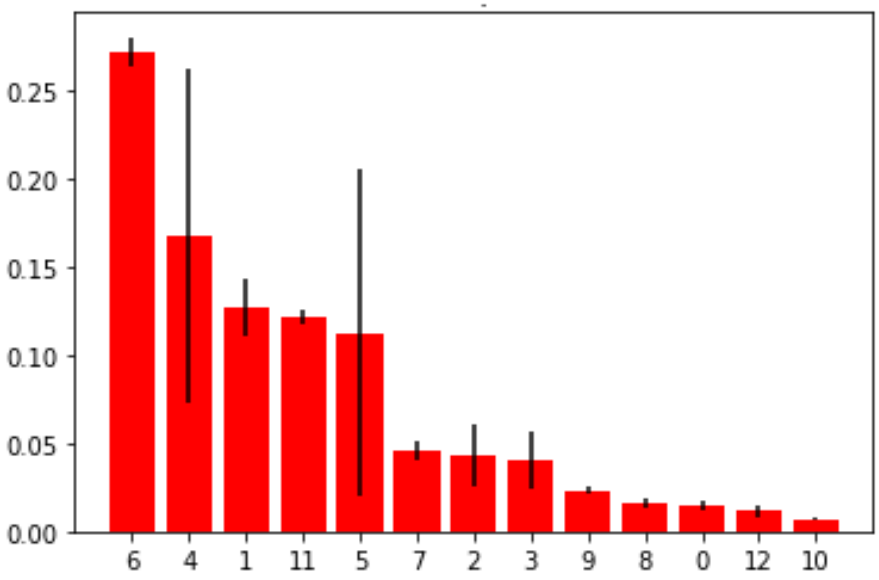

Figura 3.9: Importância dos atributos para *chikungunya*.

3.8 Informação

A informação é gerado após o pós processamento, no caso da mineração dos dados é devido a criação das matrizes de confusão e os valores das métricas para avaliar os desempenhos obtidos.

Os módulos de análise de dados, geram a informação tanto a respeito da situação dos vírus no quesito de entender o problema quanto referente a qualidade da base de dados, por exemplo, campos que poderiam ter sido utilizados na predição dos vírus entretanto, devido a sua ausência em uma grande quantidade de registros inviabiliza seu uso.

Tabela 3.6: Atributo e sua importância para zika e chikungunya.

Número	**Nome**	**Importância Zika**	**Importância Chikungunya**
0	Sexo	0.014986	0.014535
1	IBGEMunicípioSolicitante	0.078878	0.126372
2	EstadodoSolicitante	0.030598	0.042783
3	EstadodeResidência	0.028994	0.040258
4	Exame	0.221810	0.167057
5	Metodologia	0.200139	0.112497
6	Idade	0.207667	0.271591
7	AgravodaRequisição	0.032746	0.046238
8	Finalidade	0.020089	0.015780
9	IdadeGestacional	0.041641	0.023405
10	Nacionalidade	0.005534	0.006716
11	MunicipiodeResidência	0.086855	0.121481
12	MaterialBiológico	0.030061	0.011287

Capítulo 4

Resultados

4.1 Proposta

Tendo em vista a gravidade que foi a epidemia dos vírus no Brasil nos últimos 5 anos [2], surge a necessidade de um estudo com o intuito de entender como foi a propagação desde os primeiros casos até o cenário atual. Tendo em posse os dados de exames em todo o país desde o inicio dos casos e com conhecimento de técnicas de mineração e de análise de *big data* foram realizadas atividades sob os dados para extrair informação que pudesse ajudar no entendimento do problema como um todo e também no auxilio de tomada de decisão no diagnóstico.

Esse capítulo está divido em 3 seções, a 4.2 irá exibir informações referentes a qualidade dos dados, que foram analisados durante a etapa de pré-processamento da base de dados. Já na seção 4.3 será exibido os resultado obtidos referente a análise dos dados tanto pelo módulo Python tanto quanto os obtidos no software tableau. Na seção 4.4, será apresentado mapas referentes a propagação dos vírus no Brasil, mapa de calor do país com as áreas mais afetadas entre outras informações. E por último, na seção 4.5 será exibido os resultados obtidos com os algoritmos de predição.

4.2 Qualidade dos dados

Essa seção irá apresentar a análise referente a qualidade dos dados, grande parte dos resultados apresentados nessa seção foram obtidos durante o pré-processamento da base e durante a análise. Já era esperado que a base de dados não fosse perfeita, já que normalmente não são criadas para análise, entretanto a ausência de dados em alguns campos fizeram com que não pudessem ser incluídos na análise ou nas predições.

Conforme foi apresentado na tabela 3.2, é possível observar que quase toda a base de dados teve que ser processada a fim da correção de inconsistências. A tabela 4.1 apresenta

Tabela 4.1: Percentual de atributos com erro.

Erro	Percentual na base de dados
Ortografia	43.3%
Campos Nulos	70%

um percentual de atributos que possuíam erros para a base de dados completa, isto é, sem dividi-la para cada vírus.

É possível observar que uma grande parte da base de dados precisou processada. Na questão de campos nulos 70% dos atributos possuía pelo menos um percentual dos seus campos salvos como nulo. Informações podem ter sido perdidas devido a essa quantidade de dados omitidos.

Já os erros de ortografia estavam quase em 50% dos atributos, o que indica a não existência de uma padronização dos usuários. Um alternativa possível seria a modificação do sistema que armazena os dados.

Como as bases de dados foram utilizadas em algoritmos de predição de maneiras separadas, os dados sobre a qualidade das bases foram divididos em 2 subseções.

4.2.1 Zika

Os dados obtidos referente a qualidade de dados foram obtidos a partir do módulo Python desenvolvido. Ele que primeiramente tratou a ocorrência dos erros, foi utilizado para ter a noção do estado da base de dados.

A Figura 4.1 apresenta um caso no qual um atributo possui uma grande quantidade de dados ausentes. O caso apresentado é o atributo "zona"que não pode ser utilizado para análise já que quase 47% dos dados estão ausentes.

A Figura 4.2 também apresenta informação a respeito de dados que não foram inserido na base de dados e é possível notar que mais da metade dos dados estão ausentes. A raça do paciente não pode ser analisada como um fator dos vírus devido a maioria dos dados (59%) não estarem presentes na base.

4.2.2 Chikungunya

Também utilizando o mesmo módulo apresentado na subseção 4.2.1, somente alterando a base de dados para tratar dados agora da chikungunya, foi realizado o estudo a respeito a qualidade dos dados.

Novamente o atributo "zona"teve um grande percentual de dados omitidos, conforme apresentado pela Figura 4.3. No total 44% dos dados de zona são nulos. Já para o

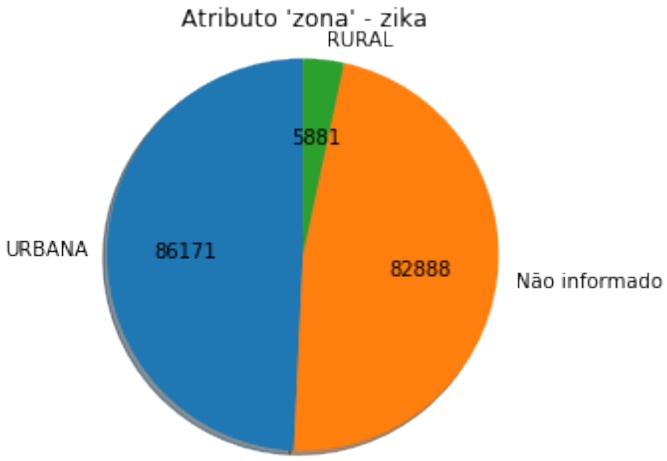

Figura 4.1: Distribuição dos dados para o atributo 'zona' para dados da zika.

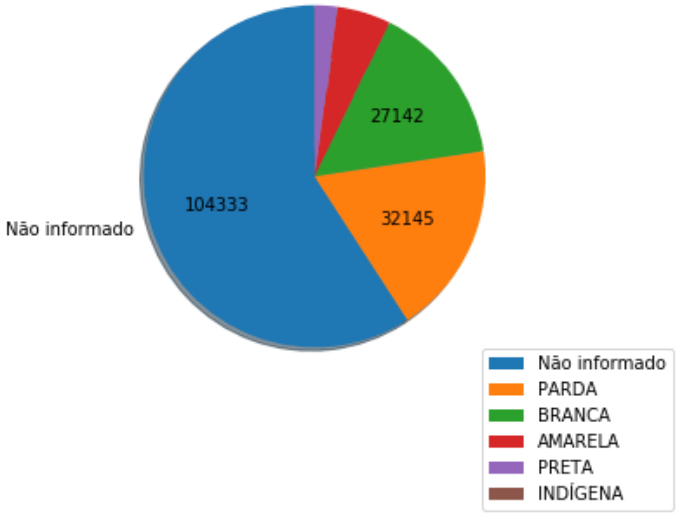

Figura 4.2: Distribuição dos dados para o atributo 'raça' para dados da zika.

atributo 'raçacor' a quantidade de dados nulos era ainda maior, 54% dos dados não estavam preenchidos.

4.3 Entendimento

Com os dados já tratados foi possível entender o cenário por completo no qual o Brasil se encontra diante dos vírus, foi também possível estabelecer o perfil por alguns atributos presentes na base. Com o uso do módulo Python e do software tableau foi possível a criação de gráfico que exibissem as informações que serão apresentadas nessa subseção.

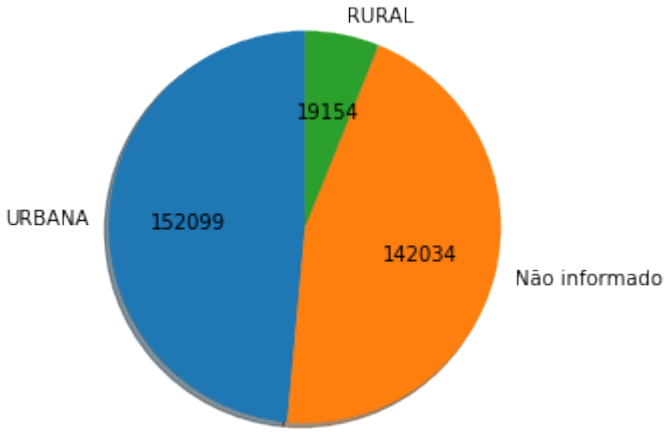

Figura 4.3: Distribuição dos dados para o atributo 'zona' para dados da chikungunya.

Primeiramente, foi estabelecido o perfil do público dos exames, com isso foi possível obter que para ambos os vírus o público em relação ao sexo, em sua grande maioria era o feminino. A Figura 4.4 exibe três gráfico mostrando o percentual para do público geral da base, para os que realizaram exame para chikungunya e na sequência para zika. Quase 70% de toda a base de dados é referente a registros de pessoas do sexo feminino.

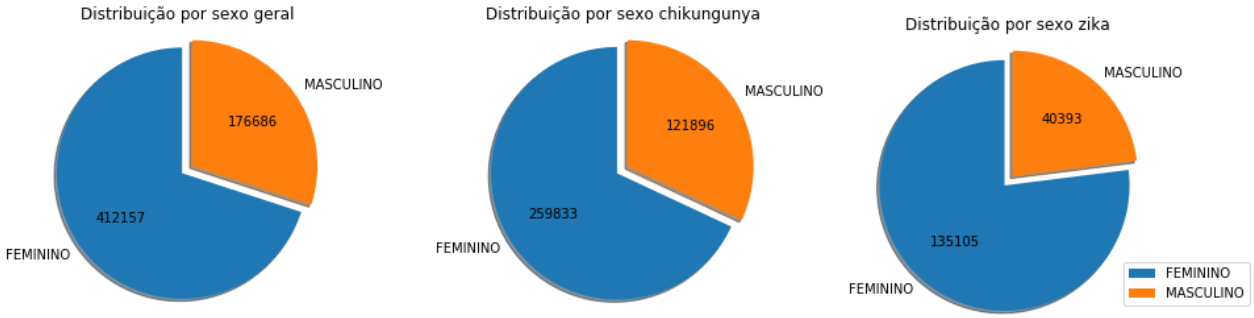

Figura 4.4: Cruzamento de dados em relação ao vírus com o resultado do exame pelo sexo do paciente.

Com essa grande maioria do sexo feminino, foi observado qual o percentual delas estariam grávida, uma vez que, os vírus estavam sendo relacionados a síndrome de Guillain–Barré (SGB) e a microcefalia. Foi observado que um percentual de 15% estavam grávidas, conforme apresentado na Figura 4.5

Com tais informações, foi relacionado o vírus, com o sexo e com o resultado exame para de tal forma ser possível ver o perfil que foi mais afetado. Esse cruzamento de dados resultou na Figura 4.6. Onde é possível observar que pacientes do sexo feminino foram o mais atingidos para a chikungunya, sendo a maior quantidade relatada como confirmado

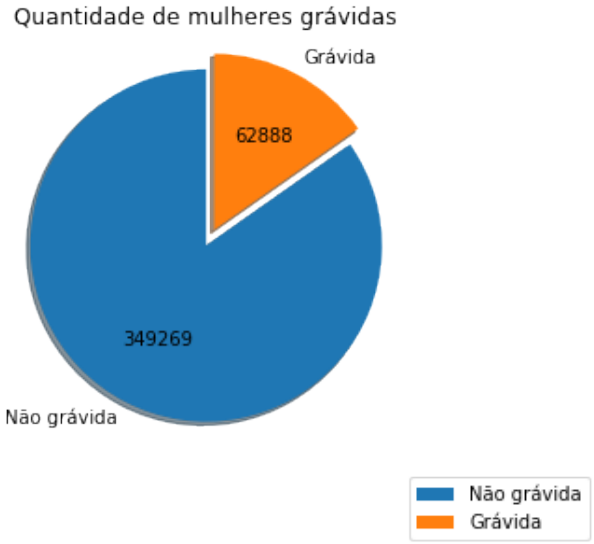

Figura 4.5: Cruzamento de dados em relação ao vírus com o resultado do exame pelo sexo do paciente.

o vírus. A escala da cor vermelha exibe que quanto mais próximo do vermelho (112.093 casos) maior foi a quantidade de registros naquela categoria. Resultados inconclusivos para zika foi a maior categoria da base de zika, o que mostra que o surto pode ter sido ainda maior, uma vez que mais de 160 mil casos não tiveram resultados definitivos para os vírus.

Com o objetivo de descobrir mais a respeito do perfil dos pacientes que realizaram o exame, foi criado o histograma dos pacientes de cada vírus a fim de verificar se havia diferença também entre os dois. A Figura 4.7 apresenta o histograma das idades para ambos os vírus, é possivel notar que a curva das idades para ambos os vírus é bem semelhante, a de chikungunya por ter tidos mais casos está registrando uma maior quantidade mas os picos foram os mesmo do vírus zika. Os *Outliers* foram removidos devido a impossibilidade de existir pacientes com 180 anos.

Por possuírem quase ou mais da metade dos dados nulos, não foi gerada as análises dos atributos "RaçaCor"para exibir as raças mais afetadas no país e também do atributo "Zona"para verificar as áreas mais afetadas. A decisão de não inserir foi devido a não representação 'confiável' do cenário devido a grande quantidade de dados ausentes.

4.3.1 Dados abertos deste trabalho

Com o objetivo de deixar todas a análises acessíveis a todos e ainda tornar possível que qualquer pessoa possa fazer as suas próprias conclusões, foi utilizado o *tableau public*, onde é possível publicar as visualizações de dados criadas através do tableau. As análises

Virus	Sexo	resultado Negativo	Positivo	Inconclusivo
Chikungunya	FEMININO	107.043	112.093	40.697
	MASCULINO	55.927	45.788	20.181
Zika	FEMININO	31.101	20.226	83.778
	MASCULINO	14.921	3.574	21.898

3.574 112.093

Figura 4.6: Cruzamento de dados em relação ao vírus com o resultado do exame pelo sexo do paciente.

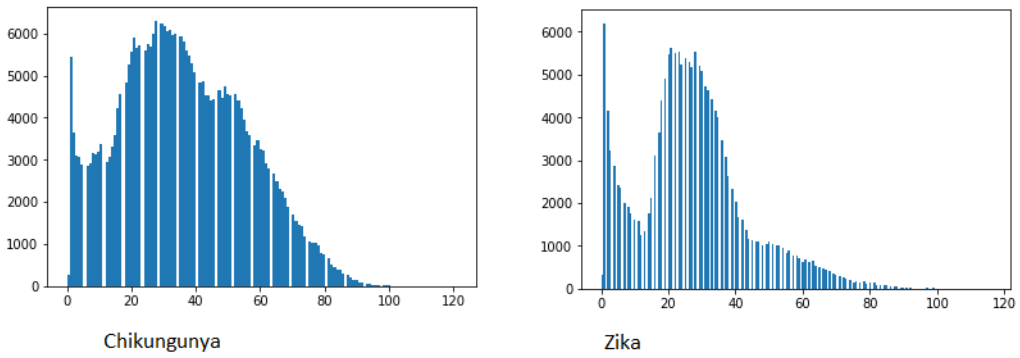

Figura 4.7: Histograma das idades dos pacientes que realizaram exames para chikungunya e zika.

estão disponíveis perfil gabriel.pereira.pinheiro[0] não sendo necessário se cadastrar para acessar.

Foram criados várias corelações entre os dados com filtros dinâmicos para deixar o usuário livre para decidir o que deseja analisar.

[0]https://public.tableau.com/profile/gabriel.pereira.pinheiro

4.4 Análise espaço-temporal

O primeiro caso encontrado na base de dados é de 2014 para o vírus *chikungunya*, pouco mais de 5 anos depois, o número de casos suspeitos chegou perto dos 600 mil, o que ratifica a epidemia, pelo aumento exponencial dos casos. Com esses dados, foi gerado a Figura apresentada pela Figura 4.8, que exibe a quantidade de casos em 6 épocas diferentes, a primeira Figura é registrando o primeiro mês onde se registrou o primeiro casos, as imagens seguintes apresentam o aumento na quantidade de casos no intervalo médio de 1 ano. Ao final é possível notar que no total até fim de 2018 foram 559.786 exames realizados no Brasil para ambos os vírus. Isso mostra como os vírus se espalharam rapidamente por todo o país.

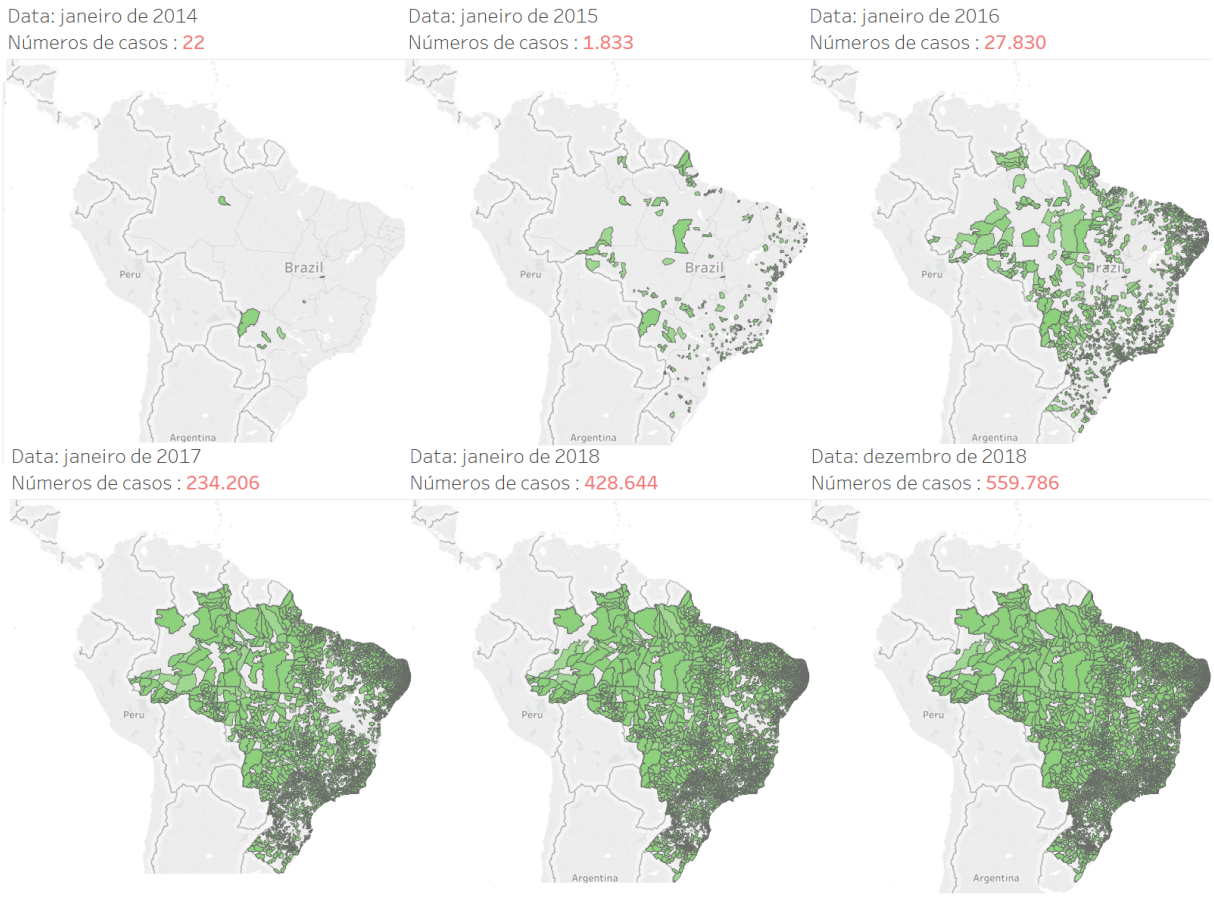

Figura 4.8: Animação da evolução do casos para ambos os vírus no Brasil .

Diferentemente da Figura 4.8, a Figura 4.9 apresenta os casos confirmados do vírus da *zika*, ou seja, apenas os que os exames deram positivos. Os 13 primeiros casos confirmados foram dia 12 de janeiro de 2015 o que ratifica com o encontrado em [12]. Já no início de 2016 já haviam sido confirmados mais de 1400 e até o final de 2018 já tinham

sido confirmados quase 24 mil casos. Esses valores podem ainda ser maiores devidos a quantidade de casos que tiveram seus resultados inconclusivos. O maior salto foi de 2016 para 2017, no qual houve um aumento de 14181 (1003%) casos confirmados.

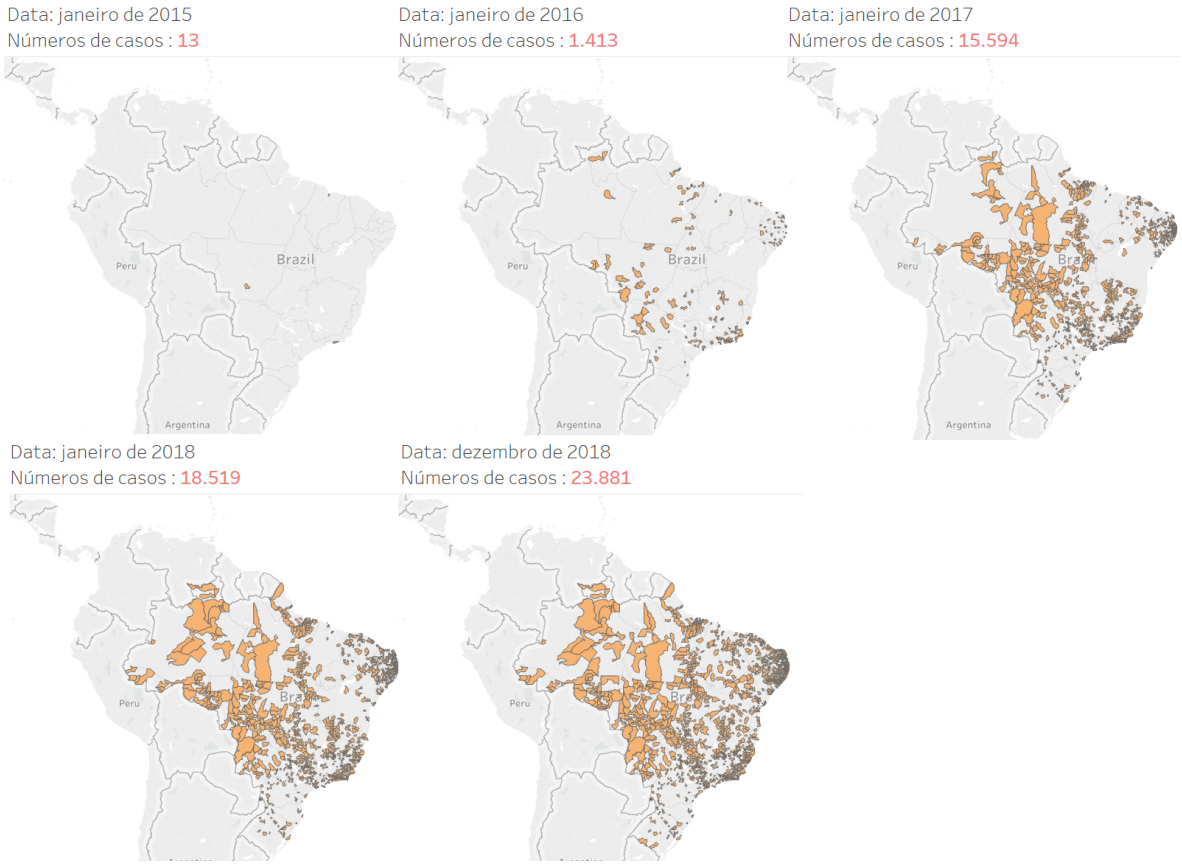

Figura 4.9: Animação da evolução do casos confirmados para o vírus *zika* no Brasil de 2015 a 2018.

A Figura 4.10 é referente a evolução do vírus *chikungunya*. Seu primeiro caso confirmado foi confirmado dia 12 de janeiro de 2014 no estado do Amazonas, em uma mulher de 47 anos. De 2016 para 2017, o número de casos confirmados cresceu de 4.283 para 60.227, mais de 55 mil novos casos. No intervalo de 2017 para 2018, mais 74.189 novos casos. Até o final de 2018, sendo esse a última data presente na base de dados, haviam sido registrados quase 160 mil casos positivos para o vírus.

Com esses dados foi possível gerar a Figura 4.11, que apresenta a quantidade de exames realizados para ambos os vírus em cada mês existente na base de dados. Quanto mais escuro o tom de azul maior foi a quantidade de casos. É possível observar uma maior concentração no meses mais próximos ao meio do ano. O mês que ocorreu a maior quantidade de exames foi agosto de 2016, que no total foram realizados 23.271 exames.

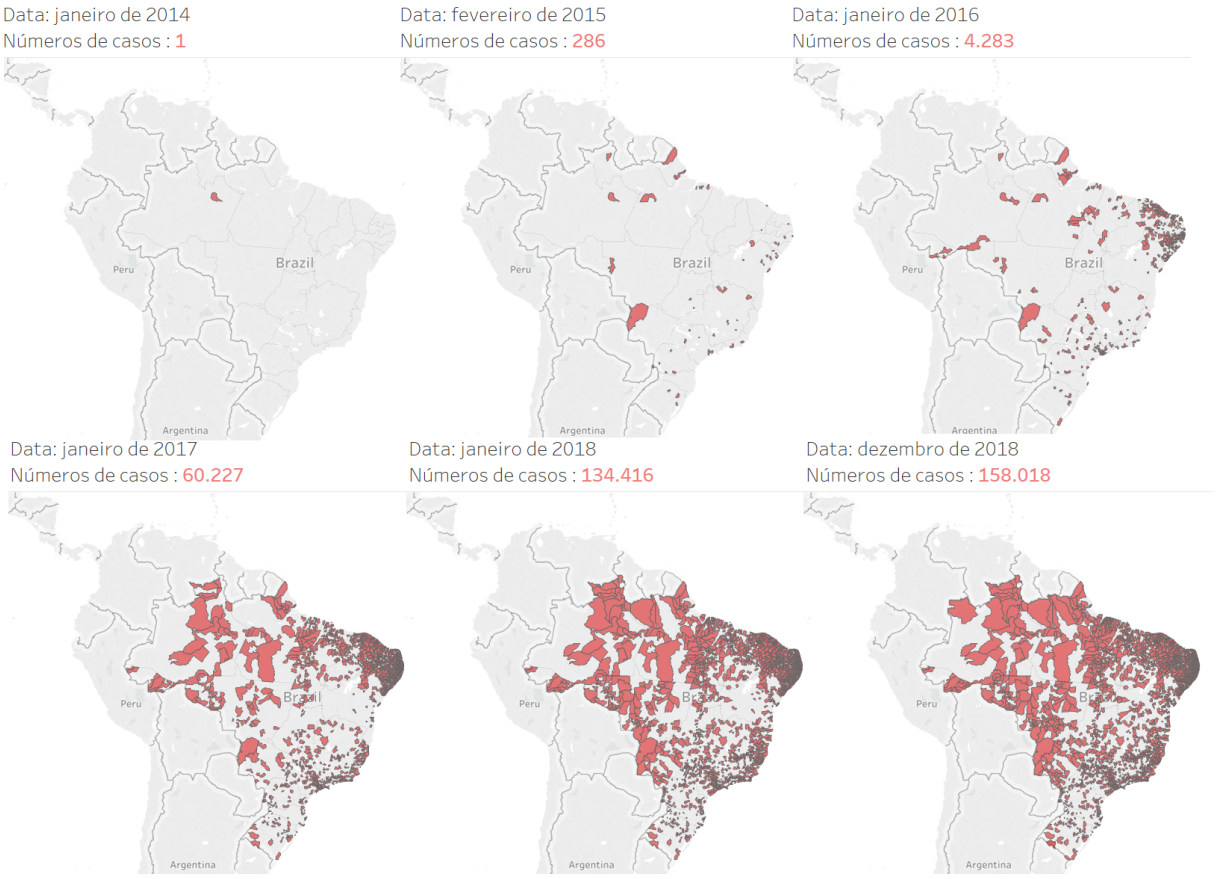

Figura 4.10: Animação da evolução do casos confirmados para o vírus chikungunya no Brasil de 2014 a 2018.

	Mes											
Ano	1	2	3	4	5	6	7	8	9	10	11	12
2014	22	9	19	4	20	16	34	20	100	344	276	285
2015	684	890	1.104	940	1.194	1.519	1.386	1.075	1.394	1.648	1.462	3.161
2016	10.224	8.396	17.562	17.399	16.359	21.657	19.275	23.271	16.585	13.500	15.388	20.662
2017	16.322	15.749	22.654	14.198	22.719	19.632	18.738	19.058	14.076	13.463	13.967	11.474
2018	11.345	13.840	14.485	21.010	20.785	17.112	15.752	17.233	14.814	11.232	8.903	4.983

Figura 4.11: Quantidade de exames por mês.

Em relação as áreas mais afetadas, foi criada a Figura 4.12, com o intuito de saber onde são os focos dos vírus. A região nordeste teve uma grande acúmulo de casos, juntamente com parte da região sudeste.

Também foi criado uma Figura que apresenta o avanço dos exames do vírus no Brasil. O vídeo está disponível no youtube [1], e apresentação o crescimento na quantidade de

[1]Link para vídeo: https://www.youtube.com/watch?v=5lVrQjMSt1g

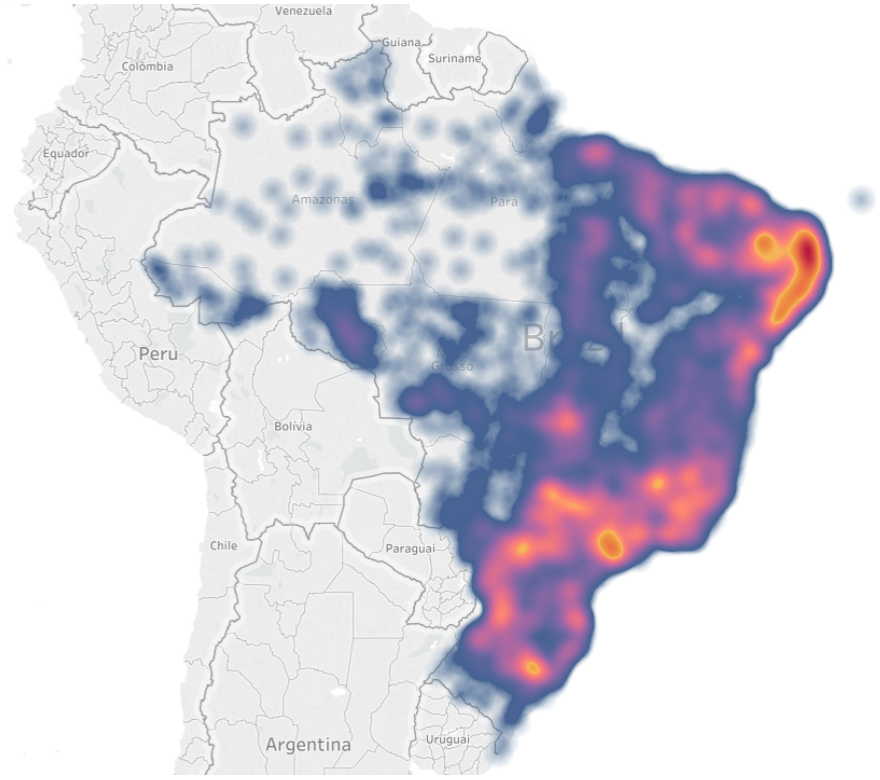

Figura 4.12: Mapa de calor das áreas com a maior incidência de exames para ambos os vírus.

Tabela 4.2: Cenários de testes executados.

Cénario	**Descrição**
Cenário 1	10% da base de dados do vírus balanceada
Cenário 2	Base de dados do vírus completa

exames de janeiro de 2014 a dezembro de 2018.

4.5 Predição

Essa seção tem como objetivo apresentar os resultados algoritmos utilizados nos diferentes cenários. No total foram testados 3 algoritmos a fim de realizar uma comparação da melhor escolha para esse caso. Como explicado os algoritmos foram treinados com 90% bases de dados para cada vírus balanceadas, apresentada pela Figura 3.6.

O objetivo no uso dos algoritmos é a tentativa de predizer o resultado de uma exame com algumas características do paciente a fim de ajudar na tomada de decisões médicas.

Para o teste foram realizados dois cenários, apresentado na tabela 4.2. No primeiro cenário de teste foram utilizados 10% da base balanceada e já no segundo cenário, foram utilizados toda a base de dados do vírus.

Tabela 4.3: Resultados dos algoritmos para cenário 1 para vírus zika

Algoritmo	Métrica	Classe		
		0	1	2
Decision tree	Recall	0.85	0.51	0.69
	Precision	0.83	0.54	0.67
	F1-score	0.84	0.53	0.68
Random forest	Recall	0.88	0.59	0.75
	Precision	0.84	0.63	0.73
	F1-score	0.86	0.61	0.74
Extra tree	Recall	0.87	0.6	0.73
	Precision	0.83	0.62	0.74
	F1-score	0.85	0.61	0.73

Tabela 4.4: Resultados dos algoritmos para cenário 2 para vírus zika

Algoritmo	Métrica	Classe		
		0	1	2
Decision tree	Recall	0.85	0.51	0.69
	Precision	0.88	0.24	0.90
	F1-score	0.87	0.33	0.78
Random forest	Recall	0.92	0.85	0.77
	Precision	0.94	0.43	0.97
	F1-score	0.93	0.57	0.86
Extra tree	Recall	0.92	0.86	0.75
	Precision	0.93	0.42	0.98
	F1-score	0.93	0.56	0.85

Utilizando-se desses dois cenários, os modelos foram treinados e testados em cada um dos cenários. A tabela 4.3 e 4.4 apresentam os resultados dos três algoritmos para a zika. A três métricas utilizadas para avaliação foram *recall*,*precision* e *f1-score* conforme apresentado na subsubseção 2.3.3.2.

A Figura 4.13 apresenta a matriz de confusão de cada cenário e de cada algoritmo juntamente com as métricas deles. Utilizando-se o *recall* como principal métrica, uma vez que, é bastante utilizado como métrica de modelos quando há um alto custo associado ao *false negative*, uma vez que no caso médico, diagnosticar um paciente com um resultado negativo sendo que o resultado correto seria positivo poderia ter consequências.

Baseando-se nas tabelas 4.3 e 4.4 e na Figura 4.13 é possível observar uma grande melhora quanto ao algoritmos que combinam árvores de decisão. A melhora foi principalmente para a classe mais rara do conjunto de dados, a positivo (1). Relativo ao cenário 1, para as classes 0 e 2, o algoritmo *random forest* mostrou um melhor desempenho, já para a classe 1 o melhor foi o algoritmo *extra tree*. Quanto ao cenário 2, o resultado foi similar no quesito melhores algortimos para a classe 1 foi novamente o *extra tree* e para a

Zika - Árvore de decisão cenário 1

```
[[2022  350    0]
 [ 386 1243  810]
 [  33  695 1634]]
              precision    recall  f1-score   support

           0       0.83      0.85      0.84      2372
           1       0.54      0.51      0.53      2439
           2       0.67      0.69      0.68      2362
```

Zika - Árvore de decisão cenário 2

```
[[39137  6968     0]
 [ 3665 12224  8019]
 [ 1546 31132 73751]]
              precision    recall  f1-score   support

           0       0.88      0.85      0.87     46105
           1       0.24      0.51      0.33     23908
           2       0.90      0.69      0.78    106429
```

Zka - Extra tree cenário 1

```
[2075  279   18]
[ 394 1452  593]
[  31  618 1713]]
              precision    recall  f1-score   support

           0       0.83      0.87      0.85      2372
           1       0.62      0.60      0.61      2439
           2       0.74      0.73      0.73      2362
```

Zika - Extra tree cenário 2

```
[[42438  3543   124]
 [ 1735 20621  1552]
 [ 1281 25458 79690]]
              precision    recall  f1-score   support

           0       0.93      0.92      0.93     46105
           1       0.42      0.86      0.56     23908
           2       0.98      0.75      0.85    106429
```

Zika - Random forest cenário 1

```
[[2084  275   13]
 [ 363 1437  639]
 [  32  553 1777]]
              precision    recall  f1-score   support

           0       0.84      0.88      0.86      2372
           1       0.63      0.59      0.61      2439
           2       0.73      0.75      0.74      2362
```

Zika - Random forest cenário 2

```
[[42303  3703    99]
 [ 1480 20225  2203]
 [ 1283 22670 82476]]
              precision    recall  f1-score   support

           0       0.94      0.92      0.93     46105
           1       0.43      0.85      0.57     23908
           2       0.97      0.77      0.86    106429
```

Figura 4.13: Matrizes de confusão por cenário e por algoritmo.

classe 2 o *random forest*. Já para classe 0 ambos obtiveram o mesmo resultado.

É possível observar que o teste com todos os dados da base obtiveram bons resultados, uma vez que para prevê se o paciente possui o vírus da *zika*, com até 86% de *recall* é uma métrica razoável. Para negativos os valores foram ainda melhores alcançando 92%.

O melhor algoritmo quando observado o *recall* e *o f1-score*, que analisa sua estabilidade, foi o *random forest* para ambos os cenários do vírus zika. A Figura 4.14 apresenta a matriz de confusão do cenário 1 para random forest no vírus zika.

As tabelas 4.6 e 4.6 apresentam os resultado dos três algoritmos para o vírus da *chikungunya*. Seguindo as mesmas métricas dos vírus da *zika*.

A Figura 4.15 apresenta todas as matrizes de confusão de cada cenário e de cada algoritmo juntamente com as métricas deles do vírus *chikungunya*.

O *recall* foi novamente utilizado como métrica principal da análise e diferentemente do que ocorreu para a base da zika, o algoritmo árvore de decisão apresentou resultados semelhantes aos demais no cenário 1. Para a classe 0, a árvore de decisão obteve o melhor

Tabela 4.5: Resultados dos algoritmos para cenário 1 para vírus *chikungunya*

Algoritmo	Métrica	Classe		
		0	1	2
Decision tree	Recall	0.69	0.66	0.84
	Precision	0.66	0.65	0.91
	F1-score	0.67	0.65	0.87
Random forest	Recall	0.67	0.67	0.85
	Precision	0.66	0.64	0.89
	F1-score	0.67	0.66	0.87
Extra tree	Recall	0.67	0.64	0.84
	Precision	0.65	0.63	0.89
	F1-score	0.66	0.64	0.86

Tabela 4.6: Resultados dos algoritmos para cenário 2 para vírus chikungunya

Algoritmo	Métrica	Classe		
		0	1	2
Decision tree	Recall	0.73	0.7	0.86
	Precision	0.74	0.71	0.80
	F1-score	0.73	0.70	0.83
Random forest	Recall	0.74	0.75	0.93
	Precision	0.80	0.74	0.81
	F1-score	0.77	0.75	0.87
Extra tree	Recall	0.76	0.73	0.92
	Precision	0.78	0.75	0.81
	F1-score	0.77	0.74	0.86

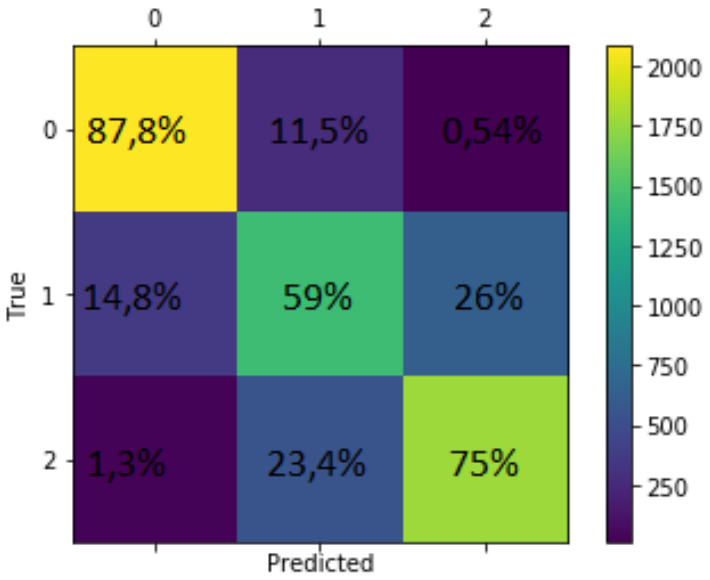

Figura 4.14: Matriz de confusão para o vírus zika utilizando algortimo random forest.

resultado, já para a classe 1 e 2, o melhor algoritmo foi o *random forest* mas com valores bem próximos dos demais.

No geral, analisando *recall* e o *f1-score*, foi possível concluir que o *random forest* foi o melhor algoritmo a ser utilizado nesse conjunto de dados. A matriz de confusão para o cenário 1 utilizando o *random forest* é apresentada pela Figura 4.16

Chikungunya - Árvore de decisão cenário 1

```
[[4121 1715  134]
 [1677 4048  410]
 [ 492  511 5315]]
              precision    recall  f1-score   support

           0       0.66      0.69      0.67      5970
           1       0.65      0.66      0.65      6135
           2       0.91      0.84      0.87      6318
```

Chikungunya - Árvore de decisão cenário 2

```
[[119402  41525   2991]
 [ 37822 109998  10198]
 [  4156   4260  52992]]
              precision    recall  f1-score   support

           0       0.74      0.73      0.73    163918
           1       0.71      0.70      0.70    158018
           2       0.80      0.86      0.83     61408
```

Chikungunya - Extra tree cenário 1

```
[[4029 1744  197]
 [1719 3951  465]
 [ 466  538 5314]]
              precision    recall  f1-score   support

           0       0.65      0.67      0.66      5970
           1       0.63      0.64      0.64      6135
           2       0.89      0.84      0.86      6318
```

Chikungunya - Extra tree cenário 2

```
[[123762  36274   3882]
 [ 33610 115249   9159]
 [  1886   2829  56693]]
              precision    recall  f1-score   support

           0       0.78      0.76      0.77    163918
           1       0.75      0.73      0.74    158018
           2       0.81      0.92      0.86     61408
```

Chikungunya - Random forest cenário 1

```
[[3988 1799  183]
 [1561 4102  472]
 [ 459  464 5395]]
              precision    recall  f1-score   support

           0       0.66      0.67      0.67      5970
           1       0.64      0.67      0.66      6135
           2       0.89      0.85      0.87      6318
```

Chikungunya - Random forest cenário 2

```
[[121522  38654   3742]
 [ 29263 118872   9883]
 [  1655   2383  57370]]
              precision    recall  f1-score   support

           0       0.80      0.74      0.77    163918
           1       0.74      0.75      0.75    158018
           2       0.81      0.93      0.87     61408
```

Figura 4.15: Matrizes de confusão por cenário e por algoritmo para o vírus chikungunya.

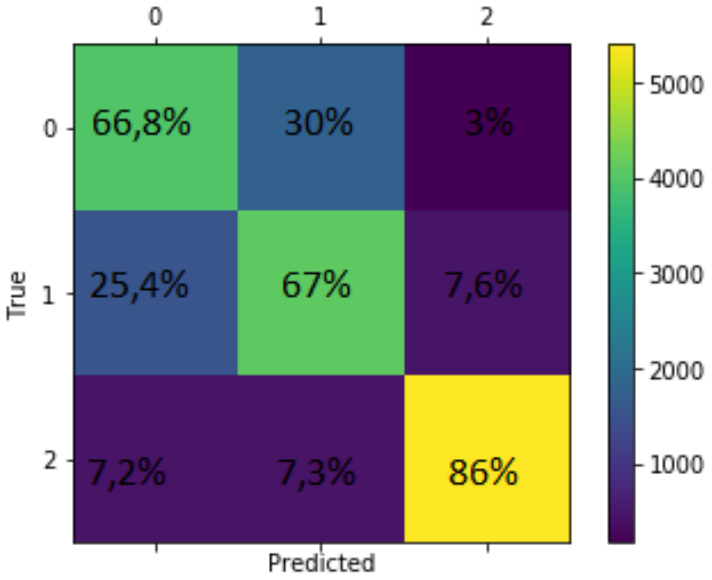

Figura 4.16: Matriz de confusão para o vírus *chikungunya* utilizando algortimo *random forest*.

Capítulo 5

Conclusão e Trabalhos Futuros

5.1 Conclusão

Com as Figuras geradas foi possível ter noção da rapidez no qual os vírus se propagaram por uma área tão extensa quanto o Brasil. Em um período de pouco mais de 5 anos, quase 80% de todo o território já tinha registrado caso de pelo menos um dos vírus. O que mostra uma fragilidade do país em combater e se proteger quanto a epidemias. Para as áreas mais afetadas, mostra que é ainda mais necessário atitudes de combate e prevenção ao vírus, uma vez que, a doença é perigosa e ainda mais para mulheres grávidas pela possibilidade de trazer sequeles irreversíveis para seus filhos.

A qualidade dos dados revelou uma possível falta de padronização do sistema que armazena os dados ou no método de inserção por parte dos usuários finais, uma vez que, foram encontrados inúmeras variações de conteúdo para se referir a uma mesma sentença. Erros de ortografias foram encontrados em vários campos da base, o que tornou obrigatório o tratamento prévio da base antes de seu uso. A não padronização acarretou na perda de dados em campos que poderiam ter usado e recuperar muitas informações, principalmente os campos de raça do paciente e a zona onde ele reside.

Quanto ao modelo de predição, foram obtidos bons resultados quando testado para todo o conjunto de dados, mostrando-se que para o cenário que o Brasil enfrentou, seria obtido bons resultados. Um ponto que pode ter influenciado o modelo a não ser mais preciso foi a falta de atributos relacionados a saúde do paciente, isto é, dados referente ao quadro clínico do pacientes, sintomas, estado de saúde entre outras possíveis informações.

Variáveis como a zona onde ele reside poderiam ter sido importantes uma vez que poderia refletir a respeito da qualidade do local onde o paciente reside, mas devido a qualidade de alguns atributos, não foi possível a utilização nos modelos. Todavia, esse trabalho poderia se tornar uma ferramente de auxilio a tomada de decisões do profissional de saúde, tendo em vista que se mostrou eficaz para o diagnóstico de resultado dos exames.

O trabalho ratifica o uso dos dados de saúde para a absorção de conhecimento, uma vez que a melhoria nos tratamentos e diagnósticos pode ajudar a salvar vidas e também pelo fato das informações que podem ser obtidas através desses dados não devem ser desperdiçadas.

5.2 Trabalhos futuros

Um possível trabalho futuro seria a tentativa de predizer se uma criança ainda não nascida iria possuir ou não sequelas devido ao fato da mãe ter contraído *zika* durante a gravidez. Para esse novo trabalho a base de dados atual não seria suficiente mas ajudaria com os dados das mulheres grávidas que foram atendidas.

Quanto ao entendimento da base de dados, existem outros vários possíveis meios de cruzamentos de dados que poderiam ter sidos realizados. A continuação desse processo é essencial para cada vez mais obter informações a respeito desse cenário.

A continuação do trabalho também poderia ser o teste com dados atuais dos treinos dos modelos obtidos a fim de verificar se para os próximos anos o modelo ainda obteria bons resultados em sua previsão.

A busca de testes em novos algoritmos ou um maior poder computacional para cada vez mais conseguir obter melhores resultados, é uma necessidade para sempre está tornando mais precisa a ferramenta. Tendo em vista que a área da saúde que lida com vidas de pessoas exige a maior precisão possível.

Referências

[1] Tan, P. N., M. Steinbach e V. Kumar: *Introduction to data mining*, volume 1. Addison-Wesley Longman Publishing Co. Inc, Boston, MA, 2005. ix, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 21, 27

[2] Garcia, Leila Posenato: *Epidemia do vÍrus zika e microcefalia no brasil: EmergÊncia, evoluÇÃo e enfrentamento*. Instituto de Pesquisa Econômica Aplicada – ipea 2018, 2018. 1, 39

[3] Raghupathi, Wullianallur, e Viju Raghupathi: *Big data analytics in healthcare: promise and potential*. Health Information Science and Systems 2, 2014. 1, 22

[4] Pandey, S. C.: *Data mining techniques for medical data: A review*. International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), 2016. 1, 22

[5] S. Jyoti, A. Ujma, S. Dipesh e S. B. Sunita: *Predictive data mining for medical diagnosis: An overview of heart disease prediction*. International Journal of Computer Applications, 2011. 1, 34

[6] G. William Moore, K. J. Cios e: *Uniqueness of medical data mining*. Artificial Intelligence in Medicine, vol. 26, no 1–2, p. 1–24, 2002. 1

[7] União, Controladoria Geral da: *Sistema eletrônico do serviço de informação ao cidadão*, 2019. `https://esic.cgu.gov.br`, Último acesso 19 maio 2019. 1, 25

[8] Pandey, Subhash Chandra: *"data mining techniques for medical data: A review"*. International conference on Signal Processing, Communication, Power and Embedded System (SCOPES)-2016, 2016. 2

[9] Lima, T., B. Barbosa C. Niquini C. Araújo e R. Lana: *Playing against dengue design and development of a serious game to help tackling dengue*. IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH), 2017. 4

[10] uadri, S. M., T. K. Prashanth S. Pongpaichet A. A. A. Esmin e R. Jain: *Targetzika: Epidemic situation detection and risk preparedness for zika virus*. 0th International Conference on Ubi-media Computing and Workshops (Ubi-Media), 2017. 4

[11] Chunxiao Dinga, Nana Taoa, Yuanguo Zhua: *A mathematical model of zika virus and its optimal control*. 35th Chinese Control Conference (CCC), 2016. 4, 5

[12] Cruz, Fundação Oswaldo: *Zika: sintomas, transmissão e prevenção*. https://www.bio.fiocruz.br/index.php/zika-sintomas-transmissao-e-prevencao, Último acesso 25 maio 2019. 5, 45

[13] Shobhit Verma, Nonita Sharma: *Statistical models for predicting chikungunya incidences in india*. First International Conference on Secure Cyber Computing and Communication (ICSCCC), 2018. 5

[14] Cruz, Fundação Oswaldo: *Chikungunya: sintomas, transmissão e prevenção*. https://www.bio.fiocruz.br/index.php/chikungunya-sintomas-transmissao-e-prevencao, Último acesso 25 maio 2019. 5

[15] Shobhit Verma, Nonita Sharma: *Chikungunya fever: An epidemiological review of a re-emerging infectious disease*. First International Conference on Secure Cyber Computing and Communication (ICSCCC), 2018. 5

[16] Fayyad, Usama: *Data mining and knowledge discovery in databases: Implications for scientific databases*. Proceedings. Ninth International Conference on Scientific and Statistical Database Management (Cat. No.97TB100150), 1997. 6

[17] Stephen Kaisler, Frank Armour, J. Alberto Espinosa William Money: *Big data: Issues and challenges moving forward*. 2013 46th Hawaii International Conference on System Sciences, 2013. 6

[18] Usama Fayyad, Gregory Piatetsky Shapiro e Padhraic Smyth: *From data mining to knowledge discovery in databases*. AI Magazine Volume 17 Number 3, 1996. 6, 22

[19] S. Swapna, P. Niranjan, B. Srinivas R. Swapna: *Data cleaning for data quality*. 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016. 9

[20] James, Gareth, Daniela Witten, Trevor Hastie e Robert Tibshirani: *An Introduction to Statistical Learning*. Springer New York, 2013. https://doi.org/10.1007/978-1-4614-7138-7. 10, 11

[21] Praciano, Bruno Justino Garcia, Joao Paulo Carvalho Lustosa da Costa, Joao Paulo Abreu Maranhao, Fabio Lucio Lopes de Mendonca, Rafael Timoteo de Sousa Junior e Juliano Barbosa Prettz: *Spatio-temporal trend analysis of the brazilian elections based on twitter data*. 2018 IEEE International Conference on Data Mining Workshops (ICDMW), 2013. 10

[22] Tahir, Nooritawati Md, Aini Hussain, Salina Abdul Samad, Khairul Anuar Ishak e Rosmawati Abdul Halim: *Feature selection for classification using decision tree*. Em *2006 4th Student Conference on Research and Development*. IEEE, junho 2006. 11

[23] *Ensemble methods*. https://scikit-learn.org/stable/modules/ensemble.html, Último acesso 27 maio 2019. 13

[24] Liang, Xin Zhou ; Jianmin Pang ; Guanghui: *Image classification for malware detection using extremely randomized trees*. 2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID), 2017. 13

[25] Usama Fayyad, Gregory Piatetsky Shapiro e Padhraic Smyth: *From data mining to knowledge discovery in databases*. AI Magazine Volume 17 Number 3 (1996), 1996. 14

[26] Almeida, Rafael Timóteo de Sousa Jr., Flávio Elias de Deus Georges Daniel Amvame Nze Fábio Lúcio Lopes de Mendonça Wesley Gongora de: *Taxonomy of data quality problems in multidimensional data warehouse models*. 2013 8th Iberian Conference on Information Systems and Technologies (CISTI), 2013. 17

[27] Hosseinkhah, Fatemeh, Hassan Ashktorab, Ranjit Veen e M. Mehdi Owrang O.: *Challenges in data mining on medical databases*. Em *Database Technologies*, páginas 1393–1404. IGI Global, 2009. `https://doi.org/10.4018/978-1-60566-058-5.ch083`. 22

[28] pydata: *Python data analysis library*, 2019. `https://pandas.pydata.org/`, Último acesso 25 maio 2019. 25

[29] Org, Jupyter: *Jupyter*, 2019. `https://jupyter.org/`, Último acesso 19 maio 2019. 26

[30] Hao Wei, Jeffrey Xu Yu e Can Lu: *String similarity search: A hash-based approach*. IEEE Transactions on Knowledge and Data Engineering (Volume: 30 , Issue: 1 , Jan. 1 2018), 2018. 26

[31] *Análise e business intelligence | tableau software*. `https://www.tableau.com`, Último acesso 27 maio 2019. 30

[32] Zheng Yao, Peng Liu, Lei Lei Junjie Yin: *Rx4.5 decision tree model and its applications to health care dataset*. Proceedings of ICSSSM '05. 2005 International Conference on Services Systems and Services Management, 2005., 2005. 34

[33] KADI, Ilham e Ali IDRI: *A decision tree-based approach for cardiovascular dysautonomias diagnosis*. 2015 IEEE Symposium Series on Computational Intelligence, 2015. 34

[34] Aiswarya Iyer, S. Jeyalatha e Ronak Sumbaly: *Diagnosis of diabetes using classification mining techniques*. International Journal of Data Mining Knowledge Management Process (IJDKP) Vol.5, No.1, January 2015, 2015,. 34

[35] Emrana Kabir Hashi, Md. Shahid Uz Zaman e Md. Rokibul Hasan: *An expert clinical decision support system to predict disease using classification techniques*. International Conference on Electrical, Computer and Communication Engineering (ECCE), February 16-18, 2017, Cox's Bazar, Bangladesh, 2017. 34

[36] *Scikit learn*. `https://scikit-learn.org`, Último acesso 28 maio 2019. 35

Apêndice A

Tabela de descrição dos atributos da base de dados

Nome do atributo	Quantidade	Tipo	Descrição
AgravodaRequisição	98	String	Exibe caso possua outras doenças que agravem a situação do paciente
Amostra	12	Inteiro	Controle de solicitação da amostra (1ª amostra, 2ª amostra.....) listada na base
DatadaColeta	11863	Data	Identifica no formato dd/mm/aaaa a data da coleta do material para análise do paciente
DatadaLiberação	4613	Data	Data da liberação do Exame pelo Analista Chefe (responsável pela análise do exame)
DatadaSolicitação	1300	Data	Identifica no formato dd/mm/aaaa a data da solicitação do material/exame para
DatadeCadastro	11931	Data	Data de cadastro da Requisição no Sistema
DatadeNascimento	125864	Data	Identifica no formato dd/mm/aaaa a data de nascimento do paciente cujo os dados serão analisados
Datado1ºSintomas	16553	Data	Identifica no formato dd/mm/aaaa a data dos sintomas relatados pelo paciente
DescriçãoFinalidade	98	String	Descrição do objetivo da Requisição de exames
EstadodeResidência	28	String	Estado de residência do paciente
EstadoSolicitante	28	String	Estado que solicitou o exame
Etnia	15	String	Indica a etnia do paciente
Exame	42	String	Nome do exame realizado pelo paciente
Finalidade	11	String	Identifica a finalidade na qual o exame foi realizado
IBGEMunicípiodeResidência	4396	Inteiro	IBGE do município de residência do paciente
IBGEMunicípioSolicitante	3499	Inteiro	IBGE do Município da Unidade de Saúde que solicitou o exame
Idade	111	Inteiro	Quanto anos o paciente tem - Data calculada pelo sistema com base na data de cadastro da requisição
IdadeGestacional	71	String	Se caso seja gestante, a idade da gestação
MaterialBiológico	102	String	Material Biológico (amostra) em que o exame foi realizado
MaterialClínico	5	String	Tipo de Material Clínica utilizado para acondicionar o transporte da amostra
Metodologia	15	String	Nome do método do exame realizado pelo paciente
MunicipiodeResidência	4207	String	Município de residência do paciente
Nacionalidade	41	String	Nacionalidade do paciente

PaisdeResidencia	15	String	Identifica o pais no qual do paciente dos quais os dados pertencem reside
RaçaCor	7	String	Raça/Cor do paciente
Sexo	15	String	Identifica o sexo do paciente dos quais os dados pertencem
TipoIdade	6	String	Tipo de data (Dias, Mês, Ano)
Zona	5	String	Zona de residência do paciente
1CampoResultado	50	String	Resultado final
2CampoResultado	8	Strnig	Resultado para alguns exames