CENTRO UNIVERSITÁRIO FEI

RODRIGO ANDRADE FRAGOSO

AVALIAÇÃO DO USO DE MODELOS DE APRENDIZADO DE MÁQUINA ESTATÍSTICO PARA PREVISÃO DE DIAGNÓSTICO PARA DOENÇAS INFECCIOSAS

São Bernardo do Campo

2022

RODRIGO ANDRADE FRAGOSO

AVALIAÇÃO DO USO DE MODELOS DE APRENDIZADO DE MÁQUINA ESTATÍSTICO PARA PREVISÃO DE DIAGNÓSTICO PARA DOENÇAS INFECCIOSAS

Dissertação de Mestrado, apresentada ao Centro Universitário FEI para obtenção do título de Mestre em Engenharia Elétrica. Orientado pelo Prof. Dr. Reinaldo A. C. Bianchi.

São Bernardo do Campo

2022

Fragoso, Rodrigo Andrade.
Avaliação do uso de modelos de Aprendizado de Máquina Estatístico para previsão de diagnóstico para doenças infecciosas / Rodrigo Andrade Fragoso. São Bernardo do Campo, 2022.
85 p. : il.

Dissertação - Centro Universitário FEI.
Orientador: Prof. Dr. Reinaldo A. C. Bianchi.

1. Aprendizado de Máquina. 2. Classificação. 3. Classificação Binária. 4. COVID-19. I. Bianchi, Reinaldo A. C., orient. II. Título.

Elaborada pelo sistema de geração automática de ficha catalográfica da FEI com os dados fornecidos pelo(a) autor(a).

centro universitário FEI

APRESENTAÇÃO DE DISSERTAÇÃO
ATA DA BANCA EXAMINADORA

Programa de Pós-Graduação Stricto Sensu em Engenharia Elétrica

Mestrado

PGE-10

Aluno: Rodrigo Andrade Fragoso **Matrícula:** 119112-1

Título do Trabalho: AVALIAÇÃO DO USO DE MODELOS DE APRENDIZADO DE MÁQUINA ESTATÍSTICO PARA PREVISÃO DE DIAGNÓSTICO PARA DOENÇAS INFECCIOSAS.

Área de Concentração: Inteligência Artificial Aplicada à Automação e Robótica

Orientador: Prof. Dr. Reinaldo Augusto da Costa Bianchi

Data da realização da defesa: 24/02/2022

ORIGINAL ASSINADA

Avaliação da Banca Examinadora:

São Bernardo do Campo, / / .

MEMBROS DA BANCA EXAMINADORA	
Prof. Dr. Reinaldo Augusto da Costa Bianchi	Ass.: _____
Prof. Dr. Ricardo de Carvalho Destro	Ass.: _____
Prof. Dr. Oscar Eduardo Hidetoshi Fugita	Ass.: _____

A Banca Julgadora acima-assinada atribuiu ao aluno o seguinte resultado:

APROVADO ☒ REPROVADO ☐

VERSÃO FINAL DA DISSERTAÇÃO

APROVO A VERSÃO FINAL DA DISSERTAÇÃO EM QUE FORAM INCLUÍDAS AS RECOMENDAÇÕES DA BANCA EXAMINADORA

_____

Aprovação do Coordenador do Programa de Pós-graduação

_____

Prof. Dr. Carlos Eduardo Thomaz

"*The world's most valuable resource is no longer oil, but data*"

The Economist

RESUMO

Nos últimos anos o mundo se deparou com a pandemia do COVID-19 e consequentemente com o avanço de tecnologias e campos de estudo em relação a doenças infecciosas. Com isso, a área de Inteligência Artificial, especificamente dentro da medicina, ganhou mais visibilidade e trouxe avanços e novos estudos, como sobre classificação de pacientes, a partir dos seus dados e exames clínicos, estarem potencialmente infectados pela doença. Isso pode ser explorada para melhorar as tomadas de decisões médicas, principalmente em situações em que os exames específicos para a doença não estejam disponíveis. Neste trabalho, é proposto o desenvolvimento de uma solução de Aprendizado de Máquina que visa realizar essa classificação entre paciente infectados e não infectados. Para isso, foram utilizados para treino e avaliação os dados disponibilizados por instituições de saúde em parceria com a (FAPESP, 2020). O processo de modelagem estatística passa por todas etapas do seu desenvolvimento, tendo o foco na métrica de sensibilidade, dando maior importância nos resultados dos exames para as pessoas que realmente estão infectadas. Os resultados obtidos mostraram que esse campo de estudo tem um potencial promissor e que com a presença de um exame de hemograma completo e outros exames complementares é possível alcançar até 80% de sensibilidade, indicando que a cada 100 pacientes realmente infectados, 80 deles seriam diagnosticados como positivos. Com isto, a medicina pode se beneficiar dessas metodologias para auxiliar no combate em situações de surtos, epidemias ou até outras pandemias.

Palavras-chave: Aprendizado de Máquina, Classificação, Classificação Binária, COVID-19.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS

FE	*Feature Engineering*
FS	*Feature Selection*
GBM	*Gradient Boost Machine*
KNN	*K-Nearest Neighbor*
LASSO	*Least Absolute Shrinkage and 81 Selection Operator*
ML	*Machine Learning*
NEC	Número de exames considerados
NMEP	Número mínimo de exames por cada paciente
NPV	*Negative Predictive Value*
PPV	*Positive Predictive Value*
RAMOBoost	*Ranked Minority Oversampling in Boosting*
RDT	Teste rápido para COVID-19
RT-PCR	*Reverse transcription polymerase chain reaction*
SHAP	*SHapley Additive exPlanations*
SMOTE	*Synthetic Minority Over-sampling Techniquen*
SQE	Soma de Quadrado do Erro
SVM	*Support Vector Machines*

LISTA DE SÍMBOLOS

β	coeficientes da regressão
s	ponto de corte
ℓ	verossimilhança
p	probabilidade
Pr	probabilidade
R	região de predição
X	variável preditora
Y	variável resposta

SUMÁRIO

1 Introdução

No final do ano de 2019, a população mundial teve os seus primeiros contatos com o SARS-CoV-2, responsável pela pandemia do COVID-19. A partir daí, a sociedade se deparou com diversos desafios para combater a dispersão desse vírus a nível global, fomentando novas discussões e possibilidades de melhoria dentro da comunidade médica, sendo uma delas no campo da Inteligência Artificial.

Ao longo da pandemia a COVID-19 se alastrou pelo mundo mudando a forma de convívio social da grande maioria das nações. Isto aconteceu, principalmente, porque esta doença possui uma capacidade de transmissão elevada, um dos fatores responsáveis pela declaração do estado de pandemia. Com isso, de acordo com Kilic, Weissleder e Lee (2020), uma das soluções para ”frear” esta contaminação seria, principalmente, o isolamento social e a rastreabilidade de contato dos já infectados através de testagem em massa, política que foi inicialmente implementada na Coreia do Sul e Alemanha.

Quando se fala de testagem em massa, principalmente no início de uma pandemia, na qual ainda estão sendo desenvolvidas tecnologias/métodos que atendam exclusivamente a ”nova doença”, é preciso uma infraestrutura e oferta mundial dos insumos necessários para atender a grande demanda de testes. Infelizmente, isto pode demorar a ser atendido integralmente ou numa parcela muito próxima ao total e aqui surge uma oportunidade do uso de Inteligência Artificial para auxiliar nesse processo de testagem.

Em relação ao desenvolvimento das tecnologias para o combate da COVID-19, surgiram metodologias de detecção que possuem números animadores: *Reverse transcription polymerase chain reaction* (RT-PCR) e o Teste rápido para COVID-19 (RDT) que registram até 89% e 100% de especificidade, respectivamente. Infelizmente, como foi dito, o número de testes necessários para realizar um rastreamento da melhor forma possível não se mostrou disponível para toda sociedade, principalmente no início da pandemia (KILIC; WEISSLEDER; LEE, 2020).

Existe uma grande deficiência na quantidade de testes específicos para essa doença, principalmente nos países em que o insumo para tais é obtido através de importações. Assim, surgem consequências disto que impactam diretamente no combate de uma pandemia, perde-se parte da rastreabilidade das infecções e afasta-se profissionais da saúde por suspeita contaminação, agravando ainda mais todo o contexto da pandemia (THORNTON, 2020).

1.1 Motivação

Considerando os problemas apresentados, faz-se necessária a criação de metodologias alternativas para auxiliar na detecção da COVID-19 e em possíveis ocasiões semelhantes. Isso pode ser feito através de diferentes modalidades de testagem que se adaptem rapidamente a cada tipo de população ou região.

Uma das principais oportunidades está na situação em que não existe a possibilidade de realização de um teste específico e é necessária uma tomada de decisão com urgência. Um possível exemplo é: um paciente está aguardando atendimento e é preciso definir se ele será encaminhado para um fluxo de pacientes com suspeita de COVID-19 ou seguirá no fluxo regular.

Sabendo que existe uma grande oferta de outros tipos de exame, como hemogramas, urina ou diversos tipos de imagens, percebe-se que são gerados dados e informações sobre cada paciente. Esses dados são úteis e podem alimentar sistemas de classificação que podem ser capazes, ou não, de identificar o potencial de um paciente estar infectado. Desse modo, surge a oportunidade de desenvolvimento de um método baseado em dados clínicos e resultados já consolidados de testagens anteriores, a fim de auxiliar no diagnóstico de uma doença infecciosa, como a COVID-19. Tudo isso, ajudando a realizar uma triagem preliminar do grau de suspeita de um indivíduo ter contraído a doença.

Vale lembrar que aqui não é proposto que os exames sejam substituídos por esse método e sim que eles sirvam de apoio em um cenário em que não existem exames específicos para todos, o que pode ocorrer novamente em outras situações de surtos, epidemias ou até pandemias.

1.2 Objetivos

O objetivo deste trabalho é desenvolver uma metodologia para detecção do potencial de infecção viral de um paciente, através dos seus dados clínicos disponíveis e utilizando aprendizado estatístico de máquina. Para isto, foi realizado um estudo utilizando informações médicas, disponibilizados por laboratórios brasileiros (FAPESP, 2020), referentes ao COVID-19. Além disso, será criada uma ferramenta que seja capaz de ser reproduzida de forma simples e eficaz para situações futuras semelhantes.

1.3 Organização do Trabalho

Neste capítulo foi apresentada uma breve introdução sobre as oportunidades de Inteligência Artificial no combate à pandemia, o que o trabalho se propõe a fazer para combater a falta de testes, assim como a sua motivação e objetivo. A seguir, este trabalho apresentará mais 4 capítulos. Sendo o capítulo 2 uma abordagem sobre os conceitos fundamentais aplicados neste trabalho, o capítulo 3 um estudo dos trabalhos relacionados com o tema abordado, o capítulo 4 possui detalhes sobre a metodologia, o capítulo 5 sobre os resultados obtidos e, por fim, o capítulo 6 demonstra as conclusões que foram obtidas através dessa pesquisa.

2 Conceitos Fundamentais

2.1 Aprendizado de Máquina

Com a evolução e a diminuição dos custos do poder computacional ao longo dos últimos anos, termos como Aprendizado de Máquina e ciência de dados se tornaram usuais em diversos ambientes corporativos, na academia e na maioria das áreas do conhecimento. As máquinas estão se aperfeiçoando no aprendizado autônomo e na medicina não é diferente: já existem diversos artigos aplicando estas técnicas para resolver problemas comuns no cotidiano dos profissionais da saúde (DEO, 2015).

Aprendizado de Máquina ou *Machine Learning*(ML) é um método de análise de dados que visa automatizar a criação de modelos analíticos e/ou estatísticos e que também se baseia na ideia de que sistemas podem aprender, principalmente através de dados, podendo identificar padrões e auxiliando humanos a tomarem melhores decisões. ML faz parte do estudo de Inteligência Artificial.

Naturalmente, esse campo possui um vínculo forte com a estatística, matemática e a ciência da computação. Essas áreas do conhecimento são capazes de criar algoritmos e soluções que conseguem lidar com pequenas ou grandes quantidades de dados, auxiliando a ciência no avanço desde tópicos simples até os mais complexos (DEO, 2015).

Dentro do ML pode-se dividir os tipos de aprendizado em três grandes grupos: Aprendizado Supervisionado, Não Supervisionado e por Reforço. Como esse trabalho fala em classificar indivíduos que podem ou não ter contraído uma doença, teve como foco os dois primeiros conceitos para afunilar entre as diversas seções dessa disciplina e chegar na melhor abordagem para o problema apresentado.

Os dados que foram utilizados possuem o formato tabular e dentro desta estrutura podem existir registros de conjunto de dados que possuem ou não uma variável alvo a ser predita. Daí, se iniciam as discussões sobre os conceitos de supervisão do aprendizado.

2.1.1 Aprendizado Supervisionado

Quando se fala em Aprendizado Supervisionado, o objetivo dos modelos/algoritmos é de prever uma variável já conhecida pelos dados coletados. Alguns exemplos de tarefas que já são bastantes conhecidas neste campo são: o reconhecimento de imagens - como dígitos manuscritos - classificação de documentos (se ele pertence ao ensaio clínico ou ao relatório financeiro, por exemplo) e, principalmente, nas atividades que envolvam regressão e classificação de dados. (DEO, 2015).

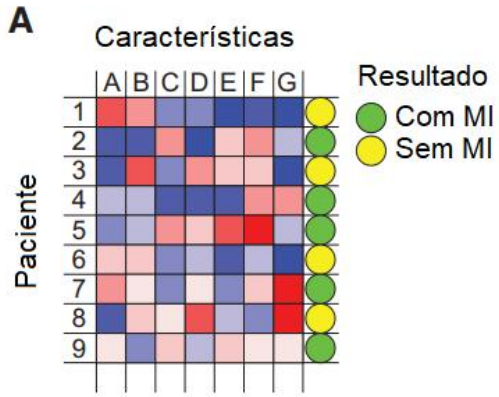
A abordagem que se mostra mais interessante neste caso é a de classificação, que funciona da seguinte maneira: já conhecendo em que subgrupos cada registro do conjunto de dados pertence, tenta-se prever em qual subgrupo os atuais e novos registros (os que ainda possuem a variável alvo desconhecida), tendem a se encaixar. Ainda dentro deste universo, podemos falar de duas ramificações: a primeira a Classificação Binária, quando apenas temos a possibilidade da resposta pertencer ou não a uma classe (0 ou 1) e a segunda, Classificação Múltipla, caso existam várias classes a serem identificadas.

A Classificação Binária se mostra mais adequada ao objetivo de simular o resultado do exame da COVID-19 ou de outras doenças infecciosas. Apesar desses exames geralmente possuírem até 3 resultados (Detectado, Não-Detectado e Indefinido) pode-se simplificar os resultados apenas entre Detectado e Não-Detectado.

Para realizar essa identificação, os modelos costumam ”treinar” com dados que tenham a variável alvo conhecida, de modo a aprender os padrões e comportamentos comuns a cada uma das classes existentes. Para isso, usa-se um outro grupo de variáveis, preditora/regressoras, e, a partir delas, os modelos criam suas ”conclusões”.

Um exemplo de um sistema de classificação binária para identificar o infarto do miocárdio é ilustrado pela figura 1. Nesse caso, as variáveis representadas pelo eixo horizontal(A, B, C ... G) são características do paciente como idade, sexo, indicadores clínicos e outros. Cada indivíduo é representado por uma linha no eixo vertical (1, 2, 3, 4 ... 9) e, por último, temos o resultado dessa classificação que pode ser o infarto (verde) ou não infarto (amarelo).

Figura 1 – Representação de exemplo de dados utilizados em classificação binária de Miocardite (MI)



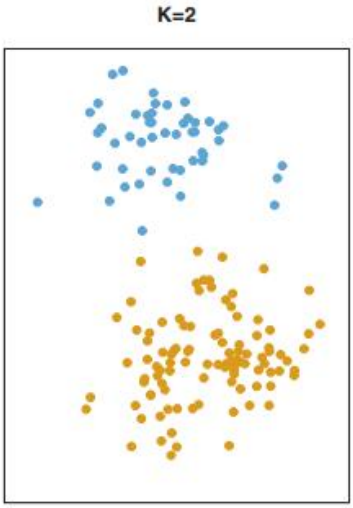
Fonte: Adaptado de (DEO, 2015)

2.1.2 Aprendizado Não Supervisionado

Em contrapartida ao tópico anterior, o Aprendizado Não Supervisionado é utilizado para detectar padrões ou agrupamentos, principalmente em dados que não temos uma variável alvo ou quando não se tem a intenção de usá-la. Por se tratar de uma tarefa desafiadora e muitas vezes subjetiva, um dos maiores desafios que encontramos neste campo é a avaliação da performance dos algoritmos por ele utilizados, já que não teremos um resultado comparativo no final da análise (DEO, 2015).

Um dos mais famosos e conhecidos algoritmos desta natureza é o *K-Means*, que, assim como foi dito, tem o objetivo de agrupar uma amostra de dados em até K grupos, com K sendo definido usuário, como demonstra a Figura 2. A partir da representação bidimensional e com o estudo proposto podemos entender que ao utilizar esta técnica é possível dividir, com eficiência, os pacientes entre o grupo infectados e não infectados, mas como os dados possuem uma variável alvo anotada, é razoável que escolhamos por abordagens que a utilizem, buscando a melhor performance para o modelo proposto. De forma geral, o aprendizado supervisionado funciona melhor para casos em que a variável do resultado já esteja definida. (JAMES et al., 2013).

Figura 2 – Aplicação do algoritmo K-Means, para K=2



Fonte: Adaptado de (JAMES et al., 2013)

Na medicina, devido a heterogeneidade inerente à maioria das doenças, os pacientes geralmente não estão em grupos bem definidos, como a partir das suas características fisiológicas e clínicas. Segundo Deo (2015), o uso destas técnicas não é muito comum por apresentar pouca eficácia e resultados inconclusivos, mas ainda tem a potencial capacidade de encontrar padrões para novos tratamentos e identificações de grupos que possuam características semelhantes e possam ser mais aderentes a certos medicamentos, por exemplo.

2.1.3 A estrutura do Aprendizado

Após conhecer os tipos de aprendizado, ainda é necessário entender como é formulado um problema de ML e o modo de utilizar esta ferramenta, de uma forma generalizada. Esse tópico terá como foco os problemas de classificação, que estão inseridos dentro do Aprendizado Supervisionado e será o principal tema abordado neste estudo.

Para exemplificar como funciona a estruturação desse aprendizado, pode-se usar o exemplo da Figura 1, de previsão do infarto do miocárdio e definir duas classes: a primeira é referente a todos indivíduos que tiveram um ou mais infartos desse tipo e a segunda quem nunca o teve. Com esse objetivo definido, é necessário criar um modelo capaz de distinguir, preferencialmente com precisão, a qual classe cada indivíduo pertence.

O primeiro passo é a coleta e escolha das nossas variáveis preditoras. Apesar de parecer uma etapa simples, é necessário um certo grau de cautela pois deve-se escolher as variáveis que,

logicamente, tem uma associação com a doença, como por exemplo a presença de hipertensão, diabetes e níveis de colesterol, sabendo que podem não representar um bom poder preditivo quando associadas em conjunto. Mesmo assim, costuma-se colher a maior quantidade distinta de dados, tendo em mente a relação de disponibilidade/acessibilidade dessa informação. Para decifrar esse problema e realizar escolhas de maneira direcionada, surgem as técnicas de seleção de variáveis ou *Feature Selection* (FS), que é um campo muito importante do ML (DEO, 2015).

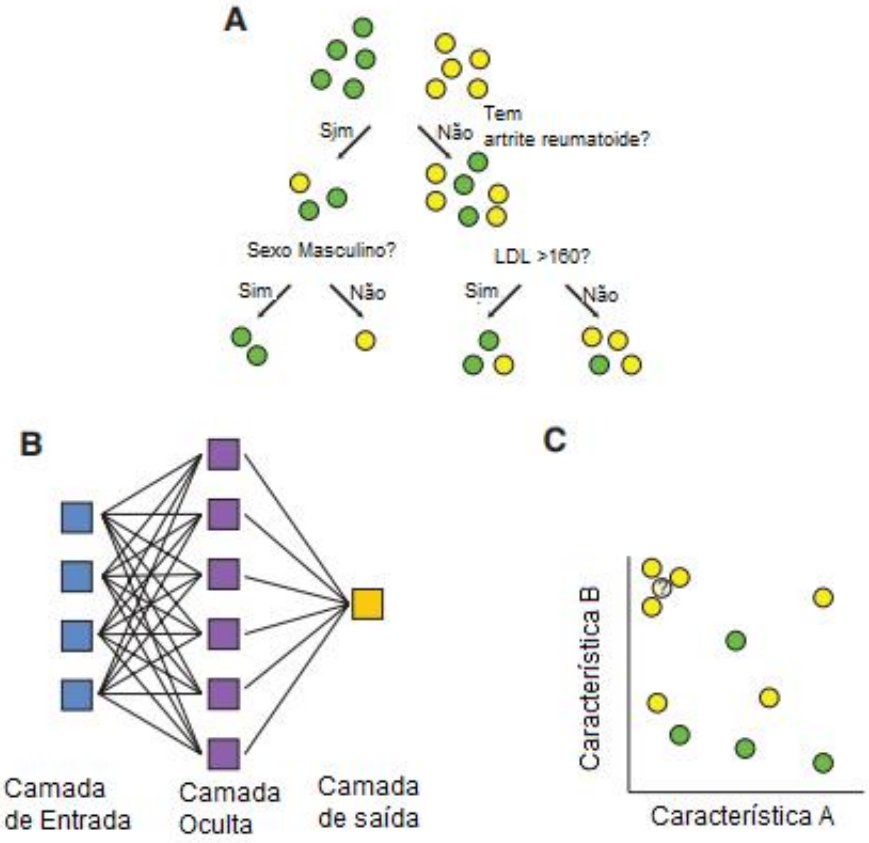
Feito isso, é necessário utilizar uma função que converta os regressores em uma das classes existentes e para isto existem diversos tipos de algoritmos, com cada um portando características próprias. Pensando na estatística clássica, o modelo de regressão logística é amplamente utilizado em atividades desse quesito, se tornando um forte candidato para se adaptar a essa tarefa, mas ainda existem outras opções, como os modelos baseados em árvores de decisão que tem uma forte tendência a flexibilizar casos de exclusividade mútua, conhecida por ser um ponto negativo da regressão logística. Ainda neste campo, as redes neurais, que possuem a ”habilidade” de realizar a transformação/seleção das variáveis de entrada, conforme a figura 3 B, costumam trazer resultados acurados, assim como a *Support Vector Machines* (SVM) que também utiliza esse conceito, mas com uma diferente formulação matemática. E por fim, modelos como o *K-Nearest Neighbor* (KNN) que utilizam a abordagem de definir a classe baseando-se em pacientes/registros com características próximas, conforme ilustrado no plano bidimensional na figura 3 no bloco C, diferente dos modelos citados anteriormente que buscam converter os regressores diretamente na classe escolhida, quando já treinados (DEO, 2015).

Falando das características de cada modelo, temos os hiper-parâmetros que são aqueles que fornecemos ao modelo, como por exemplo: número de Nós e Camadas ocultos, recursos de entrada, Taxa de Aprendizagem, Função de Ativação etc. na Rede Neural, enquanto Parâmetros são aqueles que seriam aprendidos na etapa de treinamento pela máquina como Pesos e Viés .

Todos modelos de aprendizado de máquina possuem hiper-parâmetros: aqueles que são oferecidos ao modelo pelo próprio usuário como a taxa de aprendizagem, número de camadas ocultas e a função de ativação de uma rede neural. Geralmente, o usuário tem maior controle sobre o ajuste dos hiper-parâmetros e esse ajustes costumam ser feitos durante o treinamento e otimização de cada modelo.

Para realizar esse treinamento é preciso que os dados sejam preparados, de forma que se tornem ”compreensíveis” para os modelos que serão utilizados. Sabendo que existem distintos tipos de dados tabulares (e.g. categóricos, texto, numéricos e outros) e que os algoritmos costumam ter legibilidade apenas para os numéricos, é fundamental garantir que a informação esteja

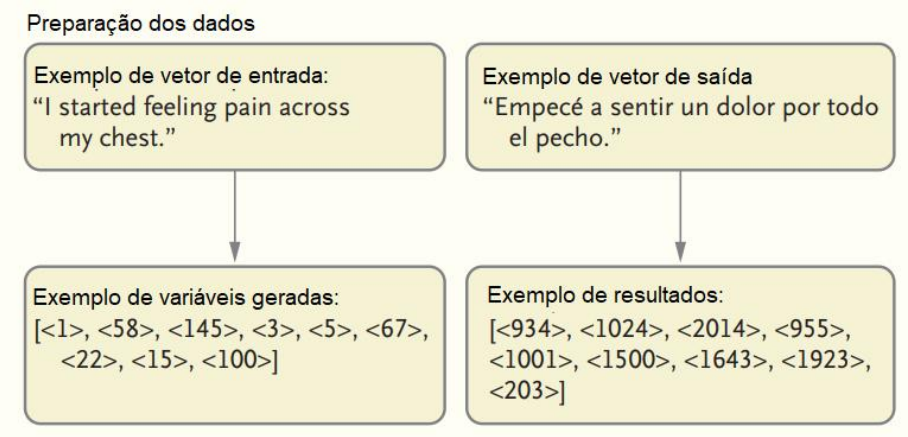
Figura 3 – Tipos de modelos utilizados para classificadores



Fonte: Adaptado de (DEO, 2015)

representada por este tipo. No caso de uma frase, representada computacionalmente por uma *string*, pode-se codificar as palavras em números, conforme a figura 4, utilizar a frequência das palavras como regressores, dentre outras técnicas. No caso das variáveis categóricas pode-se utilizar o *one-hot encoding* que transforma cada categoria em uma coluna binária, indicando se o registro pertence ou não a ela. Em resumo, dentro do ML, essas transformações e/ou adequações dos dados obtidos é conhecida como *Feature Engineering* (FE) e possui as mais diversas aplicações para cada um dos casos citados e outros mais (RAJKOMAR; DEAN; KOHANE, 2019).

Figura 4 – Exemplo de preparação dos dados do tipo texto

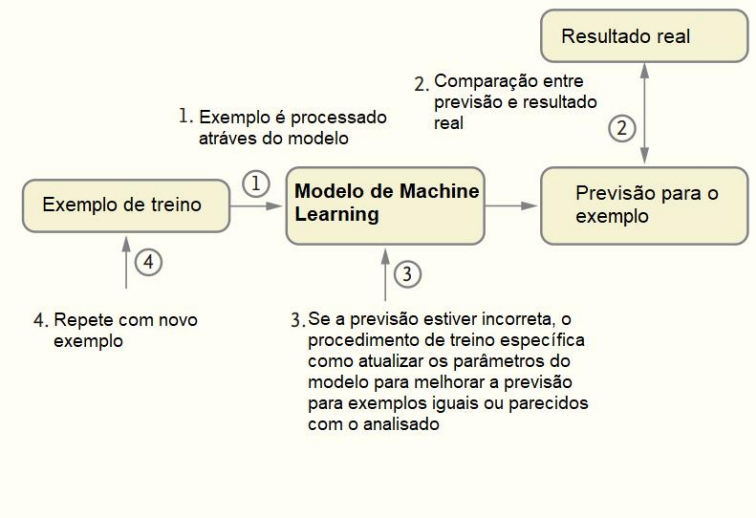


Fonte: Adaptado de (RAJKOMAR; DEAN; KOHANE, 2019)

Com os dados prontos para o ”consumo” dos modelos selecionados é possível iniciar o treinamento de cada um deles. Este processo costuma ser feito de forma iterativa e, deste modo, é possível notar a principal diferença entre modelos de aprendizado de máquina e outras técnicas convencionais: eles aprendem a partir de exemplos e não por uma cadeia de regras pré-definidas, por exemplo.

Durante a fase de treinamento, os parâmetros citados anteriormente, são inicializados randomicamente e um ou vários exemplos são enviados para o algoritmo convertê-los em uma das classes existentes, no caso da classificação. Nas primeiras iterações, a assertividade da classe predita versus a real costuma ser muito baixa, isso porque o modelo está constantemente ”aprendendo” ao comparar os resultados obtidos com os reais e este é o ponto chave, no qual o modelo define como os parâmetros devem ser modificados, a cada iteração, para aproximar de forma otimizada a conversão dos exemplos nas suas classes reais, conforme ilustrado pela Figura 5 (RAJKOMAR; DEAN; KOHANE, 2019).

Figura 5 – Etapas do treinamento no ML



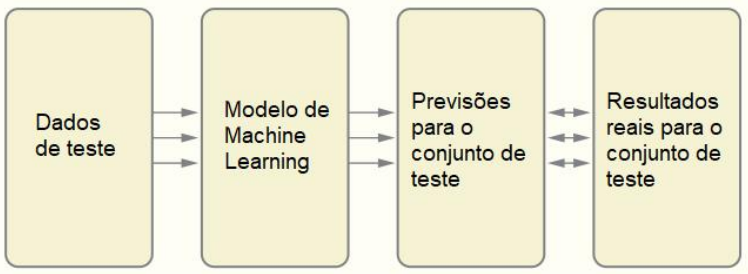
Fonte: Adaptado de (RAJKOMAR; DEAN; KOHANE, 2019)

Uma boa prática ao criar modelos de ML é fazer a divisão dos dados nos grupos de treinamento e teste. O primeiro bloco, como foi exemplificado anteriormente, é utilizado para ajustar os hiper-parâmetros e treinar os parâmetros. Sendo assim, como se tem o objetivo de simular um caso real, no qual irão surgir novos dados desconhecidos que servirão como exemplos de entrada para os algoritmos, o conjunto de teste realiza esta função, ou seja, este último bloco não é utilizado no treinamento e serve, exclusivamente, para avaliar a performance do modelo de acordo com as métricas escolhidas pelo usuário.

Uma abordagem opcional que ocorre dentro do bloco de treinamento, consiste em dividi-lo em dois blocos: dados de treino e validação, de uma forma que antes de expôr o modelo aos dados de teste, podemos ter uma breve estimativa de como ele está performando ainda dentro deste pequeno conjunto de validação que também não foi utilizado para treino.

O fluxograma representando o uso das informações de teste pode ser visualizado na Figura 6, lembrando que a métrica escolhida para relacionar os resultados entre o terceiro e o quarto bloco, assim como no treinamento, é de fundamental importância para definir as características da modelagem realizada

Figura 6 – Avaliação do modelo com o conjunto de teste



Fonte: Adaptado de (RAJKOMAR; DEAN; KOHANE, 2019)

Entendendo como é realizado o aprendizado de máquina, é interessante pontuar algumas aplicações, trazidas por Rajkomar, Dean e Kohane (2019) no campo da medicina: prevenção do infarto do miocárdio, sequenciamento dos dados genéticos para avaliar se o paciente o terá câncer ou não, realização de diagnóstico através de um *chat-bot*, identificar em tempo real o custo de internamento de um indivíduo através dos seus dados do prontuário eletrônico, dentre outras aplicações. Tudo isso, mostra como os modelos podem auxiliar os profissionais do campo da saúde a otimizar as suas decisões, sempre atentos ao senso crítico e aos seus conhecimentos da área.

2.2 Modelos de classificação

Nesta seção serão apresentados, formalmente, os modelos de ML que serão utilizados para simular o teste da presença do vírus da COVID-19. Para isso, será considerado um *dataset* padrão que possui as variáveis preditoras com suas respectivas classes anotadas: Detectado ou Não Detectado.

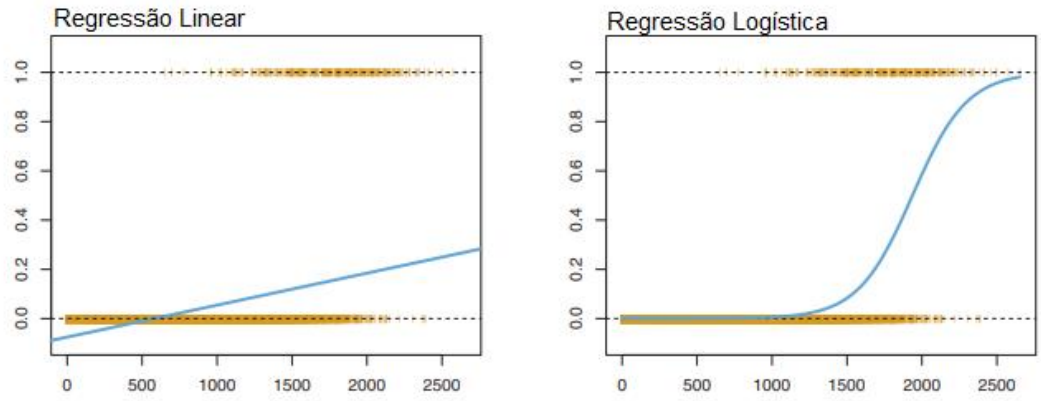
2.2.1 Regressão Logística

A regressão logística tem como objetivo quantificar a *probabilidade* da variável resposta Y(diagnóstico) pertencer a uma categoria, tudo isto dado as variáveis regressoras X_i, considerando o *dataset* padrão.

Ao comparar este modelo com a regressão linear, percebe-se porque ela é mais adequada quando se trata de problemas de classificação. Isto porque, basicamente, a função logística retorna valores entre 0 (Não Detectado) e 1 (Detectado), enquanto a linear extrapola estes limites

e traz algumas incoerências matemáticas como valores negativos ou muito maiores que 1, de *probabilidade*, conforme a Figura 7 (JAMES et al., 2013).

Figura 7 – Comparativo entre classificadores utilizando a regressão linear e logística, respectivamente



Fonte: Adaptado de (JAMES et al., 2013)

Sendo assim, a equação (1) representará a probabilidade Pr do paciente (que possui as características clínicas X_i) se encaixar no diagnóstico Detectado.

$$Pr(diagnóstico = Detectado|paciente) \tag{1}$$

Para simplificar a análise, essa probabilidade supracitada será chamada de $p(paciente)$, lembrando que ela tem o intervalo de valores entre 0 e 1. Neste caso, ainda é necessário definir um ponto de corte, podemos partir com $0{,}5$, lembrando que este número geralmente é otimizado de modelo a modelo para obter a melhor performance na métrica que está sendo otimizada(como a curva ROC, precisão e outras), em que está predição se torna verdadeira e isto pode ser selecionado como parâmetro a ser otimizado ou pré definido pelo usuário que está criando o modelo.

Ao observar a métrica a ser otimizada na identificação do exame, pode-se utilizar $p(paciente) > 0{,}9$ como ponto de corte, tornando a previsão para casos positivos probabilisticamente mais rara ou até $p(paciente) > 0{,}1$, favorecendo a detecção da classe positivo (JAMES et al., 2013).

Vale ressaltar que o ajuste deste limítrofe costuma impactar diretamente na forma em que o modelo se comporta, alterando as métricas que são utilizadas na mensuração da performance dos classificadores. Então, é uma boa prática defini-lo no momento do treinamento para o modelo não perder a ”referência” ao simplesmente alterar o ponto de corte, para obter resultados que sejam satisfatórios no momento em que ele já está treinado.

A função logística é representada pela equação (2) e, como esperado, ela possui o limite superior e inferior iguais a 1 e 0, conforme ilustrado pelo quadro direito da Figura 7.

$$p(paciente) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}} \tag{2}$$

Para ajustar o modelo e chegar nos coeficientes β, indicados pela equação (2) é utilizado o método da máxima verossimilhança. Sendo assim, será produzida uma curva em forma de S, que traz uma sensibilidade e interpretação maior para casos de classificação quando comparada a uma regressão linear, por exemplo. De modo a melhor entender o comportamento da função logística pode-se manipular a equação (2) para chegar na (3) que representa as *odds* do modelo (JAMES et al., 2013).

$$\frac{p(paciente)}{1 - p(paciente)} = e^{\beta_0+\beta_1 X} \tag{3}$$

A partir da equação (3), nota-se que o valor das *odds* é representado pelo lado esquerdo da igualdade, podendo variar entre 0 e ∞, no qual, em compração com as probabilidades, se comporta de uma forma exponencial. Por exemplo, quando tem-se $p(paciente) = 0{,}2$ as *odds* representam $\frac{0{,}2}{1-0{,}2} = \frac{1}{4}$ e para $p(paciente) = 0{,}9$ tem-se as *odds* $\frac{0{,}9}{1-0{,}9} = 9$. Como esta visão valoriza os números mais altos de probabilidade, elas costumam ser usadas em casas de apostas, como corridas de cavalo, dando mais confiabilidade ao apostador (JAMES et al., 2013).

Aplicando o logaritmo neperiano na equação (3) chega-se na equação (4).

$$ln\left(\frac{p(paciente)}{1 - p(paciente)}\right) = \beta_0 + \beta_1 X \tag{4}$$

No caso da equação (4), é representada a função $log - odds$ também conhecida como $logit$ e percebe-se que a regressão logística possui uma relação linear, dentro da função $logit$, entre as características do paciente, que são representadas por X e o seu diagnóstico. Sendo assim, sabendo que a relação entre $p(paciente)$ e o paciente não é uma linha reta, percebe-se que o valor de β_1 responde no mesmo sentido de X, caso seja positiva ela indicará uma relação de crescimento entre X e o $p(paciente)$ e, analogamente, o contrário quando for de forma inversa, podemos visualizar este efeito no quadro direito na Figura 7 (JAMES et al., 2013).

Como foi dito, para estimar os coeficientes da regressão β_0 e β_1 será utilizado o método da máxima verossimilhança, lembrando que esta etapa consiste apenas do uso dos dados de treinamento. A ideia através dessa metodologia é maximizar a função da verossimilhança ℓ,

representada pela equação (5), na qual px_i representa a *probabilidade* da classe positiva, dado as características do paciente i.

$$\ell(\beta_0,\beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})) \tag{5}$$

A intuição por trás desse método é, simplesmente, obter o valor predito o mais próximo possível do real e a cada exemplo que exista esta diferença e ela seja maior, a verossimilhança irá penalizar o treinamento de forma proporcional. Lembrando que nesta abordagem, são utilizados os valores do $p(paciente)$, ou seja não estamos falando de uma tarefa matematicamente binária, que seria a comparação pura entre 0 e 1. Para treinar os modelos são utilizadas técnicas como o algoritmo de Newton-Raphson, implementados em diversos pacotes estatísticos como o *R* e em *python* (FRIEDMAN; HASTIE; TIBSHIRANI et al., 2001).

Com os valores dos β calculados, é possível fazer as predições dadas pelo modelo através da equação (6).

$$\hat{p}(paciente) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0+\hat{\beta}_1 X}} \tag{6}$$

Sabendo que um paciente possui p características/preditores, $X = (X_1,X_2...X_p$, é necessário usar a regressão logística na sua forma múltipla. Sendo assim, pode-se generalizar a equação (2), expandindo-a e chegando à equação (7)

$$p(paciente) = \frac{e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}} \tag{7}$$

.

De forma semelhante ao modelo com uma variável preditora, é utilizado o método da máxima verossimilhança para definir os coeficientes β. Um dos benefícios deste modelo múltiplo é, também, a relação direta entre os valores dos coeficientes a cada uma das características do paciente: um valor elevado e positivo do coeficiente que está associado ao índice glicêmico relata uma relação forte entre altos índices e um diagnóstico positivo, isto mostra que de certa forma, existe uma explicabilidade inerente ao modelo apenas observando os valores dos coeficientes e que podem auxilias os especialistas na análise crítica desta ferramenta (JAMES et al., 2013).

A regressão logística é bastante utilizada em previsões de casos clínicos e, segundo Christodoulou et al. (2019), não é possível distinguir se ela possui um performance melhor ou pior do que os outros modelos de ML, mas conseguiu atender as expectativas quando utilizado.

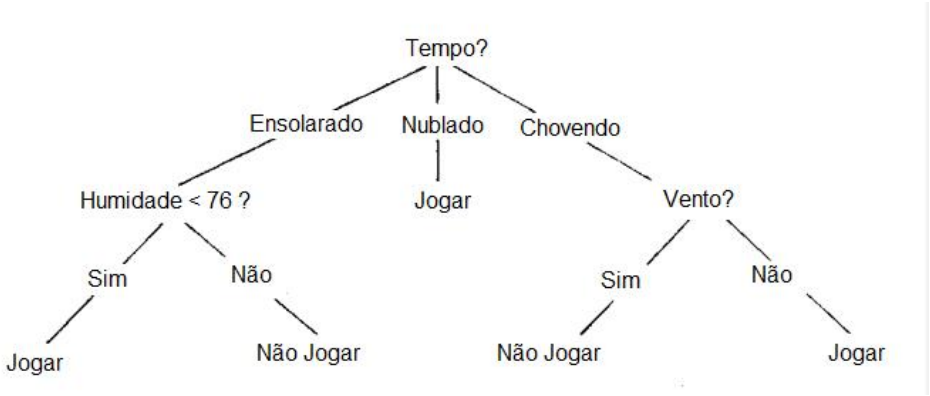
Segundo o autor desse estudo, isso se deve pelo fato de problemas de calibração e validação que não estavam presentes quando utilizadas outras técnicas de ML, mas ainda assim mostra a representatividade deste modelo no mundo da medicina.

2.2.2 Modelos de Árvore

As árvores de decisão são algoritmos que podem ser utilizados tanto nas tarefas de regressão como de classificação, e costumam ser utilizadas graças a sua interpretabilidade.

Esta família de modelos foca em descobrir um conjunto de regras que irão formar a dita árvore através de uma estratégia *top-down* de dividir e conquistar, tudo isto dentro de pequenos passos que serão apresentados posteriormente. Além do mais, ela conta com uma estrutura recursiva, para representar essas regras e associa cada ”folha” a uma certa classe, como pode ser visto no exemplo da Figura 8 que cria um conjunto de regras para definir se a pessoa irá jogar ou não *baseball* (QUINLAN, J. R., 1990).

Figura 8 – Árvore de decisão definindo se o exemplo irá jogar baseball ou não



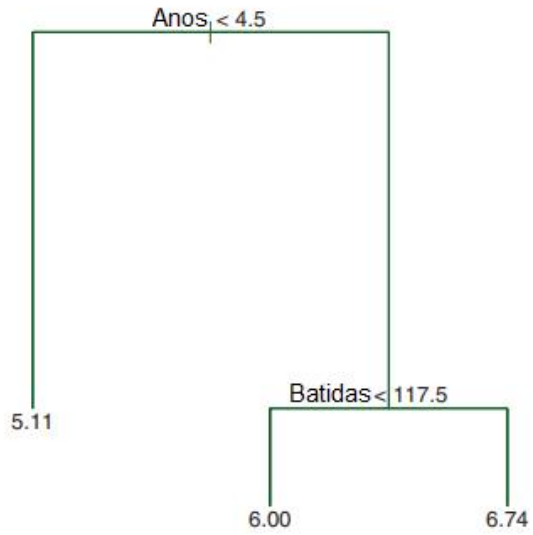
Fonte: Adaptado de (QUINLAN, J. R., 1990)

2.2.2.1 Árvores de Decisão

Primeiramente, será desenvolvida a explicação para uma árvore de regressão e feito isto, a explicação será expandida para os casos de classificação.

Tomando como exemplo a predição de salários (em log) de jogadores de *baseball* baseado no número de rebatidas e anos que um certo jogador completou nas grandes ligas, pode-se chegar no exemplo da Figura 9, no qual o salário é definido por nós que representam divisões baseadas nas informações citadas.

Figura 9 – Árvore de decisão definindo o salário do jogador de *baseball*



Fonte: (JAMES et al., 2013)

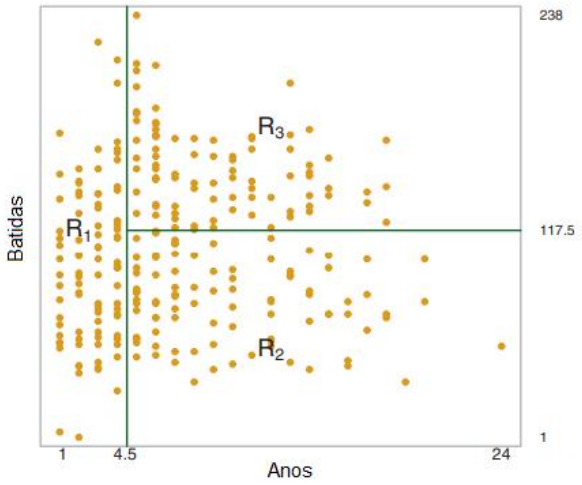
Desse modo, dado um conjunto de dados de vários jogadores de *baseball*, pode-se interpretar a árvore da Figura 9 da seguinte forma: caso ele possua menos do que $4{,}5$ anos de experiência, estará sujeito a um salário de $e^{5{,}11}$ mil dólares, caso ele tenha mais do que $4{,}5$ anos de experiência e tenha rebatido menos do $117{,}5$ vezes ele terá o salário de e^{6} mil dólares e se tiver rebatido mais do que isto terá a recompensa de $e^{6{,}74}$ mil dólares. Este é o modo de como são feitas predições utilizando as árvores de decisão (JAMES et al., 2013).

Os nós que estão relacionados diretamente com um resultado final são chamados de nós do terminal, e são eles que dividem graficamente os dados entre as diferentes classes/valores, como pode ser visto na Figura 10, no qual eles estão representados pelos espaços R_1,R_2 e R_3.

Entendendo a intuição do modelo, ainda é necessário discutir como são criadas as árvores de decisão. De forma simplificada, este passo possui as seguintes duas etapas.

1) O espaço das variáveis preditoras, os conjuntos compostos por todas combinações de diferentes X_1,X_2,...X_p é dividido em J regiões que não se sobrepõe, R_1,R_2,...,R_3 ;
2) Para todas as observações que estejam contidas na região R_j é feita a mesma predição, que é definida como a média dos valores alvos de todos indivíduos de treina presentes em R_j.

Figura 10 – Divisão bidimensional nas três regiões de salário dos jogadores de *baseball*



Fonte: (JAMES et al., 2013)

Para exemplificar o passo a passo, suponha que os dados sejam divididos em duas regiões R_1 e R_2 e que médias dos valores de treinos contida nelas sejam 10 e 20, respectivamente. Desse modo, qualquer indivíduo que esteja dentro de R_1 terá o valor previsto como 10 e caso esteja em R_2 será 20 (JAMES et al., 2013).

Teoricamente essas regiões podem possuir qualquer formato, mas são utilizados retângulos de alta-dimensionalidade com o objetivo de simplificar esta divisão e produzir resultados com maior poder de interpretação. No caso da regressão, é comum utilizar a Soma de Quadrado do Erro (SQE) como objetivo de otimização, isto para todas regiões J que é dada pela equação (8), na qual $\hat{y}_{R_j}$ representa o salário médio da região J.

$$SQR = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \tag{8}$$

.

Idealmente, a cada iteração poderia-se dividir o espaço em diversas regiões, mas, infelizmente, isso se implicaria num processamento computacional tão grande que inviabilizaria tal ação. De modo a contornar este problema, é utilizada a abordagem gulosa *top-down* através da divisão binária recursiva. Tudo começa levando em consideração que exista apenas uma região J e partir daí são feitas subdivisões em dois caminhos para cada nova divisão/nó existente, iniciando do topo até a raiz da árvore (*top-down*). A cada divisão de um nó é escolhida a repartição que obtenha o melhor valor de SQE e é por isso que ela é gulosa, sempre está pensando no passo atual e não na performance do modelo como um todo (JAMES et al., 2013).

Vale lembrar que cada divisão de um nó é considerada apenas uma variável regressora X_j e o ponto de corte s para ela. Com esses valores selecionados, a notação $\{X|X_j < s\}$ representa quais indivíduos X estarão presentes na região J dado o ponto de corte s para X_j.

Sendo assim, as duas novas divisões serão definidas pela equação (9).

$$R_1(j,s) = \{X|X_j < s\} \;\; e \;\; R_2(j,s) = \{X|X_j \geq s\} \tag{9}$$

.

Consequentemente, o valor do SQE referente a cada região será dado pela equação (10) que é o objetivo de minimização.

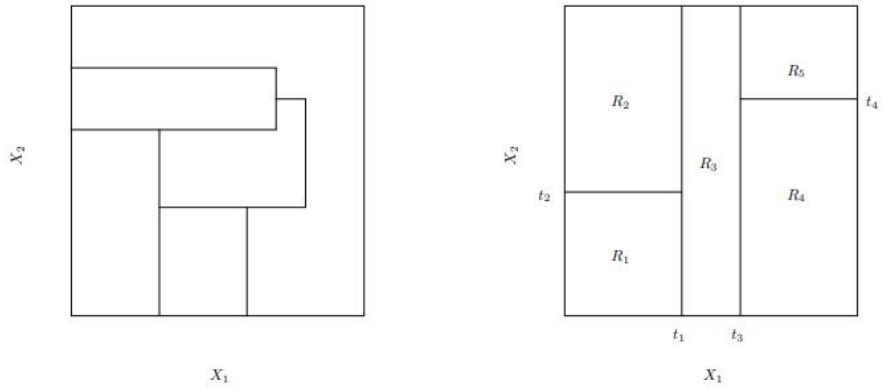
$$\sum_{i:x_i \in R_1(j,s)} (y_i + \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i + \hat{y}_{R_2})^2 \tag{10}$$

.

É de fundamental importância que cada nova região esteja inserida dentro da divisão que lhe deu origem. Feito isto, esse processo é repetido iterativamente e são criadas J regiões, R_1,R_2,...,R_j até o momento em que o critério definido seja alcançado, como cada região final ter apenas três observações, por exemplo.

A Figura 11 traz dois exemplos de subdivisões dentro de um plano bidimensional, o quadro da esquerda representa uma demarcação que não seria possível de ser feita através da metodologia das árvores de decisões, já que possui regiões que não podem ser derivadas de outras duas, ao contrário do quadro a direita que respeita todas condições supracitadas.

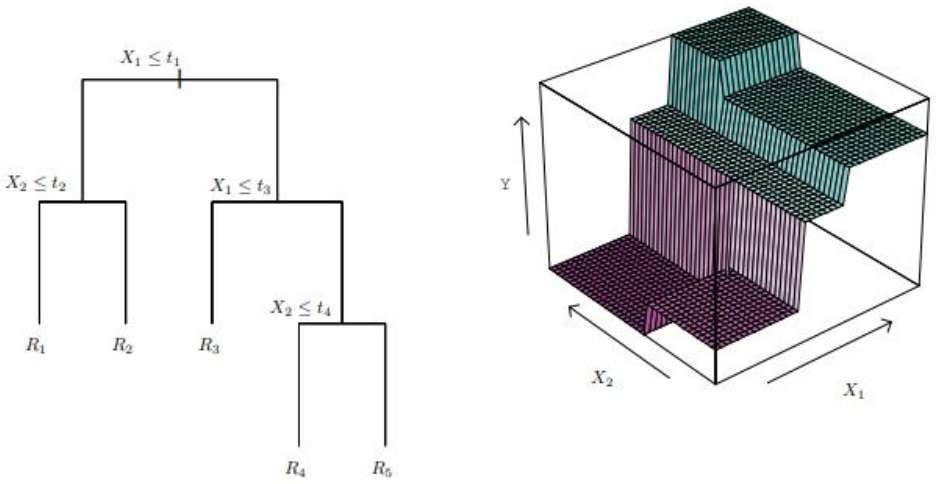
Figura 11 – Regiões demarcadas dentro de um plano bidimensional



Fonte: (JAMES et al., 2013)

Interpretando o quadro da direita presente na Figura 11, é montada a árvore de decisão que está representada na Figura 12, que ainda traz a visão valores médios de Y para cada região R_1,...,R_4.

Figura 12 – Árvore de decisão e sua perspectiva de predição



Fonte: (JAMES et al., 2013)

Através do processo que foi descrito até o momento é possível obter previsões mais acuradas em relação ao conjunto de treinamento. Em contrapartida, isto pode causar com que o modelo tenha uma queda de performance muito grande quando for validado no conjunto de teste, fenômeno conhecido como *overfitting*.

Existem duas abordagens para evitar o *overfitting*. A primeira delas é tornar o critério de parada menos específico, aumentando a capacidade de generalização do modelo para dados não vistos. Já a segunda é a chamada ”poda” das árvores, que consiste em criar uma árvore profunda e ”podá-la” em sub-árvores para avaliar sua performance no conjunto de teste ou validação. Esta última costuma ser mais custosa computacionalmente, porém tende a trazer melhores resultados (QUINLAN, J. R., 1990).

Existem diferentes metodologias para realizar a poda, e algumas delas estão brevemente descritas abaixo.

a) ***Cost complexity Pruning*** : A escolha da árvore é feita através da acurácia obtida utilizando os dados de treino. Ou seja, são utilizados dados não vistos para escolher a sub-árvore que possui melhor aderência ao problema (BREIMAN et al., 1984) ;

b) ***Minimum Description Length Pruning*** : Neste método, procura-se uma sub-árvore, na qual as classes consigam ser definidas pela menor quantidade de dados possível, com o intuito de, consequentemente, aumentar seu poder de generalização (QUINLAN; RIVEST, 1989) ;

c) ***Pessimistic Pruning*** : Esta abordagem, amplifica os erros obtidos pelas sub-árvores, para tentar refletir o tamanho e composição dos dados de treinamento, removendo todos casos em que o erro não se tornou significativamente menor (QUINLAN, J. Ross, 1987).

As árvores de classificação se comportam de forma muito similar às de regressão, tendo a principal diferença que neste caso a predição é feita para uma classe ou resposta qualitativa ao invés da quantitativa. Neste caso, quando uma região é observada, o algoritmo irá considerar como predição a classe que a "domine" (a de maior frequência).

Como a resposta é qualitativa, não há mais sentido em utilizar o SQE como critério de crescimento da árvore. Uma das alternativas mais comuns é o erro de classificação E, representado pela equação (11), na qual $\hat{p}_{mk}$ é a proporção dos dados de treino na m-ésima região e k-ésima classe (JAMES et al., 2013)

$$E = 1 - \max_{k}(\hat{p}_{mk}) \tag{11}$$

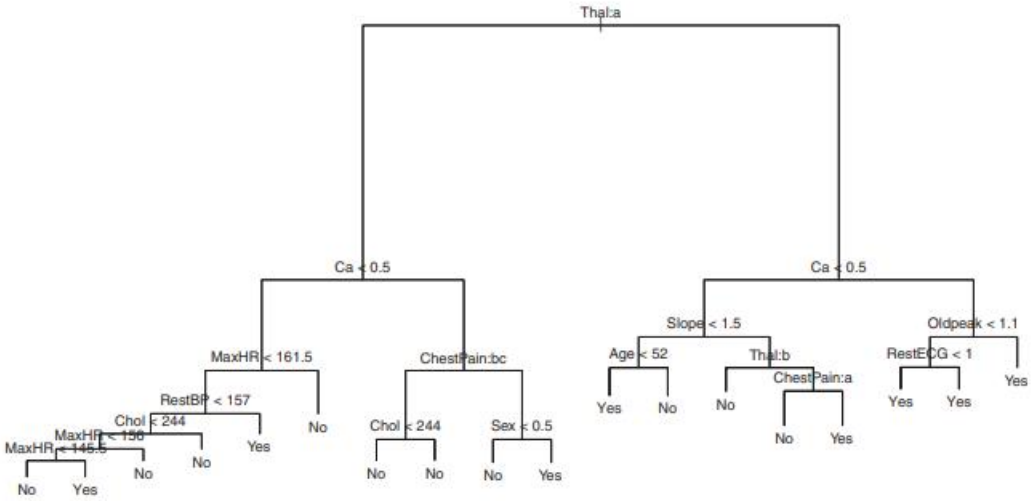
.

O erro de classificação E muitas vezes não é capaz de criar boas árvores. Dito isto, surgem outras opções como o *Gini Index* G que calcula a variância entre todas as K classes. Esta métrica é associada a "pureza" do nó, já que uma vez que ela tenha um valor baixo isso implica em uma predominância de uma classe K, conforme a equação (12) (JAMES et al., 2013).

$$\sum_{k=1}^{K} \hat{p}_{mk}\,(1 - \hat{p}_{mk}) \tag{12}$$

Para exemplificar o processo de criação da árvore de classificação, será utilizado o problema de prever se o paciente possui doenças cardíacas ou não. No primeiro passo, é criada a árvore completa, conforme a Figura 13. Feito isto, é analisado em que nível será realizada a "poda", que neste exemplo utiliza o erro nos conjuntos de treino e teste até chegar a redução final da Figura 14.

Figura 13 – Árvore de classificação para prever a presença de doenças cardíacas



Fonte: (JAMES et al., 2013)

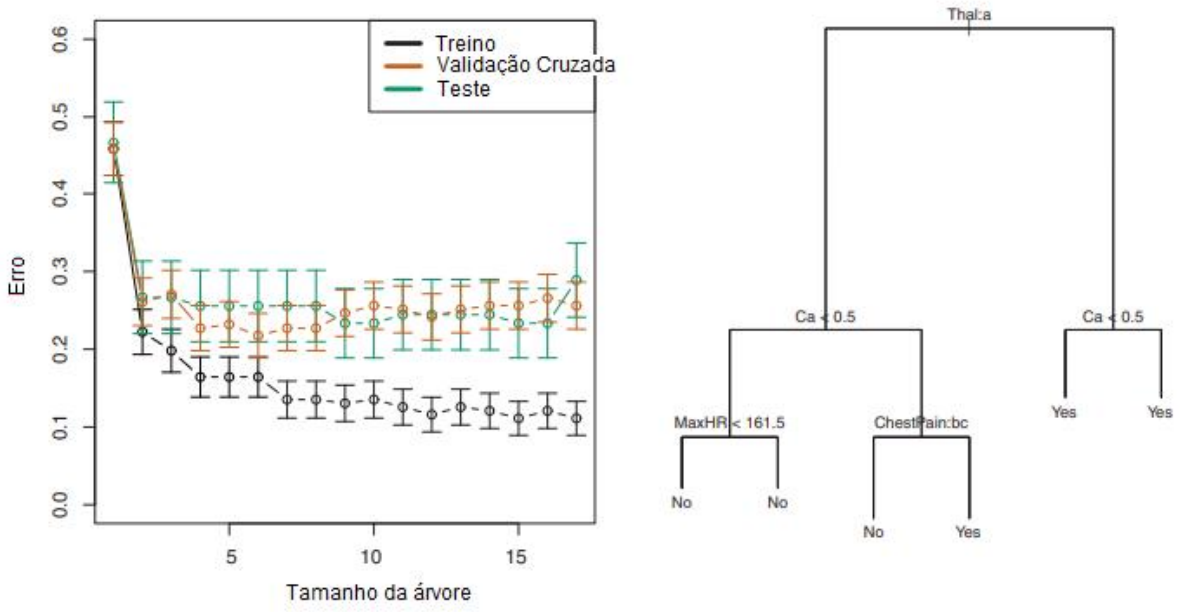
Neste exemplo, pode-se perceber como é possível criar árvores simplificadas que possuam maior poder de generalização. Ainda assim, uma das desvantagens deste tipo de modelo é que eles, geralmente, não conseguem se comparar a classificadores clássicos como a Regressão Logística, além de serem muito sensíveis a qualquer alteração simples nos dados. Para mitigar esse problema, surgem as técnicas de *Bagging* e *Boosting* que serão apresentadas no próximo tópico.

2.2.2.2 Bagging e Boosting

Como foi dito, apenas uma árvore de decisão tende a ser pior do que os outros tipos de modelagem disponíveis para o caso da classificação. Dessa forma, as técnicas de *Bagging* e *Boosting* propõem a construção de diversos classificadores para serem utilizados de uma forma conjunta e tentar incrementar, significativamente, a performance do modelo. Lembrando que esta metodologia pode ser aplicada a qualquer conjunto de algoritmos de ML que não precisam, necessariamente, ser os mesmos (QUINLAN et al., 1996).

O *Bagging* é inicializado utilizando o *Bootstrap*, que nada mais é do que a criação de N re-amostragens do conjunto de dados original com a possibilidade de repetir pontos mais do que duas vezes, com o objetivo de tentar reproduzir a distribuição de origem dos dados, vide Figura 15. Deste modo, para cada re-amostragem gerada, é treinada uma árvore, por exemplo,

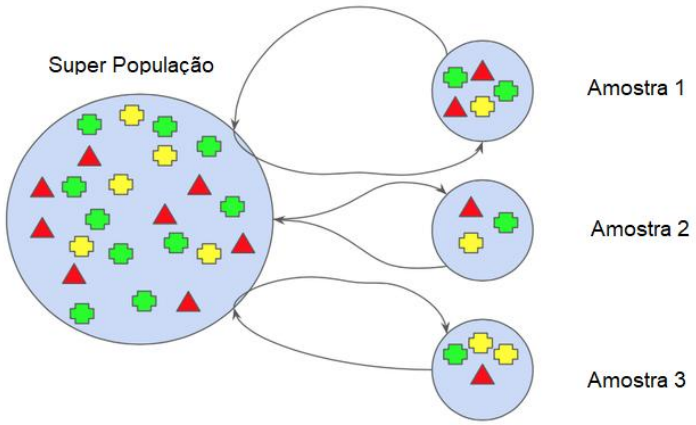
Figura 14 – Avaliação do erro e construção da árvore de classificação podada



Fonte: (JAMES et al., 2013)

para realizar a predição dentro daquele universo. Por fim, todas elas são utilizadas de forma ponderada e em paralelo fazendo a predição da classe, conforme o quadro esquerdo da Figura 16 (QUINLAN et al., 1996).

Figura 15 – Representação gráfica da técnica de *Bootstrap* com $N = 3$



Fonte: (SINGHAL, 2020)

Em árvores de decisão, o *Bagging* costuma trazer melhorias significativas. Isso porque, no primeiro passo, são criadas árvores profundas que possuem uma variância alta, porém um

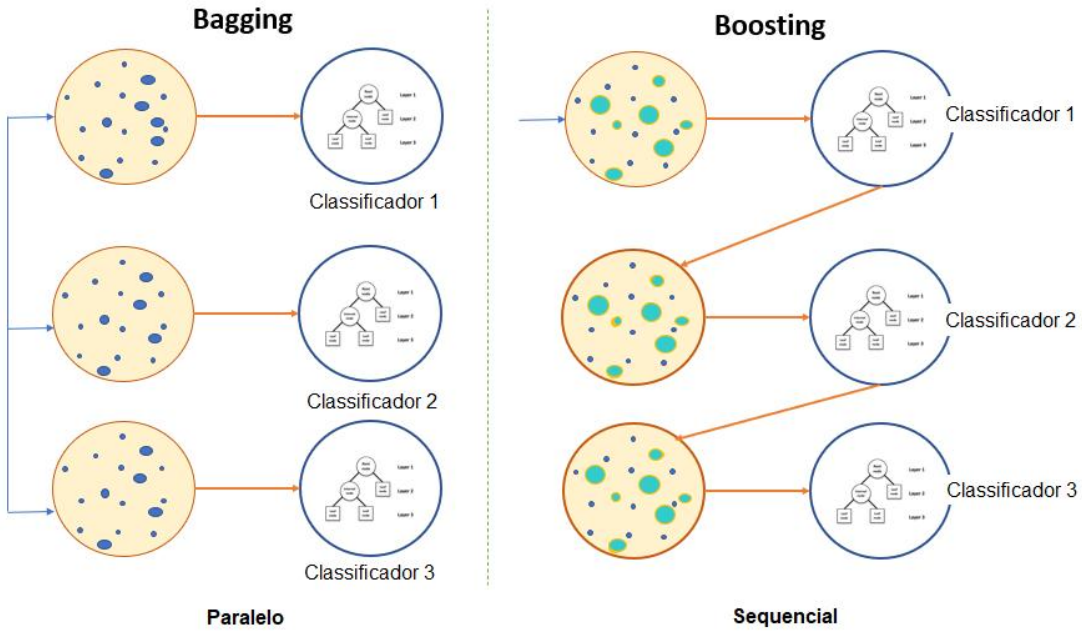
baixo bias e quando associadas em conjunto, a variância tende a diminuir, criando um cenário promissor para o modelo (JAMES et al., 2013).

Uma das implementações mais conhecidos de *Bagging* dentro da família de modelos de árvore é a *Random Forest*, que, segundo Breiman (2001), é capaz de superar os principais algoritmos de ML, o que não era observado quando utilizava-se árvores de decisão puras.

O *Boosting* também pode ser aplicado em qualquer conjunto de algoritmos de ML. De forma diferente do *Bagging*, este método utiliza diversos modelos de forma sequencial, ou seja, é utilizado o resultado obtido pelo primeiro modelo treinado como entrada do segundo e assim sucessivamente, conforme a Figura 16 lembrando que ele não utiliza o *Bootstrap* (JAMES et al., 2013).

A intuição por trás do *Boosting* é fazer com que o modelo aprenda de forma lenta, trazendo mais confiabilidade nos resultados. Nesta abordagem, o resultado que é utilizado como insumo de cada modelo, são os resíduos do anterior, ao invés da variável resultado Y. Assim como a metodologia paralela, ela também costuma trazer bons resultados, como o AdaBoost e o *Gradient Boost Machine* (GBM), que são da família dos modelos de árvore (QUINLAN et al., 1996).

Figura 16 – Representação gráfica das técnicas de *Bagging* e *Boosting*

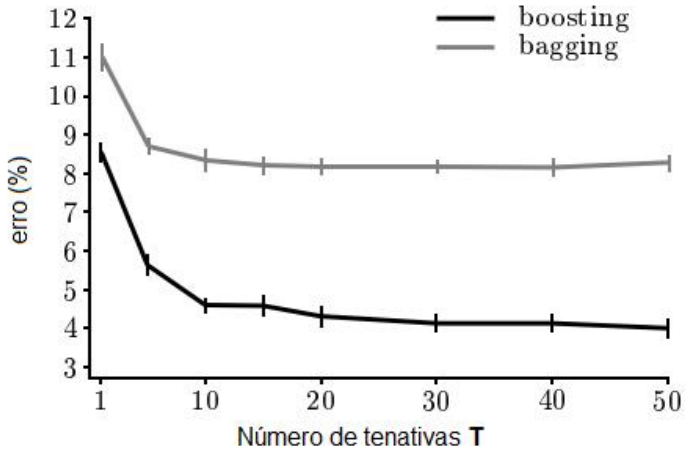


Fonte: (SINGHAL, 2020)

Em um experimento feito por (QUINLAN et al., 1996), no qual ele utiliza dados de partidas de xadrez, é possível perceber como o erro das árvores de decisão, utilizando as duas

técnicas supracitadas, tende a melhorar a performance do modelo como um todo. Na Figura 17, o *Boosting* obteve o melhor resultado para o erro, mas ,mesmo assim, ambas mostraram evolução positiva do modelo.

Figura 17 – Erro utilizando *Bagging* e *Boosting* aliada a árvores de decisão



Fonte: (QUINLAN et al., 1996)

Em suma, os modelos de árvores são grandes aliados da modelagem estatística devido a sua versatilidade e poder de generalização, quando usada da forma correta. Quando são utilizadas muitas árvores em conjunto, existe o ponto negativo da perda da simplicidade interpretabilidade já que será necessário analisar centenas ou até milhares de árvores para inferir algum comportamento ”detectável” que leve a uma classe ou valor.

2.2.3 *Support Vector Machines*(SVM)

O SVM é um tipo de algoritmo que é amplamente utilizado nas tarefas de classificação. Esta técnica é uma generalização do *maximal margin classifier* e, por isso, serão apresentados alguns conceitos prévios necessários para o entendimento do seu funcionamento.

2.2.3.1 Classificação utilizando hiperplanos

O primeiro conceito a ser utilizado é o da classificação utilizando hiperplanos. Para simplificar o entendimento é usado o plano 2D como exemplo, que define um hiperplano pela equação (13).

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0 \tag{13}$$

Expandindo para mais dimensões, podemos representá-lo pela equação (14).

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p = 0 \tag{14}$$

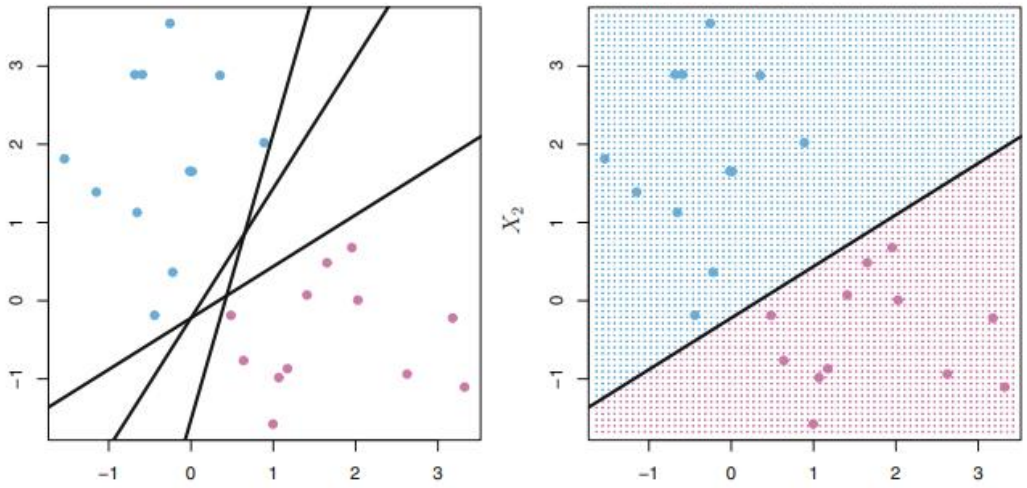
Utilizando as equações anteriores, pode-se definir um separador de dimensão p quando uma classe esteja ”sob” o hiperplano, equação (15) e ”sobre” o hiperplano, equação (16).

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p > 0 \tag{15}$$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p < 0 \tag{16}$$

Voltando ao caso 2D, o hiperplano separador se resume a um limitador de decisão linear, conforme o quadro direito da Figura 18. Como pode-se perceber, nesse exemplo, existem várias linhas que são capazes de separar perfeitamente as duas classes , conforme o quadro esquerdo da Figura 18, desse modo é necessário utilizar algum critério para definir qual linha melhor divide os dados impostos a ela e, dai, utiliza-se o *Maximal Margin Classifier* (JAMES et al., 2013).

Figura 18 – Utilização de hiperplanos para classificação binária em um espaço 2D



Fonte: (JAMES et al., 2013)

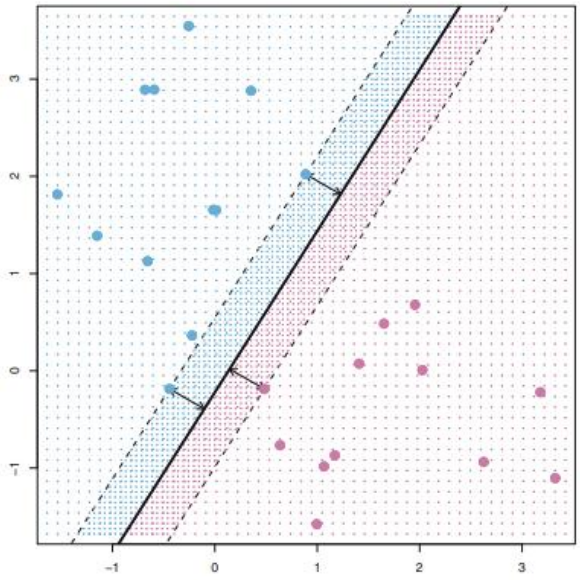
2.2.3.2 Maximal Margin Classifier e os Support Vector Classifiers

Como foi dito no tópico anterior, em casos que seja possível fazer a separação perfeita entre as duas classes é muito comum que existem infinitas limitadores lineares que sejam capazes de realizar uma classificação binária sem erro, lembrando que existem situações que não são linearmente separáveis e, consequentemente, inutilizam esta técnica (JAMES et al., 2013).

Dessa maneira, para resolver este problema pode ser utilizado o *Maximal Margin Classifier* que, basicamente, irá calcular a distância perpendicular de cada exemplo de treino até a linha/hiperplano classificador, gravando os exemplos que possuam a menor distância até a ela que será chamada de margem. Já os pontos que são utilizados para calcular a margem são chamados de *support vectors* e podem ser visualizados na Figura 19.

Ainda assim, este algoritmo procura um divisor que gere uma grande margem que, intuitivamente, irá separar da melhor forma as duas classes de modo que elas estejam mais longe uma da outra. Entretanto, é necessário ter cuidado com conjuntos de dados de alta dimensionalidade, que podem trazer ótimos resultados no treino e falhar no teste (JAMES et al., 2013).

Figura 19 – Utilização do *Maximal Margin Classifier* para escolha da reta com maior poder de generalização

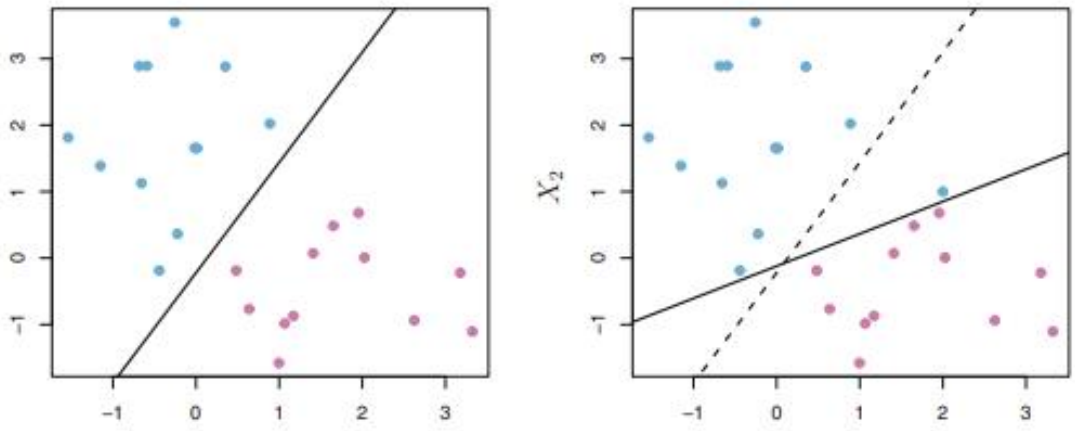


Fonte: (JAMES et al., 2013)

Outro ponto interessante é que, uma vez treinado, o *Maximal Margin Classifier* depende, exclusivamente, dos *support vectors*, fazendo com que a movimentação dos outros dados existentes não tenha nenhum efeito sobre o modelo.

Por outro lado, separar as duas classes perfeitamente com uma margem relativamente pequena, pode levar tornar o modelo frágil em algumas situações, como a da Figura 20 por exemplo, na qual a introdução de um único ponto desloca brutalmente a posição do classificador (JAMES et al., 2013).

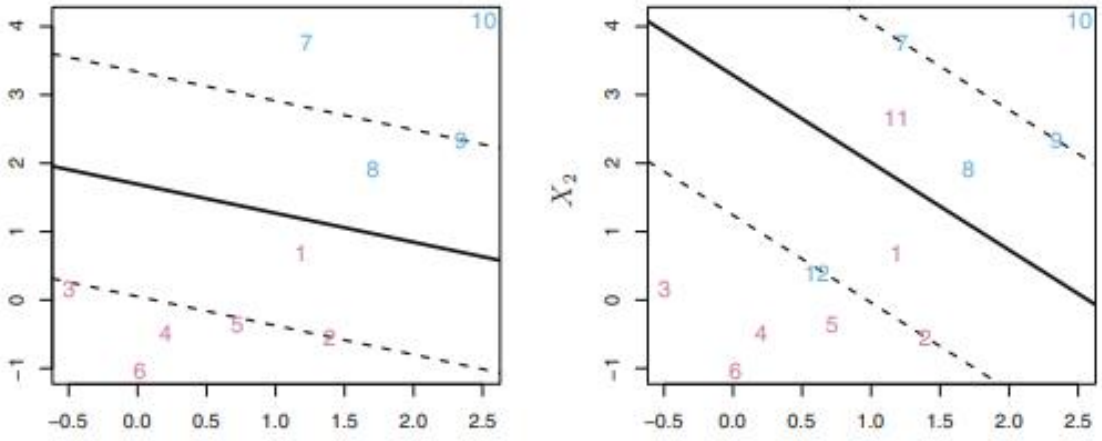
Figura 20 – Sensibilidade a novos pontos de treino no *Maximal Margin Classifier*



Fonte: (JAMES et al., 2013)

Desta maneira, os *Support Vector Classifiers* surgem com uma abordagem um pouco diferente: já não é mais necessário fazer uma divisão perfeita, ele procura por hiperplanos que formem uma margem "segura" e evitem problemas de alta variância nos dados de treino, *overfitting*. A grande vantagem desta metodologia é que ao perder um pouco de precisão, ainda é possível aumentar o poder de generalização do modelo além de lidar com casos que não sejam perfeitamente separáveis, como pode ser visto na Figura 21.

Figura 21 – Efeito da adição de novos pontos de teste no *Support Vector Classifier*



Fonte: (JAMES et al., 2013)

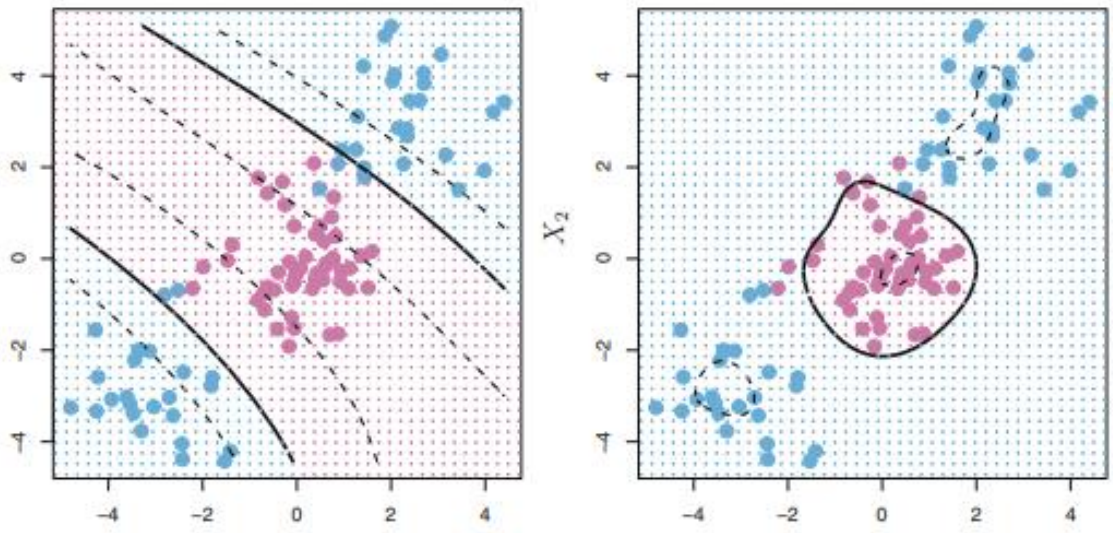
2.2.3.3 SVM e Kernels

As técnicas anteriores ainda não conseguem fazer uma boa distinção em casos que existam uma divisão não-linear entre as classes. O SVM e os *Kernels* surgem como uma alternativa para lidar com esta situação e realizar a classificação de maneira adequada nestes casos.

Deste modo, os hiperplanos são trocados por outros tipos de funções como as quadráticas e cúbicas que tem a capacidade de lidar com a não linearidade dos grupos de dados. Essas novas funções que irão definir como o limite de decisão funcionará são chamadas de *Kernels* (JAMES et al., 2013).

Utilizando o exemplo da Figura 22, no quadro esquerdo é utilizado um *Kernel* polinomial de grau 3 e no quadro direito é utilizado um *Radial Kernel*. Observando a figura, pode-se perceber que este caso não seria resolvido com um hiperplano comum e por isso é muito comum utilizar SVM para não ficar ”preso” apenas a problemas linearmente separáveis.

Figura 22 – Dois tipos de *Kernel* utilizados no SVM

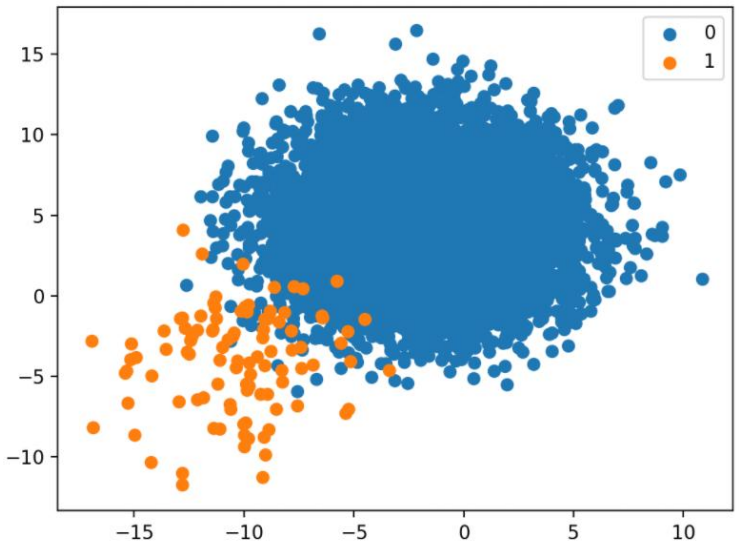


Fonte: (JAMES et al., 2013)

2.3 Re-Amostragem

Muitos *datasets* que possuem uma variável resposta qualitativa (utilizada para classificação) costumam ter suas classes desbalanceadas, como no exemplo da Figura 23 que possui uma quantidade muito maior de dados na classe 0 do que na 1. Isto pode ocorrer por alguns

Figura 23 – Conjunto de dados com classes desbalanceadas



Fonte: (BROWNLEE, 2020)

motivos, dentre eles a própria raridade do fenômeno pode implicar diretamente nessa ocasião, assim como problemas na coleta de dados.

A maior parte dos modelos de ML funciona melhor quando expostos a dados balanceados, mas há casos em que, de fato, não é possível obter informações que assim estejam . A partir daí, o campo da re-amostragem foca na pesquisa de métodos que sejam capazes de devolver o equidade das classes sem afetar, desproporcionalmente, as características daquele conjunto de dados (SHELKE; DESHMUKH; SHANDILYA, 2017).

Esta tarefa é subdividida em dois casos: a subamostragem e a superamostragem. A primeira tem como objetivo diminuir a classe dominante até que ela chegue no patamar numérico da outra, idealmente buscando uma proporção de 1:1. Já a segunda foca exatamente no contrário, ”expandir” a classe não dominante a fim de se equiparar a outra (SHELKE; DESHMUKH; SHANDILYA, 2017).

Para a tarefa de subamostragem ou *undersampling*, são listadas algumas técnicas abaixo, considerando que a classe positiva seja majoritária:

a) **Subamostragem randômica** : Neste método, são removidos exemplos de forma aleatória da classe positiva até que ela possua a mesma quantidade de indivíduos da negativa (SHELKE; DESHMUKH; SHANDILYA, 2017) ;

b) ***EasyEnsemble*** : A classe majoritária é dividida em diversas subamostras que possuam o mesmo tamanho da minoritária. Feito isto, é treinado um modelo para

cada subamostra em conjunto com os dados da classe minoritária e é feito uma agregação de todos esses modelos utilizando a técnica de *boosting*, por exemplo (LIU; WU; ZHOU, 2008).

Em termos de superamostragem, é possível citar duas técnicas que são amplamente utilizadas:

a) ***Synthetic Minority Over-sampling Techniquen (SMOTE)***

b) ***Ranked Minority Oversampling in Boosting (RAMOBoost)***

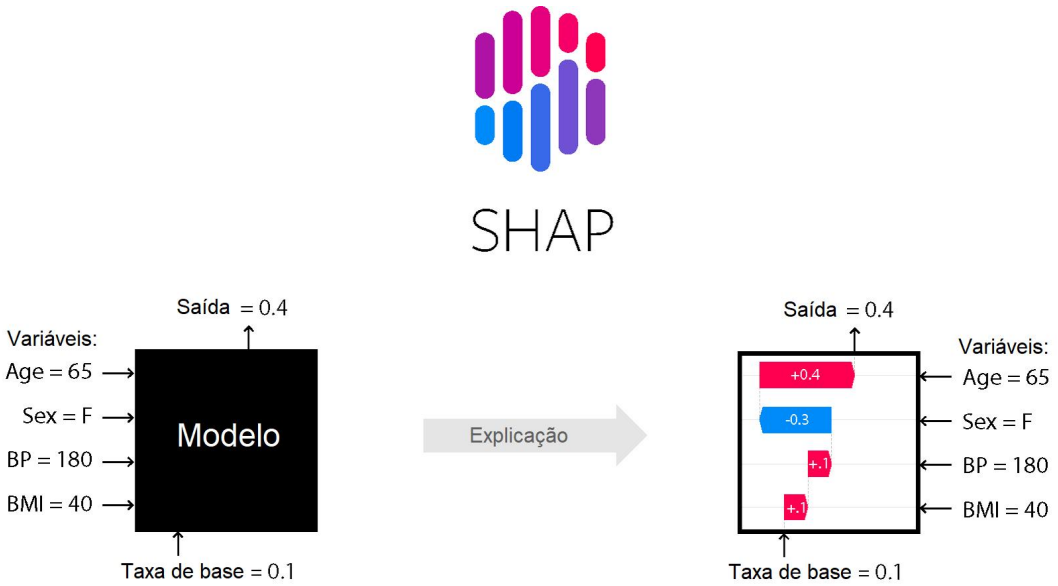
2.4 Interpretabilidade utilizando Valores SHAP

Com o advento do poder computacional e a evolução da complexidade dos algoritmos mais atuais de ML, fica cada vez mais difícil extrair interpretabilidade das soluções criadas. Para isso, existe uma vertente que vem estudando a capacidade de ”adicionar” explicabilidade para modelos de alta complexidade, como modelos de *boosting* de árvores que podem combinar milhares delas.

Dito isto, na maior parte dos casos que obtêm-se a maior acurácia em qualquer tipo de predição são utilizados algoritmos que até os maiores pesquisadores/profissionais não conseguem extrair nenhum tipo de interpretação sem um esforço gigantesco. Sendo assim, o *trade-off* entre alta performance e interpretabilidade, ajudou na aceleração do estudo de técnicas como o SHAP (LUNDBERG; LEE, 2017).

A intuição por trás desse *framework* é simplesmente ”desvendar” a caixa preta criada por alguns tipos de modelos, conforme a Figura 24.

Figura 24 – Solução no desafio de explicabilidade proposto pelo SHAP



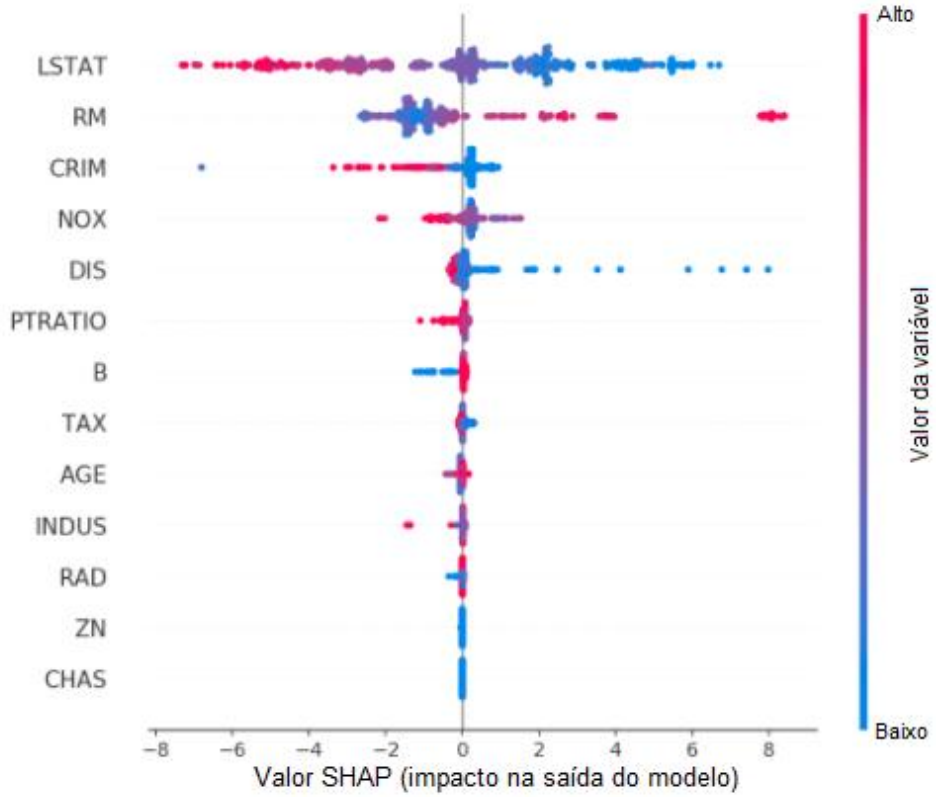
Fonte: (LUNDBERG, 2020)

Em linhas gerais, para criar uma mensuração do impacto de cada variável na predição do próprio modelo é utilizado o conceito de modelo de explicação, que nada mais é que uma variação do método original, mas de uma forma simplificada o bastante para avaliar a sensibilidade a cada regressor quando se trata da resposta (LUNDBERG; LEE, 2017).

Deste modo, é possível avaliar se as entradas impactam de uma forma positiva ou negativa na predição, uma vez que a metodologia irá dar pesos/importâncias para cada uma das variáveis utilizadas pelo modelo, como pode ser visualizado na Figura 25. Tudo isto mostra como essa ferramenta pode ser poderosíssima , já que ela une a alta performance dos modelos de alta complexidade, extraindo um bom grau de interpretabilidade.

No artigo proposto por Lundberg e Lee (2017) é possível observar como foram aliadas a teoria dos jogos com modelos de explicabilidade já existentes para esse fim. Ainda assim, há um grande desafio de processamento para realizar o diagnóstico dos modelos utilizando esta técnica, que ainda está sendo desenvolvida e otimizada.

Figura 25 – Descrição do impacto das variáveis de entrada, de acordo com o SHAP



Fonte: (LUNDBERG, 2020)

3 Trabalhos relacionados

Com a evolução do poder computacional e das aplicações de ML nas diversas áreas do conhecimento, surgiram vários estudos nesse campo para resolver problemas de variadas naturezas. Dentro destes, problemas relacionando a medicina aos modelos de aprendizado de máquina que vem sendo largamente estudados ao longo dos últimos anos.

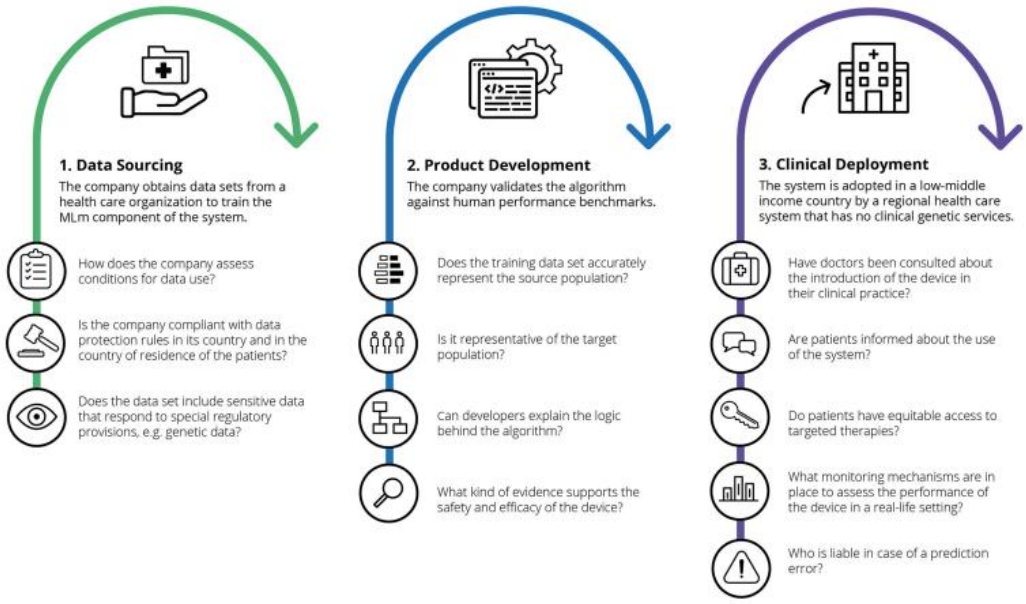
Além do mais, a quantidade de dados clínicos que vem sendo armazenados vem crescendo constantemente ao longo dos últimos anos e isso, substancialmente, pode mudar a forma de como é realizado o cuidado médico. O trabalho realizado por Deo (2015) traz exatamente essa perspectiva: de que a relação entre o paciente e o médico, assim como o seu cuidado serão fortemente beneficiadas pelas técnicas de ML, principalmente pelo fato de estarmos aumentando a coleta dos dados clínicos. Lembrando que o autor deixa claro que o aprendizado de máquina irá auxiliar o médico nas suas tomadas de decisão e não o substitui-lo na sua atividade.

A pesquisa realizada por Sidey-Gibbons e Sidey-Gibbons (2019) demonstra um caso de uso prático dentro deste universo: diagnóstico de câncer utilizando dados extraídos de exames de imagem. Apesar do número limitado de dados, que foram disponibilizados de forma pública, o modelo desenvolvido conseguiu alcançar resultados satisfatórios que teriam grandes chances de auxiliar na confecção do diagnóstico desses pacientes. A Figura 27 mostra a curva ROC comparativa entre os algoritmos utilizados(regressão linear, rede neural e SVM) e mostram a relação entre a sensibilidade e a especificidade de cada um deles e ,ainda de acordo com o autor, eles conseguiram chegar até os valores 0,99 e 0,94 destas métricas, respectivamente.

Apesar de ser uma área bastante promissora, existem diversos cuidados a serem tomados até utilizarmos o aprendizado de máquina como suporte a tomada de decisões e esse é o foco da publicação de Cabitza, Rasoini e Gensini (2017). Um dos pontos levantados neste artigo é o fato da confiabilidade dos dados utilizados e as consequências relacionadas a isso, afinal para fazer boas previsões nós precisamos de bons dados para garantir o treinamento desses agentes autônomos. Outra questão importante abordada é o fenômeno da desqualificação, que é quando o nível de habilidade, de realizar diagnóstico por parte dos médicos por exemplo, pode ser afetada no longo prazo ao semi-automatizar uma parte do seu trabalho, o que seria um cenário não desejável. Nesse caso, fica o alerta para aprimorar o monitoramento da qualidade desses sistemas, assim como medidas regulatórias a fim de utilizar essa ferramenta de forma a melhorar o setor de saúde como um todo e, segundo o autor, a pesquisa pode ser um grande aliado para esse desenvolvimento.

Ainda sobre pontos de atenção nesta área, Vayena, Blasimme e Cohen (2018) trazem informações importantes sobre o debate sobre a ética dentro do campo de ML na medicina. Dados trazidos neste artigo mostram que grande parte dos médicos acreditam que a Inteligência Artificial trará benefícios à medicina, mas ainda quase metade deles também acham que ela induzirá a erros fatais ou não irá atender a todas expectativas nelas depositadas. Para deixar a comunidade mais confortável e confiante nesse tipo de abordagem, é proposto toda uma metodologia no processo de criação desses sistemas de auxílio à tomada de decisão para viabilizar a transparência, privacidade e significância desses dados: desde representatividade estatística da população, definição de responsáveis pelos possíveis erros ocasionados pelas previsões e visibilidade do uso desses sistemas ao paciente. Algumas recomendações dadas pelo autor, para cada etapa do desenvolvimento desse tipo de solução, podem ser vistas na Figura 26.

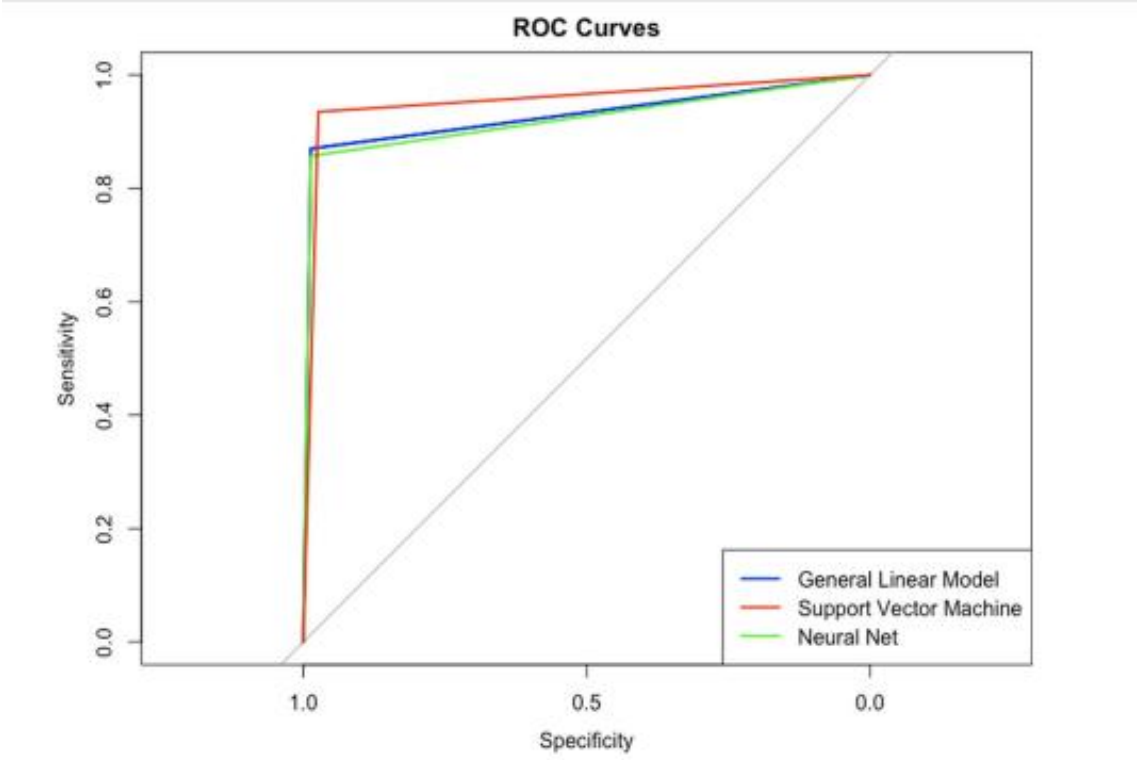
Figura 26 – Como desenvolver um *software* de machine learning em uma abordagem ética



Fonte: (VAYENA; BLASIMME; COHEN, 2018)

Esse debate é de suma importância para o amadurecimento da área, uma vez que a pesquisa trará conclusões científicas sobre o uso de ML na medicina, ajudando a comunidade médica a entender as limitações e benefícios que podem ser adquiridos a partir dessas metodologias.

Figura 27 – Curva ROC dos modelos treinados para diagnóstico de câncer de mama



Fonte: (SIDEY-GIBBONS; SIDEY-GIBBONS, 2019)

Entendendo como os algoritmos de inteligência artificial estão sendo utilizados através das pesquisas citadas e sabendo que o estudo de caso deste trabalho será relacionado ao COVID-19, o restante deste capítulo tem o foco em pesquisas de ML diretamente relacionadas com o estudo de caso proposto, a fim de se conectar de forma mais direta ao que foi desenvolvido.

No trabalho de Moraes Batista et al. (2020) foi apresentada uma metodologia para detectar o diagnóstico positivo de COVID-19 utilizando cinco tipos de algoritmo de aprendizado de máquina: redes neurais, *gradient boosted trees*, *random forests*, regressão logística e SVM. Para isto, foram utilizadas quinze variáveis para treinar os modelos, dentre resultados de hemograma e algumas variáveis demográficas como idade e gênero, possuindo informações de 256 pacientes, do Hospital Israelita Albert Einstein , uma quantidade relativamente baixa.

Ainda sobre o estudo realizado por Moraes Batista et al. (2020), foi adotado o uso de 70% dos dados para treino e 30% para teste, além de uma validação cruzada aliada a otimização bayesiana para ajustar os hiperparâmetros. Por fim, as métricas para avaliar a performance do modelo foram a areá sob a curva ROC, sensibilidade, especificidade, F1-Score, Brier score, *Positive Predictive Value* (PPV) e *Negative Predictive Value* (NPV).

Os algoritmos utilizados obtiveram resultados similares que foram animadores como área sob a curva ROC mínima de 0,84, os resultados podem ser vistos na Figura 28. Ainda assim, os dados utilizados foram considerados pelos autores uma amostra pequena e de apenas um hospital, o que pode gerar algum tipo de viés que tentaram ser mitigados apenas ao utilizar pacientes da emergência.

Figura 28 – Resultados obtidos no estudo realizado com pacientes do Hospital Albert Einstein

Table 2: Performance metrics for each machine learning algorithm for the test set.

Algorithm	AUC	Sensitivity	Specificity	F1-score	Brier score	PPV	NPV
Supp. Vec. Machines	0.847	0.677	0.850	0.724	0.160	0.778	0.773
Random Forests	0.847	0.677	0.850	0.724	0.161	0.778	0.773
Neural Networks	0.844	0.742	0.800	0.742	0.187	0.742	0.800
Logistic Regression	0.843	0.742	0.825	0.754	0.161	0.767	0.805
Grad. Boost. Trees	0.842	0.806	0.800	0.781	0.171	0.758	0.843

Fonte: (MORAES BATISTA et al., 2020)

De forma geral, o estudo de Moraes Batista et al. (2020) conclui que existem evidências que suportam que pode-se aliar o ML com medicina para realizar previsões de potencial infecção de COVID-19 em áreas que não possuam amplo acesso a testes físicos. Outro ponto importante, é tornar acessível o treinamento dos algoritmos em cada local específico, capturando um comportamento mais acurado na relação de condições clínicas e diagnóstico do COVID-19.

Já em Wynants et al. (2020) foi realizada uma revisão sistemática de diversos estudos, em todo o mundo, que utilizaram aprendizado de máquina para resolver problemas relacionados a pandemia do COVID-19. Desde detecção de diagnósticos por exames clínicos, como por imagem e até a realização de prognósticos de pacientes que já haviam contraído a doença. Isto mostra como o ML vem se tornando uma ferramenta em evidência, quando se pensa em auxiliar os profissionais de saúde para tomadas de decisão mais assertivas.

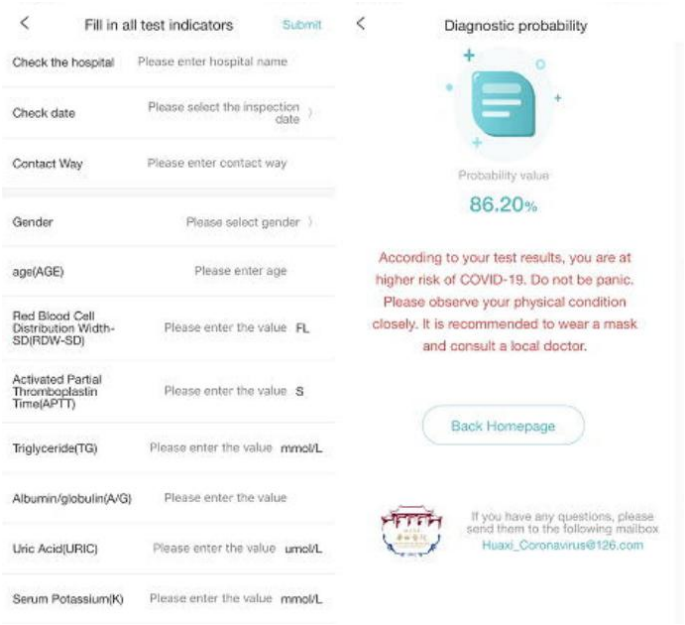
Ainda em relação a estudos específicos em prever o diagnóstico dessa doença, Meng et al. (2020) utilizaram dados de 620 pacientes chineses, com resultados positivos e negativos do exame RT-PCR, para treinar um modelo de regressão logística a fim de prever se cada um deles estaria infectado ou não.

Para isso, foram utilizados 35 indicadores referentes a hemogramas, coagulação e examinações bioquímicas. Assim como no estudo Moraes Batista et al. (2020), o conjunto de dados de treino escolhido foi de 70% do total, além de terem utilizado o modelo linear *Least Absolute*

Shrinkage and 81 Selection Operator (LASSO) para realizar a escolha das variáveis preditores mais relevantes.

Um ponto interessante, é que todo o modelo e resultados foram encapsulados numa aplicação visual e foi criado um *app* para auxiliar no combate do coronavírus naquele país, uma visualização das telas da aplicação podem ser vistas na Figura 29. Os resultados do modelo também trouxeram resultados relevantes como a área sob a curva ROC de 0,89 , mostrando, mais uma vez, as oportunidades de usabilidade desta ferramenta em situações deste tipo.

Figura 29 – Aplicativo desenvolvido para ajudar na detecção do COVID-19 com o auxilio de ferramentas de aprendizado de máquina

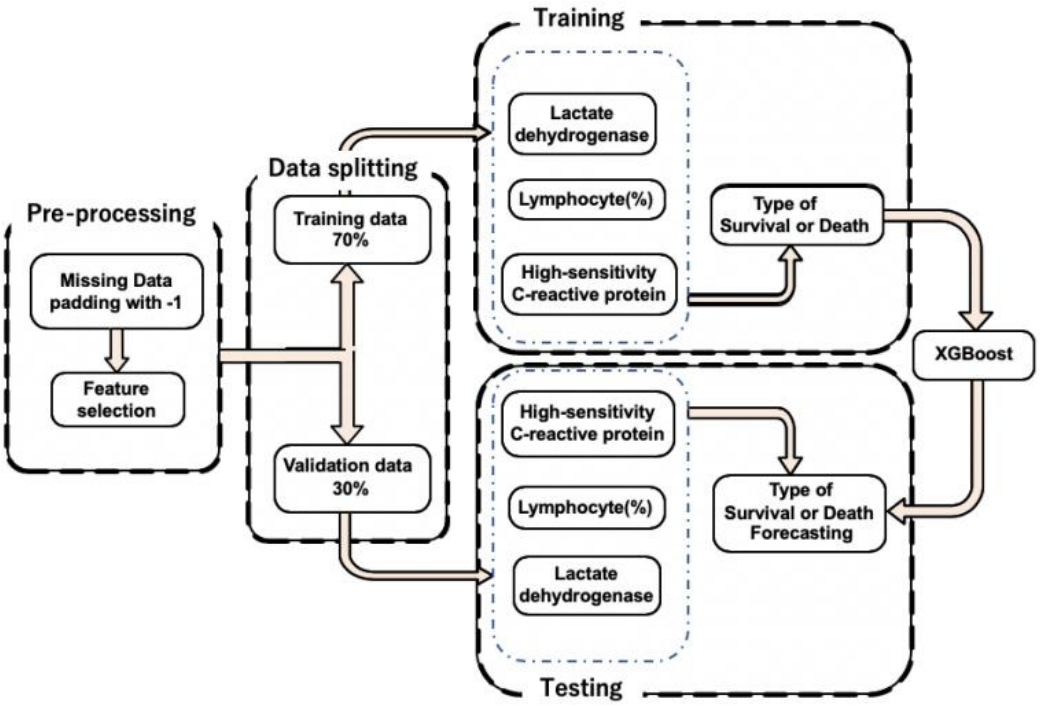


Fonte: (MENG et al., 2020)

No campo de prognósticos, mas ainda relacionados ao Sars-CoV-2, Yan et al. (2020) exploram os modelos de aprendizado de máquina para fazer previsões relacionadas ao que irá acontecer com aqueles pacientes que já foram infectados, principalmente em relação a indicadores de mortalidade.

Para isto, foram utilizados dados de prontuário eletrônico de 2779 pacientes e foi desenhado um fluxo para tratar os dados e treinar um modelo, da família dos algoritmos de árvore, o XGBoost. Neste caso, também foi feito um *split* utilizando 70% dos dados para treino, utilizando o modelo de árvore para auxiliar na interpretabilidade do problema, conforme o fluxograma apresentado na Figura 30

Figura 30 – Fluxograma indicando como foram utilizados os dados para treinar o modelo de previsão de mortalidade



Fonte: (YAN et al., 2020)

Todos estes estudos revelam que o uso do ML no campo da medicina está se disseminando e vem mostrando evolução nos resultados obtidos. Desse modo, ainda existe um grande espaço para desenvolver novas aplicações e até trazer melhorias para as já existentes, mostrando a relevância de estudos desta natureza.

4 Metodologia

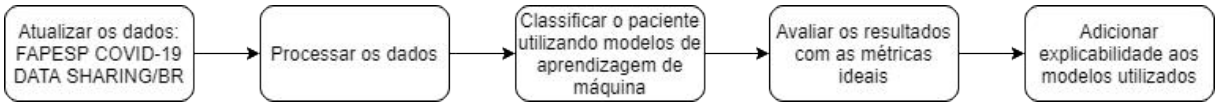
Como um estudo de caso para o tema proposto, este trabalho elaborou um sistema de classificação para indicar se pacientes que deram entrada em unidades de Saúde Hospitalares recentemente, tem alto potencial para testarem positivo para o COVID-19, que é um tipo de doença infecciosa. O principal objetivo desse sistema não seria a substituição dos exames convencionais e sim utilizar essa tecnologia para auxiliar as tomadas de decisões médicas, pensando em um contexto em que o exame da referida doença esteja com alguma dificuldade no mercado: indisponibilidade, alto prazo de liberação dos resultados, confiabilidade baixa, dentre outros.

Para realizar essa tarefa foram utilizados os dados disponibilizados pela FAPESP (2020), que estão publicados no portal ”FAPESP COVID-19 Data Sharing/BR”.

Para resumir o entendimento desse sistema ele foi dividido em cinco etapas de desenvolvimento e sustentação, que são representadas pelo fluxograma da Figura 31:

a) Obtenção e atualização dos dados;
b) Processamento dos dados;
c) Desenvolvimento da solução , utilizando modelos de aprendizagem de máquina para classificação;
d) Avaliação dos resultados obtidos;
e) Utilização de métodos de explicabilidade para interpretar modelos mais complexos.

Figura 31 – Fluxograma da criação do Sistema de Classificação



Fonte: Autor

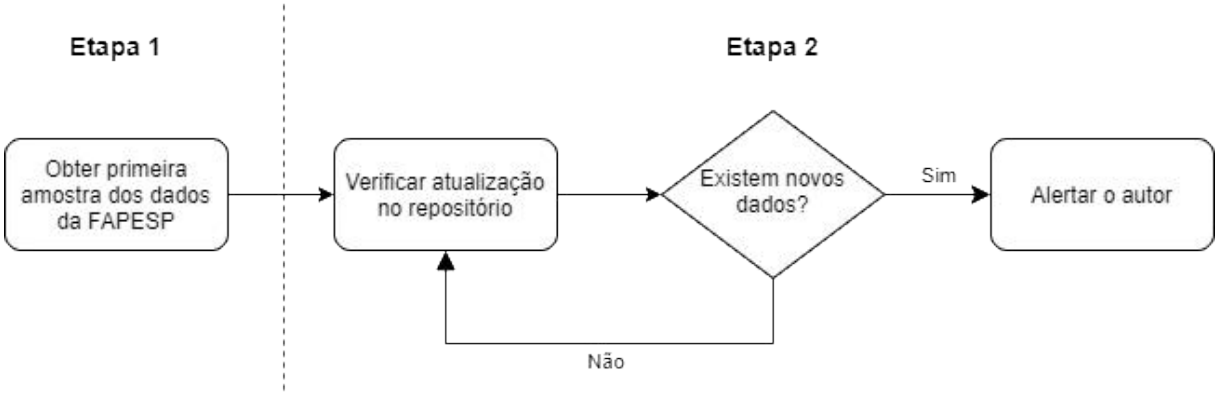
4.1 Coleta e atualização dos dados

Como mencionado, a primeira etapa foi a coleta e atualização dos dados que foram utilizados para treinar e avaliar os modelos de Inteligência Artificial. Vale ressaltar que que os dados funcionam como matéria-prima dos nossos modelos estatísticos, então ao utilizar dados

de baixa qualidade e de fontes não confiáveis podemos perder performance nos algoritmos treinados. Por isso, devemos sempre entender a origem e o método de coleta para realizar o melhor processamento possível.

O repositório citado sofreu atualizações em intervalos de tempos não definidos, sendo assim o projeto foi dividido em duas etapas, ilustradas pela Figura 32: a priori, foram utilizados os dados coletados no início da pesquisa, mas como existia uma chance deles serem atualizados durante o processo, na segunda etapa, essa nova parcela foi inserida no conjunto geral e serviu com o propósito de expandir a capacidade dos nossos modelos, aumentando o seu poder de generalização.

Figura 32 – Fluxograma da coleta de dados



Fonte: Autor

Para a segunda etapa, foi desenvolvido um robô, em python, responsável por realizar o *Web Crawling* do repositório da FAPESP. Em outras palavras, ele verifica diariamente se foi feita uma nova atualização no *dataset* da COVID-19, caso tenha ocorrido o robô alerta o autor que irá tomar a decisão do que fazer com esses novos dados.

Os dados disponibilizados pela FAPESP (2020) inicialmente foram divulgados para três fontes distintas que são: Hospital Israelita Albert Einstein, Grupo Fleury e Hospital Sírio Libanês e ao longo deste projeto foram adicionadas mais duas fontes: Hospital Beneficência Portuguesa e o Hospital das Clínicas FMUSP.

Todas as fontes de informação possuem estruturas semelhantes: um arquivo contendo informações dos pacientes como ID de identificação anonimizado, gênero, nascimento e endereço reduzido e outro material contendo o mesmo id ,para cruzamento com o primeiro, e informações sobre o exame como data da coleta, descrição do exame, local de coleta e o seu resultado. Excepcionalmente, os dados oriundos do Hospital Sírio Libanês e do Hospital Bene-

ficência Portuguesa possuem um arquivo adicional que informa qual foi o desfecho do paciente. A descrição completa do dicionário de dados pode ser visualizada no Apêndice A.

Inicialmente, a ideia era construir um modelo que unificasse todos os dados disponíveis, mas infelizmente a definição de exames mais específicos eram muito divergentes entre as instituições, tornando inviável o cruzamento de dados entre elas. Sabendo que o compartilhamento de dados é um tema sensível, principalmente na área da Saúde, o desenvolvimento foi segmentado por Instituição, de modo que possa ser replicado por qualquer outra que disponibilize suas informações, a fim de criar um modelo que seja ótimo para cada uma delas, aumentando a disponibilidade da solução.

4.2 Os dados

Como foi dito anteriormente, os dados são parte fundamental da modelagem e é necessário ter um entendimento completo do que está disponível antes de avançar qualquer etapa.

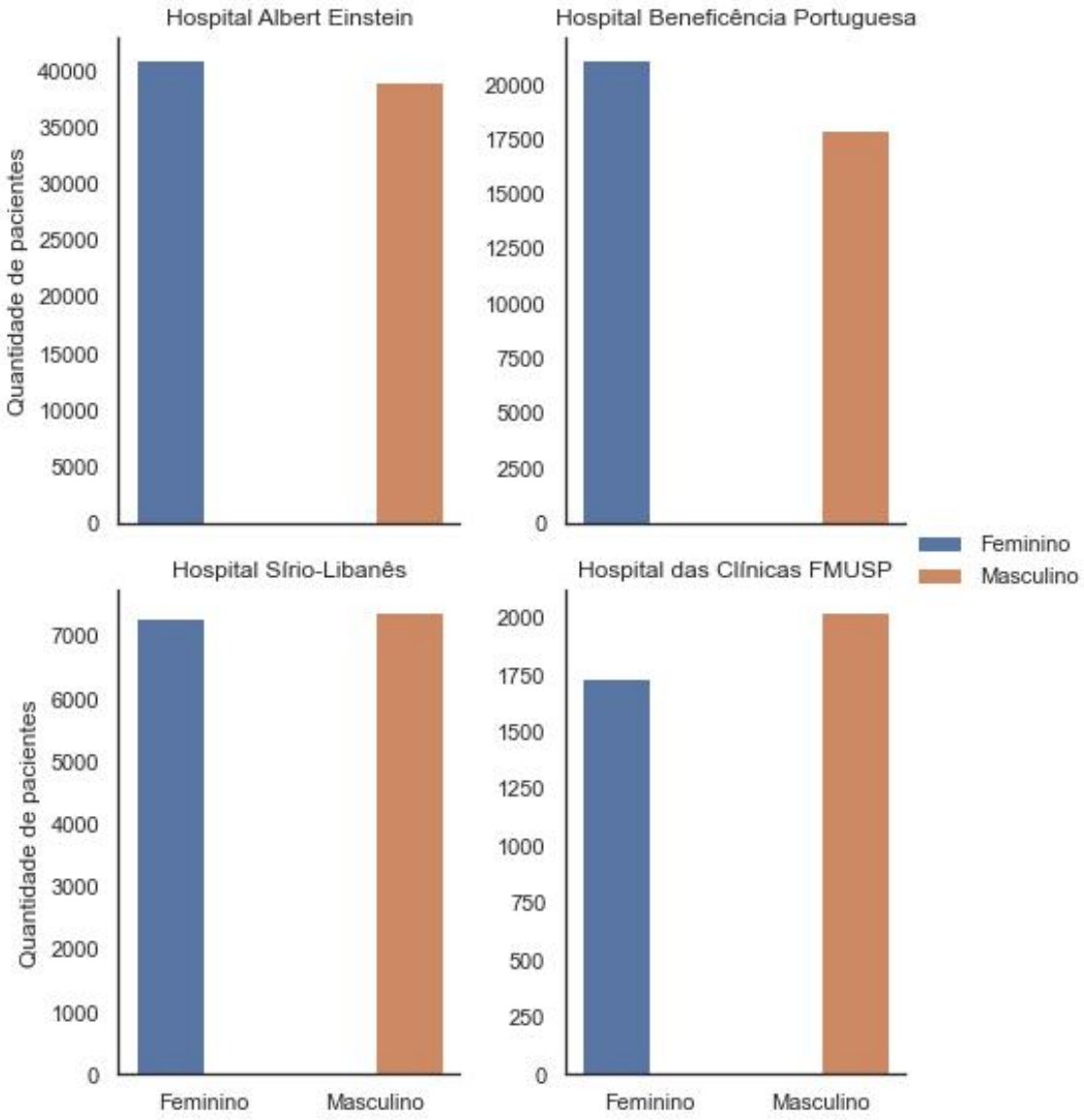
O primeiro passo foi definir quais conjuntos seriam utilizados para criar os classificadores e, como a proposta deste estudo de caso é analisar pacientes que deram entrada em unidades de Saúde Hospitalares, optou-se apenas por utilizar informações de origem hospitalar, descartando exames de origem laboratorial. Infelizmente, os dados do Grupo Fleury são apenas laboratoriais(de acordo com a variável DE_ORIGEM disponível em cada exame coletado) e por isso não foram utilizados.

Para entender o perfil dos pacientes que estão listados nas instituições remanescentes, é possível verificar se existe algum viés óbvio presente na própria coleta dos dados. O intuito dessa etapa é identificar algumas distribuições básicas dos indivíduos que estamos lidando.

Para isso, foi observado o gênero dos pacientes e como podemos observar na Figura 33, esse atributo está bem balanceado e também é possível verificar que o volume de pacientes, com dados coletados e disponibilizados para o repositório utilizado, é maior no Hospital Albert Einstein e menor no Hospital das Clínicas.

A exata quantidade de pacientes listados por instituição, pode ser visualizada na Tabela 1, o número de pacientes do Hospital das Clínicas, apenas 3751, pode ser um empecilho para a modelagem que será explorado nos próximos tópicos.

Figura 33 – Distribuição do gênero dos pacientes por Hospital



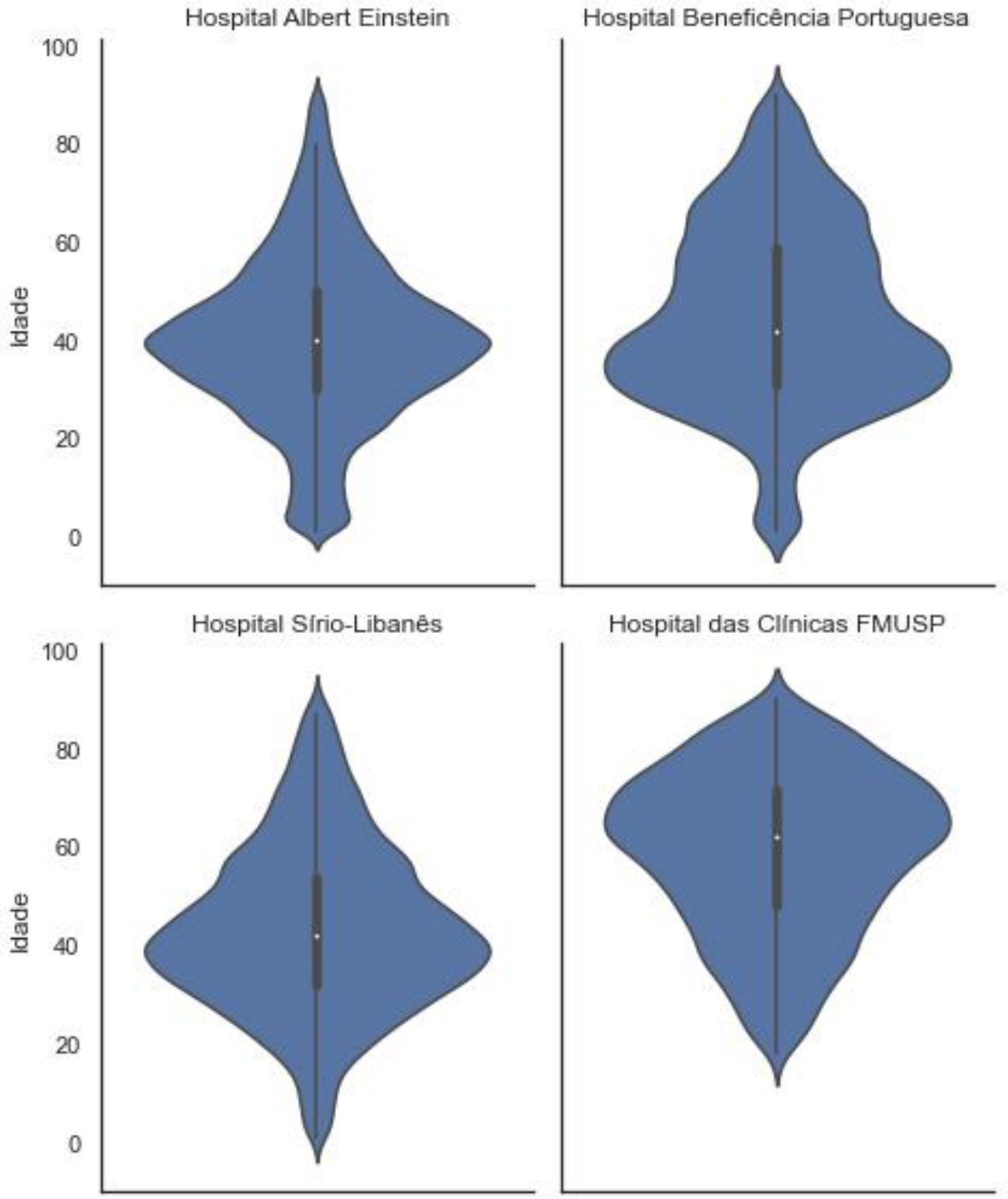
Fonte: Autor

Tabela 1 – Quantidade de pacientes por Hospital

Instituição	Número de Pacientes
Hospital Albert Einstein	79,863
Hospital Beneficência Portuguesa	39,000
Hospital Sírio-Libanês	14,673
Hospital das Clínicas FMUSP	3,751
Total:	**137,287**

Fonte: Autor

Figura 34 – Distribuição da Idade dos pacientes por Hospital - Gráfico de Violino



Fonte: Autor

Ainda em relação aos pacientes, foi observada a distribuição de idades(Figura 34) e o que pode-se concluir é que todos os hospitais possui uma distribuição semelhante, exceto o Hospital das Clínicas que lida majoritariamente com pacientes de idade mais avançada, o que pode ser confirmado por indicadores estatísticos básicos, como média e mediana, extraído das informações e sumarizados na Tabela 2.

Tabela 2 – Indicadores estatísticos básicos das idades dos pacientes por Hospital

Instituição	Média	Mediana	Desvio Padrão
Hospital Albert Einstein	40	40	17
Hospital Beneficência Portuguesa	44	42	19
Hospital Sírio-Libanês	43	42	17
Hospital das Clínicas FMUSP	59	62	16
Média Total:	**47**	**47**	**17**

Fonte: Autor

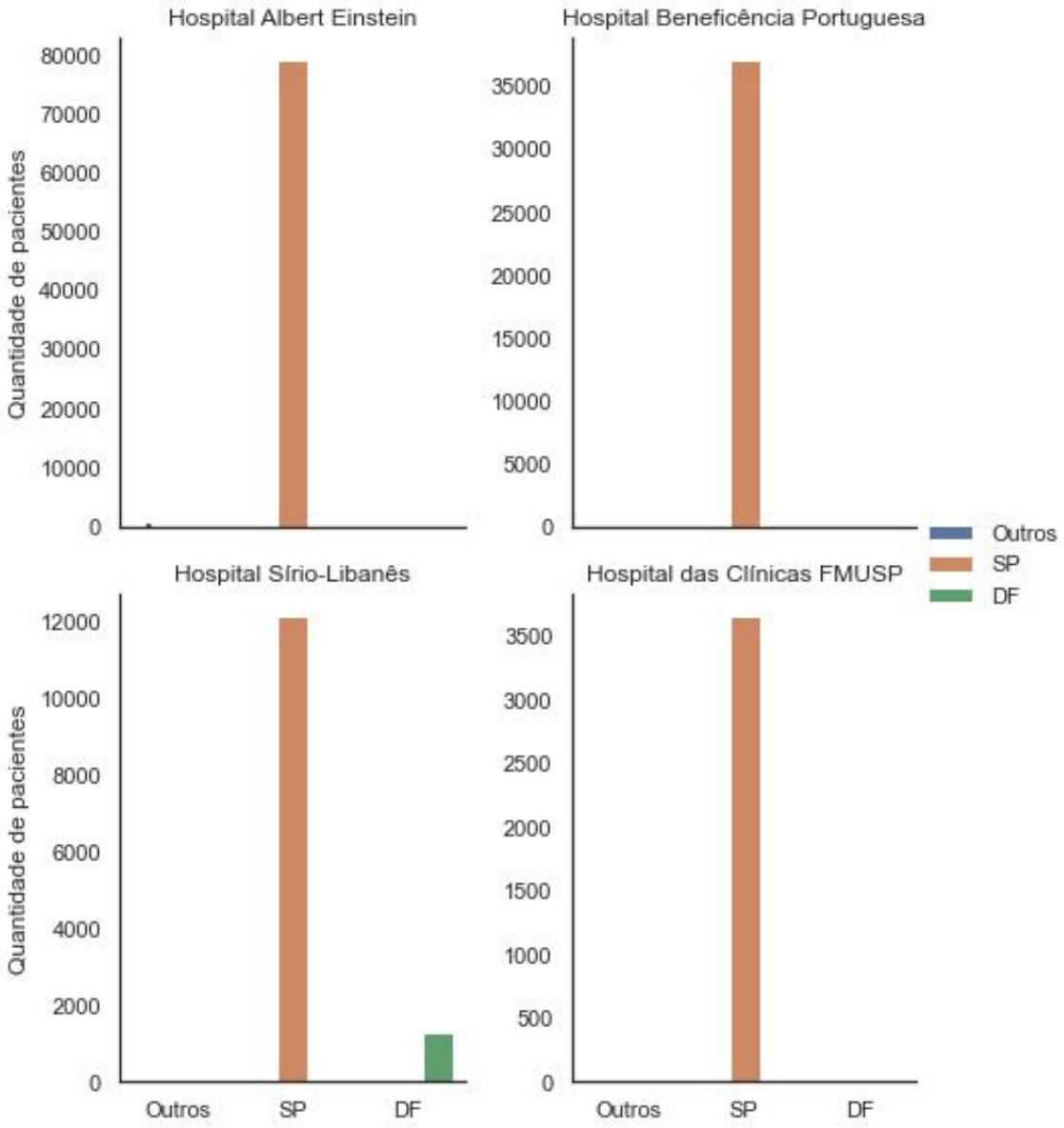
Outro fator observado foi o estado de residência dos pacientes, por se tratarem de Hospitais com sede em São Paulo era esperado que a maioria dos pacientes fossem residentes dessa região e, conforme a Figura 35, é o que realmente acontece no conjunto de dados utilizado. A quantidade de pacientes de outros estados é tão pequena que se torna imperceptível nas visualizações gráficas.

Saindo do universo dos pacientes, é importante entender a variável alvo do modelo, que nesse caso é o resultado do exame de detecção de COVID-19 por RT-PCR. O que esperava-se aqui eram três resultados: Detectado, Não-Detectado e Inconclusivo.

Quando estamos lidando com dados médicos referentes a anotações de doenças, é comum lidarmos com dados desbalanceados. Isso pode acontecer por diversos motivos: um deles seria a própria distribuição da doença na comunidade já possuir uma quantidade de positivos mais escassa, que é o caso deste trabalho, ou completamente comum entre os indivíduos, o mesmo racional pode se aplicar para os casos negativos.

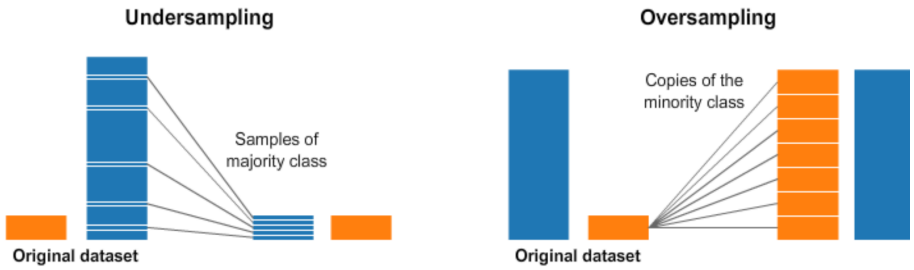
Infelizmente, modelos de *machine learning* utilizados para efetuar classificações binárias não costumam lidar bem com este fenômeno e podem acabar sendo condicionados a escolher, na maior parte das tentativas, a classe dominante. Para lidar com isso, as técnicas de re-amostragem foram utilizadas com o intuito de balancear o conjunto de dados. Lembrando que existem duas vertentes deste tipo de processamento: o *undersampling* e o *oversampling*, conforme a Figura 36.

Figura 35 – Estado de residência dos pacientes por Hospital



Fonte: Autor

Figura 36 – Demonstração visual do resultado das técnicas de balanceamento de classes: *undersampling* e *oversampling*



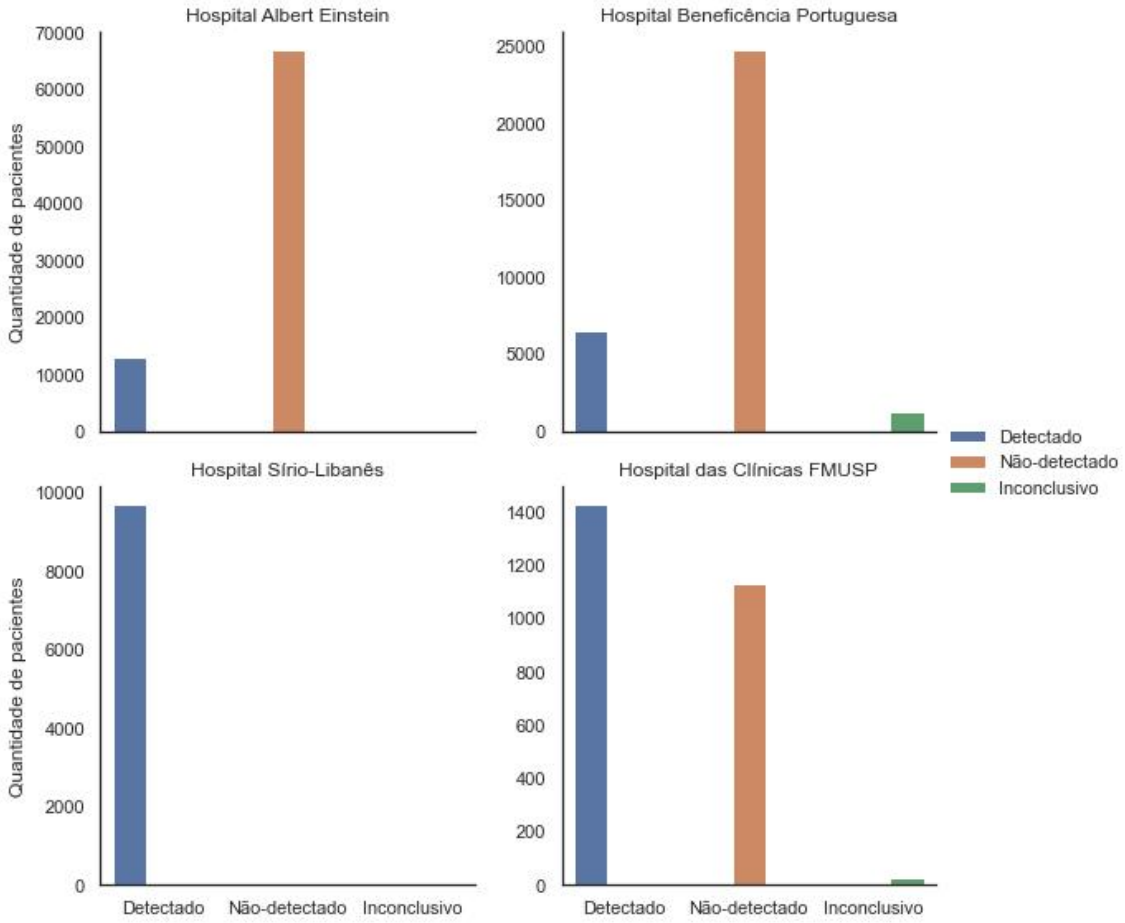
Fonte: (BADR, 2020)

A escolha entre essas duas vertentes é uma situação a ser considerada com muita cautela. Quando estamos falando em *undersampling* é possível que informações importantes sejam perdidas durante o processo, já no caso do *oversampling* as técnicas que criam novos exemplos sintéticos podem estar desconectadas da realidade e não possuir valores dentro de intervalos que sejam considerados possíveis. Tudo isto foi avaliado para garantir que a re-amostragem foi feita de forma eficaz e traga resultados satisfatórios para a modelagem do problema. Por isso precisamos verificar o balanceamento da variável.

Conforme a Figura 37, os dados do Hospital Hospital Sírio-Libanês possuem apenas exames com o resultado Detectado e, sabendo que para treinar um modelo de classificação é necessário que tenhamos todos resultados possíveis, eles foram descartados por possuir apenas uma categoria.

Nos outros hospitais as três categorias esperadas estão presentes e ainda existe uma maior quantidade de Não Detectados do que Detectados nos Hospitais Albert Einstein e Beneficência Portuguesa, sendo um comportamento esperado já que uma maior amostra irá representar uma distribuição mais próxima da realidade da doença. Vale lembrar também que por serem dados obtidos em hospitais, existe um viés de pacientes que já esperam que estejam com alguma enfermidade, possivelmente COVID-19, e isso leva a uma distribuição com uma maior porcentagem de infectados do que em relação a toda a população.

Figura 37 – Estado de residência dos pacientes por Hospital



Fonte: Autor

A partir das análises realizadas nessa seção já é possível ter um entendimento mais apurado dos dados disponíveis, descartando, de uma forma geral, os que não são adequados para este estudo de caso e selecionando os de mais valia.

4.3 Processamento dos dados e Modelagem

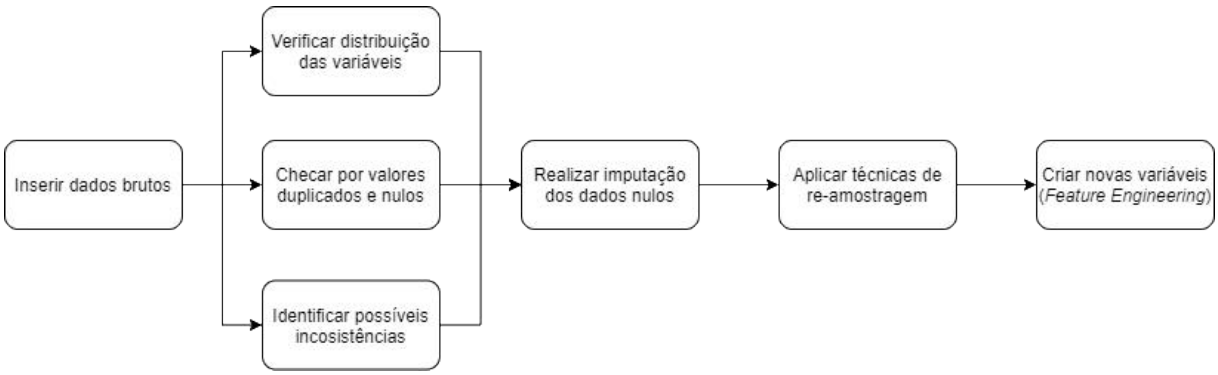
O processamento e a modelagem são processos que andam juntos, visto que a cada passo realizado no processamento podem existir ganhos de performance na modelagem. Essa linha é muito tênue e deve ser tratada com cuidado, pois cada tipo de processamento pode trazer uma melhora individual que talvez não reflita quando em conjunto com todas as outras técnicas aplicadas.

Ao lidar com dados brutos é comum lidarmos com alguns tipos de problemas que costumam ocorrer com qualquer fonte, como valores nulos, duplicados, inconsistentes e classes

desbalanceadas. Quando estamos falando de dados médicos é recorrente lidarmos com algumas destas situações de forma acentuada e para cada uma delas foram propostas técnicas que auxiliam na suavização dos danos que elas podem causar ao modelo de classificação a ser utilizado.

Desse modo, nessa etapa foram aplicadas 3 técnicas de processamento: observação de inconsistências, tratamento dos nulos e o FE. Lembrando que o principal objetivo desse passo é tornar o conjunto de informações consumível para os nossos modelos e, eventualmente, melhorar a sua performance. A metodologia utilizada pode ser verificada na Figura 38.

Figura 38 – Fluxograma Processamento



Fonte: Autor

4.3.1 Conjuntos de Validação

Outro ponto de fundamental importância é a escolha dos conjuntos de validação e como eles serão utilizados. A principal preocupação nesta etapa é garantir que o modelo treinado tenha uma boa capacidade de generalização, lembrando que previne-se os efeitos de *overfitting* e/ou *underfitting*, que são quando os dados se ajustam excessivamente ou muito pouco aos dados de treino, respectivamente.

De forma geral, a separação entre o conjunto de treino e teste é utilizada para treinar e testar o modelo, respectivamente. Para trazer mais confiabilidade, é possível adicionar mais uma camada de validação dentro dos próprios dados de treino, conforme ilustra a Figura 39.

A metodologia consiste em utilizar os dados de treino para ajustar os parâmetros dos modelos, otimizando os resultados com o alvo no conjunto de validação. Uma vez feita essa otimização, o modelo é exposto a dados nunca vistos(teste) a fim de comprovar que é capaz de

lidar com novas informações que nunca foram consideradas anteriormente para qualquer tipo de ajuste e/ou otimização.

Figura 39 – Divisões entre os conjuntos de treino, validação e teste

Fonte: Autor

Como os dados utilizados possuem marcações históricas, o método de separação escolhido para dividi-los nos conjuntos de treino, validação e teste foi a partir da data do exame. Essa escolha é dada com o intuito de simular um processo real, no qual os dados mais antigos são utilizados para prever comportamentos do futuro. O problema de utilizar uma separação randômica é que caso fossem utilizados dados do ”futuro” para fazer a previsão de fenômenos do ”passado”, os mais recentes poderiam ter informações que ainda não tinham sido percebidos e trazer melhorias injustas na performance do modelo, trazendo resultados que não iriam refletir a realidade em um caso de uso real.

Desta forma, eles foram separados, cronologicamente, em 60% para treino, 20% para validação e 20% para teste.

4.3.2 Dados faltantes e *Feature Engineering*

Uma situação que foi identificada de forma acentuada é a falta de dados em diversos exemplos do nosso banco de dados, para ilustrar este cenário é possível verificar a descrição das cinquenta variáveis que possuem maior taxa de preenchimento no Apêndice B.

Para resolver este problema, podemos substituir os dados nulos por variáveis constantes ou utilizar alguma função para realizar esta tarefa. Neste trabalho foram utilizadas quatro abordagens para lidar com os nulos: substituí-los pela média, mediana ou o dado mais frequente da coluna e a utilização do *KNN imputer* que calcula qual o dado existente mais próximo, através

do algoritmo KNN, que tenha semelhança com as colunas não nulas para realizar essa substituição (MAHBOOB et al., 2018).

O principal objetivo aqui foi testar e avaliar a performance dos modelos utilizando as diferentes técnicas escolhidas, entendendo o que faz mais sentido em cada situação. Vale ressaltar que elas foram avaliadas e otimizadas junto às diferentes técnicas de processamento, sempre buscando o conjunto de transformações que possua o melhor poder de generalização aliado a um bom valor da métrica escolhida para avaliação do resultado.

A última etapa realizada do processamento, na qual já possuíamos dados mais consistentes, foi a do *Feature Engineering*. Neste caso, são realizadas tentativas de combinar novas variáveis derivadas das originais e isso pode ser feito de diversas maneiras: operações matemáticas entre elas, criação de novas categorias para suprimir uma grande quantidade de variáveis, dentre outras.

Ainda em relação ao processamento, foi necessário definir quais pacientes deveriam ser utilizados no conjunto final de dados do modelo, já que existem alguns casos do registro ter apenas o resultado do exame de COVID-19, configurando uma situação desprezível para o treinamento dos algoritmos.

Para isso foi utilizada uma heurística na qual consiste em utilizar as variáveis que tivessem maior taxa de preenchimento, esse valores foram iterados no intervalo de 10 até 70 variáveis, respeitando os limites computacionais da máquina utilizada.

Ainda assim, foi necessário retornar ao preenchimento dos exames de cada paciente e para isso, iteramos entre pacientes que tivessem no mínimo 3 exames realizados, não incluindo o de detecção da COVID-19. Sabe-se que quanto maior o ponto de corte da quantidade de exames desses pacientes, a amostra seria reduzida gradualmente. Isto pode ser visto como um pacote de exames que seriam necessários para utilização desse recurso opcional por cada paciente.

Como cada tipo de exame pode possuir escala às vezes muito diferentes dos outros, os modelos podem considerar que alguns valores numericamente maiores sejam mais importantes. Para mitigar este tipo de problema, também foram testadas técnicas para suavizar esse efeito como a utilização de padronização, normalização ou do *MinMax Scaler* (AHSAN et al., 2021).

4.4 Modelos de classificação e métricas

Definido como seriam processados os dados a fim de torná-los consumíveis para os modelos de ML selecionados, se tornou possível treinar os classificadores, lembrando que estamos tratando de um problema de classificação por aprendizado supervisionado.

Deste modo, o foco inicial foi na utilização de modelos que já vem sendo utilizados em larga escala na medicina, como a Regressão Logística e a Árvore de Decisão. Do outro lado, também serão utilizadas abordagens mais modernas que costumam atingir boas performances em problemas de classificação binária, lembrando que a maioria deles são baseados em modelos de árvore como o *XGBoost* e a *RandomForest*.

Para realizar a otimização do conjunto de técnicas utilizadas e também dos hiper-parâmetros dos algoritmos foi utilizada a técnica de *Grid Search*, que consiste na criação e teste de uma malha contendo todas as combinações possíveis de certas variáveis. Para não tornar este processo infinito, são definidos limites superiores e inferiores para as variáveis contínuas e discretas (contendo um degrau entre cada passo), assim como a escolha das categorias disponibilizadas para as variáveis categóricas (LIASHCHYNSKYI; LIASHCHYNSKYI, 2019).

Por fim, é necessário definir a métrica que será otimizada nos processos de modelagem e ajuste dos hiper-parâmetros dos modelos. Como estamos falando de doenças infecciosas, o resultado mais indesejado seriam os Falsos Negativos, uma vez que estaríamos sinalizando aos indivíduos que estão com a infecção a retornarem ao convívio em sociedade. Por isso, o foco principal deste modelo é a otimização da Sensibilidade e também observando os valores da Especificidade, conforme as equações 17 e 18. De forma auxiliar, também foi observado a área sob a curva AUC, de modo a verificar a relação entre as duas métricas citadas anteriormente.

$$Sensibilidade = \frac{VerdadeiroPositivos}{VerdadeiroPositivos + FalsoNegativos} \tag{17}$$

$$Especificidade = \frac{VerdadeiroNegativos}{VerdadeiroNegativos + FalsoPositivos} \tag{18}$$

4.5 Explicabilidade dos modelos de inteligência artificial

Quando estamos falando de modelos mais complexos, principalmente que possuam originalmente alguma técnica de *Ensemble* se torna mais difícil avaliar porque ele está tomando cada decisão e essa explicação pode ser fundamental e até mesmo obrigatória para o consu-

midor daquela solução confiar e utilizar seus resultados . Devido a isso, a explicabilidade dos modelos utilizados pode ser tão importante quanto sua performance.

Em relação aos modelos que têm maior dificuldade em explicar as suas decisões, foram desenvolvidas algumas soluções que visam desvendar essa ”caixa preta” e explicar a importância de cada uma das variáveis ali utilizadas.

Neste caso, foram utilizadas as tecnologias que vêm sendo desenvolvidas para resolver este tipo de problema, dentre elas estão os *Shapley values*, oriundos da Teoria dos Jogos que podem auxiliar no entendimento destes modelos.

5 Resultados

Ao iterar em diversas combinações das metodologias supracitadas os modelos apresentaram diferentes resultados para cada instituição. Isso acontece pois cada informação tem uma fonte de coleta diferente, influenciando diretamente na qualidade dos dados, e também pela quantidade de amostras disponíveis para realizar o treinamento, validação e teste dos modelos.

As tabelas foram divididas de acordo com o número de exames considerados Número de exames considerados (NEC) e o número mínimo de exames por cada paciente Número mínimo de exames por cada paciente (NMEP), a fim de trazer diferentes perspectivas dessas variáveis e o seu impacto nos resultados finais, ressaltando que as técnicas de processamento foram utilizadas para otimizar os resultados em cada uma dessas combinações.

5.1 Hospital Israelita Albert Einstein

A partir das tabelas 3 a 7, percebe-se que para o Hospital Israelita Albert Einstein (HIAE) as melhores métricas alcançadas foram provenientes do modelo XGBoost, alcançando uma sensibilidade de 0.80 indica que a cada 100 pacientes testados e que de fato possuíam a doença, 80 deles terão o teste como Detectado.

Certamente o ideal seria todos serem classificados como positivos, mas devido a quantidade e qualidade das informações/dados utilizados nos modelos, este valor mostra que existe um grande potencial em otimizar ainda mais os modelos e técnicas disponíveis.

Na tabela 3 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 70 E NMEP = 15, com isso foram utilizados dados de 14.828 pacientes.

Tabela 3 – Resultados no conjunto de teste com NEC = 70 e NMEP = 15 para o HIAE

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.74	0.84	0.79	0.84
Random Forest	0.71	0.80	0.84	0.84
XGBoost	0.80	0.77	0.85	0.84
Support Vector Machine	0.68	0.71	0.73	0.79
Número de exames utilizados	**70**	**Total de pacientes pós filtros:**		**14,828**
Qtde mínima de exames por paciente	**15**			

Fonte: Autor

Na tabela 4 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 50 E NMEP = 10, com isso foram utilizados dados de 15.386 pacientes.

Tabela 4 – Resultados no conjunto de teste com NEC = 50 e NMEP = 10 para o HIAE

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.75	0.75	0.81	0.81
Random Forest	0.74	0.80	0.82	0.83
XGBoost	0.77	0.73	0.84	0.81
Support Vector Machine	0.72	0.69	0.77	0.77
Número de exames utilizados	**50**	**Total de pacientes pós filtros:**		**15,386**
Qtde miníma de exames por paciente	**10**			

Fonte: Autor

Na tabela 5 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 30 E NMEP = 7, com isso foram utilizados dados de 14.916 pacientes.

Tabela 5 – Resultados no conjunto de teste com NEC = 30 e NMEP = 7 para o HIAE

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.75	0.71	0.78	0.79
Random Forest	0.66	0.77	0.80	0.81
XGBoost	0.78	0.72	0.83	0.80
Support Vector Machine	0.71	0.67	0.75	0.70
Número de exames utilizados	**30**	**Total de pacientes pós filtros:**		**14,916**
Qtde miníma de exames por paciente	**7**			

Fonte: Autor

Na tabela 6 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 20 E NMEP = 5, com isso foram utilizados dados de 14.922 pacientes.

Tabela 6 – Resultados no conjunto de teste com NEC = 20 e NMEP = 5 para o HIAE

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.77	0.63	0.77	0.73
Random Forest	0.71	0.73	0.78	0.79
XGBoost	0.75	0.65	0.79	0.74
Support Vector Machine	0.71	0.68	0.76	0.72
Número de exames utilizados	**20**	**Total de pacientes pós filtros:**		**14,922**
Qtde miníma de exames por paciente	**5**			

Fonte: Autor

Na tabela 7 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 15 E NMEP = 3, com isso foram utilizados dados de 14.832 pacientes.

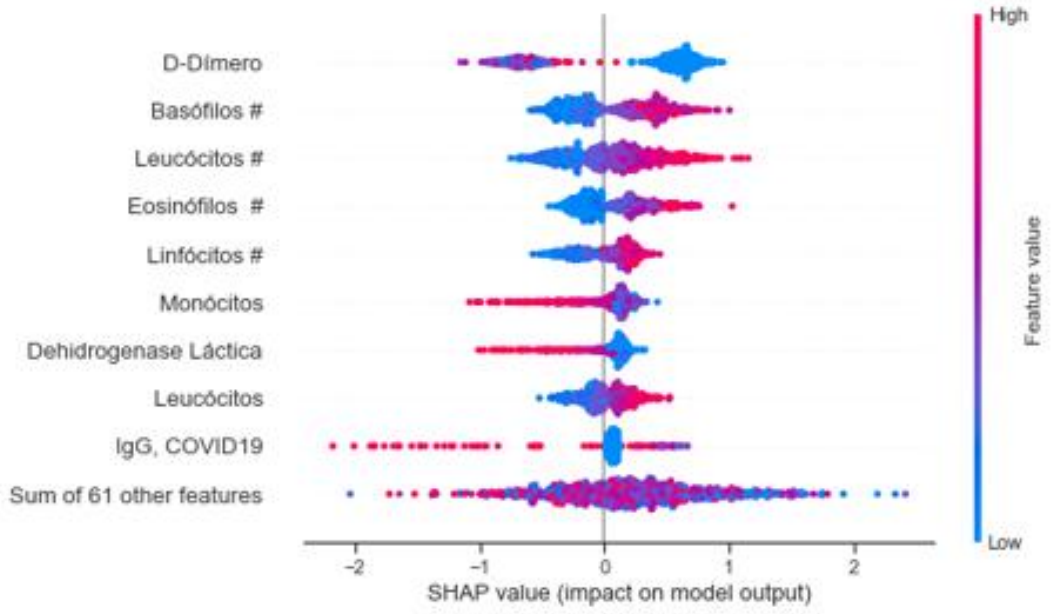
Tabela 7 – Resultados no conjunto de teste com NEC = 15 e NMEP = 3 para o HIAE

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.79	0.63	0.76	0.74
Random Forest	0.73	0.71	0.78	0.80
XGBoost	0.79	0.61	0.80	0.72
Support Vector Machine	0.72	0.70	0.75	0.76
Número de exames utilizados	**15**	**Total de pacientes pós filtros:**		**14,832**
Qtde miníma de exames por paciente	**3**			

Fonte: Autor

Por fim, os *Shapley values* adicionam uma camada de explicabilidade e trazem algumas respostas para o usuário. Podemos observar através da Figura 40 que valores baixos de D-Dímero e valores altos de Basófilos e Leucócitos tendem a convergir para um diagnóstico positivo da COVID-19. Esse tipo de informação é valiosa tanto para o entendimento do modelo como para validações conceituais junto aos especialistas da área.

Figura 40 – *Shapley values* para o modelo XGBoost com NEC = 70 e NMEP = 15 para o HIAE



5.2 Hospital Beneficência Portuguesa

A partir das tabelas 8 a 12, percebe-se que para o Hospital Beneficência Portuguesa (HBP) as melhores métricas alcançadas foram provenientes do modelo XGBoost, alcançando uma sensibilidade de 0,79.

Na tabela 8 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 70 E NMEP = 15, com isso foram utilizados dados de 16.417 pacientes.

Tabela 8 – Resultados no conjunto de teste com NEC = 70 e NMEP = 15 para o HBP

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.71	0.79	0.80	0.83
Random Forest	0.74	0.80	0.81	0.85
XGBoost	0.79	0.88	0.82	0.89
Support Vector Machine	0.68	0.69	0.71	0.75
Número de exames utilizados	**70**	**Total de pacientes pós filtros:**		**16,417**
Qtde mínima de exames por paciente	**15**			

Fonte: Autor

Na tabela 9 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 50 E NMEP = 10, com isso foram utilizados dados de 16.313 pacientes.

Tabela 9 – Resultados no conjunto de teste com NEC = 50 e NMEP = 10 para o HBP

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.71	0.79	0.80	0.83
Random Forest	0.72	0.77	0.81	0.84
XGBoost	0.77	0.85	0.82	0.88
Support Vector Machine	0.71	0.69	0.74	0.77
Número de exames utilizados	**50**	**Total de pacientes pós filtros:**		**16,313**
Qtde miníma de exames por paciente	**10**			

Fonte: Autor

Na tabela 10 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 30 E NMEP = 7, com isso foram utilizados dados de 16.265 pacientes.

Tabela 10 – Resultados no conjunto de teste com NEC = 30 e NMEP = 7 para o HBP

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.73	0.81	0.82	0.83
Random Forest	0.72	0.76	0.80	0.83
XGBoost	0.76	0.83	0.81	0.87
Support Vector Machine	0.67	0.68	0.70	0.70
Número de exames utilizados	**30**	**Total de pacientes pós filtros:**		**16,265**
Qtde miníma de exames por paciente	**7**			

Fonte: Autor

Na tabela 11 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 20 E NMEP = 5, com isso foram utilizados dados de 16.193 pacientes.

Tabela 11 – Resultados no conjunto de teste com NEC = 20 e NMEP = 5 para o HBP

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.71	0.80	0.81	0.82
Random Forest	0.71	0.75	0.78	0.82
XGBoost	0.75	0.82	0.79	0.84
Support Vector Machine	0.65	0.66	0.67	0.69
Número de exames utilizados	**20**	**Total de pacientes pós filtros:**		**16,193**
Qtde miníma de exames por paciente	**5**			

Fonte: Autor

Na tabela 12 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 15 E NMEP = 3, com isso foram utilizados dados de 16.271 pacientes.

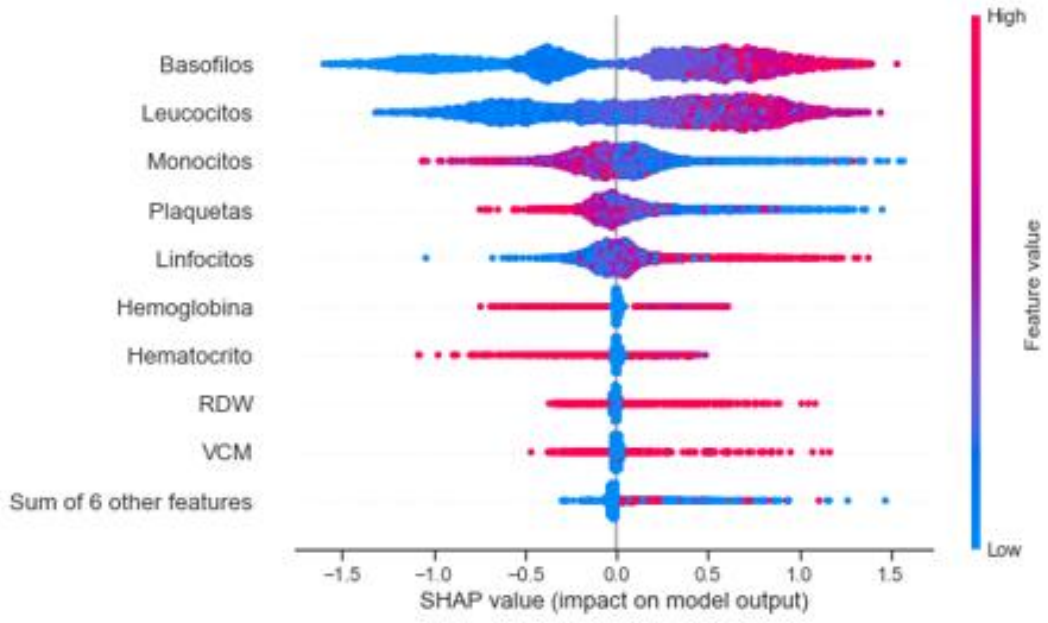
Tabela 12 – Resultados no conjunto de teste com NEC = 15 e NMEP = 3 para o HBP

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.71	0.79	0.80	0.82
Random Forest	0.68	0.75	0.77	0.79
XGBoost	0.73	0.80	0.79	0.83
Support Vector Machine	0.66	0.64	0.68	0.66
Número de exames utilizados	**15**	**Total de pacientes pós filtros:**		**16,271**
Qtde mínima de exames por paciente	**3**			

Fonte: Autor

Em relação aos *Shapley values*, podemos observar através da Figura 40 que valores altos de Basófilos e Leucócitos tendem a convergir para um diagnóstico positivo da COVID-19, assim como valores baixos de Monócitos e plaquetas.

Figura 41 – *Shapley values* para o modelo XGBoost com NEC = 70 e NMEP = 15 para o HBP



5.3 Hospital das Clínicas FMUSP

A partir das tabelas 13 a 17, percebe-se que para o Hospital das Clínicas FMUSP as melhores métricas alcançadas foram provenientes do modelo XGBoost, alcançando uma sensibilidade de 0,76.

Na tabela 13 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 70 E NMEP = 15, com isso foram utilizados dados de 2.749 pacientes.

Tabela 13 – Resultados no conjunto de teste com NEC = 70 e NMEP = 15 para o FMUSP

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.71	0.79	0.76	0.83
Random Forest	0.72	0.76	0.78	0.84
XGBoost	0.76	0.87	0.81	0.86
Support Vector Machine	0.65	0.66	0.70	0.72
Número de exames utilizados	**70**	**Total de pacientes pós filtros:**		**2,749**
Qtde miníma de exames por paciente	**15**			

Fonte: Autor

Na tabela 14 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 50 E NMEP = 10, com isso foram utilizados dados de 2.750 pacientes.

Tabela 14 – Resultados no conjunto de teste com NEC = 50 e NMEP = 10 para o FMUSP

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.68	0.76	0.78	0.81
Random Forest	0.69	0.75	0.78	0.80
XGBoost	0.73	0.83	0.80	0.86
Support Vector Machine	0.70	0.68	0.71	0.73
Número de exames utilizados	**50**	**Total de pacientes pós filtros:**		**2,750**
Qtde miníma de exames por paciente	**10**			

Fonte: Autor

Na tabela 15 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 30 E NMEP = 7, com isso foram utilizados dados de 2.747 pacientes.

Tabela 15 – Resultados no conjunto de teste com NEC = 30 e NMEP = 7 para o FMUSP

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.69	0.80	0.80	0.82
Random Forest	0.71	0.75	0.77	0.79
XGBoost	0.75	0.82	0.78	0.84
Support Vector Machine	0.64	0.66	0.68	0.66
Número de exames utilizados	**30**	**Total de pacientes pós filtros:**		**2,747**
Qtde miníma de exames por paciente	**7**			

Fonte: Autor

Na tabela 16 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 20 E NMEP = 5, com isso foram utilizados dados de 2.737 pacientes.

Tabela 16 – Resultados no conjunto de teste com NEC = 20 e NMEP = 5 para o FMUSP

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.74	0.60	0.73	0.72
Random Forest	0.70	0.69	0.77	0.78
XGBoost	0.74	0.63	0.78	0.72
Support Vector Machine	0.70	0.64	0.72	0.68
Número de exames utilizados	**20**	**Total de pacientes pós filtros:**		**2,737**
Qtde miníma de exames por paciente	**5**			

Fonte: Autor

Na tabela 17 estão demonstrados os resultados para os modelos propostos, utilizando NEC = 15 E NMEP = 3, com isso foram utilizados dados de 2.737 pacientes.

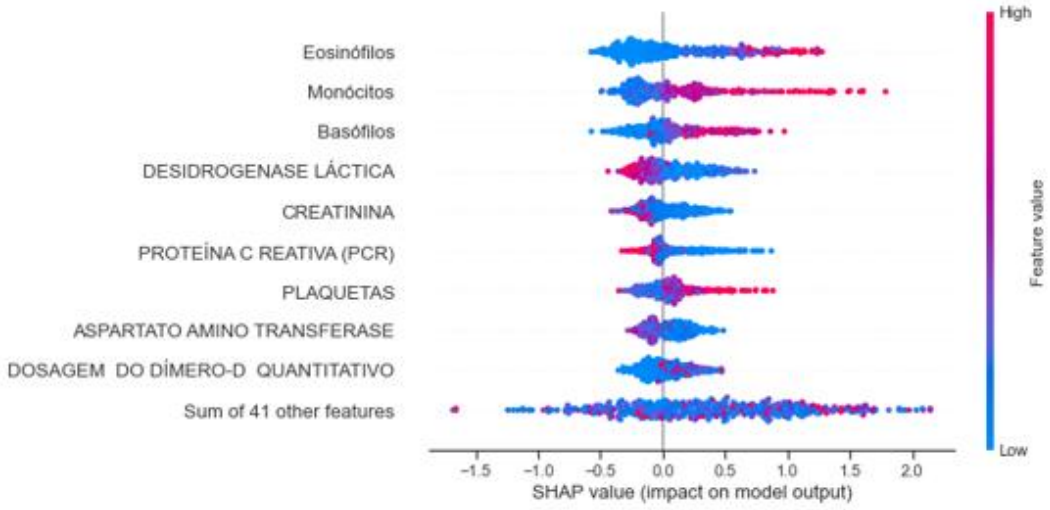
Tabela 17 – Resultados no conjunto de teste com NEC = 15 e NMEP = 3 para o FMUSP

Algoritmo	Sensibilidade	Especificidade	AUC	F1-Score
Regressão Logística	0.75	0.60	0.73	0.70
Random Forest	0.71	0.67	0.77	0.76
XGBoost	0.74	0.60	0.77	0.71
Support Vector Machine	0.71	0.67	0.74	0.73
Número de exames utilizados	**15**	**Total de pacientes pós filtros:**		**2,737**
Qtde miníma de exames por paciente	**3**			

Fonte: Autor

Em relação aos *Shapley values*, podemos observar através da Figura 42 que valores altos de Eosinófilos, Monócitos e Basófilos tendem a convergir para um diagnóstico positivo da COVID-19.

Figura 42 – *Shapley values* para o modelo XGBoost com NEC = 70 e NMEP = 15 para o FMUSP



6 Conclusão

O contexto da pandemia da COVID-19 e a evolução da ciência durante esse período foram as principais motivações para a pesquisa realizada, já que foi possível identificar uma possibilidade clara de aplicação dos conhecimentos de ML e IA na área da saúde. Com o conhecimento obtido foi possível criar classificadores que podem ser aplicados a diversas situações que se tratam de doenças infecciosas e ajudar no combate a elas.

Ao realizar a modelagem estatística percebeu-se que com quantidades limitadas de exames, principalmente aos relacionados ao hemograma completo, já foi possível chegar a métricas animadoras tanto de sensibilidade quanto de especificidade. Tudo isto mostra que essa é uma área promissora e que tem potencial para ajudar no combate em situações críticas causadas pela alta dispersão de doenças infecciosas.

Os resultados obtidos no capítulo 5 mostram que existem pontos de melhoria nas soluções propostas mas que, de forma emergencial, o autor acredita que elas possuem mais ganhos do que pontos negativos no auxílio ao combate à doenças infecciosas.

Ao observar os dados dos resultados avaliados no conjunto de teste, observa-se que foi alcançada uma sensibilidade máxima de 0,80, 0,79 e 0,76 para os Hospitais Israelita Albert Einstein, Beneficência Portuguesa e das Clínicas FMUSP, respectivamente. Isso significa que até 80% das pessoas que foram submetidas a esses modelos e estavam, de fato, infectadas foram diagnosticadas como positivas. Infelizmente, essa métrica é inferior a testes físicos que existem no mercado como o RT-PCR e o RDT, mas vale ressaltar que a proposta aqui não é substituir esses exames e principalmente auxiliar na detecção das doenças na falta deles.

Outro ponto importante a ser observado é a qualidade e quantidade dos dados que afetam diretamente no resultado final dos modelos, já que o dado é a principal matéria prima de qualquer algoritmo de ML. No Hospital Israelita Albert Einstein, por exemplo, após realizar os filtros de NEC e NMEP o número de pacientes que teriam exames elegíveis para serem utilizados no modelo matemático caiu em até 75% em relação ao conjunto de dados original.

Em termos da quantidade de dados utilizadas, temos perdas significativas que podem ocorrer principalmente por dois fatores: o primeiro é que existem diversos pacientes que possuem poucos exames e isso pode ser consequência de alguns fatores, como a falta da coleta digital dessa informação ou de realmente ele apenas possuir um número limitado de exames, acredito que a segunda é menos provável, mas não pode ser confirmado com os dados utilizados, já que nos outros Hospitais Analisados os dados remanescentes após os filtros são de cerca

de 50% dos originais. Aqui existe uma oportunidade de entender com os responsáveis pelas informações como é feita a coleta e os principais problemas que podem ser mitigados nela.

Outro ponto importante está na qualidade dos dados. Um exemplo disto seria o próprio resultado do exame da COVID-19, para cada Hospital o resultado ”Positivo” tinha até 3 grafias diferentes dentro do mesmo conjunto de teste. Esse caso traz uma insegurança para o usuário pois pode levar a questionamentos de exames que tenham os resultados numéricos e trazer a discussão se eles são confiáveis ou não. Lembrando que para este estudo eles foram considerados confiáveis.

Como o grande insumo dos modelos são os dados, sugerir melhorias no seu processo de coleta é um ponto muito importante e para a área de saúde não poderia ser diferente. Acredito que com a evolução dos prontuários eletrônicos e dos sistemas de informação internos dos hospitais isso venha a evoluir bastante e, consequentemente, auxiliará no desenvolvimento do tipo de solução que foi proposto. A iniciativa da (FAPESP, 2020) já mostra uma maior preocupação da comunidade em servir os dados e é um claro exemplo que estamos preocupados com a evolução nos pontos citados.

Tecnicamente, as técnicas de modelagem e processamento foram exauridas ao máximo neste trabalho. Disto isso, é importante reforçar que os principais ganhos nesses modelos irão surgir de dados mais robustos ou técnicas mais eficientes que serão desenvolvidas no futuro. Apesar de bons resultados de sensibilidade, lembrando que os algoritmos foram otimizados para isso, a especificidade também apresentou bons números mas que na maioria das vezes menores do que o anterior.

Outra questão importante a ser considerada leva em conta a questão do relacionamento entre a base de dados de diferentes instituições. O ponto negativo é que diversos exames não têm os nomes padronizados, o que dificulta imensamente a relação entre eles e acaba dificultando a realização de um sistema unificado para todos hospitais, por exemplo.

O uso dos *Shapley values* trouxeram reflexões importantes, como a presença de altos valores de Basófilos e Leucócitos estão associados com o diagnóstico positivo. Essas informações podem ser mais destrinchadas e levadas a discussões com especialistas da área para aumentar o entendimento e a popularidade dos modelos de ML dentro da comunidade médica.

Pensando em expandir um sistema dessa natureza em larga escala é necessário ter bastante cuidado em como transferir a informação de um local a outro, quando necessário. Imagine que em um lugar mais distante a coleta digital de dados seja inexistente ou muito pequena e seja impossível treinar o modelo com dados daquele local, nesse caso as características da população

devem ser bastante investigadas para trazer dados ”externos” que façam ”sentido”em relação a aquele conjunto de pessoas e não prejudiquem extremamente as métricas de sucesso dos exames.

Por fim, levando em consideração o objetivo proposto (que é de não reinserir pacientes com a doença infecciosa na sociedade ou até controlar a quantidade de infecção nos fluxos hospitalares) as técnicas têm potencial para auxiliar nas tomadas de decisões médicas principalmente em lugares menos privilegiados em relação a exames.

6.1 Trabalhos Futuros

A partir deste trabalho, percebeu-se a eficácia dos métodos propostos e os pontos de melhoria que podem ser endereçados a elas. Deste modo, surge a oportunidade de futuros trabalhos intimamente relacionados à qualidade dos dados, como novos métodos de coleta e a demonstração da importância desta etapa para os profissionais que a realizam diariamente.

Além da oportunidade existente na inspeção da coleta dos dados, também existe uma necessidade de integração entre os modelos de diferentes entidades médicas. Para isto, é necessário trabalhar com a padronização na nomenclatura dos exames utilizados ou até criar modelos de processamento de linguagem natural para realizar a conexão entre exames de diferentes locais. Dessa forma, será possível criar sistemas escaláveis e que possam alcançar o máximo de instituições possíveis, desde as que tenham grande quantidade de recursos até as que não o tenham.

Por último, é necessário realizar testes mais específicos em relação às novas ferramentas de explicabilidade que surgem ou poderão surgir nos próximos anos, levando em conta que a interpretação dos modelos é de suma importância para gerar mais confiabilidade em relação a eles.

REFERÊNCIAS

AHSAN, Md Manjurul et al. Effect of data scaling methods on machine learning algorithms and model performance. **Technologies**, Multidisciplinary Digital Publishing Institute, v. 9, n. 3, p. 52, 2021.

BADR, Will. **Having an Imbalanced Dataset? Here Is How You Can Fix It.** [S.l.: s.n.], 2020. https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb.

BREIMAN, Leo. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.

BREIMAN, Leo et al. **Classification and regression trees**. [S.l.]: CRC press, 1984.

BROWNLEE, Jason. **Failure of Classification Accuracy for Imbalanced Class Distributions**. [S.l.: s.n.], 2020. ://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions//. Accessed: 2020-12-17.

CABITZA, Federico; RASOINI, Raffaele; GENSINI, Gian Franco. Unintended consequences of machine learning in medicine. **Jama**, American Medical Association, v. 318, n. 6, p. 517–518, 2017.

CHRISTODOULOU, Evangelia et al. A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. **Journal of Clinical Epidemiology**, v. 110, p. 12–22, 2019. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0895435618310813>.

DEO, Rahul C. Machine Learning in Medicine. **Circulation**, v. 132, n. 20, p. 1920–1930, 2015. eprint: https://www.ahajournals.org/doi/pdf/10.1161/CIRCULATIONAHA.115.001593. Disponível em: <https://www.ahajournals.org/doi/abs/10.1161/CIRCULATIONAHA.115.001593>.

FAPESP. **FAPESP COVID-19 Data Sharing/BR**. [S.l.: s.n.], 2020. https://repositoriodatasharingfapesp.uspdigital.usp.br.

FRIEDMAN, Jerome; HASTIE, Trevor; TIBSHIRANI, Robert et al. **The elements of statistical learning**. [S.l.]: Springer series in statistics New York, 2001. v. 1.

JAMES, Gareth et al. **An Introduction to Statistical Learning: with Applications in R**. [S.l.]: Springer, 2013. Disponível em: <https://faculty.marshall.usc.edu/gareth-james/ISL/>.

KILIC, Tugba; WEISSLEDER, Ralph; LEE, Hakho. Molecular and Immunological Diagnostic Tests of COVID-19: Current Status and Challenges. **iScience**, v. 23, n. 8, p. 101406, 2020. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2589004220305964>.

LIASHCHYNSKYI, Petro; LIASHCHYNSKYI, Pavlo. Grid search, random search, genetic algorithm: A big comparison for NAS. **arXiv preprint arXiv:1912.06059**, 2019.

LIU, Xu-Ying; WU, Jianxin; ZHOU, Zhi-Hua. Exploratory undersampling for class-imbalance learning. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, IEEE, v. 39, n. 2, p. 539–550, 2008.

LUNDBERG, Scott. **GitHub - slundberg/shap: A game theoretic approach to explain the output of any machine learning model.** [S.l.: s.n.], 2020. https://github.com/slundberg/shap.

LUNDBERG, Scott M; LEE, Su-In. A Unified Approach to Interpreting Model Predictions. In: GUYON, I. et al. (Ed.). **Advances in Neural Information Processing Systems 30**. [S.l.]: Curran Associates, Inc., 2017. P. 4765–4774. Disponível em: <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>.

MAHBOOB, Tahira et al. Handling missing values in chronic kidney disease datasets using KNN, K-means and K-medoids algorithms. In: IEEE. 2018 12th International Conference on Open Source Systems and Technologies (ICOSST). [S.l.: s.n.], 2018. P. 76–81.

MENG, Zirui et al. Development and utilization of an intelligent application for aiding COVID-19 diagnosis. **medRxiv**, Cold Spring Harbor Laboratory Press, 2020.

MORAES BATISTA, André Filipe de et al. COVID-19 diagnosis prediction in emergency care patients: a machine learning approach. **medRxiv**, Cold Spring Harbor Laboratory Press, 2020. eprint: https://www.medrxiv.org/content/early/2020/04/14/2020.04.04.20052092.full.pdf. Disponível em: <https://www.medrxiv.org/content/early/2020/04/14/2020.04.04.20052092>.

QUINLAN, J Ross et al. Bagging, boosting, and C4. 5. In: AAAI/IAAI, Vol. 1. [S.l.: s.n.], 1996. P. 725–730.

QUINLAN, J Ross; RIVEST, Ronald L. Inferring decision trees using the minimum description lenght principle. **Information and computation**, Citeseer, v. 80, n. 3, p. 227–248, 1989.

QUINLAN, J. R. Decision trees and decision-making. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 20, n. 2, p. 339–346, 1990.

QUINLAN, J. Ross. Simplifying decision trees. **International journal of man-machine studies**, Elsevier, v. 27, n. 3, p. 221–234, 1987.

RAJKOMAR, Alvin; DEAN, Jeffrey; KOHANE, Isaac. Machine Learning in Medicine. **New England Journal of Medicine**, v. 380, n. 14, p. 1347–1358, 2019. PMID: 30943338. eprint: https://www.nejm.org/doi/pdf/10.1056/NEJMra1814259. Disponível em: <https://www.nejm.org/doi/full/10.1056/NEJMra1814259>.

SHELKE, Mayuri S; DESHMUKH, Prashant R; SHANDILYA, Vijaya K. A review on imbalanced data handling using undersampling and oversampling technique. **International Journal of Recent Trends in Engineering and Research**, v. 3, n. 4, p. 444–449, 2017.

SIDEY-GIBBONS, Jenni AM; SIDEY-GIBBONS, Chris J. Machine learning in medicine: a practical introduction. **BMC medical research methodology**, BioMed Central, v. 19, n. 1, p. 1–18, 2019.

SINGHAL, Gaurav. **Ensemble Methods in Machine Learning: Bagging Versus Boosting**. [S.l.: s.n.], 2020. https://https://www.pluralsight.com/guides/ensemble-methods:-bagging-versus-boosting/. Accessed: 2020-12-15.

THORNTON, Jacqui. Covid-19: Delays in getting tests are keeping doctors from work, health leaders warn. **BMJ**, BMJ Publishing Group Ltd, v. 370, 2020. eprint: https://www.bmj.com/content/370/bmj.m3755.full.pdf. Disponível em: <https://www.bmj.com/content/370/bmj.m3755>.

VAYENA, Effy; BLASIMME, Alessandro; COHEN, I Glenn. Machine learning in medicine: addressing ethical challenges. **PLoS medicine**, Public Library of Science San Francisco, CA USA, v. 15, n. 11, e1002689, 2018.

WYNANTS, Laure et al. Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal. **bmj**, British Medical Journal Publishing Group, v. 369, 2020.

YAN, Li et al. Prediction of criticality in patients with severe Covid-19 infection using three clinical features: a machine learning-based prognostic model with clinical data in Wuhan. **medRxiv**, Cold Spring Harbor Laboratory Press, 2020. eprint: https://www.medrxiv.org/content/early/2020/03/03/2020.02.27.20028027.full.pdf. Disponível em: <https://www.medrxiv.org/content/early/2020/03/03/2020.02.27.20028027>.

APÊNDICE A – DICIONÁRIO DE DADOS COVID *DATA SHARING* FAPESP

Figura 43 – Dicionário de dados - Paciente

NOME_VARIAVEL	DESCRICAO	FORMATO	CONTEUDO
ID_PACIENTE	Identificação única do paciente (correlaciona com o ID_PACIENTE do arquivo de RESULTADOS DE EXAMES)	Caracteres alfanuméricos	String, chave paciente
IC_SEXO	Sexo do Paciente	1 caracter alfanumérico	F - Feminino M - Masculino
AA_NASCIMENTO	Ano de nascimento do Paciente	4 numéricos (*)	Os 4 dígitos do ano do nascimento, ou AAAA - para ano de nascimento igual ou anterior a 1930 (visando anonimização) YYYY - qualsquer outros anos, em caso de anonimização do ano
CD_PAIS	País de residencia do Paciente	Alfanumérico	BR ou XX (país estrangeiro)
CD_UF	Unidade da Federacao de residencia do Paciente	2 caracteres alfanumérico	AC - Acre, AL - Alagoas, AM - Amazonas, AP - Amapa, BA - Bahia, CE - Ceará, DF - Distrito Federal, ES - Espirito Santo, GO - Goiás, MA - Maranhão, MG - Minas Gerais, MS - Mato Grosso do Sul, MT - Mato Grosso, PA - Pará, PB - Paraíba, PE - Pernambuco, PI - Piauí, PR - Paraná, RJ - Rio de Janeiro, RN - Rio Grande do Norte, RO - Rondônia, RR - Roraima, RS - Rio Grande do Sul, SC - Santa Catarina, SE - Sergipe, SP - São Paulo, TO - Tocantins
CD_MUNICIPIO	Municipio de residencia do Paciente	Alfanumérico	Nome do município por extenso, ou MMMM - quando houver necessidade de anonimização ou estrangeiro
CD_CEPREDUZIDO	CEP da residencia do Paciente	5 numéricos (**)	Os primeiros cinco dígitos do CEP (Código de Endereçamento Postal Brasileiro) CCCC - quando houver necessidade de anonimização ou estrangeiro

Fonte: Adaptado de (FAPESP, 2020)

Figura 44 – Dicionário de dados - Exames

NOME_VARIAVEL	DESCRICAO	FORMATO	CONTEUDO	OBSERVAÇÕES
ID_PACIENTE	Identificação única do paciente (correlaciona com o ID_PACIENTE do arquivo de PACIENTES)	Caracteres alfanuméricos	String, chave paciente	
DT_COLETA	Data em que o material foi coletado do paciente	Data (DD/MM/AAAA)	DD = Dia / MM = Mês / AAAA = Ano Exemplo: 24/06/2020	
DE_ORIGEM	Origem do Paciente	4 caracteres alfanuméricos	LAB – Exame realizado por paciente em uma unidade de atendimento laboratorial HOSP – Exame realizado por paciente dentro de uma Unidade Hospitalar	
DE_EXAME	Descrição do exame realizado	Alfanumérico	String Exemplo: HEMOGRAMA / SODIO / POTASSIO	Um exame é composto por 1 ou mais analitos
DE_ANALITO	Descrição do analito	Alfanumérico	String Exemplo: Eritrócitos / Leucócitos / Glicose / Ureia / Creatinina	Para o exame Hemograma, temos o resultado de vários elementos (analitos): Eritrócitos, Hemoglobina, Leucócitos, Linfócitos, etc. A maioria dos exames tem somente 1 analito, como exemplo a Glicose, Colesterol Total, Uréia e Creatinina
DE_RESULTADO	Resultado do exame, associado ao DE_ANALITO	Alfanumérico	Se DE_ANALITO exige valor numérico, Inteiro ou Decimal Se DE_ANALITO exige qualitativo, String com domínio restrito	Exemplo de domínio restrito - Positivo, Detectado, Reagente, não reagente, etc.
CD_UNIDADE	Unidade de Medida utilizada na Metodologia do Grupo Fleury para analisar o exame	Alfanumérico	String Exemplo: g/dL (gramas por decilitro)	
DE_VALOR_REFERENCIA	Valores de referência para DE_RESULTADO	Alfanumérico	String - Resultado ou faixa de resultados em que é considerado normal para este analito, na população 'Valor Mínimo' a 'Valor Máximo' Não Detectado/Detectado Exemplo para Glicose: 75 a 99 Exemplo para Progesterona: Até 89	

Fonte: Adaptado de (FAPESP, 2020)

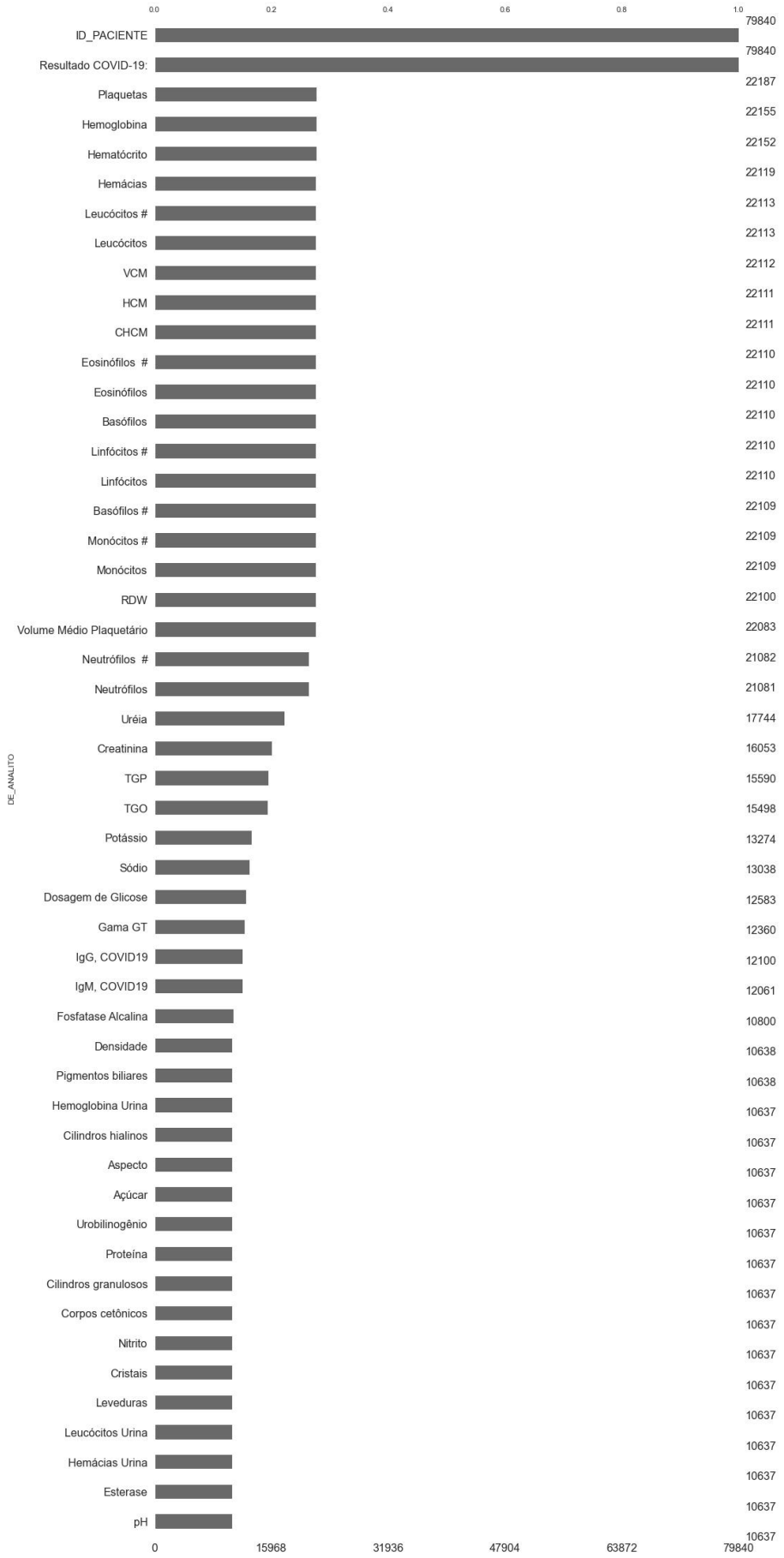
Figura 45 – Dicionário de dados - Desfecho

NOME_VARIAVEL	DESCRICAO	FORMATO	CONTEUDO
ID_PACIENTE	Identificação única do paciente (correlaciona com o ID_PACIENTE de todos os arquivos onde aparece (por exemplo, EXAMES e DESFECHOS)	32 caracteres alfanuméricos	String, anonimizado
ID_ATENDIMENTO	Identificação única do atendimento. Cada atendimento tem um desfecho.	Alfanumérico	String, anonimizado
DT_ATENDIMENTO	Data de realização do atendimento	Data (DD/MM/AAAA)	DD = Dia / MM = Mês / AAAA = Ano
DE_TIPO_ATENDIMENTO	Descrição do tipo de atendimento realizado.	Texto livre	String Exemplo: Pronto atendimento.
ID_CLINICA	Identificação da clínica onde o evento aconteceu.	Numérico	Exemplo: 1013
DE_CLINICA	Descrição da clínica onde o evento aconteceu.	Texto livre	Exemplo: Retorno Digital Adulto
DT_DESFECHO	Data do desfecho	Data (DD/MM/YYYY)	DD = Dia / MM = Mês / AAAA = Ano Exemplo: 24/06/2020
DE_DESFECHO	Descriçao do desfecho.	Texto livre	Exemplo: Alta médica melhorado

Fonte: Adaptado a partir de (FAPESP, 2020)

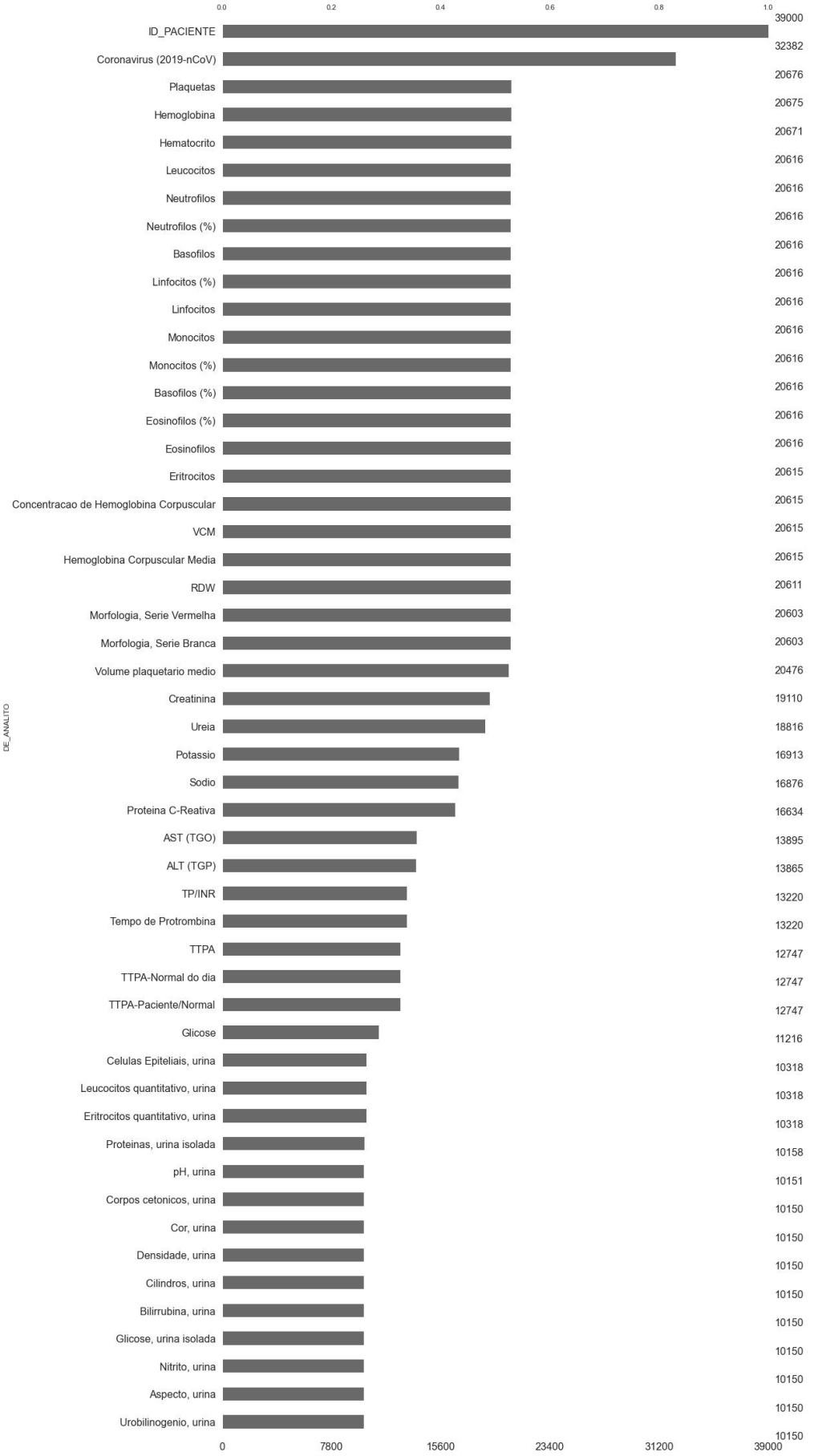
APÊNDICE B – EXPLORAÇÃO DOS DADOS FALTANTES

Figura 46 – Dados faltantes Hospital Albert Einstein (50 variáveis mais presentes)



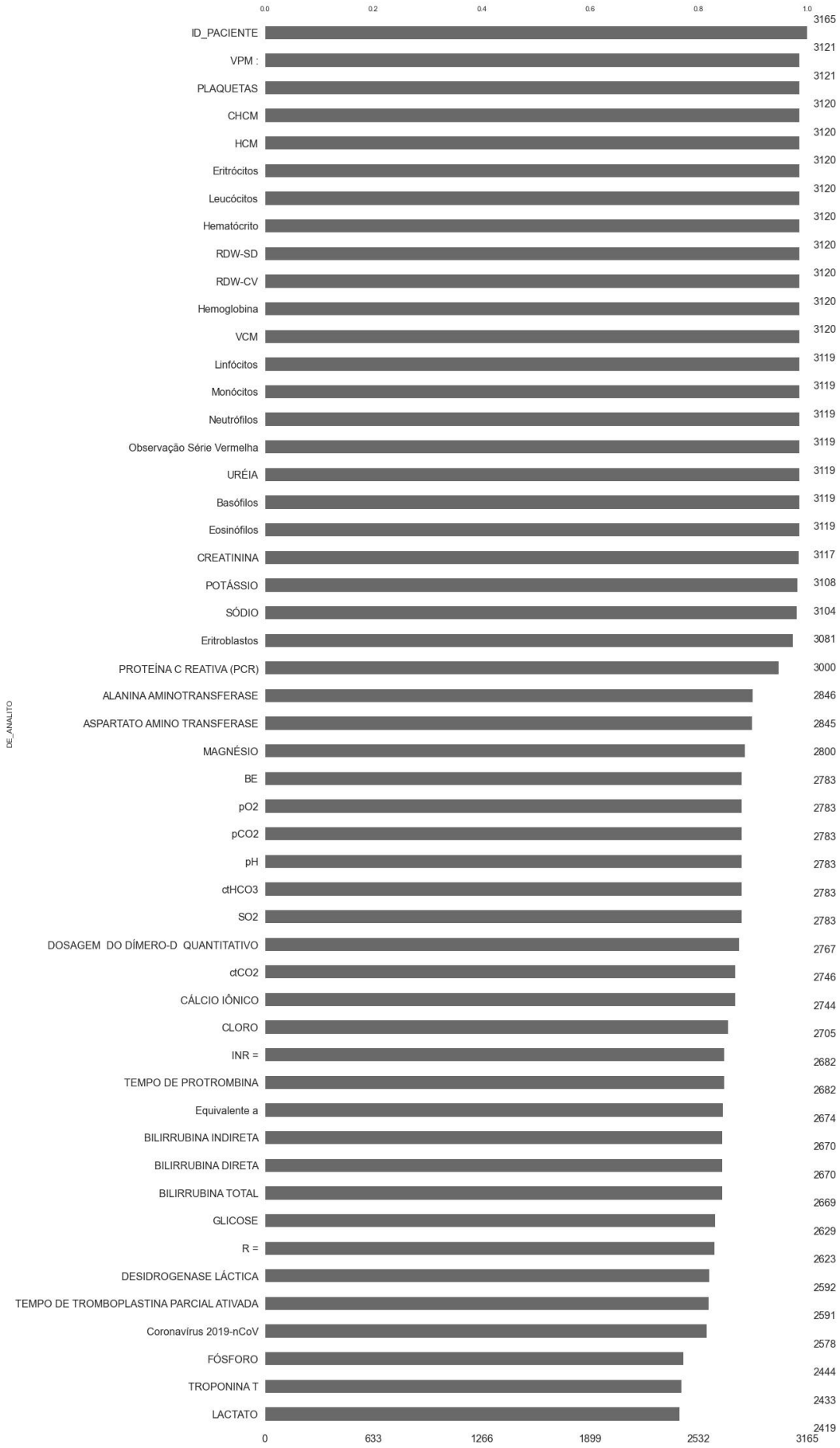
Fonte: Autor

Figura 47 – Dados faltantes Hospital Beneficência Portuguesa (50 variáveis mais presentes)



Fonte: Autor

Figura 48 – Dados faltantes Hospital das Clínicas FMUSP (50 variáveis mais presentes)



Fonte: Autor