

CENTRO UNIVERSITÁRIO FEI

ANDRÉ DE CARVALHO MADUREIRA LUIS

JOÃO VICTOR DOS SANTOS PELOSI

NATALIA CERRI CORREIA

TIAGO YUITI YAMAURA

WILLIAN TOSHIHIRO KAWANO

**PREVISÃO DE DESOCUPAÇÃO DE LEITOS HOSPITALARES POR MEIO DE
MODELOS DE *MACHINE LEARNING***

São Bernardo do Campo

2020

ANDRÉ DE CARVALHO MADUREIRA LUIS
JOÃO VICTOR DOS SANTOS PELOSI
NATALIA CERRI CORREIA
TIAGO YUITI YAMAURA
WILLIAN TOSHIHIRO KAWANO

**PREVISÃO DE DESOCUPAÇÃO DE LEITOS HOSPITALARES POR MEIO DE
MODELOS DE *MACHINE LEARNING***

Trabalho de Conclusão de Curso
apresentado ao Centro Universitário FEI,
como parte dos requisitos necessários
para obtenção do título de Bacharel em
Engenharia de Produção. Orientado pelo
Prof. Dr. João Chang Junior.

São Bernardo do Campo

2020

ANDRÉ DE CARVALHO MADUREIRA LUIS
JOÃO VICTOR DOS SANTOS PELOSI
NATALIA CERRI CORREIA
TIAGO YUITI YAMAURA
WILLIAN TOSHIHIRO KAWANO

**PREVISÃO DE DESOCUPAÇÃO DE LEITOS HOSPITALARES POR MEIO DE
MODELOS DE *MACHINE LEARNING***

Trabalho de Conclusão de Curso,
apresentado ao Centro Universitário FEI,
como parte dos requisitos necessários
para obtenção do título de Bacharel em
Engenharia de Produção.

Comissão julgadora

Prof. Dr. João Chang Junior

Prof. Dr. Dário Henrique Alliprandini

Prof. Dr. Kumiko Oshio Kissimoto

São Bernardo do Campo

2020

“O sucesso é a soma de pequenos esforços repetidos dia após dia.” (Robert Collier)

RESUMO

A busca pela melhoria da utilização dos recursos hospitalares tem sido motivada por um conjunto de fatores, como por exemplo o aumento na demanda de procedimentos cirúrgicos ou o número limitado de leitos hospitalares. Neste sentido, os pacientes que necessitam de tratamentos especiais ou de um procedimento cirúrgico, acabam submetidos a uma grande fila de espera, que infelizmente podem levar ao óbito. Portanto, o presente trabalho tem por objetivo propor modelos de machine learning para a previsão de desocupação de leitos de enfermarias e de unidades de terapia intensiva para portadores de doenças cardiovasculares congênitas, facilitando o planejamento da utilização dos leitos e o esquema de agendamento de cirurgias. Os dados levantados foram coletados por intermédio do mapeamento de processos, referentes a variáveis qualitativas e quantitativas, de modo a possibilitar a otimização da gestão dos leitos hospitalares e a aumentar o número de atendimentos com os mesmos recursos.

Palavras-chave: gestão de leitos hospitalares; modelos de machine learning para previsão; modelos de aprendizagem de máquina.

ABSTRACT

The search for improved use of hospital resources has been motivated by a number of factors, such as the increased demand for surgical procedures or a limited number of hospital beds. In this regard, patients who need special treatment or a surgical procedure, end up being subjected to a long queue, which unfortunately can lead to death. Therefore, this study aims to propose an artificial neural network model to predict the vacancy of beds in wards and intensive care units for patients with congenital cardiovascular diseases, facilitating the planning of the use of beds and the scheduling of surgeries. The gathered data will be collected through the database and process mapping, referring to qualitative and quantitative variables, in order to make possible the optimization of the bed management and increase the number of attended patients with the same resources.

Keywords: bed management; artificial neural network model; machine learning models.

LISTA DE ILUSTRAÇÕES

Figura 1: Horizontes de planejamento.....	15
Figura 2: Neurônio Biológico.....	16
Figura 3: Neurônio artificial (modelo perceptron).....	17
Figura 4: Neurônio artificial (modelo Adaline).....	18
Figura 5: Arquitetura de Rede Neural.....	19
Figura 6: Exemplo de Rede Perceptron.....	21
Figura 7: Exemplo de rede Feedforward de Camada Simples.....	22
Figura 8: Exemplo de rede Feedforward de Camadas Múltiplas.....	23
Figura 9: Diagrama em blocos da aprendizagem com um professor.....	27
Figura 10: Classificação treinamento supervisionado.....	27
Figura 11: Funcionalidade em mineração de dados.....	28
Figura 12: Exemplo de árvore de decisão.....	29
Figura 13: Exemplo de floresta aleatória.....	30
Figura 14: Processo dos pacientes neonatal.....	32
Figura 15: Processo dos pacientes da UTI Pediátrica.....	33
Figura 16: Volume Paciente x Idade na data da cirurgia.....	34
Figura 17: Demanda média x Mês.....	34
Figura 18: Distribuição Indicação Cirurgia.....	35
Figura 19: Modelo A x Acuracidade.....	39
Figura 20: Modelo B x Acuracidade.....	40

SUMÁRIO

1	INTRODUÇÃO	9
1.1	JUSTIFICATIVA DO TEMA	10
1.2	OBJETIVO GERAL	11
1.3	OBJETIVOS ESPECÍFICOS	11
2	REFERENCIAL TEÓRICO	12
2.1	CARDIOPATIA CONGÊNITA	12
2.2	FATORES QUE INFLUENCIAM FALTAS DE LEITOS DE UTI	13
2.3	PREVISÃO DE DEMANDA	14
2.4	REDES NEURAIS	16
2.4.1	Neurônio Biológico	16
2.4.2	Redes Neurais Artificiais	17
2.4.2.1	<i>Aplicações das Redes Neurais</i>	19
2.4.3	Arquitetura das Redes Neurais	19
2.4.3.1	<i>Camadas de Entrada</i>	20
2.4.3.2	<i>Camada Oculta</i>	20
2.4.3.3	<i>Camada de Saída</i>	20
2.4.4	O modelo Perceptron	20
2.4.5	Rede Feedforward	21
2.4.5.1	<i>Feedforward de Camada Simples</i>	22
2.4.5.2	<i>Feedforward de Camadas Múltiplas</i>	22
2.4.6	Função de Ativação	23
2.4.6.1	<i>Função Linear</i>	23
2.4.6.2	<i>Função Logística</i>	24
2.4.6.3	<i>Função Tangente Hiperbólica</i>	24
2.4.7	Processos de Treinamento das Redes Neurais Artificiais	25
2.4.7.1	<i>Treinamento Não Supervisionado</i>	25
2.4.7.2	<i>Aprendizagem usando lotes de padrões</i>	26

2.4.7.3	<i>Aprendizagem usando padrão por padrão</i>	26
2.4.7.4	<i>Treinamento Supervisionado</i>	26
2.4.8	Mineração de dados	28
2.4.9	Árvore de decisão	29
2.4.9.1	<i>Floresta aleatória</i>	30
3.	TRABALHO DE CAMPO	31
3.1	METODOLOGIA	31
3.1.1	Mapeamento de processos	32
3.1.2	Coleta de dados e análise	33
3.1.2.1	<i>Variáveis empregadas no modelo</i>	36
3.1.2.2	<i>Variáveis qualitativas</i>	36
3.2	DESENVOLVIMENTO DO MODELO	37
3.3	DISCUSSÃO E ANÁLISE DE RESULTADOS	39
4	CONCLUSÃO	41
	REFERÊNCIAS	42
	APÊNDICE A – DIAGRAMA DE BLOCOS A	46
	APÊNDICE B – DIAGRAMA DE BLOCOS B	47
	APÊNDICE C – PROGRAMAÇÃO EM LINGUAGEM PYTHON	48

1 INTRODUÇÃO

A cardiopatia congênita é definida como uma má-formação do coração e dos grandes vasos, presente desde o nascimento, que decorre por volta da 8ª semana de gestação, no momento da formação do coração do bebê. Trata-se da segunda maior causa da mortalidade infantil no mundo; é responsável por 3% a 5% das mortes no período neonatal e representa, aproximadamente, 40% das doenças congênitas. (BRAUNWALD; ZIPES; LIBBY, 2013; AMORIM, et al., 2008; SANTANA, 2000)

Estima-se que entre as 130 milhões, em média, de crianças nascidas por ano no mundo, 1% apresenta algum tipo de cardiopatia congênita. No Brasil, são registrados em média 2,9 milhões de nascidos vivos por ano, ou seja, cerca de 29 mil crianças apresentam algum tipo de cardiopatia congênita anualmente (DATASUS, 2017).

Por conta da gravidade das doenças e da urgência em realizar algum procedimento cirúrgico logo após o nascimento, cerca de 70% dos recém-nascidos precisam de algum tipo de procedimento ainda no período neonatal, sujeitos a grandes filas de espera nos hospitais. Infelizmente crianças vão a óbito antes mesmo de serem convocadas a uma cirurgia, portanto uma boa gestão de leitos hospitalares é fundamental para a diminuição do tempo de espera e redução do índice de mortalidade.

Segundo Barakat (2004), a falta de leitos é um problema de grande relevância que contribui com a superlotação das emergências, dificultando o atendimento e internação, provocando insatisfação dos pacientes que necessitam do serviço e causando esperas prolongadas.

De acordo com Fernandes (2010), o objetivo principal da unidade de tratamento intensivo (UTI) é suprir as necessidades dos pacientes que estão com elevado risco de óbito, porém algumas vezes pode ser usada de forma indevida, ocupando vagas não necessárias e, conseqüentemente, utilizando recursos de forma inadequada.

Verifica-se que os erros médicos são mais prováveis de ocorrer devido à baixa qualidade da informação no quesito de ter muitos dados divergentes para tentar estimar padrões de ocupações futuros, conforme afirmam os autores Littig e Isken (2007).

Taraboulsi (2009) menciona em seu livro sobre o objetivo da implementação dos sistemas de gestão hospitalar.

O objetivo dos sistemas de gestão hospitalar é otimizar todos os processos desempenhados pela instituição de saúde, que representa ganho da qualidade dos serviços hospitalares, maior visão estratégica e maior eficiência tática e operacional e, essencialmente, adequado apoio à gestão da relação com os clientes em todos os pontos de negócios. (TARABOULSI, 2009, p. 165).

Com o avanço dos sistemas de informação e da tecnologia, a comunicação entre os diversos setores de uma instituição se tornou muito mais fácil e rápida. A melhoria dos processos, velocidade de entrega e organização do ambiente de trabalho foi inevitável e tem avançado cada dia mais. Robôs realizam as tarefas, que antes eram realizadas por seres humanos, de forma rápida e mais precisa. A busca por fazer com que máquinas realizem as funções que só o cérebro humano era capaz deu início ao surgimento da inteligência artificial.

Foram observadas as primeiras manifestações elétricas de neurônios biológicos no século 19 por DuBois Reymond. Os sistemas nervosos biológicos têm a propriedade de aprender uma função, de modo que é possível ensinar uma função até que uma rede artificial a aprenda (KOVÁCS, 2006).

As redes neurais artificiais possuem a capacidade de gerar saídas adequadas para entradas que antes eram inexistentes, por meio do modelo de aprendizagem. Com isso, as redes neurais artificiais são capazes de resolver problemas que antes pareciam não haver alguma solução (HAYKIN, 2001).

Segundo Haykin (2001), no começo o modelo não é capaz de prever com perfeição, por não ser capaz de preencher algumas lacunas, assim ele começa com suposições e depois a base é realimentada, permitindo que vá aperfeiçoando o conhecimento e obtendo previsões mais assertivas.

1.1 JUSTIFICATIVA DO TEMA

Recém-nascidos com cardiopatia congênita demandam cuidados especiais dos médicos, enfermeiras e dos leitos hospitalares, como monitorização da função cardíaca, oxigenação e acúmulo de líquidos. O tempo se torna um fator importante

para eles, pois cerca de 30% dos neonatos vão a óbito por insuficiência cardíaca ou crises de hipóxia (SANTANA, 2000).

Por conta das longas filas de espera para uma cirurgia cardíaca, alguns quadros clínicos apresentam piora ou vão a óbito antes da realização do procedimento cirúrgico. Uma melhor gestão de leitos da UTI pediátrica e uma previsão mais assertiva de quando um paciente terá alta, diminuirá o tempo de espera nas filas, aumentando as chances de os recém-nascidos obterem um tratamento adequado.

1.2 OBJETIVO GERAL

Propor um modelo de rede neural artificial para a previsão de desocupação de leitos de enfermarias e de unidades de terapia intensiva para portadores de doenças cardiovasculares congênitas, facilitando o planejamento da utilização dos leitos e o agendamento de cirurgias.

1.3 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

- Mapear os processos e coletar dados a fim de utilizá-los como variáveis quantitativas e qualitativas;
- Identificar os modelos de redes neurais artificiais que apresentem melhor acuracidade;
- Desenvolver o modelo selecionado na linguagem de programação python.

2 REFERENCIAL TEÓRICO

Qualquer alteração durante a formação do feto, na gestação, pode acarretar de pequenas assimetrias até defeitos maiores com más-formações estéticas e funcionais de vários órgãos como no coração, pulmão, rins etc. Podem decorrer de fatores genéticos como também de fatores ambientais e, normalmente, a maior parte delas são identificadas nos três primeiros meses de gestação no exame pré-natal (BOTELHO, 2001).

Segundo Yaakov et al. (2006), em média, de 3% a 5% dos nascidos vivos apresentam algum tipo de doença congênita e, no Brasil, representa a segunda maior causa de mortalidade no período neonatal (DATASUS, 2017).

2.1 CARDIOPATIA CONGÊNITA

Uma doença cardíaca congênita (DCC) é uma anormalidade estrutural do coração ou de grandes vasos sanguíneos intratorácicos que tenham ou possam ter significância funcional (MITCHELL, KORONES, BERENDES, 1971). De acordo com Triedman e Newburguer (2016), a DCC é a má-formação congênita mais comum e a segunda maior causa de mortalidade entre os recém-nascidos.

Essa comorbidade pode ser dividida entre cianóticas e acianóticas, o que significa que o afetado pode ter má oxigenação do sangue arterial (cianótica) ou não (acianótica). Born (2009) diz que a cianose aumenta o risco de mortalidade materna e fetal em até 30% e que a cardiopatia congênita cianótica mais comum é a Tetralogia de Fallot. Neste caso, a mortalidade é maior do que 4% na gestação de pacientes não corrigidos cirurgicamente. Além da Tetralogia de Fallot, também existem a Anomalia de Ebstein e a Síndrome de Eisenmenger entre as cardiopatias cianóticas mais comuns.

Ainda de acordo com Born (2009), as cardiopatias congênitas acianóticas são mais comuns do que as cianóticas, sendo as de maior ocorrência a comunicação interatrial (CIA) e a comunicação interventricular (CIV). A CIA representa 35% de todos os defeitos cardíacos congênitos (LOPES; MESQUITA, 2014) e as consequências mais frequentes que ela gera são a fibrilação atrial, embolia paradoxal e insuficiência cardíaca.

No caso das CIV, o quadro clínico é determinado pelo tamanho do defeito e grau da hipertensão arterial pulmonar (HAP). Se o defeito é pequeno, não há alteração hemodinâmica importante, mas se o defeito é grande, as alterações hemodinâmicas são intensas e a evolução da gravidade pode vir acompanhada de insuficiência cardíaca, arritmia, embolia paradoxal e endocardite infecciosa. Além da CIA e da CIV, a persistência do canal arterial (PCA), a estenose pulmonar, a estenose aórtica e a coarctação da aorta também são cardiopatias congênitas acianóticas (BORN, 2009).

Estudos mostram que algumas DCC's podem mostrar uma tendência de acordo com o sexo; persistência do canal arterial (PCA), anomalia de Ebstein da válvula tricúspide e comunicação interatrial (CIA) são mais frequentes em pessoas do sexo feminino, enquanto a estenose da valva aórtica, coarctação da aorta, síndrome de hipoplasia do coração esquerdo, atresia pulmonar e tricúspide, e transposição de grandes artérias (TGA) são mais frequentes no sexo masculino. (BRAUNWALD, 2013)

2.2 FATORES QUE INFLUENCIAM FALTAS DE LEITOS DE UTI

No que se refere a assistência hospitalar do Sistema Único de Saúde (SUS), verifica-se que ocorrem inúmeros problemas devido a diversos fatores, sendo alguns deles de maior impacto (JUSTOCANTINS, 2013).

Verifica-se que os erros médicos são mais prováveis de ocorrer devido à baixa qualidade da informação no quesito de ter muitos dados divergentes para tentar estimar padrões de ocupações futuros, conforme afirmam os autores Littig e Isken (2007).

Segundo Rocco e Soares (2006), para que se possa ser mais assertivo no uso de recursos hospitalares e dos leitos de UTI, deve-se melhorar o processo de triagem na entrada do paciente, identificando a real necessidade de utilização dos recursos.

Os autores Rocco e Soares (2006) citam, ainda, que o problema de falta de leitos nas UTI's acontece por vários lugares do mundo. No Brasil esta situação não se encontra diferente, a demanda de pacientes com necessidades de utilização das unidades de terapia intensiva é muito maior do que a disponibilidade das mesmas. De acordo com a recomendação do Ministério da Saúde presente na Portaria GM/MS nº. 1.631/GM, de 1 de outubro de 2015, estima-se que a necessidade de quartos de

unidade intensiva é um percentual de 4% a 10% a mais que o total de leitos disponíveis nos hospitais.

Conforme Azambuja (2014), os diversos fatores como a falta de profissionais qualificados, a falta de equipamentos, a baixa remuneração, a utilização de medicamentos de baixa qualidade, bem como a inexistência dos serviços básicos em muitas redes públicas de serviços de saúde, pode ter grande influência na falta de leitos.

Em 2006, o Instituto de Medicina Americano (IOM) apontou outros fatores, como o aumento do número de visitas, o encerramento de hospitais, as pressões financeiras e as ineficiências operacionais (BARRETT; FORD; WARD-SMITH, 2012).

O aumento do tempo de permanência no serviço de emergência hospitalar é o principal fator que causa a falta de leitos devido a superlotação, sendo abordado por Basso (2009), Bittencourt e Hortale (2009).

Os fatores não-clínicos apresentam influências na prorrogação da utilização de leitos, mesmo no momento que o paciente já se encontra liberado pelo hospital. Existem situações em que os familiares têm dificuldades para buscar o paciente; outras vez o paciente aguarda o laudo de um exame; e, outras vezes o paciente fica por como preferências das equipes médicas (Bryan, Gage & Gilbert, 2006).

Dessa forma, verifica-se que a eficiência é impactada de forma negativa, interferindo nos custos, na qualidade de atendimento e na adequação dos cuidados.

2.3 PREVISÃO DE DEMANDA

Segundo Ritzman e Krajewski (2004) uma previsão é uma avaliação de eventos futuros utilizada para fins de planejamento, sendo indispensável para a determinação de que recursos são necessários, para o planejamento dos recursos existentes e a aquisição de recursos adicionais. As precisões assertivas direcionam os programadores e gestores a tomarem decisões para realocação de recursos eficientemente, promovendo uma redução nos tempos de produção e estoques.

Ainda segundo Ritzman e Krajewski (2004) os métodos de previsão podem se basear em modelos matemáticos que utilizam bases de dados passados, em métodos

qualitativos que aproveitam a experiência gerencial, ou em uma combinação de ambos. Eventos como campanhas promocionais, condições econômicas e ações de concorrentes são considerados como variáveis independentes. Uma variação e combinação desses métodos são valiosas para a estimativa futura dos tempos de processamento e dos efeitos da curva de aprendizagem.

De acordo com Corrêa e Corrêa (2012), para os processos decisórios é preciso ter uma boa “visão” do futuro, obtida a partir de processos de previsão. Para apoiar essas decisões se faz necessário que as previsões apresentem diferentes horizontes. Quando se considera um horizonte de curto prazo tem-se decisões de inércia pequena, que faz uso de poucos recursos e gera menos gastos, sendo que uma decisão errônea não é tão relevante. Em um horizonte de médio prazo tem-se uma inércia média e em um horizonte de longo prazo tem-se decisões de inércia grande, que demandarão um número maior de recursos e, conseqüentemente, um impacto maior em decisões equivocadas.

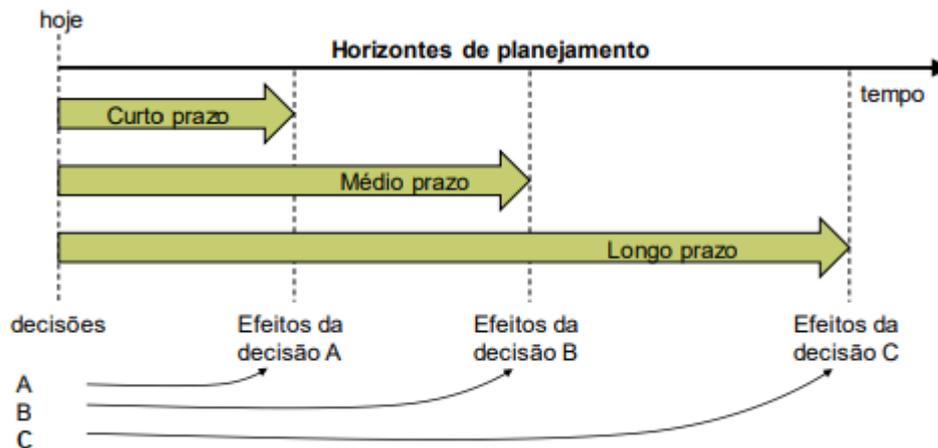


Figura 1: Horizontes de planejamento.
Fonte: Corrêa e Corrêa, 2012.

Previsões são resultantes de um conjunto de atividades que incluem: a coleta de informações relevantes, o tratamento destas informações, a consideração de fatores qualitativos relevantes, a projeção de padrões de comportamento, a estimativa de erros da previsão, a busca de padrões de comportamento fazendo muitas vezes uso de métodos quantitativos de tratamento de séries temporais de dados passados, entre outros. (CORRÊA; CORRÊA, 2012)

2.4 REDES NEURAIS

Segundo Haykin (2001) com a constante evolução da inteligência artificial (IA), houve a evolução das técnicas computacionais por meio de diversos modelos matemáticos. Analogamente ao cérebro animal as redes neurais artificiais empregam ligações entre as unidades de processamento, adquirem conhecimento pelas experiências à que são submetidas, tornando-as capazes de resolver problemas dos mais simples aos mais complexos.

2.4.1 Neurônio Biológico

O cérebro possui uma forma de processamento de informações de extrema importância no desempenho das funções usadas cotidianamente; tem-se que para o funcionamento básico do cérebro existe uma unidade básica de processamento que é o neurônio. Os neurônios são formados pelos dendritos, que são um conjunto de terminais de entrada, pelo corpo central e pelos axônios que são longos terminais de saída. Os neurônios estão relacionados entre si pelas conexões sinápticas; a sinapse é a região onde neurônios entram em contato, e por meio da qual os impulsos nervosos são transmitidos entre eles por se tratar de um sistema modulado de funcionamento. As sinapses retirar o se tornam possíveis os processos de raciocínio, memória, aprendizado, percepções e controle motor.

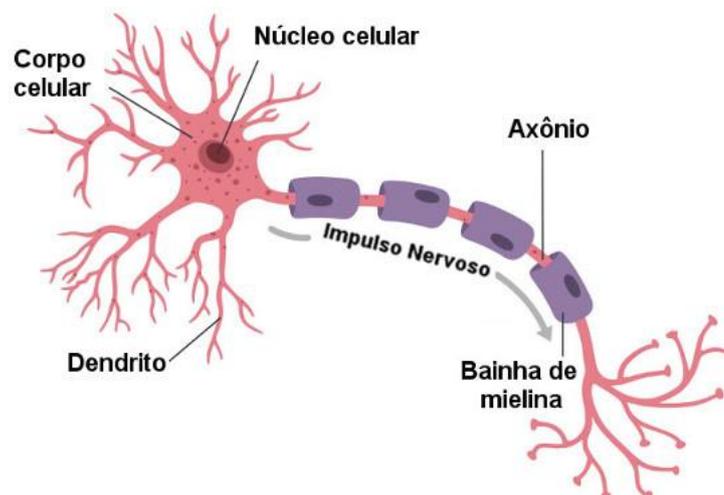


Figura 2: Neurônio Biológico

Fonte: Silva, Spatti e Flauzino (2016).

2.4.2 Redes Neurais Artificiais

De acordo com Haykin (2001) a rede neural necessita de dois parâmetros para que traga seu melhor resultado computacional: primeiro suas forças de conexões entre as unidades de processamento ou neurônios, e segundo sobre o processo de aprendizagem e generalização que se refere à capacidade das redes neurais de produzir uma saída a partir de uma entrada não mapeada durante seu período de aprendizagem. Em geral, não se verifica a capacidade de resolução de problemas apenas com uma rede neural; é necessário realizar a decomposição em tarefas mais simples e, por intermédio delas agrupa-se as coincidentes para que haja a construção de uma arquitetura computacional.

McCulloch e Pitts (1943) propuseram o primeiro modelo de redes neurais que apesar de sua simplicidade foi o início de discussões nas quais foram possíveis as criações de modelos matemáticos. O modelo proposto tratava-se de unidades binárias de processamento e foi denominado de lógica limiar, que abriu caminho para outras pesquisas mais aprofundadas sobre as redes neurais.

Hebb (1949) apresentou uma nova hipótese, mais complexa, sobre a forma de funcionamento das conexões sinápticas, trazendo a visão de que quando o cérebro dispara ligações simultaneamente proporciona uma ligação fortalecida, enquanto ligações que não são ativadas em conjunto tendem a se enfraquecer; essa teoria trouxe evoluções nas teorias das redes neurais artificiais.

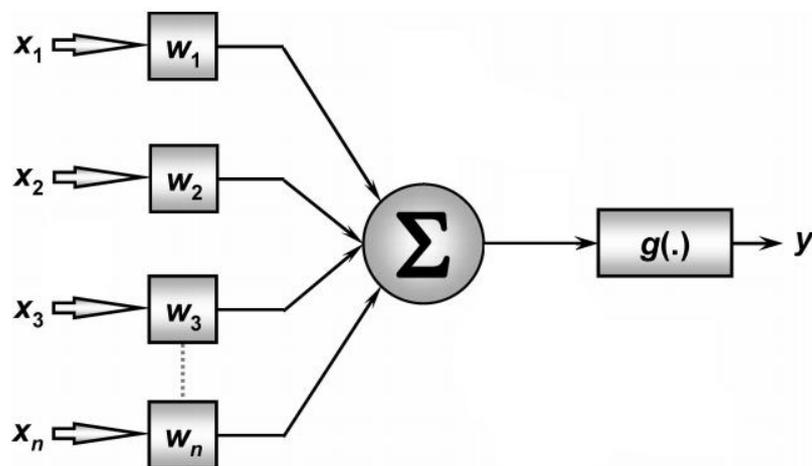


Figura 3: Neurônio artificial (modelo perceptron)

Fonte: Silva, Spatti e Flauzino (2016).

A primeira evolução do modelo inicial foi proposta em 1957: o modelo matemático denominado Perceptron. Segundo Rosenblatt (1958) o modelo apresenta de forma simples o funcionamento de uma rede neural no contexto matemático; esse mesmo modelo propõe que existe um conjunto de variáveis de entrada $\{X_1, X_2, X_3, \dots, X_n\}$, nos quais são adicionados pesos $\{W_1, W_2, W_3, \dots, W_n\}$, a cada entrada seguindo seu ranking de importância e apenas uma variável de saída, mostrada na figura 3, que é determinada por uma soma ponderada $\sum(X_i \cdot W_i - \theta)$ onde é gerado um potencial de ativação; esse valor é confrontado com o chamado valor limiar que gerará o valor binário da única saída, conforme expressão:

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

Ao decorrer do tempo percebeu-se que o Perceptron tinha limitações para resoluções com maiores níveis de complexidade, quando surge em 1959 um novo modelo. Segundo Widrow e Hoff (1960) criadores do modelo Adaline (Adaptive Linear) para que se obtivesse uma melhor condição de resultados era necessário que fosse reduzido o erro médio ao final de cada padrão apresentado, onde então foi criada a regra Delta que pela regressão linear diminuía o erro entre a saída obtida e a esperada por intermédio dos pesos utilizados.

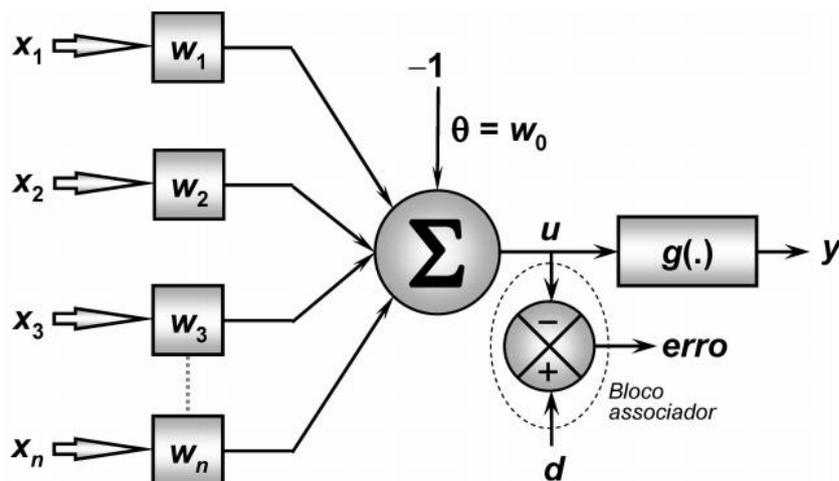


Figura 4: Neurônio artificial (modelo Adaline)

Fonte: Silva, Spatti e Flauzino (2016).

2.4.2.1 Aplicações das Redes Neurais

Existem diversas tarefas que utilizam aplicações de redes neurais, dentre elas, transformações de dados, reconhecimento de padrão e previsões de séries temporais. Todos os tipos de aplicações das redes neurais podem agrupados em dois grandes grupos:

- Redes para aproximação de função: para esse método as respostas são contínuas, e usado um modelo de aproximação de funções (interpolação);
- Rede de classificações de padrões: a classificação é considerada um caso particular de aproximação de função, onde o valor de saída não é contínuo, e sim discreto, que pertence a um conjunto finito de valores. Para realizar a discretização na saída é aplicada a lógica de limiar.

2.4.3 Arquitetura das Redes Neurais

Uma arquitetura de rede é a forma pela qual os neurônios de uma RNA estão estruturados. De acordo com Wang (2003), uma rede neural artificial consiste em uma camada de entrada, uma ou mais camadas intermediárias (camadas ocultas) e uma camada de saída, mas existem também redes sem camadas intermediárias. A figura 5 mostra uma arquitetura de rede neural.

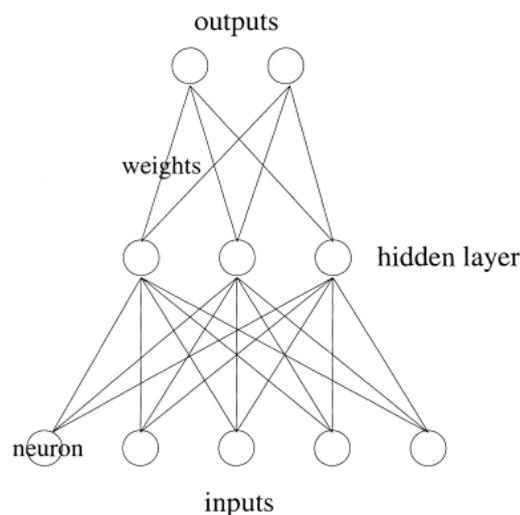


Figura 5: Arquitetura de Rede Neural
Fonte: Wang (2003)

2.4.3.1 Camadas de Entrada

Nesta camada é onde são apresentados os padrões à rede. É nela que se faz a entrada de dados, sendo esses dados os exemplos do que se quer que a máquina aprenda. O papel dos neurônios desta camada é distribuir essas entradas a todos os neurônios da próxima camada.

A camada de entrada também calcula qual o peso de cada conexão com a camada intermediária.

2.4.3.2 Camada Oculta

Esta camada é responsável pela maior parte do processamento das informações. Os padrões enviados pela camada de entrada são “decodificados”, ou seja, eles têm suas características analisadas pelos neurônios da camada intermediária e o resultado é passado para a camada de saída.

2.4.3.3 Camada de Saída

A camada de saída é a que conclui e apresenta o resultado. Ela recebe os estímulos da camada intermediária e constrói o padrão que será a resposta.

Existem certos tipos de rede que comparam o resultado obtido pela camada de entrada com um resultado desejado e caso o resultado obtido não seja satisfatório, a camada de saída propaga o erro de volta até a camada de entrada e os pesos das camadas intermediárias são recalculados até obter-se uma resposta aceitável.

2.4.4 O modelo Perceptron

Segundo Haykin (2001), o Perceptron é a forma mais simples de uma rede neural, usada para a classificação de padrões linearmente separáveis. Basicamente, ele consiste em um único neurônio com pesos sinápticos ajustáveis e bias (figura 6). O estudo dos Perceptrons iniciou-se nos anos 1950. Embora redes de todos os

tamanhos e topologias tenham sido consideradas, o único método de aprendizado efetivo naquele momento foi o das redes de camada única, por isso, a maioria dos esforços foram gastos nesse tipo de rede.

O problema que Rosenblatt propôs resolver foi o de casos simples, como a implementação das funções booleanas (E) e (OU) de duas variáveis, em que é relativamente trivial escolher os ganhos sinápticos e o valor do limiar. Porém, para a implementação de uma função discriminatória arbitrária esta escolha não é trivial e, dependendo do número de variáveis envolvidas, sem a existência de algum método, beira o impossível (KOVÁCS, 2006).

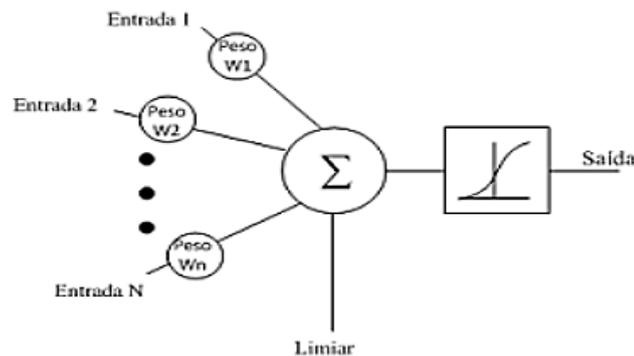


Figura 6: Exemplo de Rede Perceptron
Fonte: Adaptado de Kaykin (2001)

Em 1969, Minsky e Papert criticaram o Perceptron, demonstrando que este modelo com uma camada simples, não poderia simular o comportamento de uma função XOR (ou-exclusivo). Então na década de 1980, Rumelhart, Hinton e Williams desenvolveram um algoritmo conhecido como Backpropagation, quando foram criadas as redes Perceptron de Múltiplas camadas que, quando treinadas adequadamente, têm um grande poder de generalização, podendo aproximar quaisquer funções contínuas e não lineares.

2.4.5 Rede Feedforward

De acordo com Montana e Lawrence (1989), uma rede feedforward é aquela cuja topologia não tem caminhos fechados.

A operação de uma rede feedforward consiste em obter um resultado, dado os estados de entrada, a partir de ações que vão definindo os estados intermediários e os estados de saída.

2.4.5.1 Feedforward de Camada Simples

De acordo com Silva, Spatti e Flauzino (2016), uma rede feedforward de camada simples é aquela cuja camada de entrada está diretamente associada a um ou mais neurônios que vão gerar a resposta encontrada na camada de saída. Esse tipo de rede é usada para solucionar problemas de classificação de padrões e filtragem. A figura 7 exemplifica esse tipo de rede.

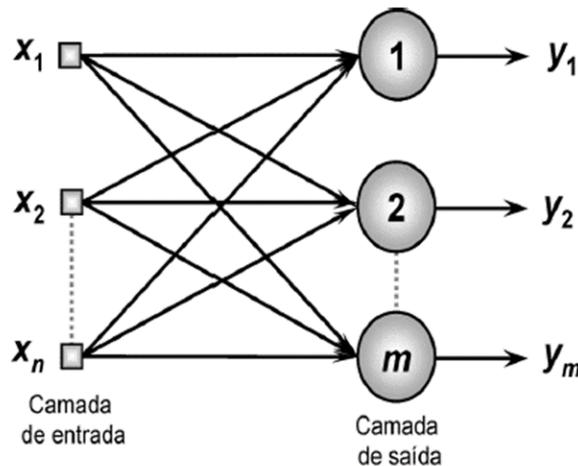


Figura 7: Exemplo de rede Feedforward de Camada Simples
Fonte: Silva, Spatti e Flauzino (2016).

2.4.5.2 Feedforward de Camadas Múltiplas

De acordo com Silva, Spatti e Flauzino (2016), a rede feedforward de camadas múltiplas é constituída por uma ou mais camadas intermediárias (ocultas) de neurônios, além das camadas de entrada e saída. Esse tipo de rede é usado para problemas de aproximação de função, classificação de padrões, identificação de sistemas, otimização, robótica e controle de processos. A figura 8 exemplifica esse tipo de rede.

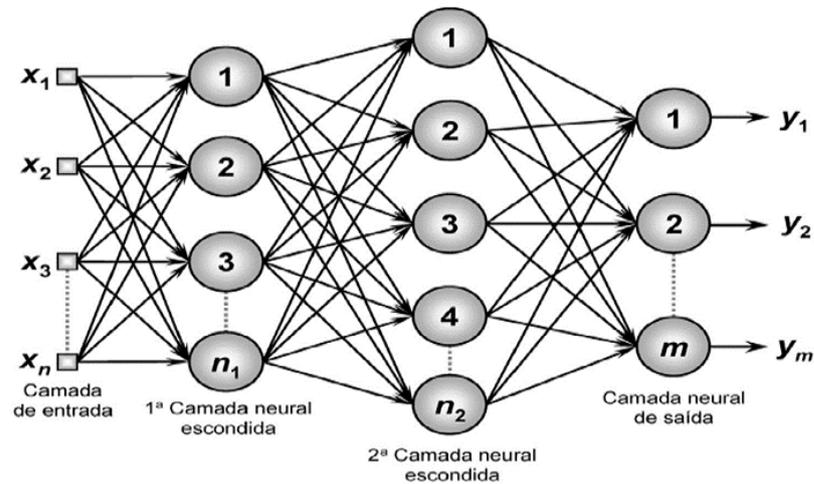


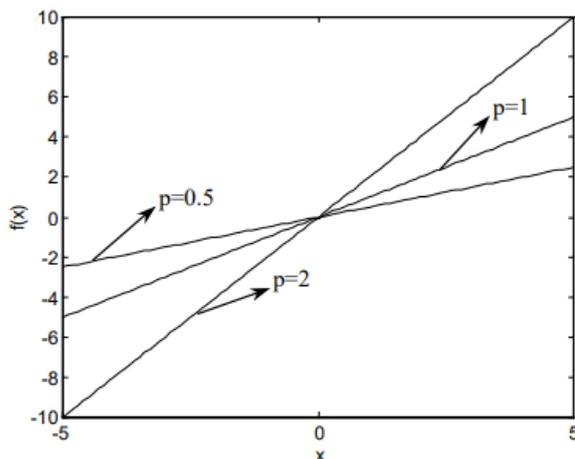
Figura 8: Exemplo de rede Feedforward de Camadas Múltiplas
 Fonte: Silva, Spatti e Flauzino (2016).

2.4.6 Função de Ativação

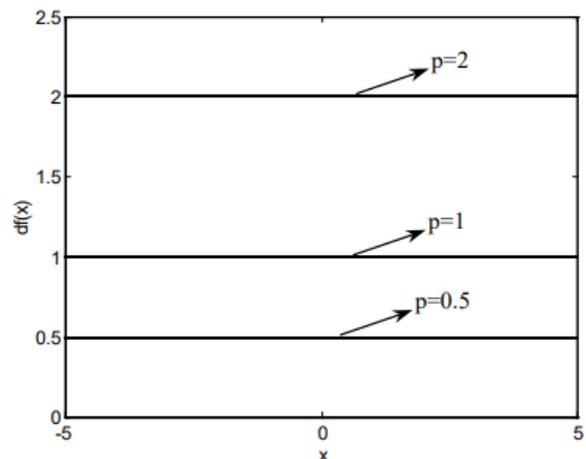
Segundo Haykin (2001) a função ativação é decisiva para que seja tomada uma decisão final com maior grau de eficiência, ou seja, essa função mostra se a informação que o neurônio está recebendo é relevante ou deve ser ignorada.

2.4.6.1 Função Linear

É comumente usada em unidades que compõe a camada de saída de uma rede de múltiplas camadas.



(a) Função linear

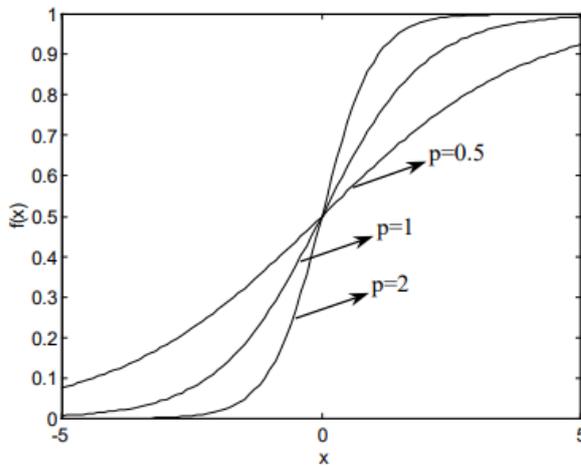


(b) Derivada

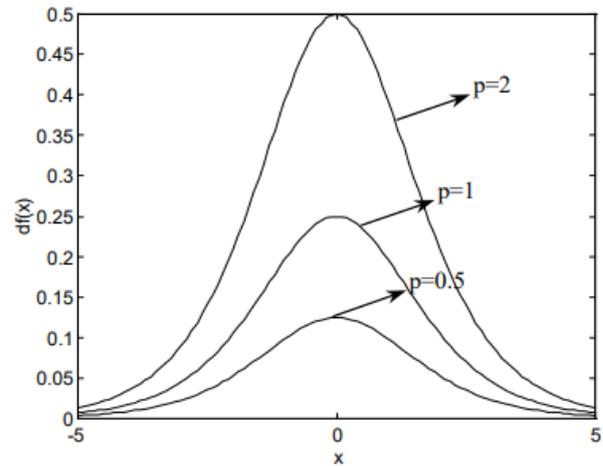
A expressão que diz respeito a essa função e a sua respectiva derivada é:

$$f(x) = p * x \quad , \quad f'(x) = p$$

2.4.6.2 Função Logística



(a) Função logística

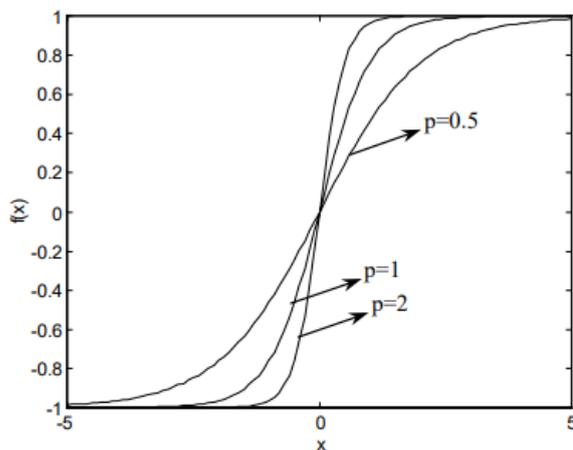


(b) Derivada

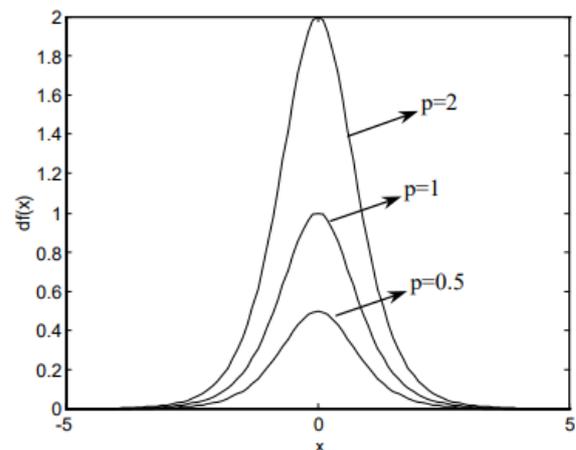
Neste tipo de função verifica-se a preocupação em limitar o intervalo de variação da sua derivada.

$$f(x) = \frac{e^{px}}{1+e^{px}} = \frac{1}{1+e^{-px}} \quad , \quad f'(x) = pf(x) * (1 - f(x))$$

2.4.6.3 Função Tangente Hiperbólica



(a) Função Tangente Hiperbólica



(b) Derivada

Diferentemente da função logística que apresenta valores no intervalo de (0,1), no caso da função tangente hiperbólica o intervalo admite valores negativos, mantendo a forma sigmoideal da função logística. A função que representa esse caso:

$$f(x) = \frac{e^{px} - e^{-px}}{e^{px} + e^{-px}} = \tanh(px) \qquad f'(x) = p(1 - f(x)^2)$$

2.4.7 Processos de Treinamento das Redes Neurais Artificiais

“Um programa de computador aprende a partir da experiência. E, em relação a uma classe de tarefas T, com medida de desempenho P, se seu desempenho em T, medido por P, melhora com a experiência E” (MITCHELL, 1997, p. 2).

O aprendizado de máquina pode ser classificado em quatro categorias, sendo elas: aprendizagem supervisionada; aprendizagem não supervisionada; aprendizagem usando lotes de padrões; aprendizagem usando lote por lote. Tais modelos devem ser detalhados ao longo do presente estudo.

2.4.7.1 Treinamento Não Supervisionado

Com base no autor Porto Filho (2017), o treinamento não supervisionado é uma técnica que, não há um professor e os grupos não são pré-definidos.

Há inúmeros exemplos de aplicação desse método que podem ser mencionados, entre eles pode-se citar as aplicações em sistemas de recomendação de filmes e músicas, e para detectar anomalias de visualização de dados.

As redes neurais artificiais, clusterização, máquinas de Boltzmann restritas, *word2vec*, *eclat*, *a priori*, são algumas das técnicas utilizadas visando a resolução de problemas. Segundo Mitchell (1997), dentre todas a mais importante é a de clusterização, em que busca encontrar grupos no conjunto de dados.

Percebe-se que devido à não aderência dos resultados de resposta com os desejados, os problemas de aprendizado não supervisionado apresentam maior complexibilidade em relação ao aprendizado supervisionado.

2.4.7.2 Aprendizagem usando lotes de padrões

Segundo Ribeiro et al. (2019), o aprendizado pode ser subdividido em online ou offline. A aprendizagem usando lotes de padrões offline é aquela que apresenta necessidade que todo o conjunto de entradas e saídas estejam disponíveis todo o tempo e todos os pesos e limiares sejam atualizados após a efetiva exposição de todo o conjunto de entradas e comparados com suas respectivas saídas.

2.4.7.3 Aprendizagem usando padrão por padrão

De acordo com Ribeiro et al. (2019), o método de aprendizagem padrão por padrão (online) necessita de uma enorme quantidade de amostras para que a saída seja mais precisa e agrega apenas um exemplo de cada vez. Os ajustes de peso e limiares são feitos após cada apresentação de amostra do conjunto, o que torna esse modelo diferente do explicado anteriormente.

Ainda com base em Ribeiro et al. (2019), tal modelo normalmente é utilizado em dois casos, o primeiro em que o sistema que está sendo mapeado varia de forma mais rápida e o segundo quando a amostra utilizada em um determinado momento pode apresentar um comportamento que não seja possível de ser detectado.

2.4.7.4 Treinamento Supervisionado

De acordo com Haykin (2001), o aprendizado supervisionado é também denominado por “Aprendizagem com um professor”, na qual requer um programador que forneça exemplos de entradas que se alinham com os resultados, vide figura 9.

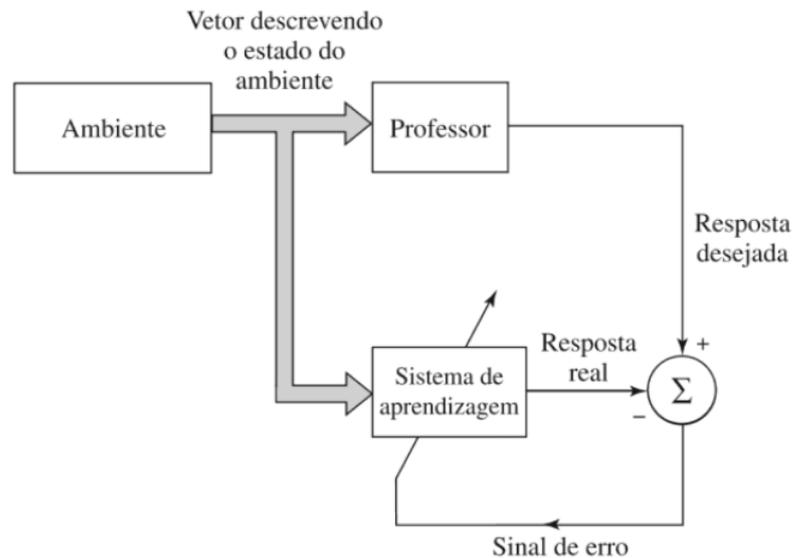


Figura 9: Diagrama em blocos da aprendizagem com um professor

Fonte: Haykin (2007).

Segundo afirmam os autores Maimon e Rokach (2005), os métodos supervisionados buscam descobrir a correlação entre a entrada de atributos e uma classe de destino. Entende-se que a entrada de atributos pode ser chamada de variáveis independentes e a classe de destino como uma variável dependente. A relação descoberta é representada em uma estrutura referida como um modelo, na qual é possível realizar a previsão da classe alvo partindo do conhecimento dos valores dos atributos de entrada.

Conforme Santos (2016), tal método pode ser classificado em problemas de regressão e classificação (figura 10).

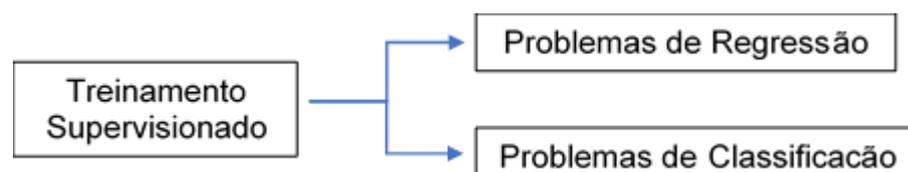


Figura 10: Classificação treinamento supervisionado

Fonte: Grupo (2019)

2.4.8 Mineração de dados

A mineração de dados é um recurso de descoberta de informações que pode agregar conhecimento e auxiliar na tomada de decisão das empresas. Sendo uma metodologia própria para preparação e exploração dos dados, interpretação de seus resultados e assimilação dos conhecimentos minerados (vide figura 11).

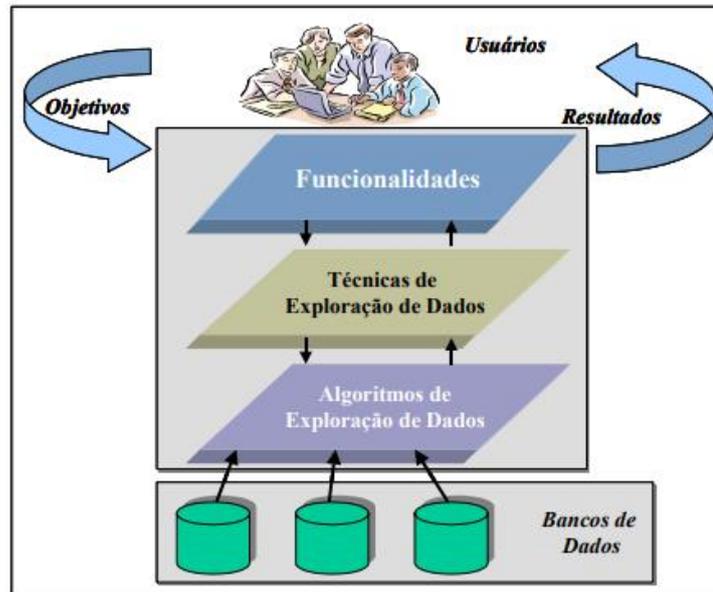


Figura 11: Funcionalidade em mineração de dados
Fonte: Côrtes *et al.* (2002)

Segundo Han et al. (2005) existem variadas funcionalidades de mineração de dados, definidas como análise de associação, clusterização, análise de *outliers* e classificação:

- A análise de associação tem como finalidade descobrir regras de associação em um conjunto de dados;
- A clusterização permite segmentar um conjunto de dados com o objetivo de formar grupos baseados em semelhanças. É importante que os dados contidos nesses grupos sejam homogêneos e heterogêneos entre os grupos;
- A análise de *outliers* permite a identificação de um conjunto de dados que não obedecem ao modelo. Quando encontrados, podem ser tratados ou descartados para que não ocorram desvios ou riscos que prejudiquem os objetivos traçados no início do *data mining*;
- A classificação consiste na análise de uma determinada característica em um banco de dados e na sua atribuição a certa classe previamente definida.

Após a definição da melhor funcionalidade do banco de dados, são estabelecidas as técnicas de *data mining* mais adequadas para a extração de conhecimento. As técnicas devem ser aplicadas de acordo com as características do banco de dados e englobam diversas áreas científicas, como estatística e *machine learning*.

2.4.9 Árvore de decisão

As árvores de decisão são métodos de aprendizados de máquinas supervisionados, sendo normalmente utilizados em tarefas de classificação e regressão. Elas são aplicadas em diversas áreas, como marketing, na predição de vendas e na classificação de grupos econômico-sociais por comportamentos, e medicina, na determinação de diagnósticos e tratamentos.

De modo geral, é uma abordagem que usa diagramas para mapear as várias alternativas e resultados de decisões. Tendo como ponto forte, sua simplicidade na estruturação, interpretação e na visualização, não necessitando de um grande esforço para preparação de dados.

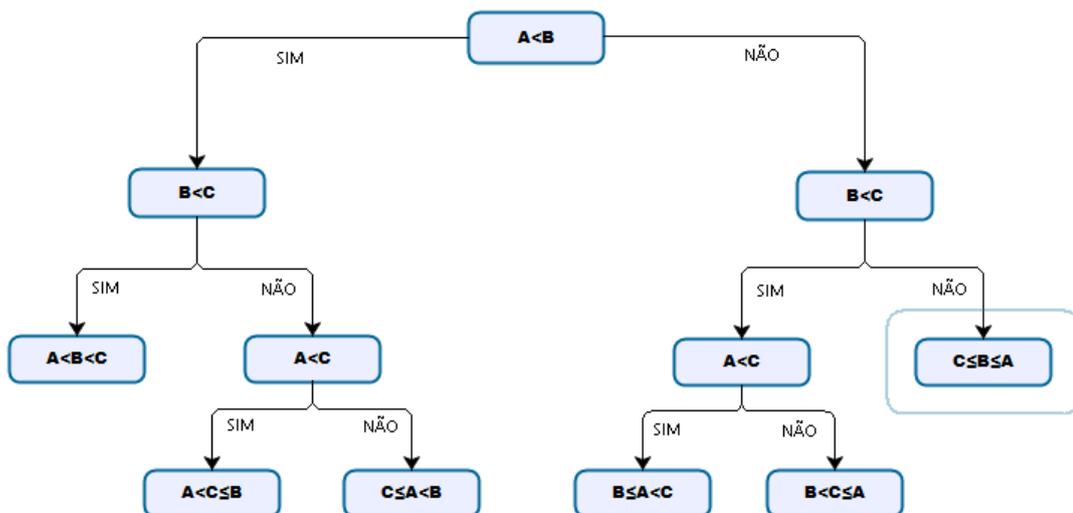


Figura 12: Exemplo de árvore de decisão
Fonte: Grupo (2019)

2.4.9.1 Floresta aleatória

A Floresta Aleatória ou conhecida como *Random Forest*, é uma técnica não linear robusta que otimiza a acurácia de predição, por realizar um ajuste sobre o conjunto de árvores no intuito de estabilizar as estimativas de um modelo. Sendo um método estatístico não paramétrico que não requer suposições distribucionais sobre a relação das covariáveis com a variável resposta. O que diferencia esse modelo é que ele realiza uma etapa a mais, após o definir da amostra, selecionando aleatoriamente variáveis predictoras para a árvore de decisão.

O nome "floresta aleatória" vem da combinação da aleatoriedade que ela usou para escolher o subconjunto de dados com um grupo de árvores de decisão, resultando em um algoritmo de aprendizagem de máquina flexível e fácil de usar, que produz excelentes resultados, na maioria das vezes.

O algoritmo de floresta aleatória não apresenta altas complexidades para sua construção, sendo acessível e trazendo ótimos resultados de predição. O algoritmo pode apresentar lentidões e ineficiências para predições de tempo real devido a grande quantidade de geração de árvores.

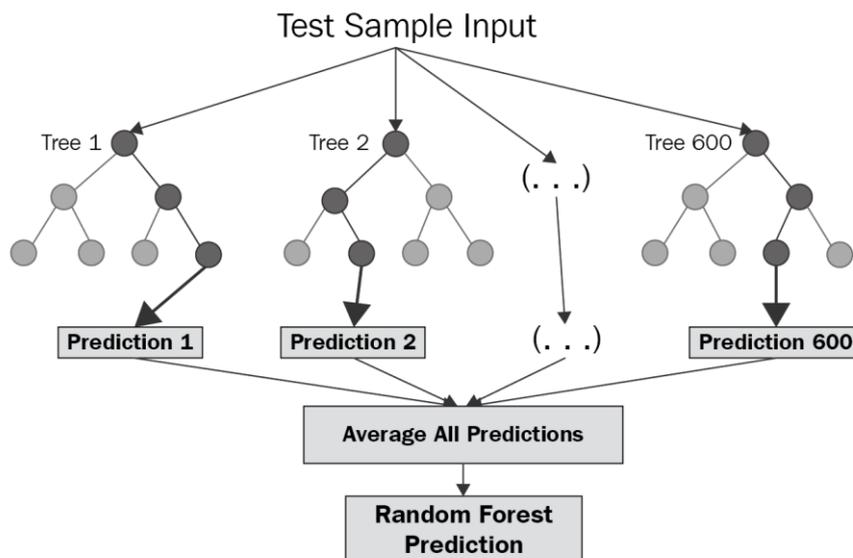


Figura 13: Exemplo de floresta aleatória
Fonte: Chakure (2019)

3. TRABALHO DE CAMPO

O trabalho contempla um desenvolvimento de um modelo de previsão de desocupação de leitos, reunindo informações através de mapeamento e coleta de dados do hospital anônimo e utilizando-os como inputs da programação, buscando obter previsões condizentes e acuradas à fim de melhorar o planejamento e alocação de recursos do hospital.

O hospital anônimo é uma instituição pública e universitária de alta complexidade, especializada em cardiologia, pneumologia e cirurgia cardíaca e torácica. Além de ser um polo de atendimento, desde a prevenção até o tratamento, a instituição se destaca como um grande centro de pesquisa e ensino.

O instituto é parte do Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo e campo de ensino e de pesquisas para outras faculdades.

Para a manutenção de sua excelência, o Instituto conta com o suporte financeiro da Fundação Zerbini, entidade privada sem fins lucrativos.

O Instituto hospitalar realiza em média por ano:

- 260 mil consultas médicas;
- 37 mil atendimentos multiprofissionais;
- 13 mil internações;
- 5 mil cirurgias;
- 2 milhões de exames de análises clínicas;
- 330 mil exames de diagnóstico de alta complexidade.

O foco do trabalho para propor um modelo de previsão de desocupação de leitos somente para os leitos que são ocupados pelos pacientes da cardiopatia congênita com a faixa etária entre 0 e 17 anos, foi devido ao aumento da taxa de mortalidade destes pacientes e à alta demanda gerando filas de espera.

3.1 METODOLOGIA

Para o desenvolvimento do modelo de previsão, foram realizadas visitas no instituto com objetivo de mapear processos, acessar o banco de dados, levantar variáveis qualitativas e quantitativas, acessar o banco de dados e identificar supostos problemas que poderiam impactar no modelo.

3.1.1 Mapeamento de processos

Com o objetivo de esclarecer todo o fluxo do processo e alinhar as expectativas dos membros da equipe, foram feitos esboços de processos com base nas visitas feitas ao instituto, mapeando as entradas, leitos e a porcentagem de distribuição dos pacientes pelas rotas. As figuras 14 e 15 mostram esses mapeamentos:

- Pacientes com 0 a 30 dias

Fluxo de Processo Neonatal (0 – 30 Dias):

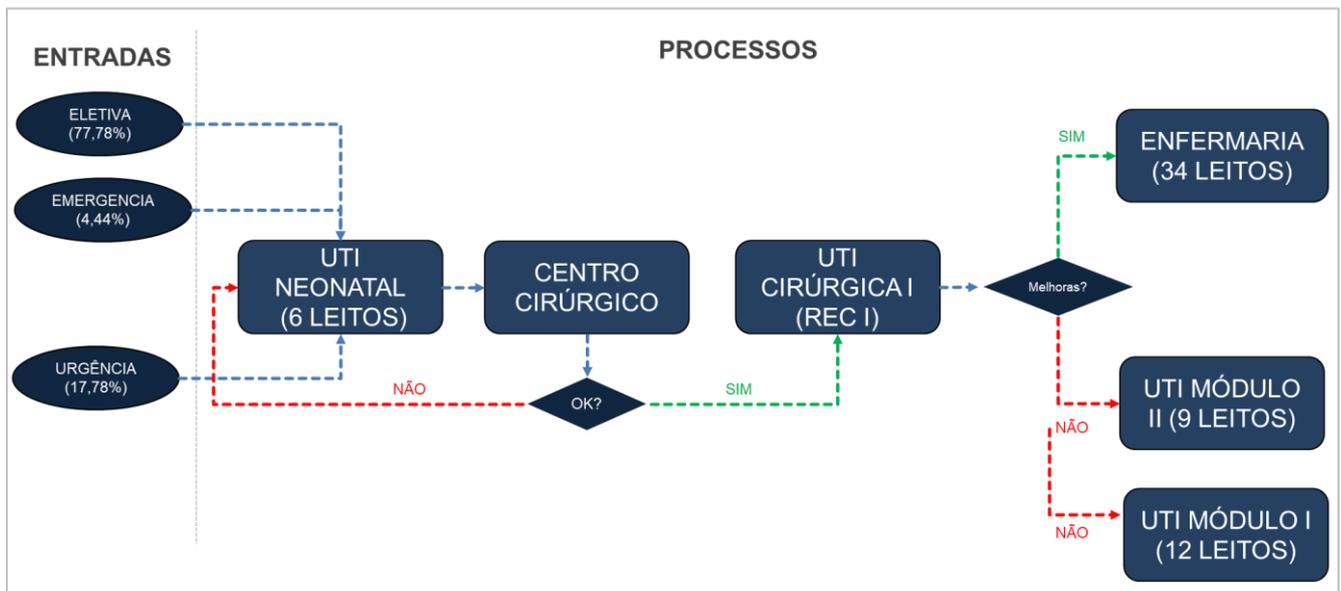


Figura 14: Processo dos pacientes neonatal
Fonte: Grupo, 2020.

- Pacientes com >30 dias

Fluxo de Processo (> 30 Dias):

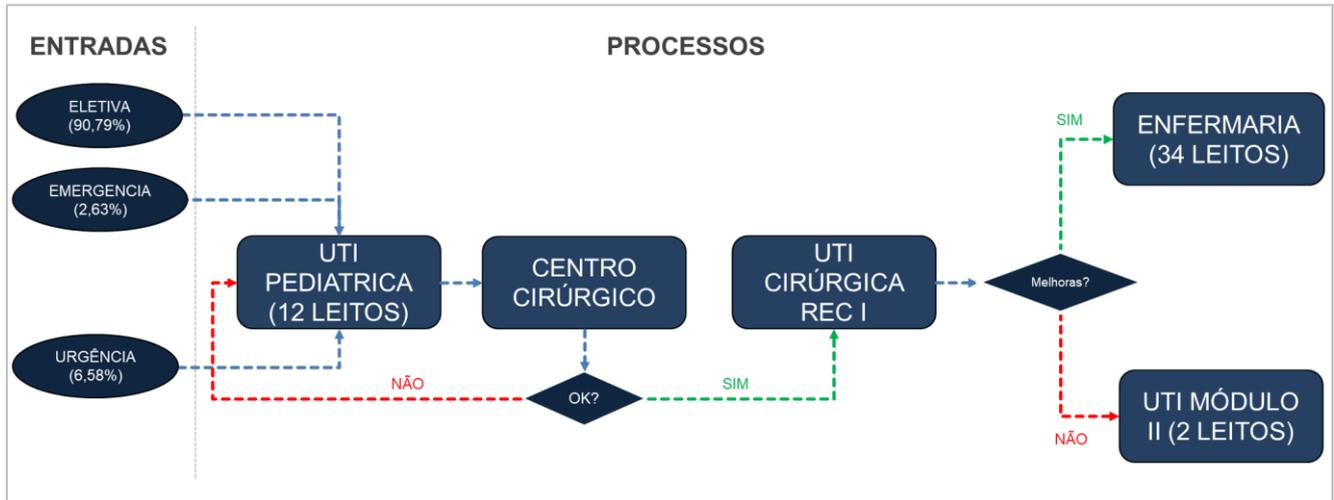


Figura 15: Processo dos pacientes da UTI Pediátrica
Fonte: Grupo, 2020.

3.1.2 Coleta de dados e análise

Por intermédio dos dados fornecidos pelos colaboradores do instituto, foram obtidos registros de todos os pacientes, contendo diversas informações como idade, sexo, procedimentos cirúrgicos, medicamento, entre outros, que serão citados nos próximos tópicos.

A figura 16 apresenta a distribuição por idade dos pacientes, onde verifica-se uma grande concentração e uma grande demanda no grupo neonatal (menor ou igual a 30 dias de idade), e o restante distribuído ao grupo pediátrico (maior que 30 dias de idade).

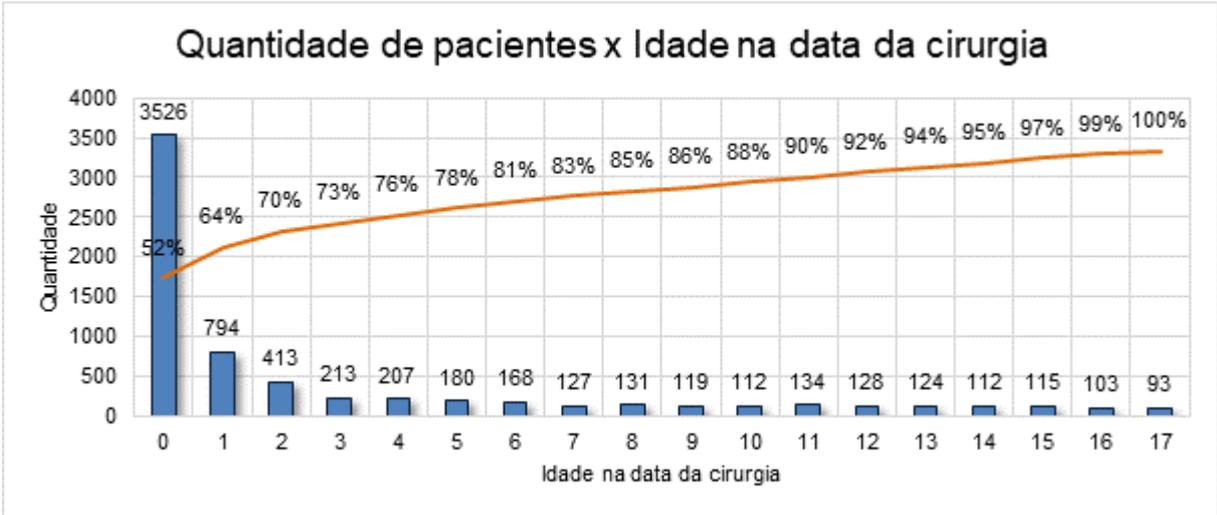


Figura 16: Volume Paciente x Idade na data da cirurgia
 Fonte: Grupo, 2020.

Com intuito de identificar sazonalidade da demanda de cirurgias, foram plotados a média da quantidade de cirurgias de acordo com seu respectivo mês. O gráfico da figura 17 apresenta um pico de solicitações em agosto, o qual não se percebe em outros meses, concluindo que o pico foi causado por fenômenos eventuais e que as demandas não seguem um padrão de sazonalidade.

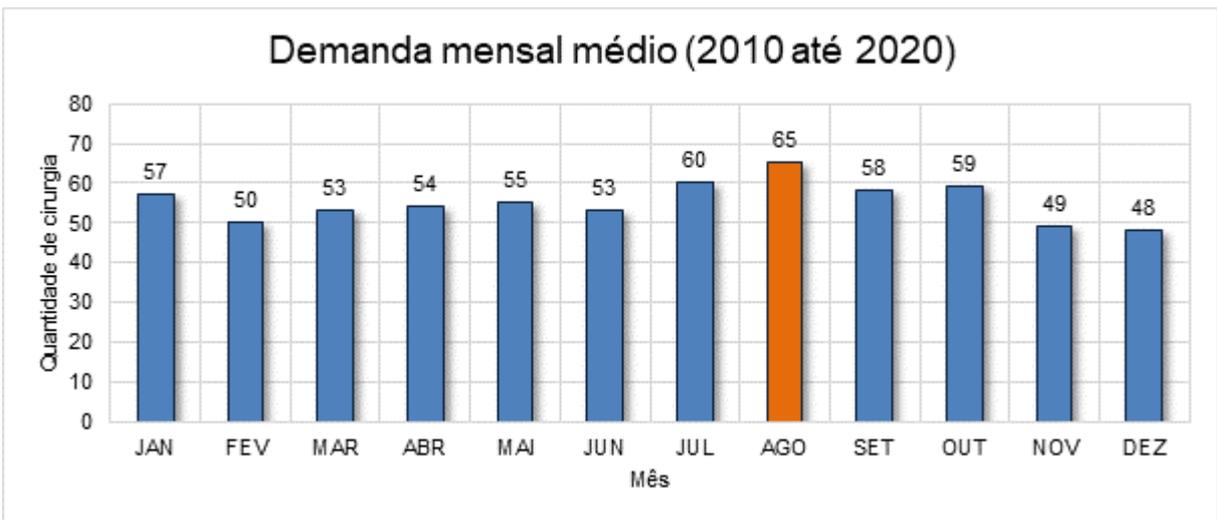


Figura 17: Demanda média x Mês
 Fonte: Grupo, 2020.

Os pacientes apresentam diferentes quadros clínicos ao chegar no hospital, os quais são classificados em quatro tipos de indicações cirúrgicas. Sendo elas:

- Eletivo: O paciente não está em quadro crítico, portanto o mesmo passa pelos procedimentos preparatórios em sua residência e é internalizado no hospital.
- Urgente: O quadro de urgência, são os pacientes que entram por qualquer canal do hospital, contando com as transferências de pacientes de outros hospitais para enfermaria, UTI ou para o Pronto-Socorro. Estes pacientes necessitam da solução da queixa inicial ou do que foi diagnosticado durante o atendimento.
- Emergência: O paciente de emergência entra via Pronto-Socorro, UTI ou enfermaria transferido de outro hospital e o problema necessita ser resolvido em 24 a 48 horas.
- Salvamento: Este quadro é formado por pacientes que estão em estado crítico e que necessitam do atendimento imediato.

A figura 18 revela os percentuais de indicações de cirurgias contemplando o período de 2010 até 2020.

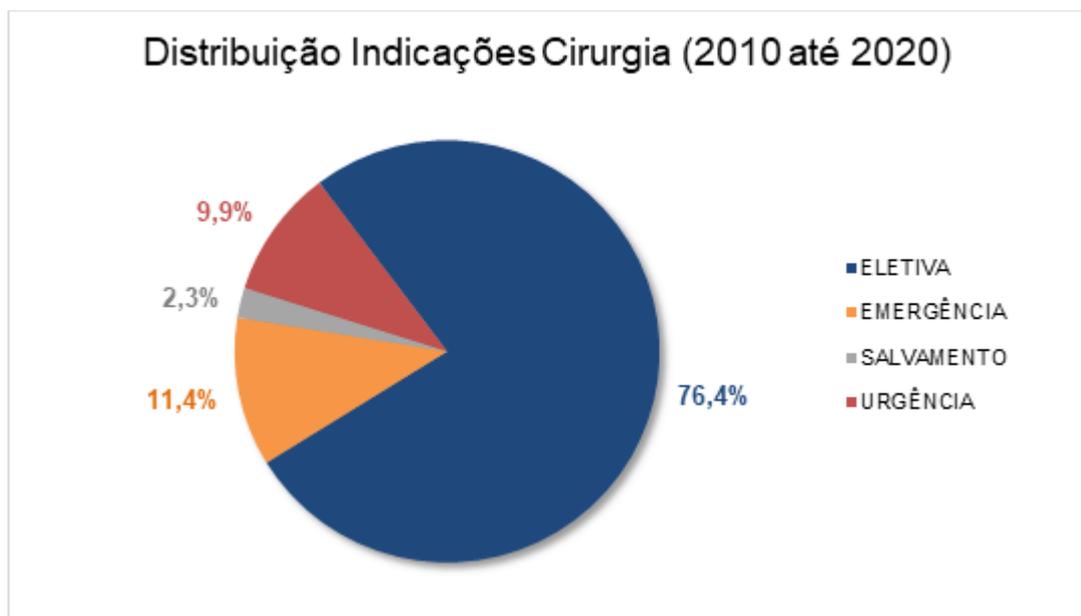


Figura 18: Distribuição Indicação Cirurgia
Fonte: Grupo, 2020.

3.1.2.1 Variáveis empregadas no modelo

Foram parametrizadas as variáveis dentro das colunas contidas na base coletada, inserindo-as no modelo de previsão como entradas à fim de aproximar o resultado do modelo mais próximo do comportamento real. As variáveis contempladas no modelo são:

- Sexo;
- Idade;
- Faixa etária;
- Solicitação (origem da solicitação cirúrgica);
- Indicação (variável indicativa do grau de prioridade);
- Nome do cirurgião;
- Planilha (observação da operação, recorrência);
- Procedimento Principal;
- Procedimento Secundário;
- Procedimentos (consolidação de todos realizados no paciente);
- Diagnóstico principal da cirurgia;
- Diagnóstico secundário da cirurgia;
- Resumo do diagnóstico principal;
- Resumo do diagnóstico secundário;
- Resumo dos diagnósticos (consolidado dos diagnósticos que o paciente recebeu);
- Infecção (data em que ocorreu a infecção);
- Infecções (descrição do tipo de infecção contraída).

3.1.2.2 Variáveis qualitativas

O modelo será alimentado com informações fornecidas pelo banco de dados, porém há outros fenômenos não registrados que podem impactar diretamente no tempo de ocupação do leito, como:

- Familiares demoram para buscar o paciente após a sua alta;
- Pacientes que necessitam de transportes da prefeitura devido a longa distância necessária para chegar à sua residência;
- Medicamentos receitados pontualmente que prorrogam o tempo de internação;
- Pacientes que aguardam o transplante;
- Prazo de internação alterado pela percepção médica pontual.

Fatores como estes podem interferir na apuração do modelo e, conseqüentemente, na obtenção dos resultados, pois estas informações apresentariam fortes correlações com o tempo de permanência.

3.2 DESENVOLVIMENTO DO MODELO

Recebendo os dados brutos pelo instituto, foi necessário realizar um tratamento através da linguagem SQL para aplicar o filtro de idade (menor ou igual à 17) e de especialidade (Cardiopatia Congênita) sobre as informações que seriam utilizadas no modelo, à fim de reduzir o tempo de processamento do modelo e trazer uma maior acuracidade.

Após a manipulação primária dos dados, a base foi chamada dentro da linguagem Python 3.7, tratados na plataforma web em Jupyter Notebook, onde foram importadas as bibliotecas pandas, matplotlib, numpy e sklearn, para realizar o tratamento e a análise de dados. Houve uma manipulação secundária que consiste em converter as informações que estavam em STRINGS para INT, indexando os textos para o modelo conseguir realizar a leitura e assim dar sequência para as próximas etapas. Foi necessário substituir todos os valores nulos contidos nas colunas pela média do tempo de permanência da UTI.

Na sequência, foram declarados dois tipos de variáveis dentro do modelo, as variáveis preditivas (X) e a variável a ser prevista (Y), sendo as variáveis preditivas as declaradas na seção 3.1.2.1 e a variável prevista é o tempo de permanência na UTI.

No modelo, a base foi dividida em duas partes, 90% para treinar o modelo e 10% para testar o modelo; assim o modelo absorve e compreende o comportamento das informações e realiza as simulações para obter a sua acurácia.

Após os preparatórios, foram executados seis modelos de previsão com o objetivo de comparar e identificar o modelo mais preciso para prever o tempo da desocupação dos leitos. Os modelos simulados pelo algoritmo descrito no apêndice A são:

- Naive Bayes;
- Regressão logística;
- XG Boosts;
- kNN (k-vizinhos mais próximos);
- SVM;
- Random Forest.

Com intuito de aumentar a acuracidade do modelo, foram definidas faixas de tempo de permanência nos leitos, no total 18 faixas, sendo as primeiras 17 classes referente aos intervalos definidos na tabela 1, e o 18º um agrupamento dos excedentes aos 30 dias. A variável preditiva tem como variações as faixas de tempo definidas na tabela 1.

Tabela 1: Definição das faixas de tempos em dias.

INDEX	FAIXA
1	1 a 2,7
2	2,7 a 4,4
3	4,4 a 6,1
4	6,1 a 7,8
5	7,8 a 9,5
6	9,5 a 11,2
7	11,2 a 12,9
8	12,9 a 14,6
9	14,6 a 16,3
10	16,3 a 17,1
11	17,1 a 18,8
12	18,8 a 21,5
13	21,5 a 23,2
14	23,2 a 24,9
15	24,9 a 26,6
16	26,6 a 28,3
17	28,3 a 30
18	> 30 dias

Para o cálculo do intervalo de classes, utilizou-se a equação abaixo:

$$Intervalo = \frac{Amplitude}{número\ de\ classes} = \frac{30 - 1}{17} = 1,7$$

Nos apêndices A e B estão os diagramas de blocos dos algoritmos de *machine learning* desenvolvidos neste trabalho.

Para a discussão e análise de dados foram elaborados gráficos por meio dos resultados e simulações dos modelos, os quais são discutidos na seção 3.3.

3.3 DISCUSSÃO E ANÁLISE DE RESULTADOS

Pela primeira simulação e seus resultados obtidos, sem a definição das faixas de classe, observa-se que nenhum dos modelos apresentaram resultados atrativos, demonstrando grandes defasagens entre o tempo de permanência real e o tempo simulado, sendo a maior acuracidade obtida de 25,15% resultante do modelo XGBoosts (vide figura 19).

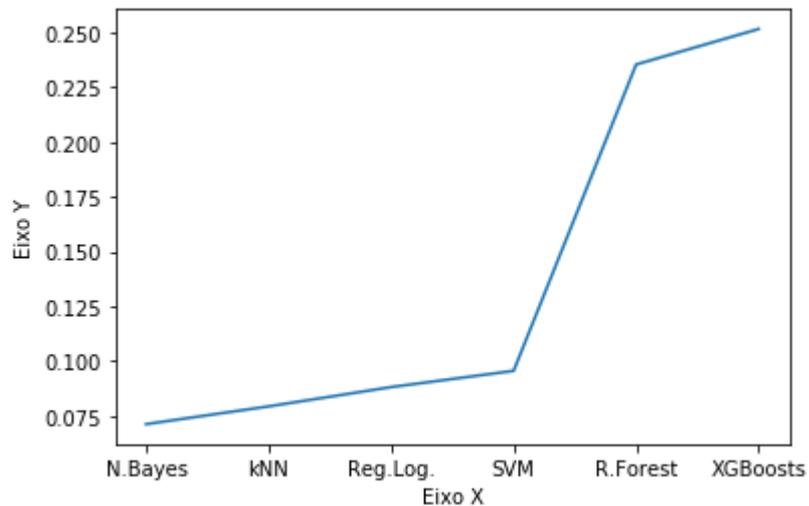


Figura 19: Modelo A x Acuracidade
Fonte: Grupo, 2020.

Após a inclusão das faixas de classes, os modelos demonstraram uma acuracidade maior em relação aos anteriores. Constatou-se que novamente, o modelo de XGBoosts obteve os melhores resultados em relação aos demais, apresentando um aumento para 36,76% (vide figura 20).

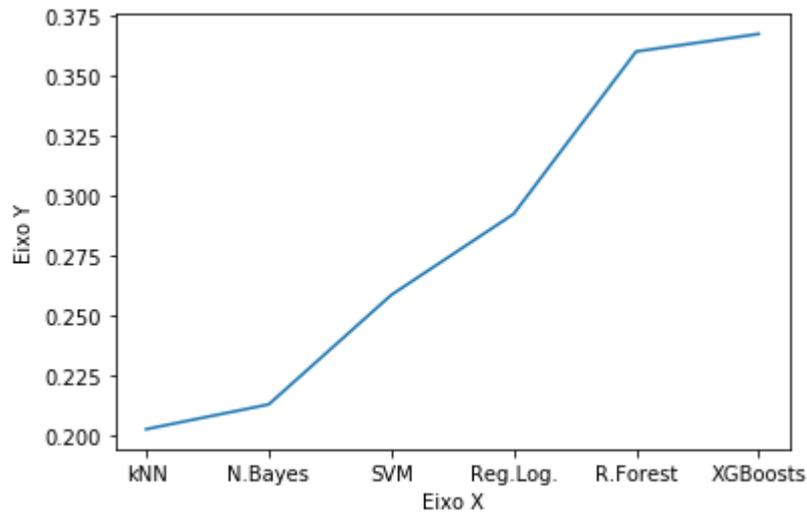


Figura 20: Modelo B x Acuracidade
Fonte: Grupo, 2020

As figuras 19 e 20 evidenciam a dificuldade de prever o tempo de permanência do paciente no leito, porque está sujeito a diversos impactos como os fenômenos qualitativos citados na seção 3.1.2.2 e até ao comportamento individual do organismo de cada um dos pacientes. Considerando também a premissa dos modelos preditivos que apresentam uma acuracidade de acordo com a volumetria de informações inseridas como entrada, ou seja, quanto mais informações são fornecidas para o modelo, maior será o seu acerto. Idealmente é necessário inserir 10000 amostras para apresentar um resultado confiável, entretanto para os modelos do caso foram inseridos 6799 registros de pacientes de cardiopatia congênita, apresentando resultados de baixa confiabilidade.

4 CONCLUSÃO

Dentre os resultados obtidos pelos modelos propostos, conclui-se que o modelo preditivo de XGBoosts apresenta uma maior acuracidade diante dos outros modelos.

Porém, os resultados apresentados por esse modelo ainda demonstram baixa acurácia devido à pouca quantidade de registros no banco de dados. Para um resultado confiável estima-se que devem ser inseridos 10.000 registros e, apesar do histórico de 10 anos do instituto, não se conseguiu alcançar esse número. A má organização das informações no banco de dados, as quais não possuem um padrão de preenchimento, é um fator que dificulta na execução dos modelos; também a falta de informações qualitativas importantes que poderiam ser agregadas aos modelos, elevando as suas precisões, como o longo tempo de espera para que familiares busquem os pacientes após alta, o prolongamento da medicação prescrita pelo médico, a individualidade do quadro de cada paciente e a resiliência dos seus respectivos organismos.

O atual cenário de pandemia teve um impacto negativo na coleta de informações, o que impediu que o desenvolvimento do trabalho fluísse conforme o esperado.

Portanto, o presente trabalho deparou com restrições de informações, o que prejudicou diretamente a construção dos modelos. Apesar dos resultados, o trabalho evidenciou que para construção de um modelo com maior acurácia é essencial a alimentação de todos os tipos de variáveis que impactariam no tempo de permanência do paciente no leito hospitalar, mostrando que é possível dar continuidade, à fim de aprimorar o sistema de armazenamento de dados do instituto e, conseqüentemente, a efetividade do modelo.

Como recomendações para trabalhos futuros, avaliar a possibilidade de aumentar o banco de dados em conjunto com outros hospitais e de alterações que permitam a padronização de todos os campos desse banco de dados. Também há a possibilidade de melhorar os resultados ao realizar a análise de conglomerados dos dados antes de rodar o *machine learning*, e testar outros modelos como o de lógica *fuzzy*.

REFERÊNCIAS

AMORIM, L.F.P. et al. **Presentation of congenital heart disease diagnosed at birth**: analysis of 29,770 newborn infants. *Jornal de pediatria*, v. 84, n. 1, 2008. p. 83-95

AZAMBUJA, C.R.C. **Importância das medidas de gestão no controle da superlotação hospitalar**. Orientador: Caroline Curry Martins. 2014. 37 f. Artigo de conclusão de curso (Especialização) - UFSM/CESNORS, Cachoeira do Sul. 2014.

BARAKAT, S.F.C. **Caracterização da demanda do Serviço de Emergências Clínicas de um hospital terciário do município de São Paulo**. 2004. Tese (Doutorado em Emergências Clínicas) - Faculdade de Medicina, Universidade de São Paulo, São Paulo, 2004. doi:10.11606/T.5.2004.tde-07112005-174138.

Disponível em:

<http://www.inicepg.univap.br/cd/INIC_2010/anais/arquivos/0246_0218_01.pdf>.

Acesso em: 06 out. 2019.

BARRETT, L.; FORD, S.; WARD-SMITH, P. **A bed management strategy for overcrowding in the emergency department**. *Nursing Economic*. Nova Jersey, v. 30, n. 2, p. 82-85, mar-abr., 2012.

BASSO, O. **Cost analysis of a system of ad hoc theatre sessions for the management of delayed trauma cases**. *J Orthopaed Traumatol*. v. 10, p. 91-96, 2009.

BITTENCOURT, R.J.; HORTALE, V.A. **Intervenções para solucionar a superlotação nos serviços de emergência hospitalar: uma revisão sistemática**.

Cad. Saúde Pública. Rio de Janeiro. v. 25, n. 7, p. 1439-1454, jul. 2009 .

Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-311X2009000700002&lng=en&nrm=iso>. Acesso em 06 out. 2019.

BORN, Daniel. **Diretriz da Sociedade Brasileira de Cardiologia para Gravidez na Mulher Portadora de Cardiopatia**. *Arq Bras Cardiol*. 2009. Disponível em:

<http://www.scielo.br/scielo.php?script=sci_issuetoc&pid=0066-782X20090013&lng=en&nrm=iso>.

Acesso em 15 out. 2019.

BOTELHO R. **Dinâmica familiar e relacionamento entre pais e filhos com malformações genéticas**. 2001. Disponível em: <<http://www.entreamigos.com.br>>

BRASIL. Ministério da Saúde. DATASUS. **Sistema de Informações de Nascidos Vivos (SINASC)**. 2017. Disponível em:

<<http://tabnet.datasus.gov.br/cgi/tabcgi.exe?sinasc/cnv/nvuf.def>>. Acesso em: 02 de nov. 2019.

BRASIL. Ministério da Saúde. DATASUS. **Sistema de Informações de Mortalidade Infantil (SINASC)**. 2017. Disponível em:

<<http://tabnet.datasus.gov.br/cgi/deftohtm.exe?sim/cnv/obt10uf.def>>. Acesso em: 26 de nov. 2019.

BRASIL. Ministério da Saúde. **Crítérios e Parâmetros para o Planejamento e Programação de Ações e Serviços de Saúde no âmbito do Sistema Único de Saúde** Secretaria de Atenção à Saúde. Departamento de Regulação, Avaliação e Controle de Sistemas. Brasília, Ministério da Saúde, 2015. Disponível em: <http://www.saude.campinas.sp.gov.br/programas/protocolos/Parametros_SUS_2015.pdf>. Acesso em: 17 de out. 2019

BRAUNWALD, E; ZIPES, DP; LIBBY P. **Tratado de medicina cardiovascular**. 9. ed. São Paulo: Editora Roca; 2013. p. 1443-1500.

BRYAN, K; GAGE, H; GILBERT, K. (2006). **Late transfers of older people from hospital: causes and policy implications**. Health Policy. Cross Ref Medline Web of Science Google Scholar, v.76. p. 194 – 201.

Chakure, A. (2019). **Random Forest and its Implementation**. Retrieved 5 December 2019. Disponível em: <<https://towardsdatascience.com/random-forest-and-its-implementation-71824ced454f>>. Acesso em 19 de out. 2019.

CORRÊA, H. L.; CORRÊA, C. A. **Administração de produção e operações: Manufatura e serviços, uma abordagem estratégica**. 3 ed. São Paula: Atlas, 2012

CÔRTEZ, S. D. C.; PORCARO, R. M.; LIFSCHITZ, S. **Mineração de Dados: Funcionalidades, Técnicas e Abordagens**, PUC - Rio Inf. MCC 10112, Rio de Janeiro. 2002.

FERNEDA, Edberto. **Redes neurais e sua aplicação em sistemas de recuperação de informação**. Ci. Inf., Brasília, v. 35, n. 1, p. 25-30, jan./abr. 2006.

FERNANDES, S. **Organização de redes regionalizadas e integradas de atenção à saúde: desafios do Sistema Único de Saúde (Brasil)**. Disponível em: <<https://www.scielo.org/pdf/csc/2011.v16n6/2753-2762/pt>>. Acesso em 03 nov. 2019.

HAN, J.; KAMBER, M. **Data mining: Concepts and techniques**, Morgan Kaufmann Publishers Inc., Burlington. 2005.

HAYKIN, Simon S. **Redes Neurais: Princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001.

Hebb, D. O., **The Organization of Behavior**. Wiley: New York, 1949.

JUSTOCANTINS. **TCU apresenta diagnóstico inédito sobre a saúde no Brasil**. Disponível em: www.justocantins.com.br/noticias-do-brasil-22131-tcu-apresenta-diagnostico-inedito-sobre-a-saude-no-brasil.html. Acesso em 16 out. 2019

KOVÁCS, Zsolt L. **Redes neurais artificiais: fundamentos e aplicações**. 4. ed. São Paulo: Editora Livraria da Física, 2006.

LITTIG, S.J.; ISKEN, M.W. **Short term hospital occupancy prediction**. Health Care Manage Sci. v. 10, p. 47-66, 2007.

LOPES, Antônio A.; MESQUITA, Sonia. **Comunicação Interatrial em Adultos: A Correção Sempre Cura?**. Arq Bras Cardiol. 2014. Disponível em: <http://www.scielo.br/scielo.php?pid=S0066-782X2014002400001&script=sci_arttext&tlng=pt>. Acesso em 17 out. 2019.

MAIMON, O.; ROKACH, L. **Data mining and knowledge discovery handbook**. Springer, 2005. v. 2.

MCCULLOCH, W. S.; PITTS, W. H. **A logical calculus of the ideas immanent in nervous activity**. Bulletin of Mathematical Biophysics, n. 5, p.115-133, 1943

MITCHELL, T. M. **Machine Learning**. Singapore: McGraw-Hill, 1997.

MITCHELL, SC; KORONES, SB; BERENDES, HW. **Congenital heart disease in 56,109 births: Incidence and natural history**. 1971. Volume 43. Disponível em: <<https://www.ahajournals.org/doi/abs/10.1161/01.cir.43.3.323>>. Acesso em 15 out. 2019.

MONTANA, David J.; LAWRENCE, Davis. **Training Feedforward Neural Networks Using Genetic Algorithms**. BBN Systems and Technologies Corp. Cambridge, MA. 1989.

PORTO FILHO, Carlos Humberto. **Técnicas de aprendizado não supervisionado baseadas no algoritmo da caminhada do turista**. 2017. Dissertação (Mestrado em Bioengenharia) – Bioengenharia, Universidade de São Paulo, São carlos, 2017. Disponível em: <https://teses.usp.br/teses/disponiveis/82/82131/tde-20082018-122603/publico/Dissert_CarlosPortoFilho_corrigida.pdf>. Acesso em: 02 novembro 2019.

RIBEIRO, Amanda Carvalheira et al. **Comparação da eficiência de modelos de previsão de demanda aplicados a uma indústria alimentícia**. São Bernardo do Campo, 2019. 88 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Produção) - Centro Universitário da FEI, São Bernardo do Campo, 2019

RITZMAN, Larry P; KRAJEWSKI, Lee J. **Administração da produção e operações**. São Paulo: Prentice Hall, 2004.

ROCCO, J. R.; SOARES, M.; GAGO, M.F. **Pacientes Clínicos Referenciados, mas não Internados na Unidade de Terapia Intensiva: Prevalência, Características Clínicas e Prognóstico**. 2006. Disponível em: <https://www.researchgate.net/publication/250053646_Pacientes_clinicos_referencia_dos_mas_nao_internados_na_Unidade_de_Terapia_Intensiva_prevalencia_caracteristicas_clinicas_e_prognostico>. Acesso em: 7 nov. 2019.

ROSENBLATT, M. **The Perceptron: A probabilistic model for information storage and organization in the Brain**. Psychological review, v.65, n.6, p. 386-408, 1958.

SANTANA, M. **Cardiopatas Adquiridas no Recém-Nascido**. São Paulo: Atheneu; 2000.

SANTOS, Rodrigo Mota dos. **Técnicas de Aprendizagem de Máquina Utilizadas na Previsão de Desempenho Acadêmico**. 2016. f. 27-37. Dissertação (Pós-Graduação em Educação) – Universidade Federal dos Vales do Jequitinhonha e Mucuri, Diamantina, 2016. Disponível em: <http://acervo.ufvjm.edu.br/jspui/bitstream/1/1327/1/rodrigo_magalhaes_mota_santos.pdf>. Acesso em: 28 outubro 2019.

SILVA, I.N., SPATTI D., e FLAUZINO, R. **Redes neurais artificiais para engenharia e ciências aplicadas: Fundamentos teóricos e aspectos práticos**. 2016. Artliber Editora Ltda, São Paulo, SP, Brasil. Disponível em: <https://artliber.com.br/amostra/redes_neurais.pdf>. Acesso em: 15 nov. 2019.

SIQUEIRA-BATISTA, Rodrigo et al. **As redes neurais artificiais e o ensino da medicina**. Revista Brasileira de Educação Médica. 2014. Disponível em: <<http://www.scielo.br/pdf/rbem/v38n4/17.pdf>>. Acesso em: 28 out. 2019.

TARABOULSI, Fadi Antoine. **Administração de hotelaria hospitalar: serviços aos clientes, humanização do atendimento, departamentalização, gerenciamento, saúde e turismo, hospitalidade, tecnologia de informação**. 4. ed. São Paulo: Atlas, 2009. p. 165.

TRIEDMAN, JK; NEWBURGER, JW. **Trends in congenital heart disease: the next decade**. 2016. Disponível em: <<https://www.ahajournals.org/doi/10.1161/CIRCULATIONAHA.116.023544>>. Acesso em 15 out. 2019.

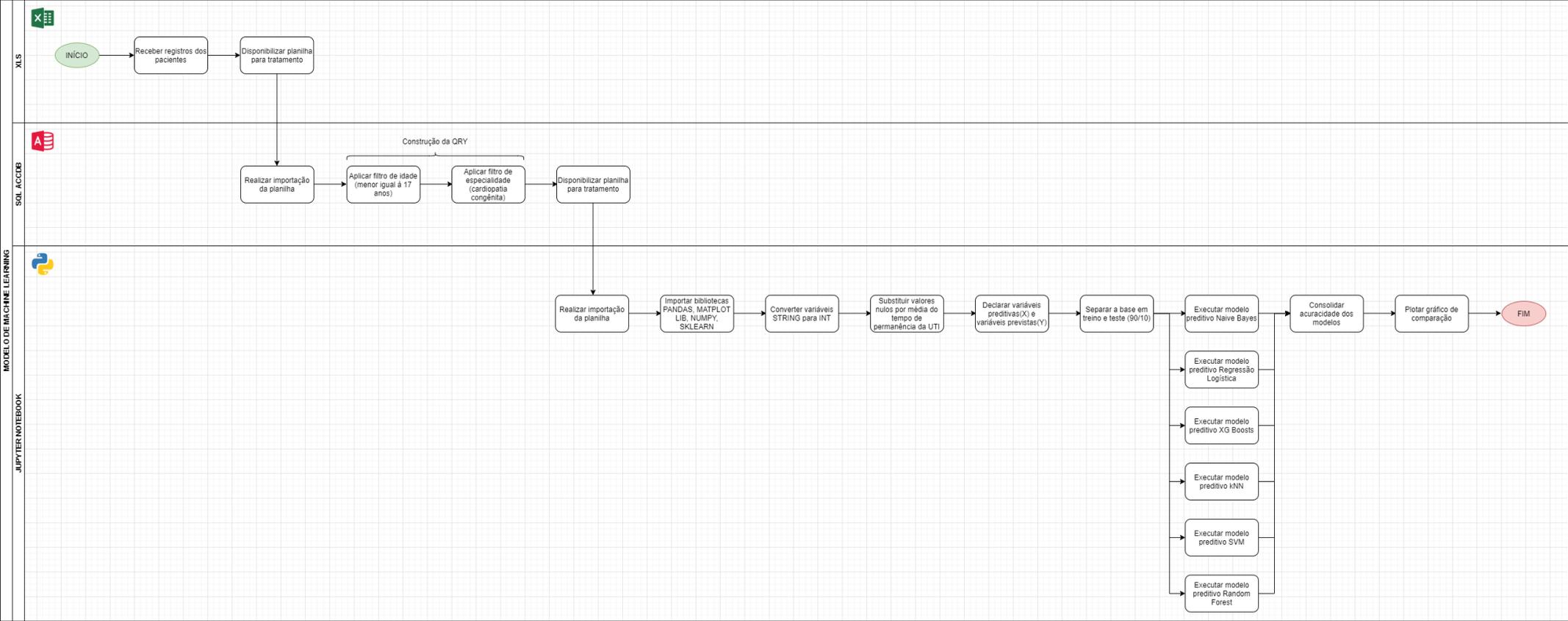
WANG, Sun-Chong. **Interdisciplinary Computing in Java Programming Language**. Kluwer Academic Publishers. Springer, 2003.

WIDROW, B.; HOFF, M.E. **Adaptive switching circuits**. Ire Wescon Conv. 1960. Rec. 4: 96-104.

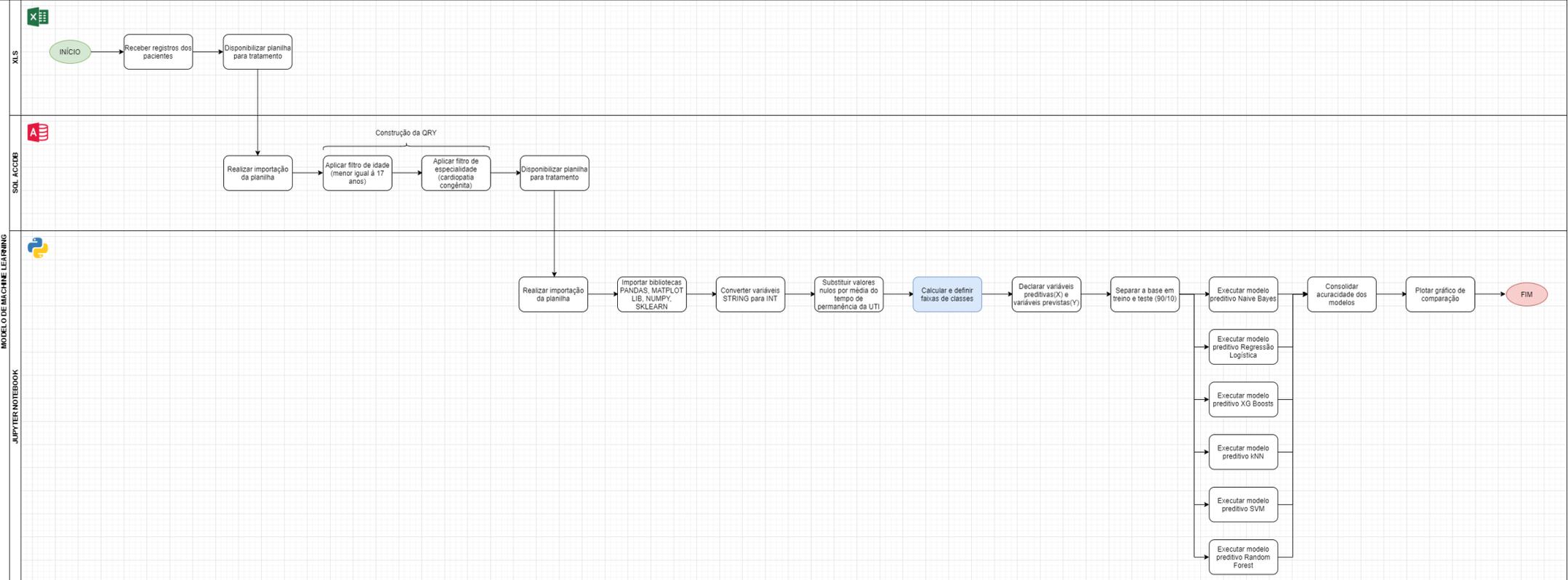
W. S. McCulloch and W. Pitts. **A logical calculus of the ideas immanent in nervous activity**. The bulletin of mathematical biophysics, 5(4):115–133, 1943.

YAAKOV, B. et al. **Major congenital malformations and residential proximity to a regional industrial park including a national toxic waste site: An ecological study**. Environmental Health: A Global Access Science Source V. 5, p. 8-15, 2006.

APÊNDICE A – DIAGRAMA DE BLOCOS A



APÊNDICE B – DIAGRAMA DE BLOCOS B



APÊNDICE C – PROGRAMAÇÃO EM LINGUAGEM PYTHON

```
#IMPORTAÇÃO

import pandas as pd
import matplotlib as mat
import matplotlib.pyplot as plt
import numpy as np
import sklearn as sk
from sklearn.model_selection import train_test_split
import math
from math import nan

#LEITURA BASE XLSX

base = pd.read_excel("QRY_BASE_INCORfaixas.xlsx")

#EXCLUINDO COLUNAS DESNECESSÁRIAS

base.drop('PACIENTE', inplace=True, axis=1)
base.drop('NASCIMENTO', inplace=True, axis=1)
base.drop('IDADE_ATUAL', inplace=True, axis=1)
base.drop('ANTERIOR', inplace=True, axis=1)
base.drop('PROGRAMACAO', inplace=True, axis=1)
base.drop('MES', inplace=True, axis=1)
base.drop('ADMISSAO', inplace=True, axis=1)
base.drop('INTERNACAO', inplace=True, axis=1)
base.drop('CIRURGIA', inplace=True, axis=1)
base.drop('SAIDA', inplace=True, axis=1)
base.drop('SAID_TP', inplace=True, axis=1)
base.drop('PROVEDOR', inplace=True, axis=1)
base.drop('CONVENIO', inplace=True, axis=1)
base.drop('PRE_OPER', inplace=True, axis=1)
base.drop('PERM_POS_UTI', inplace=True, axis=1)
base.drop('PERM_POS_ENF', inplace=True, axis=1)
base.drop('PERM_HOSP', inplace=True, axis=1)
base.drop('UTIC2', inplace=True, axis=1)
base.drop('UCO4', inplace=True, axis=1)
base.drop('UTIG4', inplace=True, axis=1)
base.drop('URES', inplace=True, axis=1)
base.drop('ESPECIALIDADE', inplace=True, axis=1)
base.drop('ANESTESISTA', inplace=True, axis=1)
base.drop('ENFERMEIRO', inplace=True, axis=1)
base.drop('SALA', inplace=True, axis=1)
base.drop('CEC', inplace=True, axis=1)
base.drop('GRUPO', inplace=True, axis=1)
base.drop('ENFE_DT_ENTR_CC', inplace=True, axis=1)
base.drop('ENFE_DT_ENTR_SC', inplace=True, axis=1)
base.drop('ENFE_DT_INI_REAL', inplace=True, axis=1)
base.drop('ENFE_DT_FIM_REAL', inplace=True, axis=1)
base.drop('ENFE_DT_SAI_SC', inplace=True, axis=1)
base.drop('ENFE_DT_SAI_CC', inplace=True, axis=1)
base.drop('FISO_IN_OBITO_SALA', inplace=True, axis=1)
base.drop('DESTINO', inplace=True, axis=1)
base.drop('UNIDADE', inplace=True, axis=1)
base.drop('CENTRO', inplace=True, axis=1)
base.drop('PROCED_SEC1', inplace=True, axis=1)

#CONVERSÃO STR>FLOAT
```

```
sexo = ["SEXO"]
col1 = base.filter(items=sexo)
v1 = col1.drop_duplicates()
v1 = v1.dropna()

a1 = v1.columns[0]
b1 = v1[a1].tolist()

lista1 = []
i = 1

for item in b1:
    lista1.append(i)
    i += 1

v1.insert(1,"SEXO_NUM",lista1,True)

idadecirurgia = ["IDADE_CIRURGIA"]
col2 = base.filter(items=idadecirurgia)
v2 = col2.drop_duplicates()
v2 = v2.dropna()

a2 = v2.columns[0]
b2 = v2[a2].tolist()

lista2 = []
i = 1

for item in b2:
    lista2.append(i)
    i += 1

v2.insert(1,"IDADE_NUM",lista2,True)

faixaetaria = ["FAIXA_ETARIA"]
col3 = base.filter(items=faixaetaria)
v3 = col3.drop_duplicates()
v3 = v3.dropna()

a3 = v3.columns[0]
b3 = v3[a3].tolist()

lista3 = []
i = 1

for item in b3:
    lista3.append(i)
    i += 1

v3.insert(1,"FAIXAETARIA_NUM",lista3,True)

solicitacao = ["SOLICITACAO"]
col5 = base.filter(items=solicitacao)
v5 = col5.drop_duplicates()
v5 = v5.dropna()

a5 = v5.columns[0]
b5 = v5[a5].tolist()
```

```
lista5 = []
i = 1

for item in b5:
    lista5.append(i)
    i += 1

v5.insert(1,"SOLICITACAO_NUM",lista5,True)

indicacao = ["INDICACAO"]
col6 = base.filter(items=indicacao)
v6 = col6.drop_duplicates()
v6 = v6.dropna()

a6 = v6.columns[0]
b6 = v6[a6].tolist()

lista6 = []
i = 1

for item in b6:
    lista6.append(i)
    i += 1

v6.insert(1,"INDICACAO_NUM",lista6,True)

cirurgiao = ["CIRURGIAO"]
col7 = base.filter(items=cirurgiao)
v7 = col7.drop_duplicates()
v7 = v7.dropna()

a7 = v7.columns[0]
b7 = v7[a7].tolist()

lista7 = []
i = 1

for item in b7:
    lista7.append(i)
    i += 1

v7.insert(1,"CIRURGIAO_NUM",lista7,True)

planilha = ["PLANILHA"]
col8 = base.filter(items=planilha)
v8 = col8.drop_duplicates()
v8 = v8.dropna()

a8 = v8.columns[0]
b8 = v8[a8].tolist()

lista8 = []
i = 1

for item in b8:
    lista8.append(i)
    i += 1
```

```
v8.insert(1,"PLANILHA_NUM",lista8,True)

procedimentoprincipal = ["PROCED_PRINC"]
col9 = base.filter(items=procedimentoprincipal)
v9 = col9.drop_duplicates()
v9 = v9.dropna()

a9 = v9.columns[0]
b9 = v9[a9].tolist()

lista9 = []
i = 1

for item in b9:
    lista9.append(i)
    i += 1

v9.insert(1,"PROCEDIMENTOPRIN_NUM",lista9,True)

procedsec2 = ["PROCED_SEC2"]
col10 = base.filter(items=procedsec2)
v10 = col10.drop_duplicates()
v10 = v10.dropna()

a10 = v10.columns[0]
b10 = v10[a10].tolist()

lista10 = []
i = 1

for item in b10:
    lista10.append(i)
    i += 1

v10.insert(1,"PROCEDSEC2_NUM",lista10,True)

procedsec3 = ["PROCED_SEC3"]
col11 = base.filter(items=procedsec3)
v11 = col11.drop_duplicates()
v11 = v11.dropna()

a11 = v11.columns[0]
b11 = v11[a11].tolist()

lista11 = []
i = 1

for item in b11:
    lista11.append(i)
    i += 1

v11.insert(1,"PROCEDSEC3_NUM",lista11,True)

procedsec4 = ["PROCED_SEC4"]
col12 = base.filter(items=procedsec4)
v12 = col12.drop_duplicates()
v12 = v12.dropna()

a12 = v12.columns[0]
```

```
b12 = v12[a12].tolist()

lista12 = []
i = 1

for item in b12:
    lista12.append(i)
    i += 1

v12.insert(1,"PROCEDSEC4_NUM",lista12,True)

procedimentos = ["PROCEDIMENTOS"]
col13 = base.filter(items=procedimentos)
v13 = col13.drop_duplicates()
v13 = v13.dropna()

a13 = v13.columns[0]
b13 = v13[a13].tolist()

lista13 = []
i = 1

for item in b13:
    lista13.append(i)
    i += 1

v13.insert(1,"PROCEDIMENTOS_NUM",lista13,True)

resumodiagpri = ["RESUMO_DIAG_PRINCIPAL"]
col14 = base.filter(items=resumodiagpri)
v14 = col14.drop_duplicates()
v14 = v14.dropna()

a14 = v14.columns[0]
b14 = v14[a14].tolist()

lista14 = []
i = 1

for item in b14:
    lista14.append(i)
    i += 1

v14.insert(1,"RESUMODIAGPRINC_NUM",lista14,True)

resumodiagsec1 = ["RESUMO_DIAG_SECUNDARIO_1"]
col15 = base.filter(items=resumodiagsec1)
v15 = col15.drop_duplicates()
v15 = v15.dropna()

a15 = v15.columns[0]
b15 = v15[a15].tolist()

lista15 = []
i = 1

for item in b15:
    lista15.append(i)
    i += 1
```

```
v15.insert(1,"RESUMODIAGSEC1_NUM",lista15,True)

resumodiagsec2 = ["RESUMO_DIAG_SECUNDARIO_2"]
col16 = base.filter(items=resumodiagsec2)
v16 = col16.drop_duplicates()
v16 = v16.dropna()

a16 = v16.columns[0]
b16 = v16[a16].tolist()

lista16 = []
i = 1

for item in b16:
    lista16.append(i)
    i += 1

v16.insert(1,"RESUMODIAGSEC2_NUM",lista16,True)

resumodiag = ["RESUMO_DIAGNOSTICOS"]
col17 = base.filter(items=resumodiag)
v17 = col17.drop_duplicates()
v17 = v17.dropna()

a17 = v17.columns[0]
b17 = v17[a17].tolist()

lista17 = []
i = 1

for item in b17:
    lista17.append(i)
    i += 1

v17.insert(1,"RESUMODIAG_NUM",lista17,True)

infeccao = ["INFECCAO"]
col18 = base.filter(items=infeccao)
v18 = col18.drop_duplicates()
v18 = v18.dropna()

a18 = v18.columns[0]
b18 = v18[a18].tolist()

lista18 = []
i = 1

for item in b18:
    lista18.append(i)
    i += 1

v18.insert(1,"INFECCAO_NUM",lista18,True)

infeccoes = ["INFECCOES"]
col19 = base.filter(items=infeccoes)
v19 = col19.drop_duplicates()
v19 = v19.dropna()
```

```

a19 = v19.columns[0]
b19 = v19[a19].tolist()

lista19 = []
i = 1

for item in b19:
    lista19.append(i)
    i += 1

v19.insert(1,"INFECCOES_NUM",lista19,True)

cirurgiadiagpri = ["CIRURGIA_DIAG_PRINCIPAL"]
col20 = base.filter(items=cirurgiadiagpri)
v20 = col20.drop_duplicates()
v20 = v20.dropna()

a20 = v20.columns[0]
b20 = v20[a20].tolist()

lista20 = []
i = 1

for item in b20:
    lista20.append(i)
    i += 1

v20.insert(1,"CIRURGIADIAPRINC_NUM",lista20,True)

cirurgiadiagsec = ["CIRURGIA_DIAG_SECUNDARIOS"]
col21 = base.filter(items=cirurgiadiagsec)
v21 = col21.drop_duplicates()
v21 = v21.dropna()

a21 = v21.columns[0]
b21 = v21[a21].tolist()

lista21 = []
i = 1

for item in b21:
    lista21.append(i)
    i += 1

v21.insert(1,"CIRURGIADIAGSEC_NUM",lista21,True)

BASE_MODELO1 = pd.merge(base, v1, on=['SEXO'], how='left')
BASE_MODELO2 = pd.merge(BASE_MODELO1, v2, on=['IDADE_CIRURGIA'], how='left')
BASE_MODELO3 = pd.merge(BASE_MODELO2, v3, on=['FAIXA_ETARIA'], how='left')
BASE_MODELO4 = pd.merge(BASE_MODELO3, v5, on=['SOLICITACAO'], how='left')
BASE_MODELO5 = pd.merge(BASE_MODELO4, v6, on=['INDICACAO'], how='left')
BASE_MODELO6 = pd.merge(BASE_MODELO5, v7, on=['CIRURGIAO'], how='left')
BASE_MODELO7 = pd.merge(BASE_MODELO6, v8, on=['PLANILHA'], how='left')
BASE_MODELO8 = pd.merge(BASE_MODELO7, v9, on=['PROCED_PRINC'], how='left')
BASE_MODELO9 = pd.merge(BASE_MODELO8, v10, on=['PROCED_SEC2'], how='left')
BASE_MODELO10 = pd.merge(BASE_MODELO9, v11, on=['PROCED_SEC3'], how='left')
BASE_MODELO11 = pd.merge(BASE_MODELO10, v12, on=['PROCED_SEC4'], how='left')
BASE_MODELO12 = pd.merge(BASE_MODELO11, v13, on=['PROCEDIMENTOS'], how='left')

```

```

BASE_MODELO13 = pd.merge(BASE_MODELO12, v14, on=['RESUMO_DIAG_PRINCIPAL'], how='left')
BASE_MODELO14 = pd.merge(BASE_MODELO13, v15, on=['RESUMO_DIAG_SECUNDARIO_1'], how='left')
BASE_MODELO15 = pd.merge(BASE_MODELO14, v16, on=['RESUMO_DIAG_SECUNDARIO_2'], how='left')
BASE_MODELO16 = pd.merge(BASE_MODELO15, v17, on=['RESUMO_DIAGNOSTICOS'], how='left')
BASE_MODELO17 = pd.merge(BASE_MODELO16, v18, on=['INFECCAO'], how='left')
BASE_MODELO18 = pd.merge(BASE_MODELO17, v19, on=['INFECCOES'], how='left')
BASE_MODELO19 = pd.merge(BASE_MODELO18, v20, on=['CIRURGIA_DIAG_PRINCIPAL'], how='left')
BASE_MODELO = pd.merge(BASE_MODELO19, v21, on=['CIRURGIA_DIAG_SECUNDARIOS'], how='left')

#TRATANDO DADOS NULOS

BASE_MODELO.isnull().sum()

BASE_MODELO.update(BASE_MODELO['PROCEDESEC2_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['PROCEDESEC3_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['PROCEDESEC4_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['PROCEDIMENTOS_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['RESUMODIAGPRINC_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['RESUMODIAGSEC1_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['RESUMODIAGSEC2_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['RESUMODIAG_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['INFECCAO_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['INFECCOES_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['CIRURGIADIAPRINC_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['SOLICITACAO_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['CIRURGIAO_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['PLANILHA_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['PROCEDIMENTOPRIN_NUM'].fillna(0))
BASE_MODELO.update(BASE_MODELO['CIRURGIADIAGSEC_NUM'].fillna(0))

BASE_MODELO['PROCEDESEC2_NUM'] = BASE_MODELO['PROCEDESEC2_NUM'].astype("int64")
BASE_MODELO['PROCEDESEC3_NUM'] = BASE_MODELO['PROCEDESEC3_NUM'].astype("int64")
BASE_MODELO['PROCEDESEC4_NUM'] = BASE_MODELO['PROCEDESEC4_NUM'].astype("int64")
BASE_MODELO['RESUMODIAGPRINC_NUM'] =
BASE_MODELO['RESUMODIAGPRINC_NUM'].astype("int64")
BASE_MODELO['RESUMODIAGSEC1_NUM'] = BASE_MODELO['RESUMODIAGSEC1_NUM'].astype("int64")
BASE_MODELO['RESUMODIAGSEC2_NUM'] = BASE_MODELO['RESUMODIAGSEC2_NUM'].astype("int64")
BASE_MODELO['RESUMODIAG_NUM'] = BASE_MODELO['RESUMODIAG_NUM'].astype("int64")
BASE_MODELO['INFECCAO_NUM'] = BASE_MODELO['INFECCAO_NUM'].astype("int64")
BASE_MODELO['INFECCOES_NUM'] = BASE_MODELO['INFECCOES_NUM'].astype("int64")
BASE_MODELO['CIRURGIADIAPRINC_NUM'] =
BASE_MODELO['CIRURGIADIAPRINC_NUM'].astype("int64")
BASE_MODELO['CIRURGIADIAGSEC_NUM'] = BASE_MODELO['CIRURGIADIAGSEC_NUM'].astype("int64")
BASE_MODELO['SOLICITACAO_NUM'] = BASE_MODELO['SOLICITACAO_NUM'].astype("int64")
BASE_MODELO['CIRURGIAO_NUM'] = BASE_MODELO['CIRURGIAO_NUM'].astype("int64")
BASE_MODELO['PLANILHA_NUM'] = BASE_MODELO['PLANILHA_NUM'].astype("int64")
BASE_MODELO['PROCEDIMENTOPRIN_NUM'] =
BASE_MODELO['PROCEDIMENTOPRIN_NUM'].astype("int64")
BASE_MODELO['PROCEDIMENTOS_NUM'] = BASE_MODELO['PROCEDIMENTOS_NUM'].astype("int64")
BASE_MODELO['UTIC1'] = BASE_MODELO['UTIC1'].astype("int64")
BASE_MODELO['UNEO'] = BASE_MODELO['UNEO'].astype("int64")
BASE_MODELO['UIG51'] = BASE_MODELO['UIG51'].astype("int64")
BASE_MODELO['FAIXADIAS_NUM'] = BASE_MODELO['FAIXADIAS_NUM'].astype("int64")

BASE_MODELO.dtypes

#CONSTRUINDO E TREINANDO O MODELO

```

```

# Seleção de variáveis preditoras (Feature Selection)
atributos = ['SEXO_NUM',
'IDADE_NUM','FAIXAETARIA_NUM','SOLICITACAO_NUM','INDICACAO_NUM','CIRURGIAO_NUM','PLANILH
A_NUM','PROCEDIMENTOPRIN_NUM','PROCEDSEC2_NUM','PROCEDSEC3_NUM','PROCEDSEC4_NUM','PR
OCEDIMENTOS_NUM','RESUMODIAGPRINC_NUM','RESUMODIAGSEC1_NUM','RESUMODIAGSEC2_NUM','R
ESUMODIAG_NUM','INFECCAO_NUM','INFECCOES_NUM','CIRURGIADIAPRINC_NUM','CIRURGIADIAGSEC_
NUM']

BASE_MODELO.isnull().sum()

# Variável a ser prevista
atrib_prev = ['FAIXADIAS_NUM']

# Criando objetos
X = BASE_MODELO[atributos].values
Y = BASE_MODELO[atrib_prev].values

# Definindo a taxa de split
split_test_size = 0.10

# Criando dados de treino e de teste
X_treino, X_teste, Y_treino, Y_teste = train_test_split(X, Y, test_size = split_test_size, random_state = 42)

# Imprimindo os resultados
print("{0:0.2f}% nos dados de treino".format((len(X_treino)/len(BASE_MODELO.index)) * 100))
print("{0:0.2f}% nos dados de teste".format((len(X_teste)/len(BASE_MODELO.index)) * 100))

#MODELO NAIVE BAYES

# Utilizando um classificador Naive Bayes
from sklearn.naive_bayes import GaussianNB

# Criando o modelo preditivo
modelo_v1 = GaussianNB()

# Treinando o modelo
modelo_v1.fit(X_treino, Y_treino.ravel())

from sklearn import metrics

nb_predict_train = modelo_v1.predict(X_treino)

print("Exatidão (Accuracy): {0:.4f}".format(metrics.accuracy_score(Y_treino, nb_predict_train)))
print()

nb_predict_test = modelo_v1.predict(X_teste)

print("Exatidão (Accuracy): {0:.4f}".format(metrics.accuracy_score(Y_teste, nb_predict_test)))
print()

#MODELO FLORESTA ALEATÓRIA</font>

from sklearn.ensemble import RandomForestClassifier

modelo_v2 = RandomForestClassifier(random_state = 42)
modelo_v2.fit(X_treino, Y_treino.ravel())

```

```

# Verificando os dados de treino
rf_predict_train = modelo_v2.predict(X_treino)
print("Exatidão (Accuracy): {0:.4f}".format(metrics.accuracy_score(Y_treino, rf_predict_train)))

# Verificando nos dados de teste
rf_predict_test = modelo_v2.predict(X_teste)
print("Exatidão (Accuracy): {0:.4f}".format(metrics.accuracy_score(Y_teste, rf_predict_test)))
print()

#REGRESSÃO LOGÍSTICA

from sklearn.linear_model import LogisticRegression

# Terceira versão do modelo usando Regressão Logística
modelo_v3 = LogisticRegression(C = 0.7, random_state = 42, max_iter = 1000)
modelo_v3.fit(X_treino, Y_treino.ravel())
lr_predict_test = modelo_v3.predict(X_teste)

print("Exatidão (Accuracy): {0:.4f}".format(metrics.accuracy_score(Y_teste, lr_predict_test)))
print()
print("Classification Report")
print(metrics.classification_report(Y_teste, lr_predict_test, labels = [1, 0]))

#MODELO SVM

from sklearn import svm

modelo_v4 = svm.SVC(kernel='rbf')

modelo_v4.fit(X_treino, Y_treino)
modelo_v4.score(X_treino, Y_treino)
#Predict Output
predicted= modelo_v4.predict(X_teste)

print("Exatidão (Accuracy): {0:.4f}".format(metrics.accuracy_score(Y_teste, predicted)))
print()

#MODELO ÁRVORE DE DECISÃO

from sklearn import tree

modelo_v5 = tree.DecisionTreeRegressor()

modelo_v5.fit(X_treino, Y_treino)
modelo_v5.score(X_treino, Y_treino)
#Predict Output
predicted= modelo_v5.predict(X_teste)

print("Exatidão (Accuracy): {0:.4f}".format(metrics.accuracy_score(Y_teste, predicted)))
print()

#MODELO kNN (k- vizinhos mais próximos)

from sklearn.neighbors import KNeighborsClassifier

```

```

modelo_v6 = KNeighborsClassifier(n_neighbors=6)

modelo_v6.fit(X_treino, Y_treino)
#Predict Output
predicted= modelo_v6.predict(X_teste)

print("Exatidão (Accuracy): {0:.4f}".format(metrics.accuracy_score(Y_teste, predicted)))
print()

#MODELO XGBoosts

from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
modelo_v7 = XGBClassifier()

modelo_v7.fit(X_treino, Y_treino)

y_pred = modelo_v7.predict(X_teste)
print("Exatidão (Accuracy): {0:.4f}".format(metrics.accuracy_score(Y_teste, y_pred)))
print()

#REALIZANDO PREVISÕES COM MODELO TREINADO

import pickle

# Salvando o modelo para usar mais tarde
filename = 'modelo_treinado_v7.sav'
pickle.dump(modelo_v2, open(filename, 'wb'))
X_teste

### Carregando o modelo e fazendo previsão com novos conjuntos de dados
# (X_teste, Y_teste devem ser novos conjuntos de dados preparados com o procedimento de limpeza e
transformação adequados)
loaded_model = pickle.load(open(filename, 'rb'))
resultado1 = loaded_model.predict(X_teste[1].reshape(1, -1))
resultado2 = loaded_model.predict(X_teste[6].reshape(1, -1))
print(resultado1)
print(resultado2)

```