UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

VINÍCIUS DE PAULA PILAN

TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA DIAGNÓSTICO DE ACIDENTE VASCULAR CEREBRAL ATRAVÉS DE IMAGENS E DADOS TEXTUAIS SOBRE POSSÍVEIS VÍTIMAS

BAURU
Janeiro/2023

VINÍCIUS DE PAULA PILAN

TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA DIAGNÓSTICO DE ACIDENTE VASCULAR CEREBRAL ATRAVÉS DE IMAGENS E DADOS TEXTUAIS SOBRE POSSÍVEIS VÍTIMAS

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.
Orientador: Prof. Dr. Clayton Reginaldo Pereira

BAURU
Janeiro/2023

Vinícius de Paula Pilan TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA DIAGNÓSTICO DE ACIDENTE VASCULAR CEREBRAL ATRAVÉS DE IMAGENS E DADOS TEXTUAIS SOBRE POSSÍVEIS VÍTIMAS/ Vinícius de Paula Pilan. – Bauru, Janeiro/2023- 56 p. : il. (algumas color.) ; 30 cm.
Orientador: Prof. Dr. Clayton Reginaldo Pereira
Trabalho de Conclusão de Curso – Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Ciência da Computação, Janeiro/2023.

Vinícius de Paula Pilan

TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA DIAGNÓSTICO DE ACIDENTE VASCULAR CEREBRAL ATRAVÉS DE IMAGENS E DADOS TEXTUAIS SOBRE POSSÍVEIS VÍTIMAS

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof. Dr. Clayton Reginaldo Pereira
Orientador
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Computação

Prof. Dra. Simone Prado
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Computação

Prof. Dr. Kelton Augusto Pontara da Costa
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Computação

Bauru, _____ de _____ de _____.

Agradecimentos

A Deus por todas as oportunidades, experiências e conquistas que vivi até o presente momento em minha vida, que me fizeram ser quem sou. À minha família, que sempre me apoiou e esteve comigo em minhas decisões. Ao professor Clayton, que me orientou para que este trabalho pudesse ser realizado. Às minhas amizades que fiz durante a graudação e aos demais professores, colegas e todos que, de alguma forma, contribuiram para meu desenvolvimento pessoal e profissional nessa ótima jornada de aprendizado.

Resumo

O Acidente Vascular Cerebral (AVC) é uma das doenças que mais matam e incapacitam no mundo todo, e quanto mais tardio é o seu diagnóstico, maiores podem ser os prejuízos para a vítima. Portanto, formas de agilizar e auxiliar o processo de diagnóstico da doença podem ser bastante relevantes e benéficas. Este trabalho abordou essa questão, aplicando técnicas de Aprendizado de Máquina e Aprendizagem profunda, sendo ambas, subáreas da Inteligência Artificial, que vem sendo amplamente aplicada em diversos segmentos na busca de otimizar tarefas, principalmente no diagnóstico de doenças. Modelos classificadores de fatores de risco foram criados a partir dos algoritmos de Regressão Logística e Floresta Aleatória, assim como uma Rede Neural Convolucional (CNN) para classificação de imagens de tomografia computadorizada da região cerebral.

Palavras-chave: AVC, Acidente Vascular Cerebral, Aprendizado de Máquina, Aprendizagem Profunda, Inteligência Artificial, classificação, CNN, Redes Neurais Convolucionais.

Abstract

Stroke (AVC) is one of the most deadly and disabling diseases worldwide, and a later diagnostic can be more harmful to the victim. Therefore, ways to speed up and help the process of diagnosing the disease can be very relevant and beneficial, and this work addressed this issue, applying Machine Learning and Deep Learning techniques, both of which are subareas of Artificial Intelligence, which has been widely applied in several segments in the search to optimize tasks, mainly in the diagnosis of diseases. Risk factor classification models were created based on Logistic Regression and Random Forest algorithms, as well as a Convolutional Neural Network (CNN) for classification of computed tomography images of the brain region.

Keywords: AVC, Stroke, Machine Learning, Deep Learning, Artificial intelligence, classification, CNN, Convolutional Neural Networks.

Lista de figuras

Lista de tabelas

Lista de abreviaturas e siglas

AVC	Acidente Vascular Cerebral.
ACVh	Acidente Vascular Cerebral Hemorrágico.
ACVi	Acidente Vascular Cerebral Isquêmico.
TC	Tomografia computadorizada.
AUC	*Area Under Curve*.
ANN	*Artificial Neural Network*.
CNN	*Convolutional Neural Network*.
IA	Inteligência Artificial.
IQR	*Interquartile range*.
FN	*False negative*.
TN	*True negative*.
FP	*False positive*.
TP	*True positive*.
FPR	*False positive rate*.
TPR	*True positive rate*.
ROC	*Receiver Operating Characteristic*.

Sumário

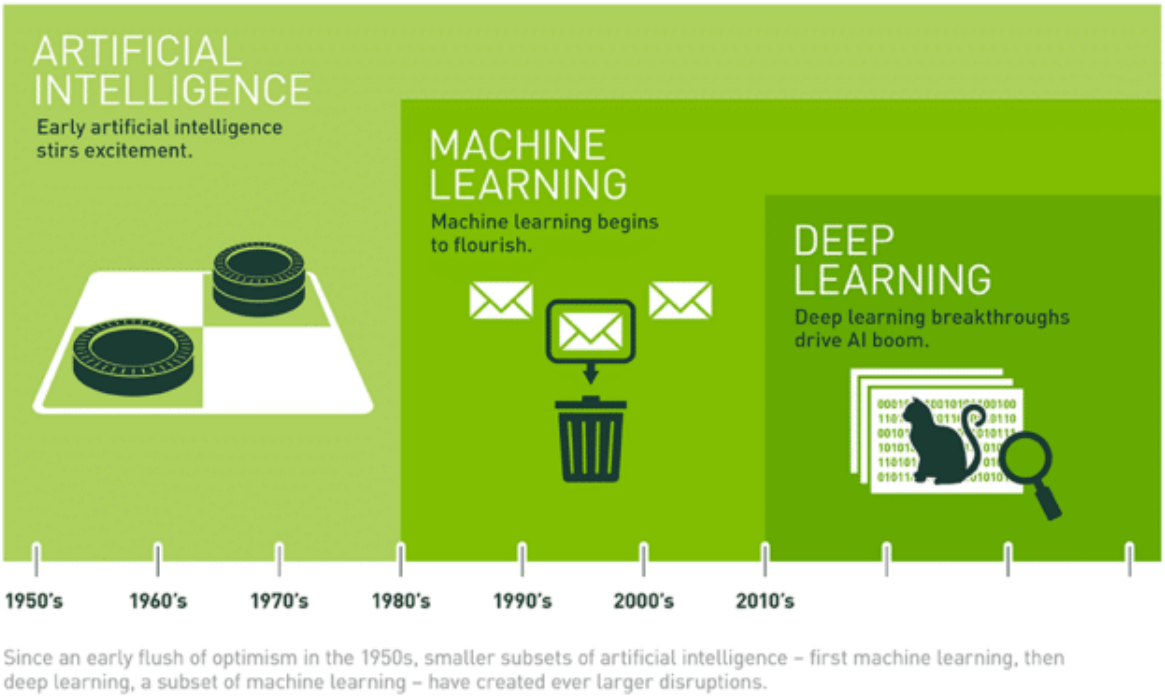
1 Introdução

O Acidente Vascular Cerebral, também chamado de derrame cerebral ou simplesmente AVC, é uma das doenças que mais causa mortes, incapacitações e internações no mundo (OLIVEIRA; ANDRADE, 2001). A doença ocorre em virtude da alteração do fluxo de sangue que chega ao cérebro. Quando acontece uma interrupção ou diminuição drástica do suprimento sanguíneo que chega na região cerebral, células nervosas dessa região são atingidas e ficam sem o abastecimento necessário de oxigênio e de nutrientes, o que as levam à morte. Tal alteração do fluxo sanguíneo pode ser causada por obstrução ou ruptura de vasos sanguíneos da região (BVS, 2015).

Quanto mais tardio é realizado o diagnóstico e tratamento, maiores são os prejuízos e sequelas, já que as células vão morrendo pela falta de abastecimento sanguíneo. Dessa forma, é de grande importância facilitar e agilizar o diagnóstico da doença, para que, assim, as consequências e sequelas sejam evitadas/diminuídas ao máximo, auxiliando na redução da quantidade de casos com maior gravidade e até mesmo casos de óbitos.

Desenvolver formas para otimizar o processo de diagnóstico do AVC no menor tempo possível pode reduzir consideravelmente as sequelas causadas no indivíduo, facilitando muito no processo de reabilitação. Utilizando a computação, especificamente as áreas relacionadas a Inteligência Artificial, ou IA, é possível sintetizar sistemas inteligentes que consigam auxiliar no diagnóstico de casos da doença. Modelos de Aprendizado de Máquina e Aprendizagem Profunda, subárea da IA e subárea do próprio Aprendizado de Máquina, respectivamente, possuem boa capacidade de reconhecimento de padrões (TIWARI; TIWARI; TIWARI, 2018) e possibilitam a criação de modelos classificadores. Com a ajuda de um classificador de fatores de risco, por exemplo, um indivíduo conseguiria ter a noção básica de sua situação atual com relação ao AVC e procurar auxílio médico de um especialista o mais rápido possível.

Figura 1 – Relação entre Inteligência Artificial (*Artificial Intelligence*), Aprendizado de Máquina (*Machine Learning*) e Aprendizado Profundo (*Deep Learning*).



Fonte: (COPELAND, 2021)

O objetivo do presente trabalho de conclusão de curso desenvolvido está diretamente relacionado a utilização de técnicas de Aprendizado de máquina e aprendizado profundo para possibilitar a classificação da situação de um indivíduo com relação a um possível quadro de AVC. Baseado nas informações do indivíduo, seus fatores de risco e, também, imagens de tomografia computadorizada (TC), será possível alertá-lo o quanto antes sobre uma possível suspeita, auxiliando o médico no processo de diagnóstico, que irá tomar as devidas providências agilizando ainda mais o processo. através do processamento de suas informações sobre fatores de risco e imagens de tomografia computadorizada.

Vale ressaltar que o intuito da utilizadas de métricas inteligentes junto ao diagnóstico de doenças não busca, em hipótese alguma, substituir qualquer opinião médica e muito menos algum profissional de saúde, mas apenas informar e conscientizar indivíduos para que busquem auxílio médico o mais rápido possível em caso de suspeita e agilizar o processo de diagnóstico da doença, auxiliando na tomada de decisões.

Baseado nessas informações, para esse trabalho de conclusão de curso (TCC), foram desenvolvidos dois modelos classificadores, usando algoritmos de Aprendizado de Máquina, para classificação de casos de AVC através da análise de fatores de risco. Tais modelos apresentaram resultados eficientes e bastante similares. Além disso, foi desenvolvido também uma rede

neural, do tipo convolucional, que consegue classificar imagens de tomografia computadorizada, rotulando-as como sendo imagens da região cerebral de um indivíduo de condição patológica vítima de AVC ou de condição normal, através do processamento dessas imagens.

2 Fundamentação Teórica

2.1 Acidente Vascular Cerebral – AVC

O AVC, causado pela alteração do fluxo de sangue do cérebro, pode ser originado da obstrução total ou parcial de alguma artéria da região, o que é conhecido como Acidente Vascular Cerebral Isquêmico (AVCi), ou simplesmente infarto cerebral. Esse tipo de AVC é o mais comum entre os casos da doença e pode ocorrer devido a um trombo (caso de trombose) ou êmbolo (caso de embolia) presente na vítima.

Outra origem da alteração do fluxo de sangue mencionada é o caso do rompimento de determinado vaso sanguíneo do cérebro, o que causa uma hemorragia na região. Esse caso caracteriza o Acidente Vascular Cerebral Hemorrágico (AVCh). Tal rompimento de determinado vaso altera o nível da pressão intracraniana e pode dificultar também a chegada de sangue em outras áreas não afetadas. Embora seja menos comum, esse tipo de AVC é o mais grave, tendo maiores índices de mortalidade quando comparado ao outro tipo abordado (AVC isquêmico) (BVS, 2015).

Com relação ao sintomas presentes nas vítimas, dor de cabeça forte sem causa aparente; fraqueza ou formigamento da face, braço ou perna (especialmente em apenas um dos lados do corpo); alteração da fala e/ou da compreensão; alteração da visão de um ou ambos os olhos; e dificuldade, ou incapacidade, de se movimentar são sintomas bastante comuns em ambos os tipos da doença.

Em específico para o caso de AVCi, tem-se de forma comum os sintomas de tontura e perda de equilíbrio ou coordenação. Já para o caso de AVCh, pode ocorrer também náusea, vômito, confusão mental e, em alguns casos, perda de consciência. Em tal caso também podem aparecer os sintomas de sonolência exagerada, alterações na frequência cardíaca e respiratória, e até mesmo convulsões. Vale ressaltar que é de grande importância que um indivíduo com os sintomas apresentados procure assistência médica o mais rápido possível. Dessa forma, a doença pode ser diagnosticada e tratada rapidamente, possivelmente resultando em maiores chances de sobrevivência da vítima e menores sequelas (BVS, 2015).

2.1.1 Fatores de risco da doença

Existem alguns fatores que aumentam a probabilidade da ocorrência do acidente vascular cerebral, facilitando o desencadeamento da doença. Estes são denominados fatores de risco e podem ser inerentes à vida humana, maus hábitos, estilo de vida inadequado ou até mesmo questões genéticas. Os principais fatores são:

- Hipertensão
- Diabetes
- Obesidade
- Tabagismo
- Consumo excessivo e frequente de álcool e drogas
- Estresse
- Idade avançada (envelhecimento)
- Histórico familiar
- Sexo masculino
- Colesterol elevado
- Doenças cardiovasculares (principalmente as que produzem arritmia cardíaca)
- Sedentarismo
- Doenças do sangue

2.1.2 Diagnóstico com tomografias computadorizadas (TC)

Além da análise de fatores de risco e sintomas, outra forma de identificar o quadro de acidente vascular cerebral em vítimas da doença é através da análise de tomografias computadorizadas. Os quadros de AVC podem apresentar características aparentes quando analisados em tais tomografias, permitindo a identificação da doença e o nível de ocorrência da mesma em determinado cérebro. Embora não seja muito fácil realizar o diagnóstico da doença em alguns casos (OLIVEIRA; ANDRADE, 2001), é possível aplicar a computação para esse tipo de situação, pois modelos de Inteligência Artificial possuem alta capacidade de reconhecimento de padrões.

De forma simplificada, a hipodensidade, presença de áreas hipodensas (mais escuras), é a representação radiológica nas TCs de um edema citotóxico[1], o qual ocorre devido ao mal funcionamento da bomba de sódio/potássio, causando uma área acinzentada mais escura que o habitual no córtex. O corpo humano resolve esse edema citotóxico deixando uma cicatriz na região, denominada de malácia cerebral.

Todo esse processo gera manchas na tomografia e essa diferença de escala de cinza pode ser um dos fatores para classificar se o quadro analisado é um AVC ou não, pois dependendo do

[1] Retenção de água e sódio no interior das células cerebrais em razão de uma anomalia no metabolismo celular. Fonte: https://saude.ccm.net/faq/4569-edema-cerebral-tipos-e-causas

caso é possível identificar um quadro da doença em fase inicial, fase subaguda ou até mesmo algum quadro de AVC com ocorrência mais antiga na vítima (RUEDA, 2022).

2.2 Aprendizado de Máquina

O Aprendizado de Máquina, ou Aprendizagem de Máquina, é uma área da inteligência artificial que lida com o a criação de programas "inteligentes", capazes de aprender determinados conceitos e comportamentos, sem serem explicitamente programados para isso, de adquirir conhecimento, de aprender e tomar decisões a partir desse aprendizado. Tal área explora também a capacidade de programas e sistemas conseguirem realizar previsões sobre determinadas situações, aprendendo através da análise de dados já existentes sobre tais situações.

Envolvendo conceitos estatísticos, reconhecimento de padrões e programação, a área de Aprendizado de Máquina possibilita que resultados sejam especulados de acordo com acontecimentos passados semelhantes, através de um processamento dos dados usando o reconhecimento de padrões. "Um sistema de aprendizado é um programa de computador que toma decisões baseado em experiências acumuladas através da solução bem sucedida de problemas anteriores"(MONARD; BARANAUSKAS, 2003).

Com relação à criação de uma solução utilizando conceitos de Aprendizado de Máquina, o processo geralmente envolve duas etapas, o treinamento e a avaliação. A etapa de treinamento é quando um determinado conjunto de dados é passado para o modelo processá-lo, aprendendo a reconhecer padrões e também a fazer previsões. A etapa de avaliação consiste em passar dados semelhantes ao determinado conjunto de dados utilizado no treino para que o desempenho do modelo seja medido, analisando os resultados esperados com os resultados obtidos.

No campo do Aprendizado de Máquina, existem basicamente três tipos diferentes de aprendizagem: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço. Os algoritmos de Aprendizado de Máquina utilizados neste trabalho pertencem à área de aprendizado supervisionado e, portanto, apenas este será abordada com mais detalhes.

2.2.1 Aprendizado Supervisionado

O Aprendizado supervisionado utiliza dados rotulados no treinamento do modelo, o qual prevê uma variável dependente a partir de uma ou mais variáveis independentes (HONDA; FACURE; YAOHAO, 2017). Existem diversos algoritmos desse tipo de aprendizado, sendo os principais: regressão linear, regressão logística, máquina de suporte vetorial, árvores de decisão, k-vizinhos mais próximos, floresta aleatória, entre outros. Os utilizados neste trabalho para desenvolver o classificador de fatores de risco foram apenas a Regressão Logística e o Classificador de Floresta Aleatória.

O modelo de Regressão Logística foi escolhido pela sua popularidade na área de

Aprendizado de Máquina e por conseguir ter capacidade de calcular, de forma eficiente, a probabilidade de determinado evento pertencer a alguma das classes do problema, e não só simplesmente classificar a entrada processada. O algoritmo de Floresta Aleatória foi escolhido pela sua popularidade na área, mas também pelo fato de ser um modelo que tem seu funcionamento bastante diferente de como o modelo de Regressão Logística processa os dados que recebe.

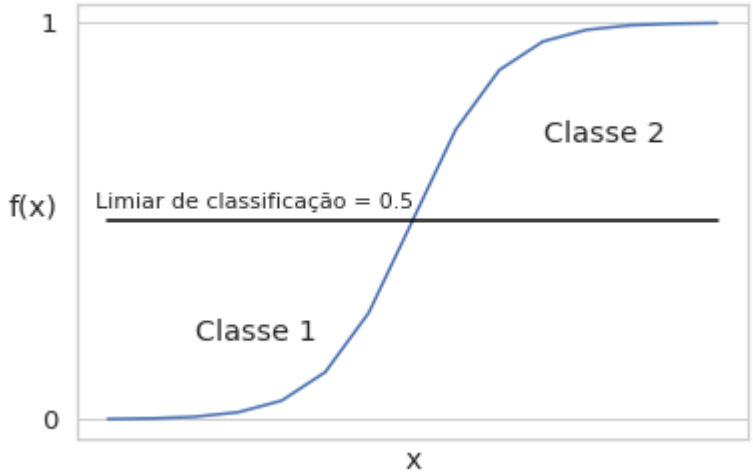
2.2.1.1 Regressão Logística

A Regressão Logística é um modelo estatístico capaz de modelar determinada probabilidade de um evento acontecer a partir da combinação linear de variáveis independentes entre si associadas a tal evento. Muito utilizado em situações de classificação binária, esse modelo combina variáveis independentes com respectivos pesos atribuídos a elas de modo a obter um valor a partir deste processo (IBM, 20–c). Tal valor é passado para uma função logística que resulta na probabilidade de algum evento acontecer. No caso de uma classificação, tal resultado indica a probabilidade de os valores analisados pertencerem a determinada classe do problema em questão.

De maneira geral e simplificada, algoritmos de regressão ajustam uma curva fazendo com que variáveis independentes consigam gerar, através de uma função, determinado valor resultante, denominado de variável dependente. O algoritmo de Regressão Logística, embora seja utilizado para classificação de dados, é um algoritmo de regressão o qual ajusta uma curva através da função logística (função sigmoide)(SCIKIT-LEARN, entre 2007 e 2022a). Exemplo dessa função:

$$\sigma(x) = \frac{1}{1 + e^{-}x} \tag{2.1}$$

Figura 2 – Gráfico de uma função logística f(x) sendo utilizada para divisão de duas classes a partir do valor 0.5 (limiar entre as classes)



Fonte: Elaborado pelo autor.

Cada variável independente passada para o modelo de Regressão Logística é interpretada como x_i, com i variando de 1 ao número total de variáveis independentes do problema abordado, e é multiplicada por um valor denominado peso. Pesos de um modelo são valores comumente descritos pelas letras θ ou w e são gerados, inicialmente, de forma aleatória, sendo otimizados durante a etapa de treinamento do modelo para gerarem menores erros de classificação (modelo mais eficiente). Através da combinação linear dos pontos $x * \theta$ (ou $x * w$), é obtida a variável dependente passada para a função logística (equação I) e o resultado desse processo é um valor que varia de 0 a 1. Esse valor, após comparado com um limiar de classificação[2], resulta na classificação da entrada recebida como uma das classes pertencentes ao problema binário abordado.

2.2.1.2 Floresta Aleatória

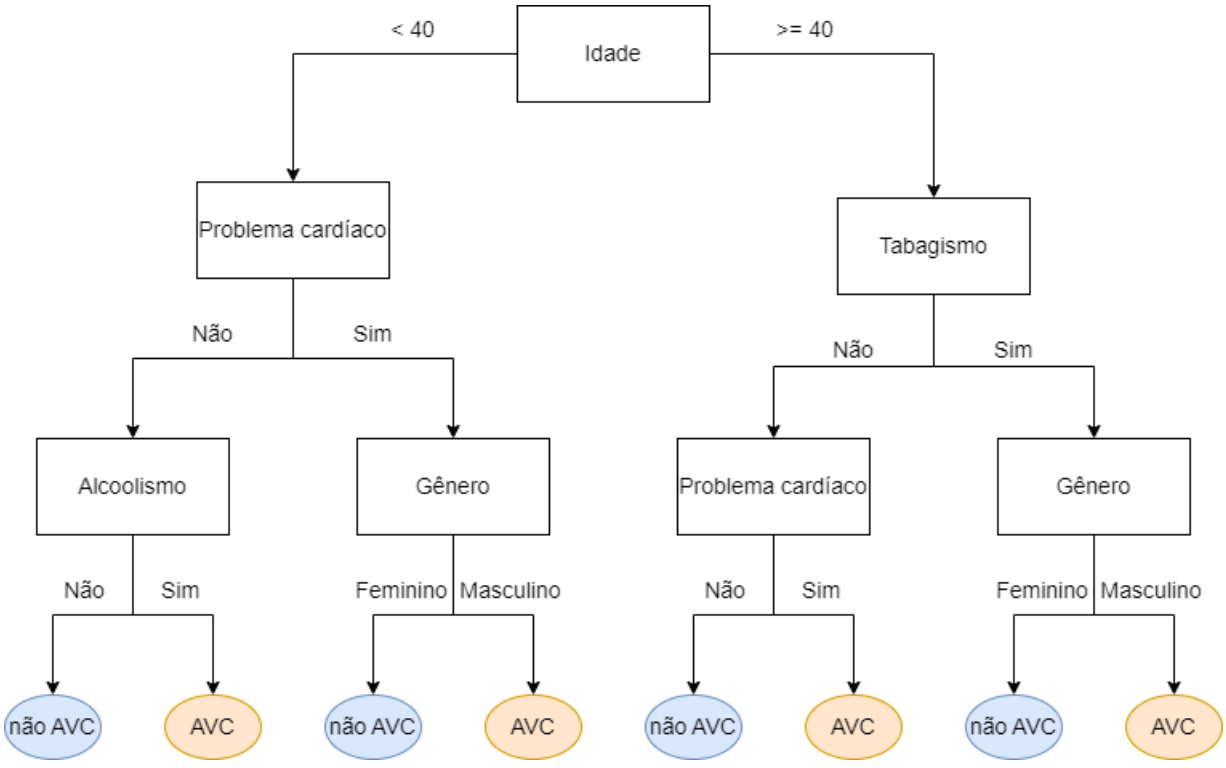
Floresta Aleatória é um modelo de aprendizado de máquina composto por um conjunto aleatório de Árvores de Decisão que pode ser usado em situações de regressão ou classificação, por exemplo. Especificamente para problemas de classificação, como o abordado neste trabalho, o modelo classifica de acordo com a classificação feita pelas árvores que o compõem. Dessa forma, se há um resultado classificativo mais frequente, isto é, se a maioria das árvores gerou determinada classificação, então esta será a que o modelo de floresta aleatória irá resultar em sua conclusão (NAVLANI, 2018).

[2] termo de valor geralmente igual a 0.5 para classificações binárias, serve como comparativo para classificação de determinado resultado que varia de 0 a 1

A estrutura de uma Árvore de Decisão é constituída por vários blocos de decisão comumente denominados de nós. Cada nó funciona como um bloco condicional que encaminha a entrada recebida para um de seus filhos de acordo com o valor que ele analisa (regra de decisão). Tal processo é feito para todo nó pertencente à árvore até chegar em determinado nó folha que, por sua vez, representa uma das possíveis classes do problema. Portanto, em uma árvore de decisão, uma entrada percorre um dos diferentes caminhos possíveis até chegar em um nó folha que a classifica como uma das possíveis rotulações da situação. Tal caminho acabada sendo definido de acordo com os valores que pertencem a entrada em questão (IBM, 20–b).

As regras de decisão utilizadas pelos nós para separar as entradas em diferentes caminhos são estabelecidas na etapa de treinamento do modelo de acordo com determinado conjunto de dados utilizado nessa fase. Por isso, existem infinitas possibilidades de estruturas para árvores de decisão, sendo cada uma delas definidas de acordo com a situação e ao conjunto de dados abordado.

Figura 3 – Exemplo de uma Árvore de Decisão para um conjunto de dados que contenha as variáveis Idade, Problema Cardíaco, Alcoolismo, Gênero e Tabagismo, com as possíveis rotulações de classe não AVC e AVC.

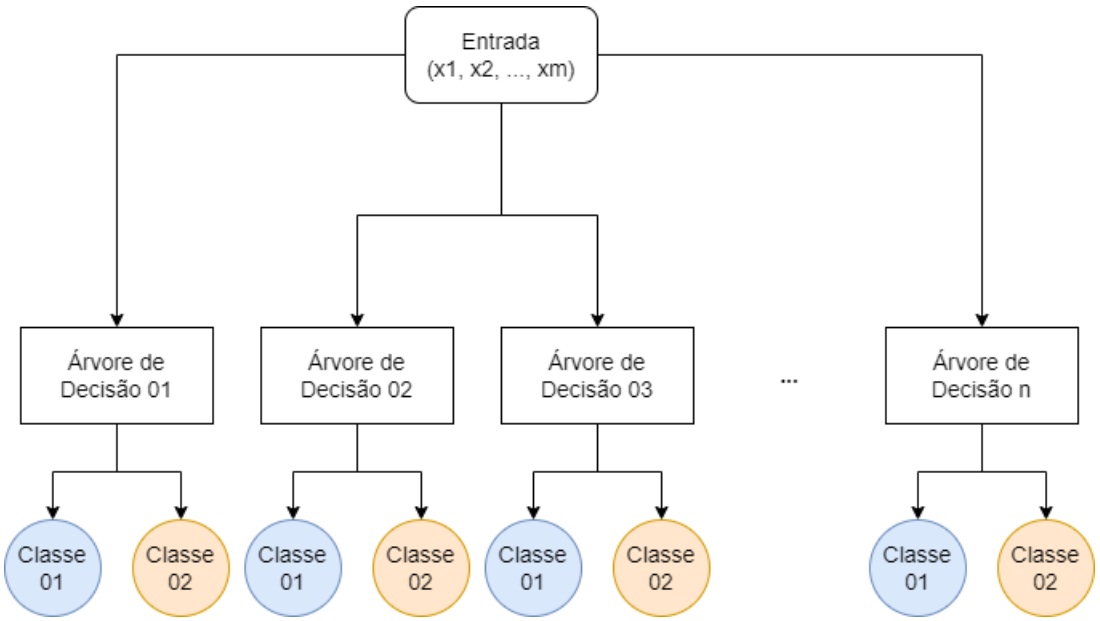


Fonte: Elaborado pelo autor.

A ideia do algoritmo de Floresta Aleatória é justamente utilizar dessas diversas possibilidades de estrutura das Árvores de Decisão e criar assim uma floresta com várias árvores diferentes abordando o mesmo problema, classificando determinada entrada recebida de acordo

com a classificação mais frequente das árvores que compõem o modelo (SCIKIT-LEARN, entre 2007 e 2022b).

Figura 4 – Estrutura de um modelo de Floresta Aleatória para m variáveis, contendo n Árvores de Decisão com duas possibilidades de classificação: Classe 01 e Classe 02.



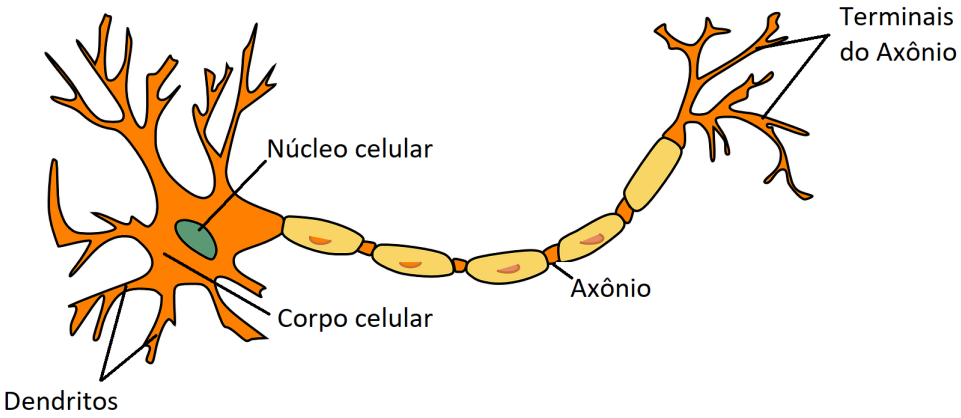
Fonte: Elaborado pelo autor.

2.3 Aprendizagem profunda

2.3.1 Neurônios artificiais

O neurônio artificial é o principal componente das redes neurais computacionais e seu funcionamento é baseado nos neurônios reais, que compõem o sistema nervoso. Os neurônios reais, base do sistema nervoso, são células que estabelecem conexões entre si para transmitir impulsos nervosos pela região cerebral (VARELLA, 20–). O neurônio artificial é uma estrutura com funcionamento que se assemelha do neurônio real, mantendo a principal característica dessas estruturas de criarem conexões entre si, recebendo informações provenientes de outros neurônios e passando elas adiante.

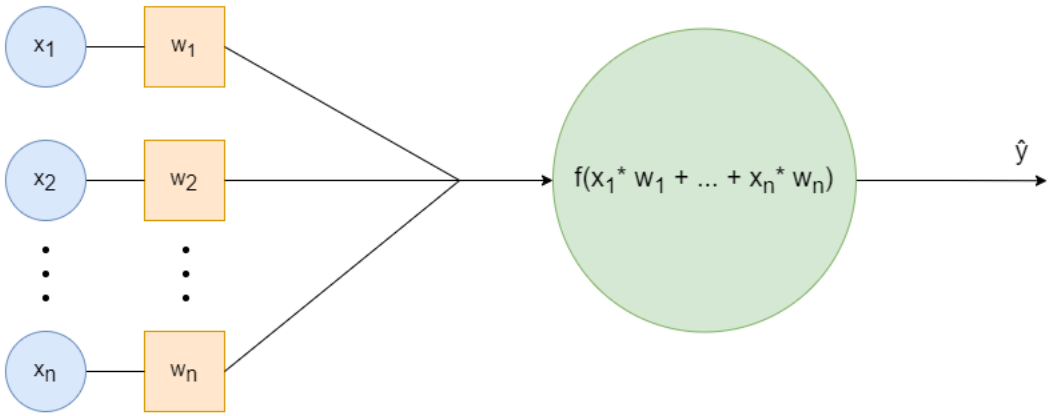
Figura 5 – Ilustração de um neurônio biológico



Fonte: Elaborado pelo autor.

Os principais componentes de um neurônio artificial são as entradas, os pesos, a função de ativação e saída. As entradas, geralmente representadas pela letra x, recebem a informação que chega para o neurônio. Tal informação é multiplicada por um determinado peso, comumente representado pela letra w, e o resultado disso é passado para uma função de ativação que processa os valores recebidos e retorna um ou mais valores, os quais servirão de entrada para um outro neurônio artificial conectado à rede (YACIM; BOSHOFF, 2018).

Figura 6 – Ilustração de um neurônio artificial simples



Fonte: Elaborado pelo autor.

2.3.2 Redes Neurais Artificiais

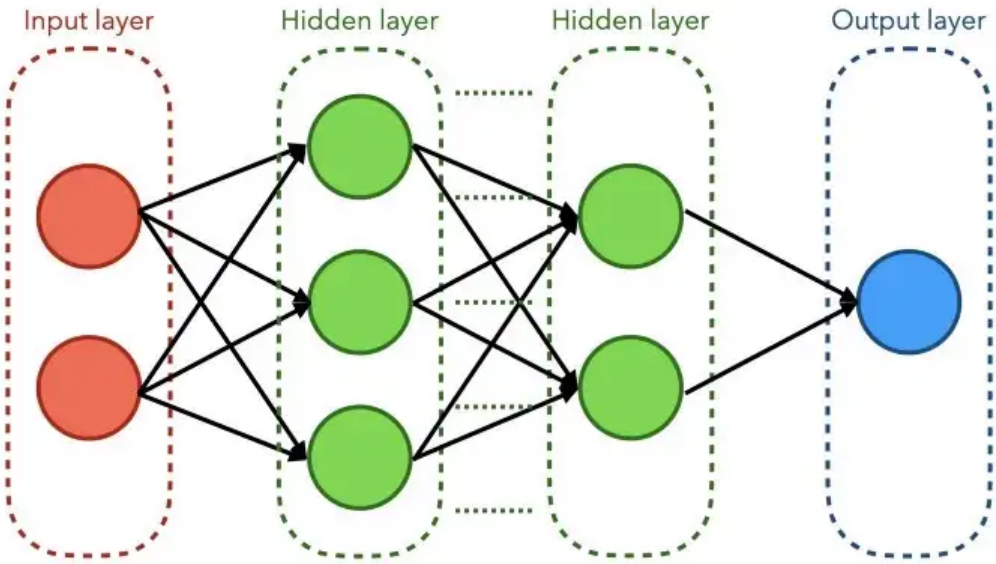
Redes neurais artificiais, ou *Artificial Neural Network* (ANN), são estruturas do campo da inteligência artificial que possibilitam os computadores a processar dados se baseando no comportamento do cérebro humano (AWS, 20–). Ao receber um conjunto de dados específico, uma rede neural artificial é capaz de se ajustar a este conjunto, aprendendo a reconhecer padrões sobre estes dados e se tornando mais eficiente à medida que os processa. A estrutura da rede é composta por um conjunto de neurônios artificiais agrupados em camadas interconectadas que possibilitam o computador a aprender, reconhecer padrões e tomar decisões inteligentes.

Uma rede neural simples possui camadas de três tipos: camada de entrada (*Input layer*), camada oculta (*Hidden layer*) e camada de saída (*Output layer*). A camada de entrada é onde se encontram os neurônios responsáveis por receber os dados que serão processados pela rede. Tal camada faz o primeiro processamento desses dados e os encaminham para o próximo conjunto de neurônios. As camadas do tipo oculta recebem dados da camada de entrada ou de outras camadas ocultas, processando novamente os dados e passando o resultado para outra camada. Redes neurais podem ter uma ou mais camadas desse tipo.

Por fim, o conjunto de neurônios da camada de saída faz o último processamento dos dados, fornecendo o resultado final de todos os dados processados pela rede. Para problemas de classificação binária, tem-se apenas um nó na camada de saída, o qual indica a probabilidade de determinada entrada analisada pela rede ser ou não da classe abordada. Já para problemas de várias classes diferentes, tem-se vários nós nessa camada (AWS, 20–).

Além dessa estrutura de rede abordada, existem outros tipos mais específicos de redes neurais que apresentam melhores resultados quando aplicados em determinadas situações. No caso de problemas que envolvem a classificação de imagens e o reconhecimento de objetos, o uso de Redes Neurais Convolucionais é mais recorrente do que o uso da rede neural padrão, justamente pela boa capacidade que estas possuem de valorizar características da imagem.

Figura 7 – Ilustração simplificada de uma Rede Neural Artificial simples



Fonte: (NURFIKRI, 2020)

2.3.3 Redes Neurais Convolucionais

Comumente chamadas de Conv-Net ou simplesmente CNN (acrônomo de *Convolutional Neural Network*), as redes neurais do tipo convolucional são muito utilizadas em tarefas que envolvem a identificação de objetos, reconhecimento de características e classificação de imagens. Esse tipo de rede tem a capacidade de valorizar detalhes da imagem que são relevantes para a classificação, realçando determinados agrupamentos de pixels através da aplicação de filtros matriciais na imagem para identificação de padrões, o que colabora para um processo de classificação mais eficiente.

As Redes Neurais Convolucionais apresentam três tipos principais de camadas em sua estrutura: camadas de convolução (*convolution*), camadas de agrupamento (*pooling*) e as camadas totalmente conectadas. Camadas de convolução, ou convolucionais, são as camadas responsáveis pelo processo de realce das características mencionado no parágrafo anterior (O'SHEA; NASH, 2015).

Esse processo se dá pela aplicação de um filtro, comumente chamado de *kernel*, na imagem, vista como uma matriz de pixels, que está sendo processada pela rede, funcionando como um detector de recursos. Esse filtro tem formato matricial com tamanho típico de 3x3, podendo ser maior dependendo da rede, e o processo de aplicação se dá de forma iterativa via a realização do produto escalar entre seus valores e os da matriz da imagem processada, finalizando-se com a soma dos resultados dessa multiplicação (IBM, 20–a).

Tal processo é iterativo, acontecendo até que o filtro multiplique toda a imagem. Alguns

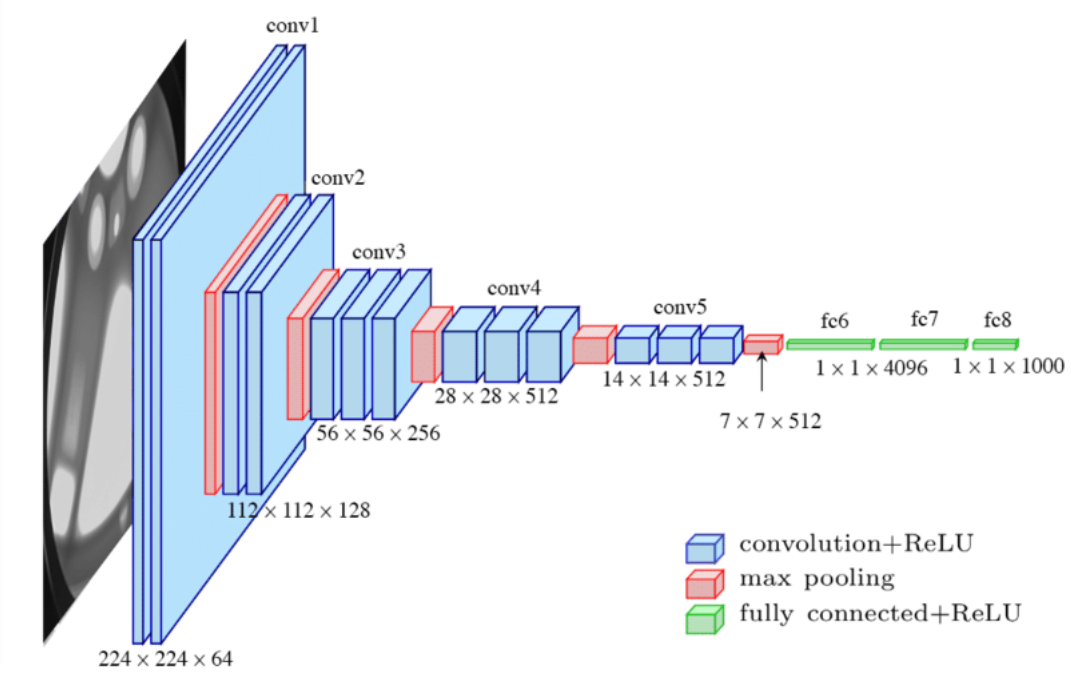
hiperparâmetros que podem variar o volume da imagem resultante são: o número de filtros que serão aplicados a imagem, o número de passos denominado *stride* que o *kernel* percorre para se deslocar de uma multiplicação para a outra e o *padding*, que adiciona zeros na matriz da imagem para chegar a determinado tamanho que favoreça a aplicação do *kernel*.

As camadas de agrupamento irão pegar o resultado da camada de convolução e reduzir o número de valores recebidos, mantendo as características realçadas pelos filtros da convolução, porém eliminando áreas vazias de informação relevante para a classificação. O processo de pooling destas camadas se dá por dois tipos diferentes que podem ser utilizados, o agrupamento máximo ou o agrupamento médio.

O agrupamento máximo é feito através da aplicação de um filtro que se desloca pela imagem que simplesmente pega o maior valor dentre os valores analisados na imagem e o coloca em uma matriz de saída, descartando os demais valores. O agrupamento médio se dá pela aplicação de um filtro que também se move pela imagem analisada, porém calculando o valor médio dos valores analisados pelo filtro, colocando-o em uma matriz de saída de forma semelhante ao outro agrupamento descrito. O agrupamento máximo tem maior frequência de utilização quando comparado com o agrupamento médio (IBM, 20–a).

Por fim, nas camadas totalmente conectadas tem-se diversos neurônios totalmente conectados que irão processar os valores recebidos aplicando alguma função de ativação, gerando valores de saída para as próximas camadas ou para a camada de saída. Toda rede neural convolucional tem seu formato base composto por essas camadas, porém existem diversos tipos de CNN diferentes que vão variar o número de instâncias dessas camadas mencionadas e os hiperparâmetros presentes nas mesmas. A arquitetura de rede convolucional denominada de VGG-16, utilizada no desenvolvimento deste trabalho, apresenta treze camadas de convolução e cinco camadas de agrupamento, assim como três camadas totalmente conectadas (JORDAN, 2018).

Figura 8 – Representação de uma rede da arquitetura de Rede Neural Convolucional VGG-16



Fonte: (FERGUSON et al., 2017)

2.4 Métricas para avaliação do desempenho de modelos classificadores

Com o modelo classificador ajustado de acordo com a situação abordada, torna-se fundamental a realização de uma boa avaliação de seu desempenho. "O objetivo da fase de avaliação é estimar os resultados de mineração de dados de forma rigorosa e obter a confiança de que são válidos e confiáveis antes de avançar." (PROVOST; FAWCETT, 2016). Avaliar de forma adequada permite a identificação de problemas, como sobreajuste ou excesso de generalização, por exemplo, assim como visualizar de forma geral como seria a performance do modelo quando colocado em produção na prática.

Para realizar tal processo de avaliação nos modelos criados neste trabalho, foram definidas algumas métricas utilizadas na avaliação de modelos classificadores, as quais serão abordadas logo abaixo.

2.4.1 Acurácia

A acurácia é uma das métricas mais simples de se medir e justamente por isso sua utilização é muito popular na avaliação de classificadores. Porém, por ser demasiadamente simples, tal métrica pode esconder deficiências do modelo analisado e transparecer um resultado

não muito confiável. O cálculo da acurácia é dado pela divisão do número de acertos do modelo pelo número total de casos analisados, ou seja, pelo número de acertos somados com o número de erros (DEVELOPERS, 2022):

$$acuracia = \frac{acertos}{acertos + erros} \tag{2.2}$$

Como o cálculo dessa métrica não leva em conta fatores além de acertos e erros, como o tipo de erro, por exemplo, é bastante interessante utilizar a Acurácia em conjunto com outras métricas para fazer uma avalição mais eficiente, levando em conta também os tipos de valores falsos que o modelo gera.

2.4.2 Matriz de confusão

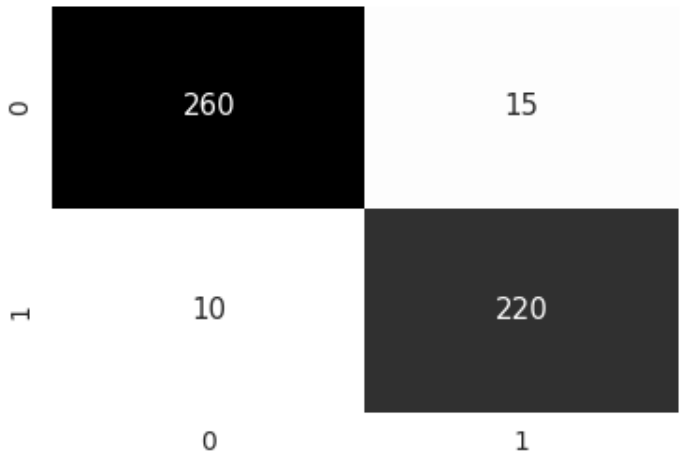
Na fase de avaliação de um modelo também pode ser interessante olhar para o desempenho do modelo com relação a erros específicos, e a matriz de confusão apresenta de forma simplificada esse comportamento. "Uma matriz de confusão separa as decisões tomadas pelo classificador, tornando explícito como uma classe está sendo confundida com outra. Desta forma, diferentes tipos de erros podem ser tratados separadamente" (PROVOST; FAWCETT, 2016).

Na matriz de confusão temos dois tipos de acertos e erros, os verdadeiros positivos e negativos, e os falsos positivos e negativos, respectivamente. Valores verdadeiros ocorrem quando o modelo classifica corretamente valores negativos (0) ou positivos (1) e valores falsos ocorrem quando se tem a classificação incorreta desses mesmos valores, negativos ou positivos. Analisar a taxa de falso positivo e de falso negativo permite melhor visualização de qual tipo de erro é predominante na questão, assim como ter uma ideia percentual probabilística de como o modelo em questão reagiria em situações reais com relação aos tipos de erros.

É comumente utilizado para simplificar a nomenclatura das métricas de falsos e verdadeiros (erros e acertos, respectivamente) as seguintes siglas: FN para falso negativo (*false negative*), TP para verdadeiro positivo (*true positive*), TN para verdadeiro negativo (*true negative*) e FP para falso positivo (*false positive*).

A matriz de confusão é uma métrica que permite a análise desses valores de falsos e verdadeiros de uma forma bastante simplificada. Ela é frequentemente utilizada na avaliação de classificadores pois fornece uma visão sobre a confusão entre as classes que um determinado classificar realizou na análise de casos passados para ele. Uma matriz de confusão é uma matriz de dimensão n x n sendo que n é o número de classes do problema abordado.

Figura 9 – Exemplo de matriz de confusão para um classificador binário em que as classes são rotuladas como 0 (negativo) e 1 (positivo). No eixo vertical tem-se a rotulação 0 ou 1 para os valores previstos pelo modelo e no eixo horizontal, a rotulação dos valores reais



Fonte: Elaborado pelo autor.

2.4.3 Taxa de FP, taxa de TP, Curva ROC e AUC ROC

A taxa de FP, ou FPR, e a taxa de TP, ou TPR, são métricas usadas em conjunto para avaliar e descrever a eficiência de um modelo classificador. A FPR mede a fração dos valores negativos que foram previstos incorretamente, julgados como positivos pelo classificador em questão. Para realizar o cálculo dessa métrica, a seguinte equação é utilizada:

$$FPR = \frac{FP}{FP + TN} \tag{2.3}$$

A TPR, também conhecida como sensibilidade, mede a fração dos valores positivos que foram previstos de forma correta, julgados também como positivos pelo modelo (DEEPCHECKS, 20–). O cálculo da TPR é feito da seguinte forma:

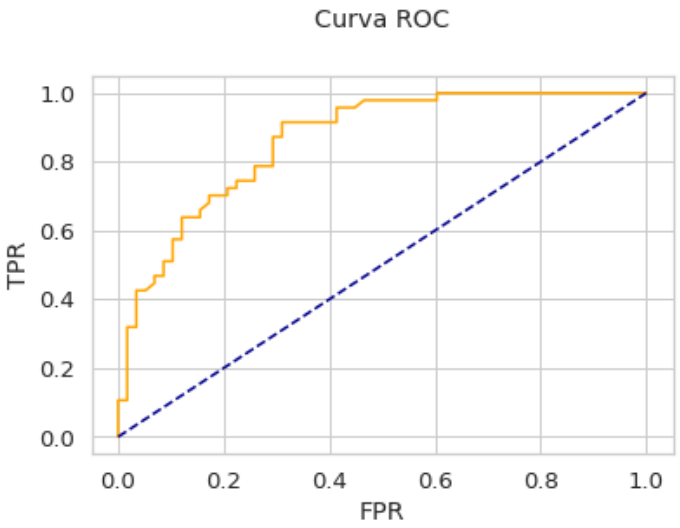
$$TPR = \frac{TP}{TP + FN} \tag{2.4}$$

É de grande importância que sejam analisadas ambas as métricas descritas acima em conjunto, pois assim torna-se possível ter um diagnóstico confiável e saber se o modelo está extremamente tendencioso ou não. Por exemplo, em uma situação em que determinado classificador rotula sempre todos os dados como sendo de uma classe positiva, sua TPR terá valor máximo (1.0), pois para todo valor positivo recebido, ele acertará todos. Porém, este mesmo modelo, como consequência, errará todos os casos negativos, e isso só poderá ser enxergado se a FPR for analisada, pois seu valor também será 1.0.

Nesse cenário descrito, analisar apenas a TPR indicaria um modelo muito eficiente, já que ele não errou nenhum caso positivo que recebeu. Dessa forma, estaria sendo omitido o alto viés e a baixa capacidade de generalização do classificador em questão. Um modelo tendencioso assim possivelmente apresentaria péssimos resultados na prática.

Para facilitar a análise em conjunto de ambas as métricas, existe uma outra métrica muito poderosa denominada de *Receiver Operating Characteristic Curve*, ou simplesmente *ROC Curve*. Em português, Curva de Característica de Operação do Receptor, a Curva ROC é uma ferramenta gráfica que permite verificar de forma conjunta TPR e FPR à medida que o limiar entre as classes (probabilidade que separa as classes) do modelo varia. Tal variação é feita para diferentes valores do limiar que geram diferentes valores de TPR e FPR como resultado. Estes valores formam pontos em um gráfico TPR x FPR e a junção desses pontos formam a curva ROC (BAGHERI, 2019).

Figura 10 – Exemplo Curva ROC (em laranja) constituída a partir da variação de TPR e FPR do modelo de Regressão Logística criado. A curva azul é uma curva de referência, indicando pontos de TPR e FPR de valores iguais.



Fonte: Elaborado pelo autor.

Outra forma de utilizar a curva ROC para avaliação de determinado modelo é calculando sua área. A partir deste processo, tem-se um número que pode variar entre 0 a 1.0, sendo que, quanto mais próximo de 1.0, mais eficiente o modelo seria, com máximo TPR e mínimo FPR. A área abaixo da curva ROC é uma métrica denominada de Pontuação AUC (*area under curve*) ROC.

3 Trabalhos Correlatos

O tema deste trabalho foi inspirado no trabalho de conclusão de curso ANÁLISE DE DADOS PARA AUXILIAR NO DIAGNÓSTICO PRECOCE DE ACIDENTE VASCULAR CEREBRAL - AVC (TAMAKE, 2020), também orientado pelo Professor Clayton, que orientou o presente trabalho. (TAMAKE, 2020) apresenta uma abordagem diferente sobre o problema de Acidente Vascular Cerebral, pois trata da classificação de casos de AVC usando modelos de Aprendizado de Máquina sem a criação de redes neurais, com implementações e resultados diferentes.

Além do fato de que (TAMAKE, 2020) não aborda o subcampo do Aprendizado de Máquina, a Aprendizagem profunda, outro fator significante que difere ele deste trabalho é a forma como os dados foram pré-processados e processados. Diferentes técnicas foram utilizadas aqui assim como a criação dos classificadores a partir de outros modelos de Aprendizado de máquina.

Todos esses fatores mencionados resultaram em novos resultados e diferentes abordagens, caracterizando um novo trabalho sobre o tema da classificação de casos de Acidente Vascular Cerebral, o qual é muito importante devido a gravidade e alta ocorrência dessa doença.

4 Metodologia

A metodologia deste trabalho se consistiu no primeiro momento a entender mais sobre os assuntos abordados, através de um levantamento bibliográfico sobre os temas de Aprendizado de Máquina e Aprendizado Profundo, estudando os principais modelos e técnicas dessas áreas para criar uma solução mais robusta e eficiente do problema abordado. Para ambos os modelos criados aqui, foram seguidas as etapas: levantamento e pré-processamento, aplicação de técnicas das áreas dos dados que envolvem o problema (fatores de risco e imagens de tomografia computadorizada), processamento de dados para criação do modelo classificador e fase final de avaliação do mesmo, para entender o desempenho do modelo criado.

4.1 Base de dados para os fatores de risco

Os dados utilizados na construção do modelo de Aprendizado de Máquina para fatores de risco foram retirados da plataforma *Kaggle*[1], sendo este um dos conjuntos de dados disponibilizados para propósitos educacionais. O conjunto original conta ao todo com 5110 registros, em que cada um contém informações de uma pessoa diferente, rotulada como vítima de AVC ou não. O conjunto original estava desbalanceado, ou seja, com uma grande quantidade de valores sendo de pessoas rotuladas como não AVC, sendo essas as características presentes no conjunto de dados:

4.1.1 Variáveis categóricas

- Sexo: Masculino ou feminino, em que o sexo masculino é um fator de risco
- Hipertensão: Indivíduo é ou não hipertenso. Hipertensão é um fator de risco.
- Doença cardíaca: Indivíduo tem ou não alguma doença cardíaca, o que é um fator de risco.
- Estado civil de ser ou já ter sido casado: Sim ou não, tal variável pode estar ligada a estilo de vida e estresse, o qual é um fator de risco da doença.
- Tipo de residência: Urbana ou rural, tal variável está ligada a estilo de vida, que pode ser um fator considerável para a doença
- Vítima de AVC: Sim ou não, essa é a variável alvo do trabalho, sendo utilizada para rotular os dados para previsão.

[1] https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

- Tipo de emprego: empregado privado, funcionário público (governo), empreendedor ou não trabalha.
- Status com relação a tabagismo: indivíduo nunca fumou, fuma formalmente, fuma regularmente ou situação desconhecida.

4.1.2 Variáveis quantitativas

- Idade: idade avançada é um fator de risco para a doença
- Nível de glicose: diabetes é um fator de risco para a doença
- BMI: Índice de massa corpórea no padrão americano, utilizado para identificar obesidade.
- Id: Identificação de cada registro. Tal variável foi desconsiderada por não ter impacto na variável alvo do trabalho que rotula os casos de AVC.

4.2 Base de dados para as imagens de tomografia computadorizada

Para criação do modelo classificador de imagens de TC, foi utilizado um conjunto de dados com 475 imagens no total, divididas em três classes diferentes: AVC Isquêmico, AVC Hemorrágico ou não AVC. Entretanto, como o objetivo deste trabalho é de apenas identificar a ocorrência de Acidente Vascular Cerebral em determinada vítima, sem especificar o tipo de AVC de fato, este conjunto foi redistribuído para apenas duas classes.

Dessa forma, o conjunto original passou a ter apenas as classes AVC e não AVC, com as seguintes distribuições:

- AVC possui 301 imagens
- AVC possui 174 imagens

4.3 Ferramentas utilizadas

4.3.1 Python

Python[2] é uma linguagem de programação de alto nível, gratuita e muito utilizada no desenvolvimento de aplicações modernas. Criada por volta de 1990, tal linguagem apresenta um uso mais simples quando comparado com o uso de outras linguagens do mercado, trazendo mais facilidade para quem não é necessariamente do ramo da programação. Ela prioriza a legibilidade do código, combinando uma sintaxe clara com recursos poderosos de sua biblioteca padrão (recursos nativos) e de outras bibliotecas desenvolvidas pela comunidade.

[2] https://www.python.org/

Por ser acessível e amigável, a linguagem de programação Python se tornou bastante popular no mercado, com muitos frameworks e módulos prontos desenvolvidos e disponibilizados para os usuários da ferramenta. Bibliotecas de áreas como Ciência de Dados, Automação, Aprendizado de Máquina, Aprendizado profundo, Redes Neurais, são bastante famosas e utilizadas pelos usuários da linguagem, podendo ser facilmente importadas para que não haja a necessidade de implementar funções complexas e componentes desde o início.

A alta disponibilidade de bibliotecas e recursos para as áreas de Aprendizado de Máquina e Aprendizado profundo, assim como a facilidade de uso da linguagem foram os principais critérios para escolha do uso dessa linguagem para o desenvolvimento deste trabalho. Abaixo estão listadas as principais bibliotecas Python criadas por terceiros e utilizadas para o desenvolvimento da solução proposta.

4.3.2 Bibliotecas utilizadas da linguagem

- Numpy[3]: utilizada para trabalhar com vetores e matrizes de forma eficiente e facilitada, possibilitando operações matemáticas velozes com essas estruturas em grandes dimensões e complexidades. Otimizada com a linguagem de programação C, Numpy traz velocidade e performance bastante agradável para essas operações no Python.

- Pandas[4]: biblioteca Python para análise e manipulação de dados. Permite gerenciar dados em grandes volumes, aplicando operações matemáticas de diferentes graus de complexidade de uma forma bastante flexível, fácil, rápida e poderosa. Diversas operações presentes nesta biblioteca são construídas em cima de componentes Numpy.

- Matplotlib[5]: biblioteca base para muitas bibliotecas de visualização de dados gráfica, compreende a criação de gráficos estáticos, animados e interativos em Python.

- Seaborn[6]: biblioteca Python para visualização de dados em alto nível. Baseada na biblioteca Matplotlib, Seaborn traz muitas implementações de gráficos prontas como: gráfico de linhas, gráfico de barras, gráfico de pontos de calor, etc.

- Tensor Flow[7]: plataforma de código aberta que facilita a criação de modelos de Aprendizagem de Máquina e Aprendizagem profunda. Tensor Flow traz possibilidades para que o usuário do framework possa trabalhar com modelos prontos ou cria-los de acordo com as necessidades. Com esta biblioteca é possível facilmente estruturar redes neurais simples e complexas, alterando o número de neurônios, função de ativação, adicionando diferentes tipos de camadas, camadas de pré-processamento, entre outros.

[3] https://numpy.org/
[4] https://pandas.pydata.org/
[5] https://matplotlib.org/
[6] https://seaborn.pydata.org/
[7] https://www.tensorflow.org/

- Scikit-learn[8]: também conhecida como Sk Learn, é uma das principais ferramentas para se trabalhar com Aprendizado de Máquina na linguagem Python. Tal biblioteca traz diversas implementações de modelos de regressão, classificação e agrupamento, assim como a implementação de várias métricas de avaliação bastante conhecidas como: acurácia, precisão, matriz de confusão, erro quadrado médio, curva ROC, etc. Dessa forma, o programador que utiliza o Sk Learn pode se preocupar com o uso dos modelos e seus resultados mais do que com a criação do modelo em si.

  Construída a partir da junção de outras bibliotecas como Numpy, Matplotlib e SciPy[9] (biblioteca que implementa métodos estatísticos em Python), a Scikit-learn traz consigo também diversos componentes para otimização/ajuste de hiperparâmetros, possibilitando que modelos eficientes sejam criados e aperfeiçoados para contextos mais específicos.

  Sk Learn traz também algumas implementações de componentes para divisão de dados inteligente de forma mais inteligente e elaborada, como algoritmos para validação cruzada de dados, e algoritmos para normalização dos dados, como normalização padrão, normalização min-max, entre outras. Com essa biblioteca também é possível gerar *datasets* e aplicar técnicas para correção de conjuntos já existentes, por meio de técnicas como a *OneHotEncoder*, por exemplo, a qual é facilmente aplicada com a Sk Learn.

[8] https://scikit-learn.org/stable/index.html
[9] https://scipy.org/

5 Desenvolvimento

5.1 Classificador para fatores de risco

5.1.1 Pré-processamento: preparação dos dados

Antes de iniciar o processo de criação de um modelo preditivo, tanto para classificação quanto para regressão, é necessário preparar os dados que serão utilizados nesse processo. Isto é necessário para que o melhor desempenho deste modelo a ser criado seja alcançado.

> As tecnologias analíticas que podemos utilizar são poderosas, mas impõem determinados requisitos sobre os dados que usam. Com frequência, elas exigem que os dados estejam em uma forma diferente de como são fornecidos naturalmente, e alguma conversão será necessária (PROVOST; FAWCETT, 2016, p. 29-30).

Dessa forma, algumas alterações nos dados foram feitas com intuito de prepara-los para uso correto na criação do classificador. Ao analisar a distribuição da variável alvo do conjunto de dados, foi encontrado a princípio desbalanceamento dos valores. Com 4861 casos de não ocorrência de AVC e apenas 249 casos para ocorrência da doença, seria sintetizado um classificador enviesado para casos de não AVC, classificando dados que receberia de forma tendenciosa.

Para realizar o balanceamento dessas duas classes (avc e não avc) foi realizado o seguinte procedimento:

1. Separação dos elementos do conjunto original em duas classes: avc e não avc.
2. Redução aleatória do conjunto de dados da classe não avc: de 4861 elementos para 275.
3. Junção do conjunto de dados da classe avc (249 elementos) com o novo conjunto de dados reduzido da classe não avc (276 elementos), totalizando 525 elementos.

Portanto, com o balanceamento das classes presentes na variável alvo, tem-se um novo conjunto de dados de 525 elementos com 52.6% dos dados sendo casos de AVC e o restante para casos de condição normal (não AVC). Esse novo conjunto de dados criado foi utilizado ao decorrer de todo este trabalho para desenvolver o modelo classificador.

Com relação ao formato de cada atributo que compõe o conjunto de dados, alguns não apresentavam formato numérico e, portanto, tiveram que ser corrigidos para que não houvesse erros na criação do classificador. As variáveis contínuas presentes no conjunto de dados já estavam com o formato adequado numérico e não precisaram de correção.

Tabela 1 – Frequência de cada possível valor dos atributos categóricos presentes no conjunto de dados.

Atributo	Distribuição das frequências
Gênero	Masculino: 41.71% Feminino: 58.29%
Hipertensão	0: 82.48% 1: 17.52%
Doença cardíaca	0: 88.76% 1: 11.24%
É / já foi casado	Sim: 80% Não: 20%
Tipo de emprego	Setor privado: 62.86% Autônomo: 22.09% Cargo público: 14.29% Nunca trabalhou: 0.76%
Tipo de residência	Urbano: 50.86% Rural: 49.14%
Tabagismo	Nunca fumou: 37.33% Fuma formalmente: 24.57% Fuma constantemente: 22.48% Situação desconhecida: 15.62%
AVC	0: 52.57% 1: 47.43%

Fonte: Elaborado pelo autor.

Para corrigir o formato textual de variáveis categóricas que apresentam apenas duas opções de valores, basta apenas substituir uma delas pelo dígito “1” e a outra pelo ”0”. Já para casos em que as colunas apresentam mais do que dois possíveis valores, é possível realizar a correção das mesmas através da técnica de *One-Hot Encoding*[1]. Tal transformador recebe uma coluna com diferentes possíveis valores textuais e cria novas colunas binárias para cada valor diferente.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

Figura 11 – Exemplo de correção para a coluna com apenas duas opções de valores (variável binária).

Gênero
Masculino
Masculino
Feminino
Feminino
Feminino
Masculino
Feminino

→

Masculino
1
1
0
1
0
0
0

Fonte: Elaborado pelo autor.

Figura 12 – Exemplo de aplicação da técnica *One-Hot Encoding* para correção de uma coluna com quatro opções de valores diferentes.

Tipo de emprego
Privado
Privado
Autônomo
Privado
Criança
Cargo público
Autônomo

One-Hot Encoding →

Privado	Autônomo	Cargo público	Criança
1	0	0	0
1	0	0	0
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0

Fonte: Elaborado pelo autor.

Como visto acima, toda linha obrigatoriamente contém o dígito "1" em pelo menos uma das novas colunas criadas. Isso ocorre, pois, a criação dessas colunas é feita a partir da ocorrência de todos os diferentes possíveis valores, de forma a ter uma nova coluna para cada valor diferente presente. Por essa razão, pode-se deduzir o valor de uma coluna a partir dos valores das demais, já que os valores são exclusivos.

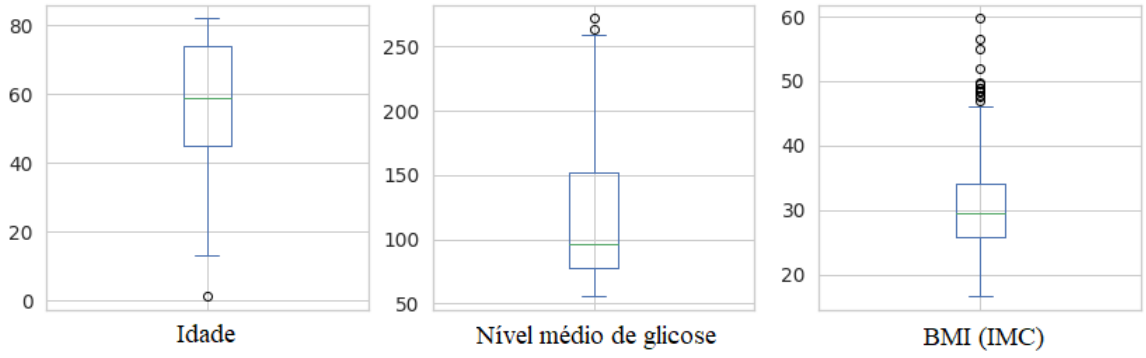
Além do formato numérico das variáveis, também é necessário lidar com a presença de dados nulos no conjunto de dados para evitar erros durante o processo de ajuste do modelo. Dessa forma, cada variável presente no conjunto de dados foi analisada com relação a presença desse tipo de dado e a única que apresentou os dados nulos foi BMI, com 7.6% (40 elementos em um total de 525 registros) de valores desse tipo.

Para correção dos valores nulos encontrados em BMI fora calculada a mediana dos valores não nulos dessa coluna e depois feita a substituição dos valores nulos por essa mediana calculada. A mediana foi escolhida para substituição dos valores nulos pois gerou melhores resultados em comparação com a substituição desses mesmos valores pela média da coluna.

Após tal procedimento, todos os valores nulos do conjunto de dados foram corrigidos, já que só existiam valores assim nessa coluna.

Outro aspecto importante que pode acabar atrapalhando a criação do modelo é a presença de anomalias, ou *outliers*, no conjunto de dados. Tais valores podem atrapalhar a análise de determinado conjunto, gerando ruídos às medidas estatísticas da respectiva distribuição de valores que pertencem. Para identificação desse tipo de valor, foi realizada a análise gráfica das distribuições das variáveis quantitativas do conjunto de dados utilizado.

Figura 13 – Gráfico de distribuição dos atributos contínuos presentes no conjunto de dados com a presença de anomalias, representadas por círculos pretos.



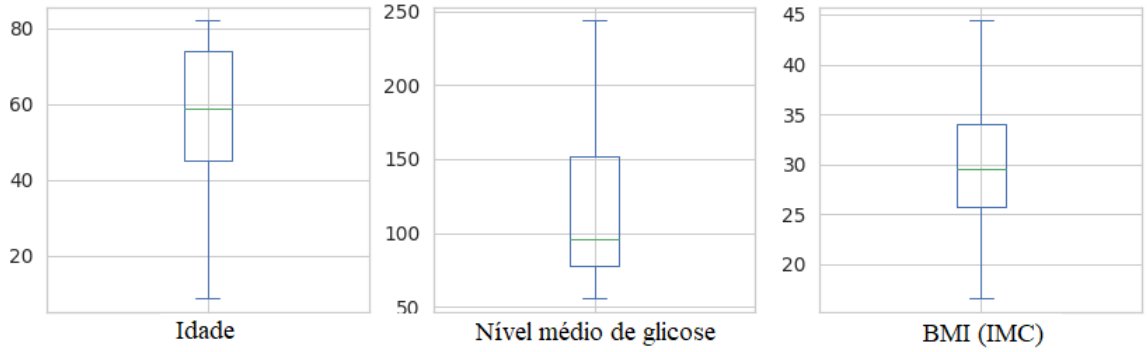
Fonte: Elaborado pelo autor.

Com a identificação das anomalias, agora pode-se iniciar a etapa de tratamento desses valores. Para isso, fora calculado o *Interquartile range* (Amplitude interquartil, em português), ou simplesmente IQR, dos atributos quantitativos. O IQR é utilizado para avaliar o grau de dispersão dos valores em torno da medida de centralidade do conjunto. Com tal valor calculado (IQR = 3° quartil – 1° quartil), foram determinados o limite superior e o limite inferior dos valores de cada atributo da seguinte forma:

1. Limite superior: Mediana + 1.5 x IQR
2. Limite inferior: Mediana - 1.5 x IQR

Com tais medidas calculadas, tornou-se possível substituir todo valor maior que o limite superior pelo próprio limite superior e todo valor menor que o limite inferior pelo próprio limite inferior. Após a realização desses procedimentos, as anomalias foram corrigidas e o resultado obtido está expresso logo abaixo:

Figura 14 – Gráfico de distribuição dos atributos contínuos presentes no conjunto de dados sem a presença de anomalias, após as alterações realizadas nos valores.



Fonte: Elaborado pelo autor.

Como algumas variáveis numéricas apresentaram ainda intervalos diferentes, a normalização foi realizada para que tal fato não prejudicasse a criação do classificador. Modelos de aprendizado de máquina lineares podem acabar sofrendo com a falta de padronização dos intervalos. Uma variável que varia de 0 a 80, enquanto outra varia de 50 a 200, por exemplo, pode acabar prejudicando o modelo linear. Por essa razão, nesses casos é interessante se realizar a normalização.

> A normalização é uma técnica geralmente aplicada como parte da preparação de dados para o aprendizado de máquina. O objetivo da normalização é mudar os valores das colunas numéricas no conjunto de dados para usar uma escala comum, sem distorcer as diferenças nos intervalos de valores nem perder informações. A normalização também é necessária para alguns algoritmos para modelar os dados corretamente. (MICROSOFT, 2022)

A normalização evita na modelagem problemas com tais diferenças nos intervalos das variáveis. Ela mantém os valores originais, porém em uma escala padrão. Como a normalização utilizada aqui foi a de máximos e mínimos, através da função *sklearn.preprocessing.MinMaxScaler()*[2], o padrão de escala ficou entre 0 e 1.

5.1.2 Modelagem: criação do classificador

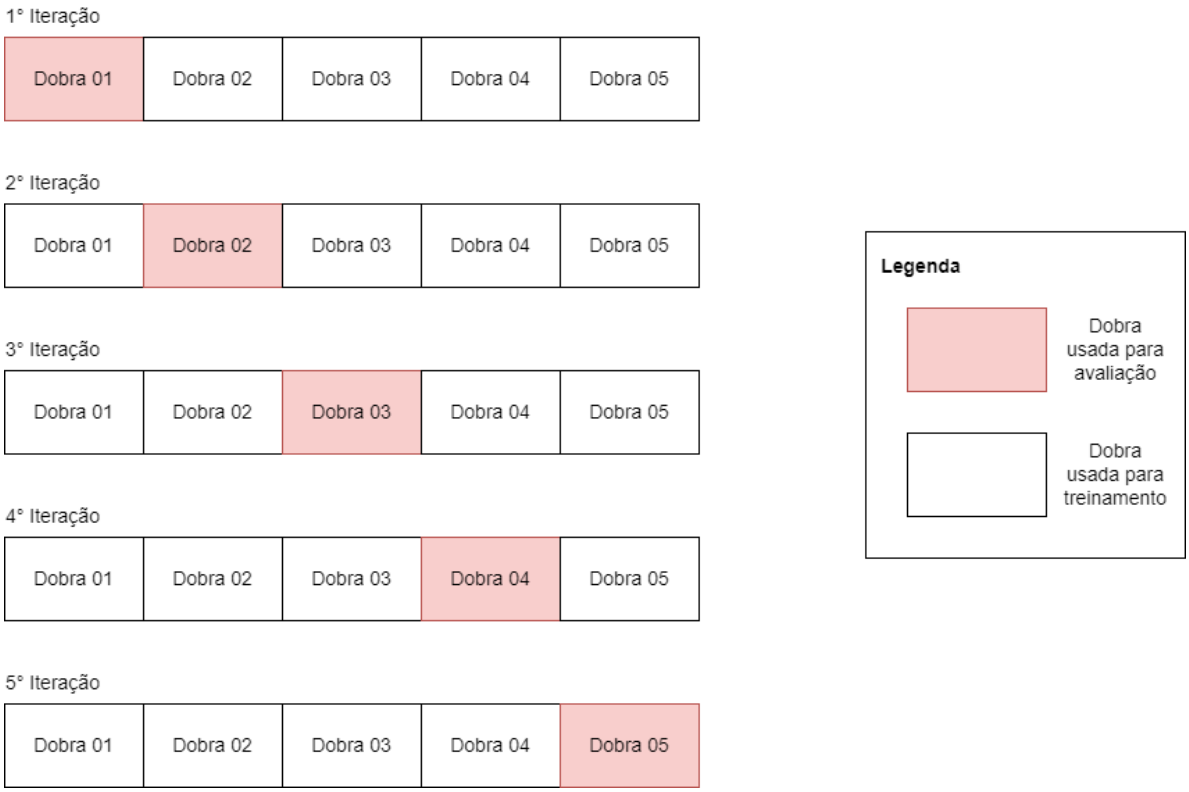
Com a etapa de preparação dos dados feita, o conjunto de dados está pronto para ser processado. Para iniciar o processo de criação do modelo, a modelagem, o conjunto total de dados disponível foi dividido em partes distintas de treino e teste, como forma de evitar o sobreajuste do modelo criado a partir da análise de tais dados. Utilizando uma parte diferente para treino e uma diferente para teste, a avaliação do modelo é feita a partir de dados desconhecidos pelo modelo, mostrando um resultado mais verdadeiro e confiável.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Porém, apenas essa divisão pode não ser suficiente para conjuntos pequenos, pois mesmo criando os conjuntos de treino e teste de forma aleatória, pode ocorrer a seleção de valores mais representativos dependendo da semente definida para aleatoriedade. Por essa razão, uma boa maneira de lidar com conjuntos pequenos é fazer várias divisões de treino e teste no conjunto, pois assim, treina-se o modelo e o avalia com diferentes subconjuntos de dados, trazendo resultados ainda mais confiáveis. Uma forma bastante comum de se fazer isso é através do uso da Validação Cruzada e essa foi a forma utilizada aqui.

A divisão dos elementos foi feita com cinco dobras: os 525 elementos presentes no conjunto de dados total foram divididos em cinco grupos de 125 elementos cada e, então, foram separados quatro deles para treinamento do modelo e um para teste. Esse processo de separação foi repetido por cinco vezes de forma que o modelo fosse novamente treinado e testado com diversas combinações de conjuntos. Essa validação cruzada foi utilizada para criação de dois modelos classificadores diferentes, um a partir do algoritmo de Regressão Logística e outro a partir do algoritmo de Floresta Aleatória.

Figura 15 – Ilustração do processo de validação cruzada, com cinco dobras, feito para treinamento e avaliação dos dois classificadores criados. A junção das cinco dobras é o conjunto de dados total utilizado.



Fonte: Elaborado pelo autor.

O classificador com Regressão Logística foi criado a partir da classe *LogisticRegression()*[3]

[3]https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

do Scikit-Learn, com o valor 7 definido como semente para geração de valores aleatórios e os demais parâmetros com valores padrões da biblioteca. O modelo para classificação com Floresta Aleatória foi criado a partir da classe *RandomForestClassifier()*[4],com a semente para geração de valores aleatórios também sendo 7, com duzentas árvores de decisão e com os demais parâmetros com valores padrões da biblioteca.

5.2 Rede Neural classificadora de imagens de tomografia computadorizada

5.2.1 Correção da rotulação dos dados e criação dos conjuntos de treino e de validação

O conjunto de dados original divide as imagens de tomografia computadorizada nas categorias de condição normal, condição de avc hemorrágico e condição de avc isquêmico. Porém, para abordar de forma mais condizente com o trabalho proposto, uma nova rotulação foi feita. As imagens foram movidas para uma nova distribuição de duas classes: avc (hemorrágico e isquêmico) e não avc (condição normal).

Sobre o tamanho desse novo conjunto gerado a partir da nova rotulação das classes feita, a classe avc possui 299 imagens enquanto que a classe não avc possui 174, sendo que a dimensão de cada imagem do conjunto é de 512 x 512 pixels. A respeito da distribuição dessas imagens para treino e validação, com intuito de evitar sobre ajuste, o conjunto de imagens foi distribuído como 65% (309 imagens) dos dados para treinamento e 35% (166 imagens) para validação. Dessa forma, a rede pode aprender com os dados de treinamento e ser avaliada com os dados de validação, os quais ela ainda não conhece por não terem sido usados na etapa de treino.

Tabela 2 – Distribuição das imagens do conjunto de dados antes da nova rotulação.

	Condição	Imagens
Rotulação original	Normal	174 (36.8%)
	AVC Hemorrágico	142 (30%)
	AVC Isquêmico	157 (33.2%)

Fonte: Elaborado pelo autor.

[4]https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees

Tabela 3 – Distribuição das imagens do conjunto de dados depois da nova rotulação.

	Condição	Imagens
Nova rotulação	AVC	299 (63.2%)
	não AVC	174 (36.8%)

Fonte: Elaborado pelo autor.

5.2.2 Pré-processamento: redimensionamento dos valores e aumento dos dados do conjunto de dados original

Como o conjunto de dados utilizado aqui é composto por imagens, tem-se então um agrupamento de pixels, uma matriz, para cada imagem. Um pixel possui sua coloração definida a partir de um valor, para cada canal de cor, que varia de 0 a 255, totalizando 256 elementos. Embora seja possível realizar o treinamento de uma rede com imagens compostas por pixels de valores entre 0 a 255, redimensionar esse intervalo para valores de 0 a 1 trazem melhores resultados para o modelo, pois os valores que serão multiplicados pelos pesos dos nós acaba sendo menor.

Portanto, foi realizado o redimensionamento dos dados por meio da adição de uma camada inicial na rede do tipo *Rescaling*[5] com valor para divisão de 255. Dessa forma, toda imagem (matriz) que entra na rede já tem seus pixels redimensionados, com seus valores divididos por 255, que é o maior valor possível presente nos pixels.

Outra etapa do pré-processamento realizado foi o aumento de informação do conjunto de dados para que melhores resultados fossem obtidos. Como mencionado na seção anterior, o conjunto de dados disponível conta com 473 imagens no total. Embora seja possível realizar o treinamento de uma rede neural com esse número de registros, aumentar o número de imagens, e consequentemente aumentar a quantidade de informação passada para a rede, pode gerar resultados mais satisfatórios e uma rede neural mais eficiente.

Uma forma bastante comum no campo do Aprendizado Profundo de aumentar os dados de um conjunto, quando não se tem a possibilidade de adicionar novas imagens externas ao conjunto em questão, é utilizar técnicas de aumento dos dados (*data augmentation*[6]), de modo que a quantidade de informação no conjunto cresça a partir das imagens já presentes no conjunto, sem que de fato sejam adicionadas novas imagens no conjunto.

[5] https://www.tensorflow.org/api_docs/python/tf/keras/layers/Rescaling
[6] https://www.tensorflow.org/tutorials/images/data_augmentation

Para realizar o aumento da informação do conjunto de dados foram adicionadas camadas no início da rede neural que fazem algumas alterações no conteúdo da imagem, caracterizando uma nova imagem, porém sem prejudicar as informações usadas para a classificação. Ao todo foram três camadas que aplicam inversão, rotação e ajustes de contraste nas imagens recebidas, respectivamente.

Tabela 4 – Alterações feitas para aumento dos dados do conjunto de imagens utilizado.

Alteração aplicada	Fator	Método utilizado
Inversão (*flip*)	Horizontal/ vertical	*tensorflow.keras.layers.RandomFlip*("horizontal_and_vertical", seed=10)
Rotação	0.2	*tensorflow.tf.keras.layers.RandomRotation*(0.2, seed=7)
Contraste	Valor entre [0.3, 0.5]	*tensorflow.keras.layers.RandomContrast*(factor=(0.3, 0.5), seed=(7,17))

Fonte: Elaborado pelo autor.

5.2.3 Estrutura da Rede Neural Convolucional desenvolvida

Para desenvolver a solução proposta neste trabalho, foi criada uma Rede Neural Convolucional para classificação binária das imagens de TC nas classes de *avc* e *não avc*. A estrutura dessa rede conta com as camadas que implementam o pré-processamento, comentado nas seções anteriores, e com a transferência de aprendizado de uma rede VGG-16 pré-treinada, contendo as camadas de convolução e de agrupamento dessa arquitetura, com parâmetros já definidos e pesos já gerados a partir do conjunto de dados *imagenet*[7], que contém diversas imagens pré-rotuladas para treinamentos avançados de Aprendizado Profundo.

Além da VGG-16, a rede desenvolvida neste trabalho conta também com uma camada de achatamento (*flatten*), com duas camadas densas de neurônios com ativação *ReLU*, com 64 e 32 neurônios, respectivamente, e com uma última camada contendo apenas um neurônio, tendo este como função de ativação a função sigmoide, a qual gera o valor usado para classificação final de determinada imagem processada pela rede.

[7] https://www.image-net.org/

Tabela 5 – Estrutura final da Rede Neural Convolucional criada para imagens RGB de dimensão 512x512 pixels.

	Componentes da rede	Principal função do componente	Tamanho da saída
1	Camada de entrada e redimensionamento	Redimensionamento de cada pixel de 0 a 255 para 0 a 1	(512, 512, 3)
2	Camada de inversão	Inversão aleatória (horizontal/vertical)	(512, 512, 3)
3	Camada de rotação	Rotação de 0.2 com sentido aleatório	(512, 512, 3)
4	Camada de contraste	Contraste aleatório com fator de 0.3 a 0.5	(512, 512, 3)
5	Camadas de convolução e agrupamento VGG-16	Transferência de aprendizado de uma CNN VGG, com pesos fixos pré-gerados a partir do conjunto *imagenet*	(16, 16, 512)
6	Camada de achatamento	Redimensiona saída para apenas um valor	(131072)
7	Camada densa de neurônios	64 neurônios com função de ativação ReLU	(64)
8	Camada densa de neurônios	32 neurônios com função de ativação ReLU	(32)
9	Camada densa de neurônios	Apenas 1 neurônio com função de ativação sigmoide	(1)

Fonte: Elaborado pelo autor.

5.2.4 Treinamento da rede estruturada

Para realizar o treinamento da rede criada, fora feita antes a compilação do modelo, definindo a forma de calculo do erro como entropia cruzada binária e uma taxa de aprendizado de 0.001. Ademais, fora definida uma função *call-back* para interromper o treinamento caso o modelo alcançasse acurácia de validação de 95% e AUC ROC de validação de 0.9.

O treinamento foi feito a partir do conjunto de dados de treino, composto por dados desconhecidos pelo modelo, com 12 épocas (iterações que percorrem todo o conjunto de dados) e com tamanho do lote (*batch size*) de 32 imagens por iteração do treinamento de cada época. As camadas da rede pré-gerada VGG-16, integrada na rede via transferência de aprendizado, não tiveram seus pesos alterados durante o treinamento feito, pois manter os pesos originais resultou em melhores resultados em comparação com treiná-los novamente a partir do conjunto de dados utilizado neste trabalho.

6 Resultados

O processo de avaliação é uma das principais partes de qualquer projeto que envolva a criação de modelos de preditivos. Após a criação dos classificadores, é preciso analisar seus desempenhos e, assim, medir a eficiência de cada um. É de grande importância que este processo seja feito de forma correta para que os resultados da avaliação não sejam enganos e não escondam falhas do modelo.

6.1 Resultados dos modelos criados para classificação de fatores de risco

Como forma de medir o desempenho geral dos classificadores criados, a acurácia de cada um foi calculada a partir da função *accuracy_score()*[1], da biblioteca Scikit-Learn, para cada dobra de teste utilizada na avaliação de ambos os modelos. Como consequência, tem-se, portanto, uma sequência de cinco valores e a acurácia final foi definida como a média desses valores.

Tabela 6 – Valor médio e desvio padrão das acurácias calculadas em cada iteração de teste da validação cruzada, para ambos os classificadores criados.

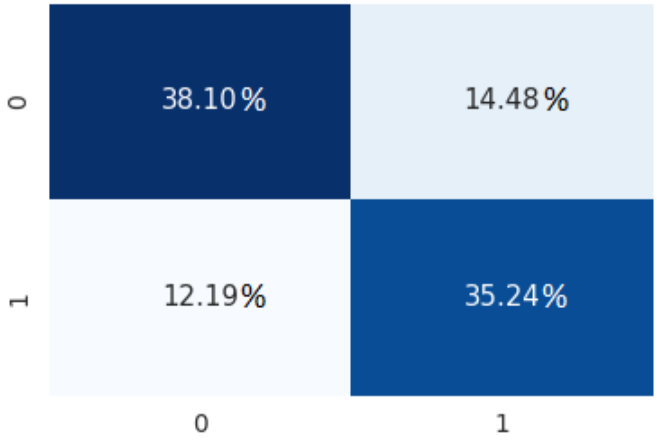
Acurácia do classificador	Regressão Logística	Média: 73.33% Desvio padrão: 2.48%
	Floresta Aleatória	Média: 73.14% Desvio padrão: 1.40%

Fonte: Elaborado pelo autor.

Também foi gerada e armazenada, para cada dobra de teste da validação cruzada feita para criação dos modelos, uma matriz de confusão. Após a execução de todas as iterações da validação, a média e o desvio dos valores foram calculados e os resultados, para o modelo de Regressão Logística e para o modelo de Floresta Aleatória, podem ser vistos nas figuras a seguir.

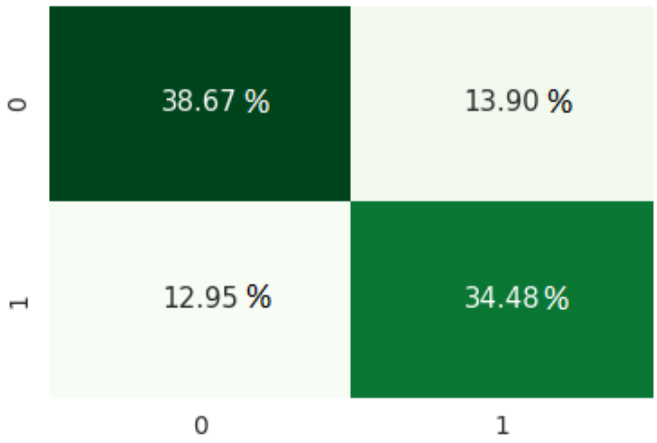
[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

Figura 16 – Matriz de confusão para o modelo de Regressão Logística, criado a partir da média dos valores obtidos nos testes da validação cruzada.



Fonte: Elaborado pelo autor.

Figura 17 – Matriz de confusão para o modelo de Floresta Aleatória, criado a partir da média dos valores obtidos nos testes da validação cruzada.



Fonte: Elaborado pelo autor.

Além da visualização da frequência de cada tipo de erro e acerto feita pela geração das matrizes de confusão de cada modelo, foi feita também a análise das taxas de falso positivo e verdadeiro positivo para os dois modelos:

Tabela 7 – Média e desvio padrão da taxa de valores verdadeiros positivos gerados em cada iteração de teste da validação cruzada.

TPR	Regressão Logística	Média: 74.53% Desvio padrão: 74.53%
	Floresta Aleatória	Média: 72.95% Desvio padrão: 4.11%

Fonte: Elaborado pelo autor.

Tabela 8 – Média e desvio padrão da taxa de valores falsos positivos gerados em cada iteração de teste da validação cruzada.

FPR	Regressão Logística	Média: 27.28% Desvio padrão: 5.27%
	Floresta Aleatória	Média: 26.23% Desvio padrão: 3.99%

Fonte: Elaborado pelo autor.

Para saber sobre o desempenho dos modelos com relação a TPR e FPR a partir da variação de limiares de classificação para o modelo, ou seja, se o modelo apresenta boa capacidade de classificação com relação a dados positivos e negativos, foi calculada a área sob a curva ROC de cada modelo, através da função *roc_auc_score()*[2], após a validação cruzada feita:

Tabela 9 – Média e desvio padrão dos valores de AUC ROC calculados em cada iteração de teste da validação cruzada para ambos os modelos.

AUC ROC	Regressão Logística	Média: 0.8029 Desvio padrão: 0.0506
	Floresta Aleatória	Média: 0.7982 Desvio padrão: 0.0349

Fonte: Elaborado pelo autor.

[2]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html

6.2 Resultados da rede neural criada para classificação de imagens de tomografia computadorizada

Para medir a eficiência da rede neural desenvolvida, as mesmas métricas da seção anterior foram calculadas novamente para o novo cenário. A acurácia atingida pelo classificador demonstrou boa capacidade de acerto geral do mesmo, com relação a ambos os conjuntos, de treino e de validação, processados.

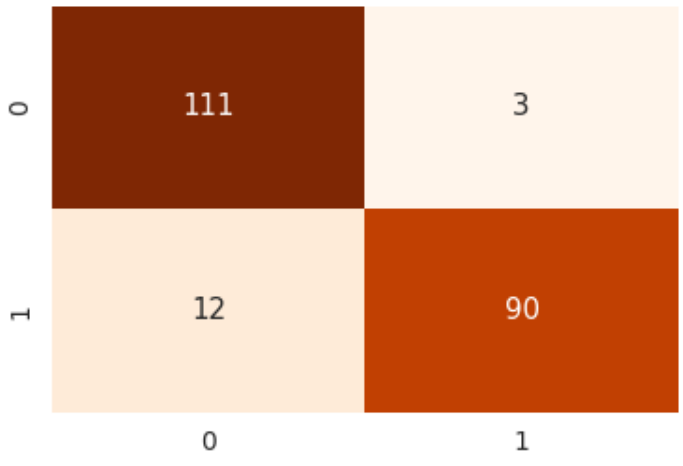
Tabela 10 – Acurácia alcançada na classificação dos dados utilizados para treinamento e validação da rede neural desenvolvida.

Acurácia da Rede	Dados de treino	93.06%
	Dados de validação	97.39%

Fonte: Elaborado pelo autor.

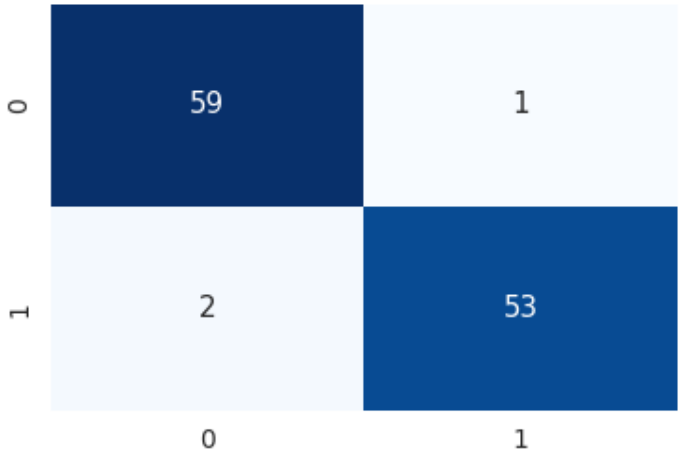
Com relação aos tipos de erros, o classificador de imagens de TC apresentou bons resultados quanto ao equilíbrio de ambos, mantendo casos de falso positivo e falso negativo com baixas frequências de ocorrência. Duas matrizes de confusão foram calculadas para análise desses valores, uma para dados de treinamento, já conhecidos pelo modelo por conta da fase de ajuste, e uma para dados de validação, com as rotulações não conhecidas pela rede neural.

Figura 18 – Matriz de confusão com os valores gerados na classificação feita pela rede a partir do conjunto de dados de treino, com valores conhecidos pelo classificador.



Fonte: Elaborado pelo autor.

Figura 19 – Matriz de confusão com os valores gerados na classificação feita pela rede a partir do conjunto de dados de validação, com valores desconhecidos pelo classificador.



Fonte: Elaborado pelo autor.

As taxas de verdadeiro positivo e falso positivo também foram calculadas para os dados de treinamento e para os dados de validação, com intuito de fornecer uma análise mais precisa desses casos. Os valores obtidos indicam bons resultados, com grandes taxas de acertos para casos positivos e baixas taxas de falsos positivos.

Tabela 11 – Taxa de valores verdadeiros positivos alcançada na classificação dos dados utilizados para treinamento e validação da rede neural desenvolvida.

TPR	Dados de treino	88.24%
	Dados de validação	96.36%

Fonte: Elaborado pelo autor.

Tabela 12 – Taxa de valores falsos positivos alcançada na classificação dos dados utilizados para treinamento e validação da rede neural desenvolvida

FPR	Dados de treino	2.63%
	Dados de validação	1.67%

Fonte: Elaborado pelo autor.

Para analisar o desempenho da rede neural criada com relação ao cálculo de TPR e FPR a partir da variação de limiares de classificação, foi calculada a área sob a curva ROC para os dados, conhecidos, de treino e também para os dados, não conhecidos, de validação. Os resultados mostram que a rede apresenta bons acertos mesmo com a variação do limiar, indicando boa capacidade de classificação da mesma.

Tabela 13 – AUC ROC calculadas na classificação dos dados utilizados para treinamento e validação da rede neural desenvolvida.

AUC ROC	Dados de treino	0.9802
	Dados de validação	0.9944

Fonte: Elaborado pelo autor.

7 Conclusão

O trabalho de conclusão de curso desenvolvido neste documento teve como principal intuito propor formas de classificar casos de Acidente Vascular Cerebral a partir da análise de informações, fatores de risco e imagens de tomografia computadorizada (TC), sobre vítimas da doença. Para isso, em um primeiro momento foi realizado o levantamento bibliográfico do problema, realizando estudos sobre a doença em si para saber como ela se caracteriza no indivíduo, quais são os motivos para sua ocorrência, qual o comportamento das vítimas e os principais fatores de risco englobados na doença.

Além de conhecer sobre o Acidente Vascular Cerebral, também foram realizadas pesquisas bibliográficas sobre as áreas de Aprendizagem de Máquina (*Machine Learning*) e Aprendizagem Profunda (*Deep Learning*), conhecendo sobre técnicas importantes dessas áreas para o desenvolvimento da solução proposta. Ademais, foram pesquisados conjuntos de dados sobre fatores de risco e imagens de TC para serem utilizados no treinamento dos modelos classificadores, que compõem a principal parte da solução desenvolvida.

Para o caso da classificação de fatores de risco, dois modelos foram desenvolvidos utilizando os classificadores de Regressão Logística e Floresta Aleatória, e seus resultados foram bastante próximos em suas avalições de desempenho. A acurácia calculada foi de 73.33% para o modelo de Regressão Logística com desvio padrão de 2.48% para cada caso de teste avaliado. Já para o modelo de Floresta Aleatória, a acurácia calculada foi de 73.14% com desvio de 1.4% para cada teste avaliado. Esses valores indicam que os mesmos acertam uma quantidade significativa dos casos que avaliam com boa capacidade de generalização e suas performance foram semelhantes. O valor médio da *AUC ROC* de 0.8 calculado para ambos os modelos demonstra que as decisões de classificação feitas pelos modelos desenvolvidos são eficientes mesmo variando os limiares de classificação.

Os valores próximos das Taxas de Verdadeiros Positivos, *TPR*, e de Falsos positivos, *FPR*, calculados para os classificadores sugerem que as frequências de ocorrências de *TP*, *FP*, *TN* e *TP* foram parecidas para ambos os modelos, e as matrizes de confusão geradas na avaliação dos dois classificadores comprovam isso. Além disso, a *TPR* média calculada de 74.5% para o classificador de Regressão Logística e a média de 73% para o modelo de Floresta Aleatória demonstram bons desempenhos dos modelos criados em relação ao acerto de casos positivos, o que é bastante interessante no contexto do problema abordado.

Para o caso de processamento e classificação de imagens de TC, a Rede Neural do tipo Convolucional desenvolvida apresentou bons resultados na classificação dos dados processados. A acurácia de $96,76\%$ para dados de treinamento, conhecidos, e $99,13\%$ para dados de teste, não conhecidos, indicam que a rede criada consegue acertar a grande maioria dos dados que

analisa. O valor calculado de AUC ROC de 0.9874 para treinamento e 0.9997 para teste indicam que o alto desempenho da rede se mantém mesmo com a variação do limiar de classificação, indicando que a mesma consegue distinguir bem imagens de TC dos casos de *avc* e dos casos de *não avc*.

Portanto, foram desenvolvidos modelos classificadores de fatores de risco e de imagens de tomografia computadorizada que apresentaram resultados satisfatórios, abrangendo técnicas da área de Inteligência Artificial e cumprindo com o intuito deste trabalho. Embora bons resultados tenham sido gerados neste trabalho, vale ressaltar que os classificadores desenvolvidos são apenas demonstrações de formas de diagnóstico da doença abordada, existindo apenas para fins demonstrativos e não de fato como soluções do problema em si.

Como o problema em questão, o AVC, é ligado diretamente a questões de medicina e saúde, os classificadores criados não são soluções para autodiagnostico da doença e nem são substitutos de profissionais da saúde. É extremamente recomendado que em caso de qualquer suspeita, o indivíduo procure assistência médica e não negligencie a situação, para que assim problemas e prejuízos possam ser evitados.

Referências

AWS. *O que é uma rede neural?* 20–. Disponível em: https://aws.amazon.com/pt/what-is/neural-network/. Acesso em: 28 novembro 2022.

BAGHERI, R. *ROC Curve, a Complete Introduction*. 2019. Disponível em: https://towardsdatascience.com/roc-curve-a-complete-introduction-2f2da2e0434c. Acesso em: 10 novembro 2022.

BVS, B. V. em S. *Acidente vascular cerebral (AVC)*. Ministério da Saúde, 2015. Disponível em: https://bvsms.saude.gov.br/avc-acidente-vascular-cerebral/. Acesso em: 27 outubro 2022.

COPELAND, M. *Qual é a Diferença entre Inteligência Artificial, Machine Learning e Deep Learning?* NVIDIA, 2021. Disponível em: https://blog.nvidia.com.br/2021/03/10/qual-e-a-diferenca-entre-inteligencia-artificial-machine-learning-e-deep-learning/. Acesso em: 24 dezembro 2022.

DEEPCHECKS. *False Positive Rate*: What is the false-positive rate in machine learning? 20–. Disponível em: https://deepchecks.com/glossary/false-positive-rate/. Acesso em: 23 dezembro 2022.

DEVELOPERS, G. *Classificação: acurácia*. 2022. Disponível em: https://developers.google.com/machine-learning/crash-course/classification/accuracy. Acesso em: 23 dezembro 2022.

FERGUSON, M.; AK, R.; LEE, Y.-T.; LAW, K. Automatic localization of casting defects with convolutional neural networks. In: . [S.l.: s.n.], 2017. p. 1726–1735.

HONDA, H.; FACURE, M.; YAOHAO, P. *Os Três Tipos de Aprendizado de Máquina*. 2017. Disponível em: https://lamfo-unb.github.io/2017/07/27/tres-tipos-am/. Acesso em: 10 novembro 2022.

IBM. *Convolutional Neural Networks*. 20–. Disponível em: https://www.ibm.com/topics/convolutional-neural-networks. Acesso em: 6 dezembro 2022.

IBM. *What is a Decision Tree?* 20–. Disponível em: https://www.ibm.com/topics/decision-trees. Acesso em: 20 dezembro 2022.

IBM. *What is logistic regression?* 20–. Disponível em: https://www.ibm.com/topics/logistic-regression. Acesso em: 10 novembro 2022.

JORDAN, J. *Common architectures in convolutional neural networks*. 2018. Disponível em: https://www.jeremyjordan.me/convnet-architectures/. Acesso em: 6 dezembro 2022.

MICROSOFT. *Componente Normalizar Dados*. 2022. Disponível em: https://learn.microsoft.com/pt-br/azure/machine-learning/component-reference/normalize-data. Acesso em: 11 dezembro 2022.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: *Sistemas Inteligentes Fundamentos e Aplicações*. 1. ed. Barueri-SP: Manole Ltda, 2003. p. 89–114. ISBN 85-204-168.

NAVLANI, A. *Understanding Random Forests Classifiers in Python Tutorial*: Learn about random forests and build your own model in python, for both classification and regression. 2018. Disponível em: https://www.datacamp.com/tutorial/random-forests-classifier-python. Acesso em: 17 novembro 2022.

NURFIKRI, F. *An Illustrated Guide to Artificial Neural Networks*. 2020. Disponível em: https://towardsdatascience.com/an-illustrated-guide-to-artificial-neural-networks-f149a549ba74. Acesso em: 28 novembro 2022.

OLIVEIRA, R. de Magalhães Carneiro de; ANDRADE, L. A. F. de. Acidente vascular cerebral. *Rev Bras Hipertens*, v. 8, n. 10, p. 280–285, 2001. Disponível em: http://departamentos.cardiol.br/dha/revista/8-3/acidente.pdf. Acesso em: 25 outubro 2022.

O'SHEA, K.; NASH, R. *An Introduction to Convolutional Neural Networks*. arXiv, 2015. Disponível em: https://arxiv.org/abs/1511.08458.

PROVOST, F.; FAWCETT, T. *Data Science para Negócios: O que Você Precisa Saber Sobre Mineração de Dados e Pensamento Analítico de Dados*. 1. ed. Rio de Janeiro: Alta Books, 2016. 383 p.

RUEDA, F. *Como identificar o AVC isquêmico na tomografia computadorizada? [vídeo]*. 2022. Disponível em: https://pebmed.com.br/como-identificar-o-avc-isquemico-na-tomografia-computadorizada/. Acesso em: 25 outubro 2022.

SCIKIT-LEARN. *Linear Models*: Logistic regression. entre 2007 e 2022. Disponível em: https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression. Acesso em: 12 novembro 2022.

SCIKIT-LEARN. *Random Forest Classifier*. entre 2007 e 2022. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Acesso em: 20 novembro 2022.

TAMAKE, B. L. *ANÁLISE DE DADOS PARA AUXILIAR NO DIAGNÓSTICO PRECOCE DE ACIDENTE VASCULAR CEREBRAL - AVC*. 52 f. Monografia (Graduação) — Faculdade de Ciências, Universidade Estadual Paulista Júlio de Mesquita Filho UNESP, Bauru, 2020.

TIWARI, T.; TIWARI, T.; TIWARI, S. How artificial intelligence, machine learning and deep learning are radically different? *International Journal of Advanced Research in Computer Science and Software Engineering*, v. 8, p. 1, 03 2018.

VARELLA, M. *NEURÔNIO*. 20–. Disponível em: https://drauziovarella.uol.com.br/corpo-humano/neuronio/. Acesso em: 27 novembro 2022.

YACIM, J.; BOSHOFF, D. Impact of artificial neural networks training algorithms on accurate prediction of property values. *Journal of Real Estate Research*, v. 40, p. 375–418, 11 2018.