Uso de redes neurais para o problema de previsão de pacientes de alto custo

Franklin Messias Barbosa

Uso de redes neurais para o problema de previsão de pacientes de alto custo

Franklin Messias Barbosa

Orientador: *Profº Drº Renato Porfirio Ishii*

Monografia entregue a Faculdade de Computação da Universidade Federal de Mato Grosso do Sul - FACOM-UFMS como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

UFMS - Campo Grande
Julho/2021

Agradecimentos

Agradeço aos meus pais, por terem sempre me apoiado e me motivado a seguir em frente.

Ao meu orientador Renato Porfirio Ishii, pelos conselhos, ajuda e tempo dedicado.

A todos os professores e funcionários da Facom que contribuíram de forma direta ou indireta com a minha formação e meu aperfeiçoamento profissional e pessoal.

Aos amigos que ganhei durante o tempo passado na UFMS.

Aos professores, colegas e funcionários que fazem parte dos laboratórios CTEI e PET-Sistemas.

Ao projeto InterSCity, do qual este trabalho faz parte, pelo apoio financeiro na publicação dos resultados aqui obtidos.

Gostaria também de agradecer pela bolsa concedida pela CAPES durante o período de desenvolvimento deste trabalho.

Abstract

The growing aging of the world population, along with several environmental, social, and economic factors, end up posing major challenges for public health in general. Within this scenario, it is of interest for both private health insurance operators and public health managers to better manage available resources to reach the largest possible share of society. To do so, keeping in mind the amount of information produced daily, it is also clear the need for data processing and decision support technologies so that such management can be done satisfactorily.

This study aims to analyze the application of machine learning and deep learning techniques in health care scenarios. One of the possible applications includes the detection of possible high-cost patients from historical data, to better target interventions that may prevent the transition of regular patients into high-cost ones or, in the case of those who are already in this condition, to allow appropriate approaches, rather than generic ones. In both cases, the detection of such patients can be beneficial, reducing avoidable costs and improving patients' condition.

The final model, chosen to predict the high-cost condition was a fully connected sequential network, with 3 hidden layers and 3 dropout layers. That network had 88% on accuracy and f1 score metrics, 91% on recall, 86% on precision and 84% specificity, showing the model's capacity to correctly classify examples from both classes. This work also aimed to make the creation and testing of such networks easier, by providing the tools developed during its evolution on GitHub.

Keywords: Machine Learning, Deep Learning, Big Data, Classification

Resumo

O crescente envelhecimento da população mundial, juntamente com diversos fatores ambientais, sociais e econômicos, acabam gerando grandes desafios para a saúde pública em geral. Dentro deste cenário é de interesse tanto para operadoras de planos de saúde privados quanto para gestores da saúde pública um melhor gerenciamento dos recursos disponíveis, a fim de atingir a maior parcela possível da sociedade. Para isso, tendo em mente a quantidade de informações produzidas diariamente, fica evidente também, a necessidade do uso de tecnologias de processamento de dados e auxílio à tomada de decisões para que tal gerenciamento seja feito de maneira satisfatória.

Este trabalho tem como objetivo analisar a aplicação de técnicas de machine learning na área da saúde. Uma das possíveis aplicações inclui a detecção de possíveis pacientes de alto custo a partir de dados históricos, a fim de melhor direcionar intervenções que venham a evitar a transição de pacientes regulares em pacientes de alto custo, ou, no caso daqueles que já estão nessa condição, permitir abordagens apropriadas ao invés de genéricas. Em ambos os casos, a detecção de tais pacientes pode ser benéfica, reduzindo custos evitáveis e melhorando a condição dos pacientes.

A fim de realizar tais detecções, este trabalho se concentrou no uso de técnicas de machine learning, especificamente, Redes Neurais, juntamente com um conjunto de dados composto por respostas de pesquisas aplicadas pelo governo dos Estados Unidos, denominado *Medical Expenditure Panel Survey* (MEPS) e atributos coletados a partir da literatura.

O modelo final escolhido para prever a condição de alto custo foi uma rede neural sequencial totalmente conectada, com 3 camadas ocultas e 3 camadas de dropout. Esta rede obteve 88% nas métricas de *accuracy* e *f1 score*, 91% na métrica de *recall*, 86% de *precision* e 84% de *specificity*, demonstrando a capacidade do modelo de prever corretamente exemplos de ambas as classes. Este trabalho teve também como objetivo facilitar a criação e o teste dessas redes, disponibilizando as ferramentas desenvolvidas durante sua evolução no

GitHub.

Palavras-chave: Machine Learning, Deep Learning, Big Data, Classificação

Lista de Figuras

Lista de Tabelas

Conteúdo

CAPÍTULO

1

Introdução

1.1 Contextualização

A saúde pública é uma área de grande importância tanto no cenário mundial quanto nacional, o que torna importante também o gerenciamento eficiente dos recursos disponíveis nessa área. Um dos desafios que pode ser observado nesse quesito, tanto em contextos públicos quanto privados, são os pacientes de alto custo, uma pequena parcela dos usuários que representa uma grande parte dos gastos. Tal fenômeno pode ser observado não apenas no Brasil [4], mas também em diversos outros países, como Canadá [5] e EUA [6].

Em relatório publicado no ano de 2013, a Kaiser Family Foundation[1] aponta que no Medicaid [6], plano de saúde público dos Estados Unidos, 5% dos beneficiados de maior custo somam 54% das despesas totais [7].

Dessa forma, uma possível abordagem seria a identificação de tais pacientes enquanto eles ainda apresentam custos médios, ou seja, antes que se tornem de alto custo, como apontado por Chechulin [8]. Tal identificação pode ser útil como guia em intervenções direcionadas, com o intuito de prevenir a transição do paciente, prevenindo a complicação de quadros já presentes e evitando o aparecimento de novos problemas quando possível, melhorando assim a eficiência dos agentes responsáveis pela saúde e prevenindo gastos.

De forma similar, a identificação de pacientes que já são de alto custo é benéfica, como colocado por Blumenthal [9], "pacientes de alto custo e alta necessidade são, normalmente, pessoas que mesmo recebendo serviços subs-

[1] A Kaiser Family Foundation é uma organização americana sem fins lucrativos, que atua em questões de saúde pública nos EUA

tanciais, possuem necessidades críticas de saúde que não são atendidas. Essa parcela da população frequentemente recebe cuidados ineficientes, como internações desnecessárias". Segundo Blumenthal, dando uma prioridade alta aos cuidados desses indivíduos, os recursos podem ser utilizados onde provavelmente produzirão melhores resultados, com custos mais baixos [9], tendo isso em mente, é valido ressaltar a necessidade do uso da tecnologia e dos dados de forma ética, com o objetivo de melhorar o tratamento e a condição de saúde da população de alto custo.

1.2 Problema

Dado o cenário econômico atual, além de fatores como o crescimento e o envelhecimento populacional, torna-se cada vez mais importante o gerenciamento adequado de recursos, principalmente em áreas críticas como a saúde. Com o intuito de auxiliar a tomada de decisões e, por consequência, otimizar esforços e recursos, tecnologias de pré-processamento de dados adequadas são fundamentais.

Nesse cenário, a utilização de algoritmos de aprendizado de máquina fornecem uma melhoria na detecção de pacientes de alto custo se comparados aos métodos utilizados anteriormente. Uma possível alternativa é a utilização de abordagens *Deep Learning* neste problema. Como tal técnica acaba por descobrir inter-relações até então desconhecidas nos conjuntos de dados em que são aplicados, tem-se como hipótese que seu uso pode apresentar uma melhoria nos resultados encontrados em trabalhos anteriores, principalmente quando consideramos a opção de sua utilização em conjuntos de dados que sofrem atualização contínua, como o MEPS, o que nos permite utilizar um dataset atual e abrangente.

Com isso, temos a possibilidade de oferecer modelos que consigam realizar a detecção confiável de pacientes de alto custo, e, como descrito na seção anterior, auxiliar na capacidade de tomada de decisões de gestores da área de saúde, tanto em contextos públicos quanto privados.

Desta forma, temos como problema principal neste trabalho a previsão de pacientes de alto custo, com base em dados históricos encontrados no dataset MEPS.

1.3 Objetivos

O objetivo geral deste trabalho é a avaliação de modelos de redes neurais para o problema de previsão de pacientes de alto custo. Como objetivos secundários podem ser listados:

1. A comparação dos resultados obtidos nos modelos gerados com aqueles encontrados em trabalhos anteriores;

2. O desenvolvimento das ferramentas necessárias para a criação desses modelos, a fim de simplificar trabalhos futuros nessa área

3. Validar o conjunto de atributos definidos por Seyed [2] em um dataset maior e mais recente.

1.4 Metodologia

A metodologia geral utilizada para o desenvolvimento deste trabalho foi dividida em cinco etapas, que incluíram:

1. A análise dos conjuntos de dados disponíveis, incluindo a motivação e impedimentos quanto aos datasets considerados;

2. O pré-processamento do conjunto de dados escolhido;

3. A implementação do pré-processamento, utilizando o dataset definido na etapa anterior;

4. A modelagem, que inclui a construção dos modelos de redes neurais;

5. E por fim, uma análise e discussão dos resultados obtidos.

Inicialmente, diversos *datasets* foram considerados, incluíndo dados do sistema único de saúde (SUS), dados cedidos por um plano de saúde privado e dados disponíveis no dataset do MEPS (*Medical Expenditure Panel Survey*). Após análise dos conteúdos desses conjuntos de dados, a disponibilidade de cada um e a facilidade de acesso aos dados, o dataset do MEPS foi escolhido, enquanto as outras fontes foram descartadas. No total, foram utilizados arquivos referentes à um período de 11 anos (2006 – 2016), resultando em um dataset final com 125.457 amostras.

Após a seleção do dataset, foi necessária uma fase de pré-processamento dos dados, de forma a aumentar as chances de que os modelos gerados possam atingir resultados adequados, isso inclui a transformação dos tipos de dados, reduções no número de classes em atributos relevantes e normalizações, detalhados na seção de pré-processamento. A etapa de implementação inclui a criação e aperfeiçoamento das ferramentas necessárias para a automação desse processo.

Em sequência, diversos modelos foram gerados, a fim de identificar um que apresente resultados aceitáveis para o problema em questão, este passo engloba a criação de modelos com parâmetros diferentes, com variações no

número de camadas ocultas, neurônios em cada camada e aplicação de dropout com diversas taxas.

Por fim, métricas relevantes são selecionadas e experimentos foram conduzidos à partir do modelo escolhido, a fim de atestar o seu desempenho, fundamentar as conclusões apresentadas e motivar possíveis trabalhos futuros.

Os modelos gerados à partir deste processo se mostraram capazes de gerar previsões adequadas para o problema proposto, obtendo-se 88% nas métricas de *accuracy* e *f1 score*, 91% na métrica de *recall*, 86% para *precision* e 84% *specificity*, esses resultados se mostram compatíveis com trabalhos anteriores [2], apresentando inclusive, resultados mais altos em algumas das métricas, como a *precision* o que, apesar de não ser uma comparação direta, dadas as diferenças entre os datasets, é um indicativo da qualidade dos modelos gerados.

1.5 Trabalhos relacionados

Diversos trabalhos foram encontrados discutindo o uso de técnicas de aprendizado de máquina no cenário da saúde.

Meehan [3] apresenta um trabalho no qual os autores aplicam três algoritmos: Regressão Logística, Árvore de decisão e *Naive Bayes*, ao problema de previsão de pacientes de alto custo, a fim de determinar qual modelo é mais adequado. São definidos também, alguns dos atributos mais importantes, que incluem: idade, estado de saúde percebido e número de meses desde a última verificação de pressão sanguínea.

Em outro trabalho, Seyed [2] define um conjunto mínimo de atributos para o problema de previsão de pacientes de alto custo utilizando o dataset do MEPS. Neste trabalho, são utilizados dados de três anos, e, à partir de técnicas estatísticas, é definido um conjunto mínimo de atributos que geram um resultado aceitável. Este conjunto mínimo foi utilizado neste trabalho para gerar os modelos avaliados. Com isso, o conjunto definido por Seyed pode ser testado em um dataset que engloba um período maior de tempo, neste caso, dados referentes à 11 anos, confirmando não apenas que o conjunto apresenta bons resultados, como continua a apresentá-los, mesmo considerando dados com 10 anos de diferença.

Boscardin [10] aponta o uso de dados auto-relatados como forma de aprimorar modelos de previsão de pacientes de alto custo. O trabalho em questão utiliza regressão logística multivariada por etapas, e utilizou como entradas os dados de uso de um serviço de saúde, dados sociodemográficos e, por fim, os dados auto-relatados. Como resultado, foi notado que a adição de tais dados foi benéfica, aumentando a taxa de acerto do modelo.

Lu [11] apresenta uma abordagem similar, tendo como foco a identificação de possíveis pacientes de alto custo dentre aqueles que apresentam custo médio. Com base nos dados dos anos de 2010 a 2013, os autores utilizam modelos de regressão logística para identificar os indivíduos que estarão entre os 10% com maiores gastos, no ano seguinte. Como resultados principais, além dos modelos para previsão, foi identificada uma série de condições crônicas e diagnósticos que são comumente associadas à transição de um paciente de custo médio em alto custo, o que pode indicar que a inclusão de tais informações em um modelo aplicado a este problema pode oferecer uma melhoria na qualidade das previsões.

É possível observar também, diversos outros trabalhos, que possuem objetivos semelhantes, mas direcionados à pacientes que possuem alguma condição específica, como doenças inflamatórias do intestino [12], carcinoma hepatocelular em pacientes com quadro de cirrose [13], esquizofrenia [14], doença pulmonar obstrutiva crônica [15], entre outras. Esses resultados servem para demonstrar que mesmo em um conjunto de pacientes com condições similares, nem todos são necessariamente de alto custo.

Por fim, podem ser encontradas também, algumas publicações que reforçam a importância de pesquisas nesta área. Segundo Blumenthal [16] o foco nos cuidados dessa parcela da população faz sentido por motivos humanitários, demográficos e financeiros. Segundo ele, tais pacientes merecem uma atenção maior, pois são os que mais entram em contato com o sistema de saúde, e, por consequência, são mais suscetíveis a serem afetados por problemas de saúde e de segurança que podem ser prevenidos. McCarthy [17] aponta que modelos de cuidados para pacientes de alto custo e alta necessidade possuem o potencial de reduzir custos ao mesmo tempo que melhoram as experiências e saúde do paciente.

1.6 Estrutura do documento

O restante do trabalho está organizado da seguinte forma: No Capítulo 2 é descrito o Referencial Teórico, que apresenta conceitos fundamentais sobre as áreas de Aprendizado de Máquina e as ferramentas utilizadas. No Capítulo 3, são apresentados os processos seguidos no desenvolvimento deste trabalho e os resultados obtidos. Por fim, no Capítulo 4 são apresentadas as conclusões obtidas à partir dos resultados encontrados, as limitações observadas no decorrer deste trabalho e ideias sobre possíveis trabalhos futuros.

CAPÍTULO

2

Referencial Teórico

2.1 Dados Abertos

O termo *open data* tem se tornado popular nas últimas décadas, e se refere, de forma abreviada, ao uso e redistribuição livre de conjuntos de dados. Uma definição bem aceita para o termo é atribuída a *Open Knowledge International* [18]: "Dados abertos são dados que podem ser usados livremente, reutilizados e redistribuídos por qualquer pessoa - apenas sujeito, no máximo, ao requisito de atribuição e *share-alike*[1]".

A *Open Knowledge International* indica também, diversos critérios que devem ser seguidos para que um conjunto de dados seja considerado aberto. Esses critérios visam garantir que os dados tenham sido disponibilizados de forma integral, sem modificações e o mais rápido possível, que o acesso a eles esteja disponível ao maior número de pessoas possível, sem que essas precisem se registrar para obter acesso.

Para que sejam considerados abertos, tais dados não podem estar sujeitos à qualquer tipo de patente ou registro de marca, e, ao mesmo tempo, nenhuma entidade deve ter o controle exclusivo sobre eles. Por fim, os dados devem estar em um formato que possa ser processado de forma automática por uma máquina.

A iniciativa de abertura de dados pode gerar grandes benefícios para a sociedade em geral, seja por meio da simplificação e viabilização de pesquisas aca-

[1] *Share-alike*, neste contexto, é um termo indicando que cópias ou adaptações de um trabalho ou conjunto de dados abertos devem ser liberados com a mesma licença do original, ou, ao menos uma licença similar, impedindo que novos usuários restrinjam o acesso a eles. Atribuição, por sua vez, indica que créditos apropriados devem ser dados ao criador, sendo necessário prover um link para acesso a licença e indicar se alguma alteração foi feita.

dêmicas, no desenvolvimento de aplicações e novas tecnologias ou mesmo contribuindo com a participação popular em questões de administração pública. No contexto de dados governamentais, o Tribunal de Contas da União [19] lista 5 motivos a favor da abertura. São citados como motivadores o aumento da transparência na gestão pública, o aprimoramento na qualidade dos próprios dados, a viabilização de novos negócios, facilitar contribuições por parte da sociedade com serviços inovadores ao cidadão e por fim, a obrigatoriedade legal.

Segundo a LAI, ou Lei de Acesso à Informação (art. 8º da Lei 12.527/2011), "É dever dos órgãos e entidades públicas promover, independentemente de requerimentos, a divulgação em local de fácil acesso, no âmbito de suas competências, de informações de interesse coletivo ou geral por eles produzidas ou custodiadas" [20].

2.2 Aprendizado de Máquina

Aprendizado de Máquina é uma subárea da Inteligência Artificial, que tem como principal objetivo e motivação a criação de agentes que tenham a capacidade de simular a inteligência humana. Em outras palavras, algoritmos que consigam utilizar informações históricas, experiências e conhecimento sobre o meio no qual estão para guiar alguma escolha ou processo de tomada de decisão, sem que seu comportamento tenha sido fixado previamente. O processo de criação de tal agente é dinâmico, ou seja, o modelo gerado ao fim do processo de criação pode variar com base nos dados com os quais ele entra em contato. A partir desses dados de entrada, se espera que o agente consiga devolver uma hipótese com uma taxa de acerto aceitável. Mitchell [21] apresenta a seguinte definição para Aprendizado de Máquina:

> "Um programa de computador é dito aprender com a experiência E em relação a alguma tarefa T e alguma medida de desempenho D, se seu desempenho em T, conforme medido por D, melhora com a experiência E."

Um exemplo apresentado por Mitchell[21] é o de um programa que aprende a jogar damas. Neste cenário, a tarefa é jogar partidas de damas, a medida de desempenho é a capacidade de vencer tais partidas. Essa capacidade, por sua vez, seria aprimorada conforme o agente ganha experiência através de partidas simuladas contra si mesmo.

Tendo em vista as características desse tipo de abordagem, são claras as vantagens de seu uso no mundo real. Com a utilização de algoritmos de aprendizado de máquina, temos a possibilidade de utilizar de forma mais eficiente a grande quantidade de dados gerados diariamente por diversas fontes. Seu uso

não se limitando apenas a contextos acadêmicos, temos diversos exemplos de aplicações populares que implementam técnicas de aprendizado de máquina, como recomendações de produtos em lojas de *e-commerce*, reconhecimento de voz, assistentes digitais, entre outros.

No geral, os algoritmos de Aprendizado de Máquina são divididos em três paradigmas: aprendizado supervisionado, aprendizado por reforço e aprendizado não supervisionado. Devido ao problema abordado, suas características e objetivos associados, neste trabalho será tratada apenas a abordagem supervisionada.

2.2.1 Aprendizado Supervisionado

O paradigma de aprendizado supervisionado se diferencia do não supervisionado pela forma com que as entradas são apresentadas ao algoritmo. Nesta abordagem, temos um conjunto de dados identificados e classificados, e, a partir destes, um algoritmo supervisionado deve gerar um modelo que consiga fazer previsões aceitáveis sobre dados similares que ainda não foram vistos, devolvendo a classe correta de uma dada amostra ou um valor próximo do esperado para ela.

O conjunto de dados apresentado a um algoritmo de aprendizado supervisionado é composto por um conjunto de amostras, que por sua vez, incluem uma coleção de uma ou mais características $X_{1...n}$ referentes a cada amostra, e uma sequência de valores alvo Y, de forma que cada valor em Y esteja associado à uma amostra específica.

Tabela 2.1: Exemplo de entrada para um algoritmo de aprendizado de máquina, contendo m amostras com n características cada, e um valor alvo y associado a cada amostra

$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{\ldots}$	$\mathbf{X_n}$	$\mathbf{Y}$
$x^{(1)}{}_1$	$x^{(1)}{}_2$	$\ldots$	$x^{(1)}{}_n$	$y^{(1)}$
$x^{(2)}{}_1$	$x^{(2)}{}_2$	$\ldots$	$x^{(2)}{}_n$	$y^{(2)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x^{(m)}{}_1$	$x^{(m)}{}_2$	$\ldots$	$x^{(m)}{}_n$	$y^{(m)}$

A partir do processamento de tal conjunto de dados, é gerado um modelo que tem a capacidade de, dada uma amostra x^k, $k = 1 \ldots m$ ainda não vista, devolver uma hipótese $h(x)$, que representa o valor y^j, $j = 1 \ldots m$ previsto para a amostra x^k de forma que o valor previsto seja próximo ao correto, em casos onde o modelo prevê valores contínuos, ou seja a classe correta, em modelos usados para classificação. Tal processo pode ser visto na Figura 2.1.

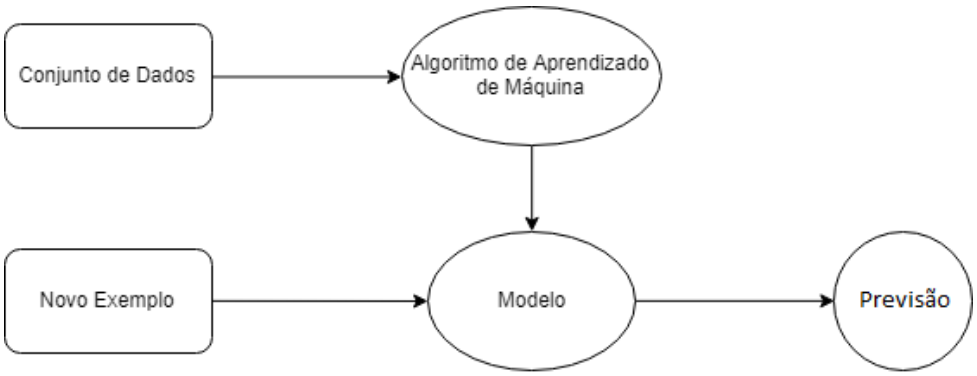

Figura 2.1: Representação visual do processo realizado por um algoritmo de aprendizado de máquina supervisionado.

2.2.2 Classificação

Uma das classes mais comuns de problemas é a associação de um dado elemento a uma entre duas ou mais classes já conhecidas. Neste tipo de problema, é recebido como entrada um conjunto de dados similar ao exibido na Tabela 2.1, em que cada amostra é associada a um valor de y, representando sua classe. Dada uma nova entrada x o algoritmo de aprendizado deve ser capaz de devolver uma das classes possíveis, com base em observações passadas. As classes, de forma geral, são representadas por valores discretos e finitos.

Dentre os problemas de classificação, é possível observar dois casos gerais: São chamados de problemas de classificação binária aqueles em que dada uma entrada x qualquer, é possível ter como resultado da classificação uma entre duas classes possíveis. Neste caso, como resposta para x, é devolvida uma hipótese $y \in [0, 1]$, onde $y = 0$ indica que x pertence à classe 0 e, de forma similar, $y = 1$ indica que x pertence à classe 1.

Em alguns algoritmos, como a regressão logística, o valor de y pode ser interpretado como o nível de certeza de que x pertence à classe 1, logo, é possível definir um limiar, $0,5$ por exemplo, para decidir à qual classe x pertence. Para casos em que o valor de y seja maior que tal limiar, x será considerado da classe 1, da mesma forma, para valores de y inferiores ao limiar, x será classificado como sendo da classe 0. Exemplos deste tipo de classificação incluem detecção de spam, classificar expressões ou textos como positivas ou negativas, decidir se uma pessoa aparece em determinada foto ou nao, entre outras. Nesses casos, dependendo do algoritmo utilizado, é possível associar a classe 0 as respostas negativas, como não spam, emoção negativa e não presente, e associar a classe 1 as respostas positivas.

O segundo caso é chamado Classificação Multiclasse e engloba problemas nos quais existem mais do que duas classes possíveis, e uma entrada x deve ser classificada com apenas uma delas. De forma geral, é possível tratar a

classificação multiclasse de forma similar à binária, devolvendo como hipótese, para cada entrada x, um vetor y com n posições, sendo n equivalente a quantidade de classes. Cada posição desse vetor possuirá um valor que, de forma semelhante a classificação binária, indica o nível de certeza de que x pertence à classe representada pela posição. Em outras palavras, em um cenário onde podemos atribuir uma entre 5 classes à uma entrada qualquer, o vetor y terá 5 posições, aqui, a primeira posição indica a probabilidade de x ser da classe 1, a segunda indicando a probabilidade de x pertencer a classe 2, e assim sucessivamente.

2.3 Deep Learning

Deep Learning, ou aprendizado profundo, é uma subárea do aprendizado de máquina que tem recebido grande atenção atualmente, vendo uso em diversas aplicações, como o *Google Translate* [22] e a assistente virtual Cortana [23], da Microsoft. De forma geral, *deep learning* implementa algoritmos chamados RNA, ou Redes Neurais Artificiais, que são inspirados pelo funcionamento do cérebro humano [24].

Uma rede neural é composta de nós, ou neurônios, que são representações matemáticas dos neurônios biológicos. Cada nó é capaz de receber um conjunto de entradas e, a partir delas, calcular um valor de saída. Tal cálculo envolve a multiplicação de cada valor recebido como entrada com um peso associado a ele. Quando todas as entradas são multiplicadas por seus pesos e somadas, o valor resultante é somado com um outro valor, independente da entrada, chamado *bias*, o resultado é então passado para uma função de ativação, que determinará a saída geral do nó [25], este processo pode ser observado na Figura 2.2.

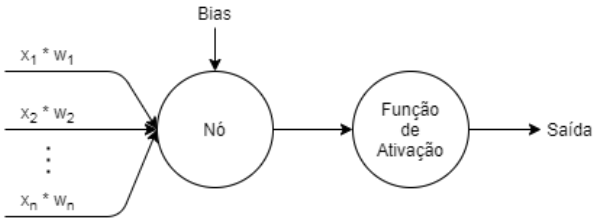

Figura 2.2: Exemplo do funcionamento de um neurônio em uma rede neural

Nós chamados de entrada, ou *input nodes*, são aqueles ativados à partir de sensores que percebem o meio, ou, em outras palavras, são aqueles que recebem diretamente algum tipo de entrada externa à rede, como uma imagem, sons ou qualquer outro conjunto de dados. Outros nós são ativados através de conexões com nós previamente ativos [26]. O treinamento de uma rede envolve a comparação do valor que está sendo gerado como saída, dada

uma amostra qualquer, e a resposta considerada correta para aquela amostra, dessa forma, os pesos associados a cada entrada serão atualizados, de forma a minimizar essa diferença. Com esse processo, é possível utilizar nós para realizar classificações binárias, adequando os pesos associados a cada entrada de forma a reconhecer um padrão ou classe do que é recebido.

É chamada rede neural uma estrutura composta por vários nós conectados, cada qual com suas entradas, sejam elas internas ou externas à rede, os pesos associados a cada valor de entrada e o valor de bias para cada um deles. Dessa forma, com cada nó individual aprendendo um modelo diferente, temos, com uma rede neural, a habilidade de gerar modelos complexos, com cada nó sendo capaz de detectar características e padrões distintos.

No contexto de redes neurais, é chamada de camada de entrada, ou *input layer* o conjunto de nós que recebe uma entrada externa à rede. Os nós conectados à camada de entrada formam a primeira camada oculta, ou *hidden layer*, de forma similar, os que estão conectados à primeira camada oculta formam a segunda camada oculta, e assim sucessivamente. Por fim, a camada que devolve a resposta final da rede é chamada de camada de saída, ou *output layer*. Em uma rede neural existe apenas uma camada de entrada, da mesma forma, existe apenas uma camada de saída, entretanto, a quantidade camadas ocultas entre as duas é variável, e deve ser definida de acordo com o problema que está sendo abordado[2].

Tendo em mente as características e a variabilidade no tamanho de uma rede neural, é possível definir de forma mais clara *deep learning*. Segundo Jeff Dean, líder da divisão de inteligência artificial do Google [27], "Ao ouvir o termo deep learning, pense em uma grande rede neural profunda. Profundidade se referindo normalmente ao número de camadas" [28].

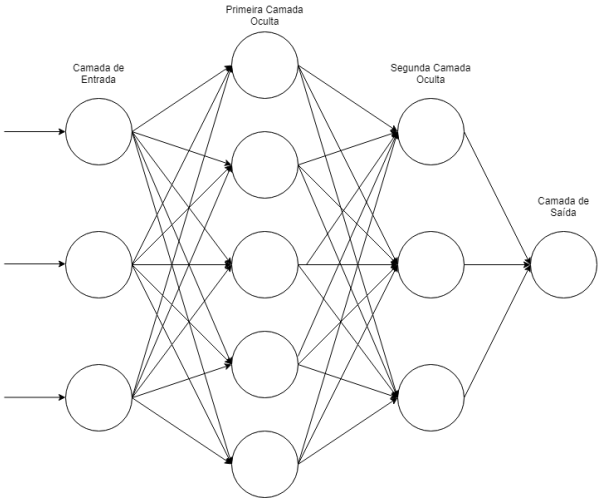

Figura 2.3: Exemplo de uma rede neural com duas camadas ocultas

[2]Da mesma forma, a quantidade de neurônios em qualquer uma das camadas, incluindo as de *input* e *output*, é variável

Apesar de serem capazes de modelar funções complexas, um dos problemas observados em redes neurais profundas é o *overfitting* [1], que pode ser definido em termos simples como a possibilidade da rede se adaptar de forma exagerada ao conjunto de treinamento apresentado, modelando uma função que tente, no caso de modelos para classificação, separar totalmente as amostras observados, ao invés de criar um modelo que generalize os dados para o problema. Neste caso o que acaba acontecendo é que o modelo gerado apresenta resultados excelentes ao considerar o conjunto de treinamento, mas que não se sustentam quando ele é exposto à novos dados, como por exemplo o conjunto de validação ou dados reais [29].

Existem diversas estratégias para evitar o *overfitting*, ou sobreajuste, em redes neurais. Uma delas, utilizada neste trabalho se chama *dropout*. A técnica de *dropout* consiste, basicamente, em ignorar ou remover temporariamente nós (e consequentemente suas conexões) da rede de forma aleatória, durante o processo de treinamento [1], apesar de parecer possivelmente contra intuitivo à primeira vista, a técnica produz resultados positivos, e tende a melhorar a qualidade dos modelos gerados [1].

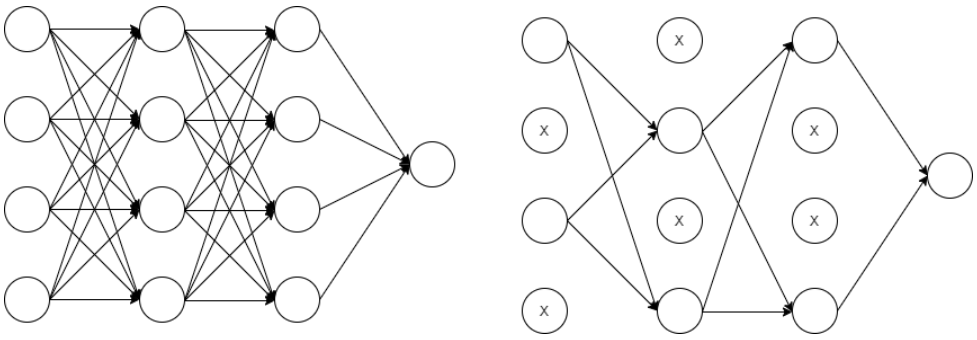

Figura 2.4: Representação visual do processo de *dropout* [1].

O processo de *dropout* pode ser observado na Figura 2.4, nela, é apresentada à esquerda uma rede neural com duas camadas ocultas, e a direita, uma segunda rede gerada à partir da primeira, após a applicação do *dropout*, nós ignorados neste passo do treinamento estão marcados com X [1].

2.4 Mineração de Dados

2.4.1 Pré-Processamento dos Dados

A fase de Pré-Processamento de Dados é composta por um conjunto de técnicas e atividades sobre os dados que serão utilizados no treinamento dos algoritmos. Esse processo tem o intuito de prevenir uma série de problemas que podem surgir futuramente devido ao estado no qual os dados podem ser recebidos em um primeiro momento, além de garantir que uma maior quanti-

dade de informações relevantes possa ser extraída do conjunto original. Isso pode vir a levar, por sua vez, a um melhor resultado por parte dos modelos gerados. No geral, um conjunto pode apresentar dados de diversas formas: números inteiros, decimais, variáveis booleanas, categóricas e até mesmo a ausência de dados em certos pontos da amostragem. Mesmo quando não existem dados ausentes, as diferenças de magnitude entre eles podem prejudicar o desempenho dos modelos gerados.

A Tabela 2.2 contém parte dos dados de um dataset [30] com informações sobre clientes de um restaurante, e serve como ilustração para algumas das formas citadas.

Tabela 2.2: Exemplo de conjunto com dados ausentes

smoker	**drink_level**	**dress_preference**	**birth_year**	**height**
false	abstemious	informal	1989	1,77
false	abstemious		1990	1,58
false	abstemious	informal	1981	1,6
true	social drinker	no preference	1989	1,8
false	abstemious	informal	1990	1,87

Utilizar dados como os apresentados na Tabela 2.2 sem qualquer tipo de pré-processamento pode ser desafiador, já que, em geral, algoritmos não possuem a capacidade de lidar com eles de forma satisfatória. Dessa forma, se torna clara a importância desta fase. Existem diversas formas de lidar com tais problemas. No caso de dados ausentes, por exemplo, o mais simples seria, ao encontrar uma amostra com uma característica ausente, descartar a amostra completamente, ou de forma alternativa, desconsiderar tal característica em todas as outras. Essa abordagem oferece desvantagens claras, já que o descarte levaria à uma grande perda de informação, prejudicando assim o desempenho de qualquer algoritmo aplicado sobre esses dados, como é apontado por Raschka [31]. Uma solução proposta pelo mesmo [31], é o uso de uma técnica chamada *mean imputation*, que consiste em inserir no lugar dos valores ausentes, o valor médio daquele campo, considerando todas as amostras disponíveis. Outras alternativas incluem também a substituição pela mediana ou pelo valor mais frequente, da mesma forma, considerando o dataset disponível. Ao aplicar a abordagem de substituir os dados ausentes pelo valor mais frequente, é gerada a Tabela 2.3.

Outro caso que deve ser considerado é a presença de dados categóricos, ou seja, aqueles que apresentam algum tipo de informação de forma não numérica, como *drink_level* e *dress_preference* na Tabela 2.2 por exemplo. Tais variáveis podem fornecer informações importantes para a criação do modelo. Em alguns casos, pode ser útil realizar a conversão dessas para valores nu-

Tabela 2.3: Exemplo de conjunto com valores ausentes substituídos

smoker	**drink_level**	**dress_preference**	**birth_year**	**height**
false	abstemious	informal	1989	1,77
false	abstemious	informal	1990	1,58
false	abstemious	informal	1981	1,6
true	social drinker	no preference	1989	1,8
false	abstemious	informal	1990	1,87

méricos, caso o algoritmo utilizado trabalhe melhor com esse tipo de dado.

Um dado categórico pode ser classificado como nominal ou ordinal, dependendo do tipo de informação contida nele. Dados ordinais apresentam uma ordenação natural quando consideramos seus possíveis valores. Na Tabela 2.2, por exemplo, *drink_level* pode ser visto como ordinal, pois seus valores podem ser ordenados na forma $abstemious < social\ drinker < casual\ drinker$. O campo *dress_preference*, por outro lado, não apresenta qualquer tipo de ordenação em seus valores. Dados com essa característica são chamados nominais, e representam categorias diferentes sem apontar relações de ordem entre seus possíveis valores.

A conversão de tais dados para valores numéricos é diferente, conforme o tipo da variável. Para ordinais, é possível utilizar a técnica de *integer encoding*, atribuindo um valor inteiro ou decimal à cada valor possível do campo original. No caso de *drink_level* por exemplo, é possível substituir os valores *abstemious* por 0, *social drinker* por 1 e *casual drinker* por 2. Como os números mantêm naturalmente a ordem presente nas variáveis originais, nenhuma informação é perdida com tal substituição. Dados nominais, por outro lado, não devem ser substituídos diretamente por números, já que isso criaria uma relação entre eles que, a princípio, não existe, o que pode prejudicar o desempenho dos modelos gerados futuramente. Neste caso, é aplicada a técnica de *hot encoding*, que consiste em criar um novo atributo booleano para cada valor possível do dado original, definindo como 1 a presença daquele valor no original, e 0 sua ausência. Utilizando novamente o exemplo da Tabela 2.2, ao converter *dress_preference*, seriam criados quatro campos novos: *no_pref*, *informal*, *casual* e *formal*. Valores são então atribuídos para as características criadas, de acordo com o exemplo original. Dessa forma, o dado nominal é transformado em um valor numérico, sem criar qualquer tipo de relação ou ordenação. Depois da aplicação desses processos, é obtido como resultado a Tabela 2.4.

Por fim, é possível encontrar em um mesmo dataset dados numéricos que possuem magnitudes diferentes. Na Tabela 2.2, por exemplo, é possível notar a diferença nos valores presentes nos campos *birth_year* e *height*. No caso

Tabela 2.4: Exemplo do conjunto após a transformação

smoker	**drink_level**	**no_pref**	**informal**	**casual**	**formal**	**birth_year**	**height**
false	0	0	1	0	0	1989	1,77
false	0	0	1	0	0	1990	1,58
false	0	0	1	0	0	1981	1,6
true	1	1	0	0	0	1989	1,8
false	0	0	1	0	0	1990	1,87

de *birth_year*, que representa um ano, sempre serão encontrados valores na casa dos milhares, por outro lado, como *height* representa uma altura, seus valores estão dentro de um intervalo muito menor. Tal diferença pode afetar negativamente diversos algoritmos, já que algumas informações teriam maior influência sobre a decisão do modelo, mesmo que tal comportamento não seja observado na função real. Para resolver esse problema, é necessário o processo de normalização, ou, em outras palavras, transformar os valores do dataset de forma que todos estejam dentro de um mesmo intervalo. É possível aplicar diversas abordagens no processo de normalização, dependendo do resultado esperado. *MinMax scaling*, por exemplo, é uma técnica simples, que redimensiona os valores de um campo de forma que eles fiquem limitados por um novo intervalo. Ela consiste na aplicação da Equação 2.1 em todos os elementos numéricos do dataset que serão normalizados.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{2.1}$$

Na Equação 2.1 x representa o valor original do campo, $x_{\max}$ representa o maior valor encontrado para aquele atributo dentro de todo o dataset, de forma similar, $x_{\min}$ representa o menor valor encontrado. Por fim, x' é o valor normalizado que será substituído no *dataset*. Dessa forma, temos para o maior valor do *dataset* a Equação 2.2, fazendo com que o maior elemento seja substituído por 1.

$$x' = \frac{x_{\max} - x_{\min}}{x_{\max} - x_{\min}} = 1 \tag{2.2}$$

Por outro lado, para o menor elemento do conjunto, temos a Equação 2.3, fazendo com que ele seja substituído por 0.

$$x' = \frac{x_{\min} - x_{\min}}{x_{\max} - x_{\min}} = 0 \tag{2.3}$$

Todos os outros valores serão, de forma similar, mapeados para o novo intervalo $[0, 1]$, mantendo sua ordem original. Aplicando a normalização, a Tabela 2.5 é obtida como resultado.

Tabela 2.5: Exemplo de conjunto com valores normalizados

smoker	**drink_level**	**no_pref**	**informal**	**casual**	**formal**	**birth_year**	**height**
false	0	0	1	0	0	0,88	0,65
false	0	0	1	0	0	1	0
false	0	0	1	0	0	0	0,06
true	1	1	0	0	0	0,88	0,75
false	0	0	1	0	0	1	1

2.4.2 Avaliação de Modelos

Supondo um classificador binário qualquer que deve tomar uma amostra e devolver como hipótese uma entre duas classes possíveis, "A" e "B", por exemplo. Toda previsão devolvida por esse classificador deve se encaixar em um entre quatro possíveis cenários:

- O modelo previu a classe "A" para a amostra, e ele realmente era da classe "A";
- O modelo previu a classe "A", mas a amostra pertencia à classe "B";
- O modelo devolveu "B" como previsão e a amostra era da classe "A";
- Ou, por fim, o modelo devolveu "B" e a amostra realmente pertencia a classe "B"

Visualizando as classes "A" e "B" como positiva e negativa, os casos podem ser representados com a matriz 2.6, onde a primeira linha representa a previsão devolvida pelo classificador, e a primeira coluna representa a classe verdadeira da amostra. Essa tabela tem o nome de matriz de confusão [25]. Seguindo este modelo, ao fim do processo de teste, a matriz é preenchida de

Tabela 2.6: Exemplo do modelo de uma matriz de confusão

	A	**B**
A	Verdadeiro positivo	Falso negativo
B	Falso positivo	Verdadeiro negativo

acordo com os resultados encontrados, e a partir dela são calculadas algumas métricas que podem ser úteis. A primeira delas é a *accuracy*, que representa a taxa com que o modelo conseguiu prever corretamente a classe de uma amostra qualquer, e é definida em 2.4.

$$accuracy = \frac{\text{Verdadeiro positivo} + \text{Verdadeiro negativo}}{\text{Quantidade total de amostras}} \tag{2.4}$$

A *accuracy* apresenta uma métrica válida para a análise de modelos, porém, não é confiável em certos casos. Mais especificamente, ela não representa bem a capacidade de previsão de um modelo cujo dataset seja desbalanceado. Em outras palavras, se a maioria das amostras no conjunto de teste forem da mesma classe, é possivel que o modelo gerado classifique todas as entradas como sendo da classe majoritária. Como a maioria das amostras terá uma previsão correta, a *accuracy* desse modelo será alta, mesmo que ele não esteja fazendo qualquer tipo de previsão propriamente dita [32].

Dessa forma, existem outras métricas, que são menos sensíveis a esse tipo de cenário e podem ser usadas em conjunto com ela para representar melhor a capacidade do modelo avaliado. Uma dessas métricas é o *recall*, que mede a taxa de acertos da classe positiva. Em outras palavras, de todas as amostras que pertencem à classe positiva, o *recall* indica quantas foram classificados corretamente. Ele pode ser calculado utilizando a Equação 2.5. Um modelo que não produz falso negativos apresenta um valor de *recall* 1.

Essa métrica pode ser útil em problemas cuja classificação correta de amostras da classe positiva são cruciais. Em problemas relacionados ao diagnóstico de doenças, câncer por exemplo, classificar um paciente que possui determinada condição como não a possuindo é muito mais prejudicial do que prever como positivo aquele que não a possui, já que esse paciente poderia ter seu tratamento adiado, colocando em risco ainda maior sua saúde.

Uma terceira métrica que pode ser considerada é a *precision*, definida na Equação 2.6, que indica a proporção de previsões para a classe positiva que estavam corretas. Um modelo que não produz falso positivos apresenta *precision* 1. Por fim, a *specificity* é definida na Equação 2.7 [25].

$$\text{Recall} = \frac{\text{Verdadeiro positivo}}{\text{Verdadeiro positivo} + \text{Falso negativo}} \tag{2.5}$$

$$\textit{Precision} = \frac{\text{Verdadeiro positivo}}{\text{Verdadeiro positivo} + \text{Falso positivo}} \tag{2.6}$$

$$\text{Specificity} = \frac{\text{Verdadeiro Negativo}}{\text{Verdadeiro Negativo} + \text{Falso Positivo}} \tag{2.7}$$

A partir dessas três métricas uma quarta pode ser derivada, o *f1 score* [33]. O *f1 score* é a média harmônica entre a *precision* e o *recall* e é calculado de acordo com a Equação 2.8. Ele dá um peso igual para ambas as medidas, dessa forma, o *f1 score* de um modelo é alto se ambos *recall* e *precision* forem altos. Por outro lado, caso qualquer um dos dois diminua, o *f1 score* final também será menor. Essa métrica é importante para a avaliação de modelos onde ambas as métricas possuem importância similar, ou seja, em problemas

onde não é vantajoso aumentar alguma das duas em detrimento da outra.

$$f1\ score = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.8}$$

2.5 Ferramentas utilizadas

2.5.1 Scikit-Learn

`Scikit-Learn` [34] é uma biblioteca de código aberto distribuída sobre a licença BSD, que oferece suporte ao desenvolvimento de aplicações na área de inteligência artificial utilizando a linguagem de programação Python [35]. Sua API conta com diversas funções que auxiliam no pré-processamento de dados, seleção de modelos e redução de dimensionalidade, além de implementações otimizadas de diversos algoritmos para problemas de regressão, classificação e clustering. Foi desenvolvido inicialmente em 2007 por David Cournapeau como projeto na Google Summer of Code. No mesmo ano, Matthieu Brucher começou a participar do projeto como parte de sua tese. No ano de 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort e Vincent Michel do INRIA assumiram liderança do projeto, lançando sua primeira versão pública. Desde então, diversas releases tem sido feitas, seguindo ciclos de três meses. Atualmente, o projeto tem sido desenvolvido pela própria comunidade [34].

2.5.2 Tensorflow

`Tensorflow`, segundo descrição oficial, "é uma plataforma completa de código aberto para machine learning" [36]. Sendo uma biblioteca de código aberto, o `tensorflow` conta com uma comunidade ativa, que permite uma grande quantidade de projetos, ferramentas, bibliotecas e recursos disponíveis, que dão suporte a diversos projetos na área de machine learning.

No `tensorflow` é suportado também a possibilidade da criação e utilização de modelos de machine learning de diversas formas não tradicionais, como dispositivos móveis, utilizando o `tensorflow` lite, que converte modelos já treinados em formatos compactos e otimizados para funcionar em cenários com recursos limitados.

2.5.3 Keras

`Keras` [37] é uma biblioteca de código aberto distribuída sobre a licença MIT escrita em Python, que dá suporte ao desenvolvimento de aplicações que utilizam deep learning.

`Keras` pode ser executado em conjunto com o `Tensorflow` [38], da Google, `CNTK`, da Microsoft, ou `Theano`, desenvolvido principalmente pelo grupo de aprendizado de máquina da Universidade de Montréal. Foi desenvolvido com o intuito de permitir desenvolvimento e experimentações rápidas e demonstra ser bem aceito, tendo mais de 200 mil usuários em novembro de 2017, além de ser o segundo mais citado em artigos sobre deep learning, ficando atrás apenas do `Tensorflow`.

Seu desenvolvimento é apoiado por diversas empresas participativas no cenário de deep learning, como Google, Microsoft, Uber, Apple, nVidia e Amazon. E é utilizado em empresas como Netflix, Uber, Yelp e outras.

2.5.4 SAS

SAS ou *Statistical Analysis System* é um software desenvolvido pelo *SAS Institute* e oferece suporte em tarefas de análise e gerenciamento de dados, *business intelligence*, entre outros.

Ele oferece também uma linguagem própria para a manipulação e análise de conjuntos de dados. Para o escopo deste trabalho, o uso do SAS se limitou à conversão dos conjuntos de dados do MEPS, obtidos em formato sas em um formato mais simples de ser usado na linguagem de programação escolhida. O SAS está atualmente disponível em versão paga, destinada principalmente à empresas, e em versão acadêmica sem custos, destinada à educadores, estudantes e pesquisadores [39].

CAPÍTULO

3

Metodologia para detecção de pacientes de alto custo

A metodologia geral utilizada para o desenvolvimento deste trabalho foi dividida em cinco etapas, que incluíram a análise dos conjuntos de dados disponíveis, incluindo a motivação e impedimentos quanto aos datasets considerados. O pré-processamento do conjunto de dados escolhido. A implementação do pré-processamento, utilizando o dataset definido na etapa anterior. A modelagem, que inclui a construção dos modelos de redes neurais, e por fim, uma análise e uma discussão sobre os resultados obtidos. Todas as etapas são detalhadas a seguir.

3.1 Análise dos conjuntos de dados

Em primeiro lugar, foi necessária uma fase para a análise dos conjuntos de dados disponíveis e a seleção do dataset que seria utilizado para a geração dos modelos. Alguns dos conjuntos de dados inicialmente considerados, incluiam aqueles disponibilizados pelo departamento de informática do SUS [40], operadoras de plano de saúde privadas, datasets gerados a partir de projetos internacionais, como o *EU Open Data Portal* [41], MEPS [42] e *eHealth Ireland* [43], e por fim, aqueles disponíveis na plataforma Kaggle [44].

Dentre as fontes de dados citadas, a escolhida foi o MEPS [42], ou *Medical Expenditure Panel Survey*. Ele é um projeto do governo dos Estados Unidos iniciado em 1996, e consiste na aplicação e análise de questionários em larga escala, direcionados a famílias, indivíduos, empregadores e provedores de ser-

viço na área da saúde, como hospitais, farmácias e médicos. Os resultados e análises derivados desses questionários são liberados periodicamente como dados de domínio público, e tem como objetivo apresentar informações sobre quais serviços de saúde são mais utilizados, com qual frequência são utilizados, quais são os custos associados, como esses serviços são pagos, além de diversas outras informações que podem ser importantes para pesquisas nessa área, incluindo, principalmente, informações a nível individual. No total, o relatório referente ao ano de 2015, o mais recente disponível, inclui 1831 variáveis para 35.427 indivíduos. Dentre as informações coletadas, são apresentados indicadores das condições físicas e psicológicas de cada entrevistado, além de dados demográficos, socioeconômicos, geográficos e informações médicas, como o diagnóstico de doenças e histórico de internações. De forma mais específica, os dados são classificados em diversos tipos pelo próprio MEPS [45], são eles:

- Variáveis de administração do questionário, que incluem identificadores para cada indivíduo, identificadores e tamanho dos grupos familiares, e outras informações sobre os indivíduos que foram entrevistados. Essas variáveis podem ser utilizadas para organização interna e como ferramenta para a associação das informações de cada questionário com outros, disponibilizados também pelo MEPS;

- Variáveis demográficas, que incluem informações sobre idade, sexo, etnia, estado civil e escolaridade;

- Variáveis de renda, que incluem informações sobre a renda, tanto individual quanto do grupo familiar, valores recebidos através de várias fontes, benefícios recebidos do governo e número de dependentes, quando aplicável;

- Variáveis de condição de nível pessoal, que dizem respeito ao estado de saúde do entrevistado, incluindo indicadores sobre a presença de alguma doença ou sintoma, como angina, enfisema pulmonar, bronquite crônica, colesterol alto, câncer, diabetes, histórico de ataques cardíacos, artrite, asma e TDAH. Além desses, é encontrada também a idade do entrevistado quando a doença foi diagnosticada. Para os casos que se aplicam, são informados também se a doença persiste, se o tratamento foi concluído recentemente ou se ele ainda continua. Casos que incluem o diagnóstico de doenças mais raras [1] são apresentados da mesma forma, a fim de proteger o anonimato dos entrevistados;

[1] Mais especificamente, para tipos de câncer que aparecem na lista de doenças raras do *National Institutes of Health*, como câncer de esôfago, pâncreas, laringe, leucemia, entre outros

- Variáveis sobre o estado de saúde incluem informações sobre o condicionamento físico do indivíduo, abordando questões como a dificuldade sentida ao levantar 10 libras, subir 10 degraus, andar 3 quadras, andar 1 milha, permanecer em pé por 20 minutos e outras atividades que são comuns no dia a dia, como se agachar, ou pegar objetos elevados, além de outras limitações físicas, como baixa visão ou audição. Também é indicado se existe alguma limitação social ou cognitiva, bem como fatores comportamentais. Por fim, são incluídas nessa classe de variáveis informações sobre o histórico clínico, indicando o tempo desde que determinados procedimentos foram realizados, incluindo mamografia, colonoscopia e histerectomia, além de outros;

- Variáveis indicadoras de dias de incapacidade são um pequeno grupo de questões indicando o número de dias de trabalho ou escola perdidos devido a doenças ou ferimentos. Também é indicado se algum dia foi perdido cuidando de uma outra pessoa nessa situação;

- Variáveis de acesso visam indicar se a pessoa possui acesso básico a serviços de saúde, qual o provedor e se ela encontra alguma dificuldade quando precisa de tais serviços;

- Variáveis de emprego indicam se o entrevistado trabalha atualmente em mais de um emprego, qual sua renda por hora, horas trabalhadas por semana e a área de atuação. Em geral, dão uma ideia das condições de trabalho e remuneração do indivíduo;

- Variáveis de plano de saúde indicam qual plano de saúde o entrevistado possui, se aplicável, e informações sobre o histórico de planos utilizados;

- Por fim, variáveis de utilização, gastos e fontes de pagamento contém informações como o número de vezes que cada serviço de saúde foi utilizado, como visitas ambulatoriais, serviços dentais, remédios prescritos, atendimentos de emergência, entre outros. Qual o custo total dos procedimentos associados ao entrevistado e como eles foram pagos. Para o problema de pacientes de alto custo, essas variáveis podem ser utilizadas para definir as classes de cada exemplo no conjunto de treinamento;

A primeira fase teve como principal resultado a definição do MEPS como dataset a ser utilizado a fim de alcançar os objetivos deste trabalho, esta decisão se deu por diversos fatores, como qualidade dos dados presentes em cada dataset, facilidade no acesso aos dados, a fim de facilitar a reprodução dos resultados encontrados e quantidade de dados disponíveis para análise.

3.2 Pré-processamento do conjunto de dados

O segundo passo é o pré-processamento dos dados contidos no dataset escolhido. Neste passo, para os propósitos deste trabalho, os processos descritos por Seyed [2], em um estudo que definiu um conjunto mínimo de atributos no dataset do MEPS, foram seguidos. Mais especificamente, apenas indivíduos com mais de 17 anos foram considerados, aqueles que possuiam apenas valores não negativos para os atributos de gasto (nenhum dos dois valores ausentes) e aqueles com valores não negativos para a variável de peso de nível pessoal (*person-level weight*), uma vez que os registros com peso zero não devem ser considerados em estudos conduzidos na população domiciliar dos Estados Unidos [2]. No total, 39 atributos foram escolhidos para compor o dataset final, divididos em 5 categorias, descritas nas Tabelas 3.1, 3.2, 3.3, 3.4, 3.5.

Tabela 3.1: Variáveis Demográficas

Nome	Significado	Tipo de Dado
Age	Idade	Inteiro (1-3)
Sex	Sexo	Flag Binária
Race	Raça	Inteiro (1-3)
HIDEG	Maior grau de educação alcançado	Flag Binária
Region	Região	Inteiro (1-4)
Marry	Estado civil	Inteiro (0-2)
POVCAT	Renda familiar em relação à linha de pobreza	Inteiro (1-3)

Tabela 3.2: Variáveis de Estado de Saúde

Nome	Significado	Tipo de Dado
RTHLTH	Estado de saúde física relatada	Inteiro (1-5)
MNHLTH	Estado de saúde mental relatada	Inteiro (1-5)
ANYLIM	Certa limitação na execução de tarefas diárias	Flag Binária
BMINDX	Índice de massa corporal	Inteiro (1-4)

Tabela 3.3: Variáveis de Cuidado Preventivo

Nome	Significado	Tipo de Dado
Check	Checkup de rotina	Inteiro (0-2)
BPCHEK	Teste de pressão arterial	Inteiro (0-2)
CHOLCK	Exame de colesterol	Inteiro (0-2)
NOFAT	Recomendado a reduzir o colesterol	Flag Binária
EXRCIS	Recomendado a se exercitar mais	Flag Binária
ASPRIN	Uso regular de aspirina	Flag Binária
BOWEL	Sigmoidoscopia ou colonoscopia	Flag Binária
STOOL	Exame de sangue oculto nas fezes	Flag Binária
DENTCK	Check-up odontológico	Inteiro (0-2)

Tabela 3.4: Variáveis de Condições Prioritárias

Nome	Significado	Tipo de Dado
HIBPDX	Hipertensão	Flag Binária
CHDDX	Doença arterial coronariana	Flag Binária
ANGIDX	Angina	Flag Binária
MIDX	Infarto do miocárdio	Flag Binária
OHRTDX	Outras doenças cardíacas	Flag Binária
STRKDX	Derrame	Flag Binária
ASTHDX	Asma	Flag Binária
EMPHDX	Enfisema	Flag Binária
CANCERDX	Câncer	Flag Binária
DIABDX	Diabetes	Flag Binária
ARTHDX	Artrite	Flag Binária
CHOLDX	Colesterol alto	Flag Binária
PC	Presença de condição prioritária	Flag Binária
PCCOUNT	Número de condições presentes	Inteiro (0-4)

Tabela 3.5: Variáveis de Contagem de Visitas

Nome	Significado	Tipo de Dado
OBTOT	Visita a consultorios	Contínuo (0-1)
OPTOT	Atendimento ambulatorial	Contínuo (0-1)
ERTOT	Sala de emergência	Contínuo (0-1)
IPDIS	Hospitalizações	Contínuo (0-1)
RXTOT	Quantidade de medicamentos prescritos	Contínuo (0-1)

Esses atributos também foram pré-processados de acordo com os resultados de Seyed [2]. Para as variáveis do grupo demográfico, "Age" trocado por uma variável categórica com 3 valores possíveis, as idades na faixa de $18 - 49$ foram substituídas por 1, a faixa de $50 - 65$ por 2 e aquelas acima de 65 por 3.

Além disso, todos os indivíduos com a idade inferior a 18 foram removidos, já que muitos dos outros atributos não são aplicáveis nesses casos. "Race", originalmente uma variável categórica com 6 valores possíveis foi reduzida para 3 categorias, com possíveis valores 1, 2 ou 3 representando brancos, negros e outros, respectivamente. "HIDEG" foi alterado de uma variável categórica que representava o mais alto grau de educação (ensino médio, bacharelado, mestrado, doutorado, etc) para uma *flag* binária, com o valor 1 representando que o indivíduo possui o ensino superior completo e 0 em caso contrário.

A variável "Marry" sofreu uma redução na quantidade de valores possíveis, de 10, originalmente, para um intervalo de $0 - 2$, onde 0 significa que o indivíduo nunca foi casado, 1 para os que atualmente são casados e 2 para viúvo, separado ou divorciado. Os valores excluídos se referiam a pessoas menores de idade e pessoas que mudaram seu estado civil durante a realização da pesquisa (entre os dois paineis do MEPS).

Finalmente, POVCAT (*poverty category*), representa a renda familiar em relação à linha de pobreza. Originalmente, este atributo possuía 5 valores possíveis, variando entre $1 - 5$ e representando "pobre/negativo", "quase pobre", "baixa renda", "renda média" e "alta renda", respectivamente. No conjunto de dados final, esses valores foram reduzidos ao intervalo entre $1 - 3$, com 1 representando as classes "pobre / negativo / quase pobre", 2 representando "baixo/médio" e 3 "alta renda". Os atributos restantes da categoria demográfica não foram alterados.

Na categoria Estado de saúde, além de BMINDX, o indicador de massa corporal, todos os atributos permaneceram inalterados. BMINDX originalmente continha um valor contínuo, referente ao IMC do indivíduo, sendo alterado para um atributo ordinal assumindo os valores entre $1 - 4$ e representando "baixo peso", "peso normal", "sobrepeso" e "obeso", de acordo com os valores aceitos para cada uma dessas classes [46].

RTHLTH e MNHLTH representam o estado de saúde físico e mental relatado pelo próprio indivíduo, em uma faixa de $1 - 5$, representando "excelente", "muito bom", "bom", "razoável" e "saúde debilitada". Por fim, ANYLIM é um atributo binário que indica se o indivíduo apresenta alguma limitação nas atividades da vida diária, como comer ou vestir-se, ou limitações nas atividades instrumentais da vida diária, como cuidar da casa ou fazer compras.

A categoria de cuidados preventivos contém atributos que indicam o tempo

desde que o indivíduo realizou diversos exames. Como o tempo desde o último *checkup* de rotina (Check), teste de pressão arterial (BPCHEK), verificação de colesterol (CHOLCK), se o indivíduo já precisou se submeter à uma sigmoidoscopia ou colonoscopia (BOWEL), se um exame de sangue oculto nas fezes foi necessário (STOOL) e frequência de check-up odontológico (DENTCK). Esta categoria também inclui informações sobre eventuais recomendações médicas recebidas pelo indivíduo, que incluem a redução do consumo de alimentos com alto teor de colesterol (NOFAT), a se exercitar mais (EXRCIS) e, por fim, se o indivíduo faz uso regular de aspirina (ASPRIN).

Todos os atributos nesta categoria ou são binários, caso em que foram deixados inalterados, ou seguem um padrão semelhante para seus valores possíveis, variando de $1-6$, onde cada valor representa o número de anos desde que o referido evento ou exame ocorreu, com exceção de 5, que representa 5 ou mais anos e 6, representando nunca. Esses valores foram reduzidos para um intervalo de $1-3$, onde 1 representa o ano anterior, 2 representa 2 anos ou mais e 3 representa nunca. Além disso, DENTCK foi alterado para um intervalo de $0-2$, onde 0 representa nunca, 1 duas vezes por ano ou mais e 2 menos de duas vezes por ano.

A categoria de condições prioritárias contém informações sobre o diagnóstico de várias condições de saúde para cada indivíduo, como hipertensão (HIBPDX), doença arterial coronariana (CHDDX), angina ou angina de peito (ANGIDX), ataque cardíaco ou infarto do miocárdio (MIDX), outras doenças cardíacas (OHRTDX), asma (ASTHDX), enfisema (EMPHDX), câncer (CANCERDX), diabetes (DIABDX), artrite (ARTHDX) e colesterol alto (CHOLDX). Como todos os atributos são binários, todos foram deixados inalterados. Os atributos PC e PCCOUNT não estão presentes no conjunto de dados original, sendo derivados dos atributos anteriores, PC é um atributo binário que indica se o indivíduo possui alguma das condições listadas e PCCOUNT aceita valores em um intervalo de $0-4$, onde $0-3$ é o número de condições e 4 representa a presença de 4 ou mais delas.

Por fim, a categoria de contagem de visitas indica o número de passagens que o indivíduo teve por diversos provedores médicos, como visitas a consultorios (OBTOT), atendimento ambulatorial (OPTOT), sala de emergência (ERTOT), hospitalizações (IPDIS). Ela inclui também a quantidade de medicamentos prescritos para o indivíduo (RXTOT). Todos os atributos desta categoria foram normalizados utilizando o MinMax 2.1.

3.3 Implementação do Pré-processamento

Como um dos objetivos deste trabalho, o processo de pré-processamento e preparação do *dataset* foram automatizados, a fim de simplificar trabalhos futuros utilizando o *dataset* do MEPS. Este processo foi dividido em vários passos, detalhados a seguir. Os únicos pré-requisitos necessários para utilizar a implementação descrita neste trabalho são os arquivos relevantes do MEPS, em formato CSV e um ambiente Python 3 funcional.

- O primeiro passo consiste em aplicar as condições mencionadas anteriormente em cada arquivo de ano (*full year consolidated data files*). Este passo exclui os indivíduos com idade inferior a 18, aqueles com algum dos valores de gasto ausentes e aqueles com peso pessoal igual a 0. Como nesse ponto o *dataset* se encontra completo, a adição ou remoção de restrições para quais indivíduos irão compor o dataset final podem ser feitas de forma simples, sem influenciar o resto do processo;

- O segundo passo é necessário apenas para os anos de 2006 e 2007. Em anos anteriores a 2008, o diagnóstico de câncer nao estava presente no mesmo arquivo que o resto das informações do indivíduo, sendo encontrado, ao invés disso, em um arquivo que continha apenas as condições médicas (*medical conditions file*). Este passo busca o arquivo de condições médicas e anexa o atributo de diagnóstico de câncer ao arquivo principal. Para os arquivos de 2008 em diante esta etapa deve ser ignorada;

- O terceiro passo cria os atributos PC e PCCOUNT, contando quantas condições de prioridade estão presentes em cada indivíduo. As condições consideradas de prioridade podem ser facilmente alteradas nesta etapa, simplesmente adicionando ou removendo seus nomes da lista no arquivo relevante;

- O quarto passo busca os gastos referentes ao segundo ano de cada indivíduo, para isso é necessário o arquivo longitudinal para o painel em questao (*longitudinal data file*). Indivíduos sem um valor de despesa para o segundo ano são excluídos;

- O quinto passo cria o atributo *HIGHCOST* para cada indivíduo, neste ponto o usuário pode escolher se considera as despesas do primeiro ou do segundo ano para determinar a população de alto custo, além do percentual do conjunto de dados que será considerado de alto custo, ambas as informações são tomadas como entrada neste passo. Em outras palavras, o usuário pode escolher considerar os 5% no topo como a população

de alto custo considerando suas despesas do segundo ano, neste cenário, 5% do conjunto de dados, aqueles com os valores mais altos para suas despesas totais no ano 2, terá o atributo *HIGHCOST* definido como verdadeiro, enquanto o restante o terá como falso;

- O sexto passo aplica alterações nos valores dos registros, aplicando normalização minmax, ou alterando intervalos de valores, de acordo com as especificações listadas acima. Ao longo dos anos, diversos campos no conjunto de dados tiveram seus nomes, ou mesmo valores, alterados. HIDEG, por exemplo, o campo que representa o mais alto grau de educação que um indivíduo alcançou, não está presente nos arquivos de 2014 e 2013, em vez disso, para obter a mesma informação, é necessário utilizar um campo diferente, chamado EDUYRDG (anos de educação ou grau mais alto). Nesta etapa levamos tais mudanças em consideração, garantindo que cada uma das 39 variáveis descritas anteriormente estejam presentes;

- O sétimo passo lida com valores ausentes ou inválidos em cada um dos campos relevantes no conjunto de dados. Por padrão, são utilizados certos valores para perguntas que o indivíduo não conseguiu ou se recusou a responder. Por exemplo, se o entrevistado não souber a resposta de uma pergunta, o valor para aquele campo é definido como -8, se ele se recusa a responder, o valor é definido como -7 e assim por diante por vários outros motivos. Para os propósitos deste trabalho, tais diferenças não são relevantes, como tal, para cada campo onde essas notações são utilizados, os valores abaixo de 0 são substituídos por -1;

- O oitavo passo pode ser subdividido em 4 partes, primeiramente, ele permite ao usuário adicionar campos que não foram incluídos nos 39 atributos originais utilizados neste trabalho e, dependendo do formato em que os dados estão, normalizá-los, reduzir seu intervalo (para as variáveis que se referem ao tempo desde o último evento) e remover os valores negativos conforme mencionado no passo anterior. Isso permite que o usuário altere facilmente os campos no conjunto de dados final, enquanto aplica o pré-processamento de dados necessário. No final desta etapa, todos os campos não usados são removidos do conjunto de dados;

- O último passo trata do conjunto de dados final, nela o conjunto de dados é dividido em alto e baixo custo, com base na variável criada anteriormente. Para manter o conjunto de dados equilibrado, o número de amostras no conjunto de dados de alto custo são contados, e o mesmo número de amostras no conjunto de dados de baixo custo são escolhidas aleatoriamente. Uma vez que os arquivos contendo os conjuntos de

dados de baixo custo e alto custo não são excluídos no processo, este processo pode ser repetido várias vezes, cada vez criando um novo conjunto de dados de baixo custo com amostras aleatórias.

3.4 Modelagem

O passo seguinte na metodologia é a modelagem, que envolve a análise e construção dos modelos que mais se adequam ao problema. Para este fim, foi utilizado o `Keras` [37], em conjunto com o `Tensorflow` [38].

Sobre seu funcionamento, o modelo é definido inicialmente como sequencial ou funcional, ambos acessados a partir do módulo `keras.models`. A API funcional é utilizada para construção de modelos complexos, como modelos de múltiplas saídas, grafos direcionados acíclicos ou modelos com camadas compartilhadas [47]. Neste trabalho, foi utilizado o modelo sequencial, que é utilizado para gerar redes compostas por camadas de forma sequencial. Definido o modelo, as camadas são adicionadas uma a uma, através do método `add`. Neste passo, é possível definir aspectos como a quantidade de elementos na camada que está sendo adicionada, qual a função de ativação será usada e a dimensionalidade dos dados de entrada e saída.

Adicionadas as camadas, é necessário compilar o modelo, isso é feito através da função `model.compile`. O passo de compilação necessário para definir a função de perda, seu algoritmo de otimização e o parâmetro `metrics`. Após o passo de compilação, o modelo pode ser treinado utilizando a função `fit`, que recebe os conjuntos X e Y, contendo as amostras e as classes correspondentes, o número de épocas consideradas para o treinamento e o tamanho de cada `batch`. Depois de treinado, novas amostras são enviados para o modelo através da função `predict` [47].

O dataset resultante do pré-processamento foi utilizado para treinar diversas redes neurais, mais especificamente, foram utilizados arquivos referentes à um período de 11 anos (2006 – 2016) do dataset do MEPS, o que resultou em um dataset final com 125.457 amostras. Cinco por cento (5%) desse total foram considerados como alto custo, desta forma, o dataset pode ser dividido em alto custo, contendo 6.280 indivíduos, e baixo custo, contendo o restante das amostras.

Inicialmente, a fim de demonstrar a inviabilidade da utilização deste dataset sem mais alterações, foram realizados alguns testes, que comprovaram, como esperado, que o uso de um conjunto de dados tão desbalanceado quanto o descrito acima tende a gerar um modelo ineficaz. Apesar de mostrarem resultados altíssimos nas métricas de *accuracy* e *specificity*, ambos acima de 95%, a *precision* se mostra baixa, apresentando resultados próximos de 60%.

Na métrica *recall*, por sua vez, foram observados resultados ainda piores, ficando abaixo de 5% nos testes realizados. Isso leva, por consequência, a uma queda no valor do *f1 score*, ficando abaixo de 10%.

Dessa forma, passos adicionais devem ser tomados, graças à natureza desbalanceada do problema. Para isso foram consideradas, inicialmente, alguma possíveis soluções. Uma delas seria criar exemplos de indivíduos de alto custo artificiais, até que ambos os conjuntos se tornassem equivalentes. Outra possibilidade é manter os exemplos de alto custo já presentes no dataset, e selecionar o mesmo número de pacientes de custo médio, criando um dataset balanceado e que contém apenas casos reais. Para os propósitos deste trabalho, a segunda opção foi escolhida.

Assim, foram seguidos os passos listados anteriormente para a criação dos conjuntos de dados que seriam utilizados no treinamento dos modelos, mantendo os 6.280 indivíduos de alto custo e selecionando aleatoriamente a mesma quantidade de pessoas de baixo custo. Durante o processo de treinamento, o dataset resultante foi dividido aleatóriamente em conjuntos de treinamento e validação, com o conjunto de treinamento contendo 80% dos indivíduos, e o conjunto de validação contendo os 20% restantes.

Cada modelo foi treinado separadamente 100 vezes, cada vez sorteando novos indivíduos para compor o conjunto de baixo custo. Os resultados obtidos em cada um desses treinamentos foram registrados e um resultado médio para cada métrica foi calculado, usando os resultados encontrados em cada iteração.

O processo descrito foi realizado duas vezes, uma usando as despesas do mesmo ano como parâmetro para definir os indivíduos de alto custo e em seguida, usando as despesas do segundo ano para definir quais amostras são consideradas de alto custo. Os modelos criados foram todos formados por redes sequenciais totalmente conectadas. Com a diferença entre elas sendo o número de camadas, numero de nós em cada camada, aplicação de *dropout* e probabilidade de *dropout*.

Os resultados obtidos à partir das redes de teste podem ser observados na Tabela 3.7, nela, cada linha representa um modelo diferente, treinado e testado seguindo os mesmos processos descritos anteriormente. A rede 1 foi a escolhida no final e é discutida com mais detalhes na Seção 3.5, as demais redes são detalhadas na Tabela 3.6, o campo camadas ocultas se refere a quantidade de camadas presente na rede, quantidade de nós detalha o número de nós em cada uma das camadas ocultas e dropout por camada indica em quais camadas foi aplicado o *dropout*, com a taxa entre parênteses .

Tabela 3.6: Arquiteturas das redes geradas

Rede	Camadas Ocultas	Quantidade de Nós	Dropout por camada
1	3	32, 64, 32	0,5 em cada
2	5	32, 64, 128, 64, 32	1 (0,1), 3(0,25), 5 (0,1)
3	7	16, 32, 64, 128, 64, 32, 16	2 (0,1), 4 (0,25), 6 (0,1)
4	3	58, 116 e 58	2 (0,1)
5	3	58, 116 e 232	2 (0,1)
6	3	58, 116 e 232	1 (0,1), 2 (0,1)
7	3	58, 116 e 232	1 (0,1), 2 (0,25)
8	3	58, 116 e 232	1 (0,1), 2 (0,25), 3 (0,5)
9	3	32, 64, 32	Sem Dropout

Tabela 3.7: Metricas obtidas nas redes de teste para a previsão de pacientes de alto custo no ano subsequente.

Rede	**Accuracy**	**Recall**	**Precision**	**Specificity**	**F1 Score**
1	88%	91%	86%	84%	88%
2	86%	83%	88%	88%	85%
3	83%	79%	85%	87%	80%
4	83%	80%	81%	86%	87%
5	80%	73%	80%	88%	72%
6	86%	90%	83%	82%	86%
7	85%	88%	82%	82%	84%
8	87%	90%	85%	84%	87%
9	83%	83%	82%	84%	80%

3.5 Benchmarking e Análise dos resultados

O penúltimo passo é o processo de benchmarking, que consiste em avaliar os resultados obtidos pelos modelos gerados na fase anterior. Para o processo de benchmarking, inicialmente, serão consideradas métricas relativas à taxa de acerto dos algoritmos, como *accuracy*, *precision*, recall e as métricas geradas a partir delas, como apresentadas na seção 2.4.2.

O modelo utilizado para gerar os resultados exibidos nas Tabelas 3.8 e 3.9 é composto por 3 camadas ocultas, com 32, 64 e 32 nós respectivamente, com *dropout* aplicado em cada uma das camadas, com uma taxa de eliminação de 0,5, uma representação desta rede pode ser observada na Figura 3.1. A função de ativação usada nas camadas foi a Unidade Linear Retificada (`relu`), para as camadas ocultas, e a função tangente hiperbólica (`tanh`) para a camada de saída.

Ao compilar o modelo, o algoritmo `RMSprop` foi utilizado como otimizador e a função de perda utilizada foi a entropia cruzada, ou *binary cross entropy*. Por fim, o modelo foi treinado por 100 *epochs*, com um *batch size* de 32.

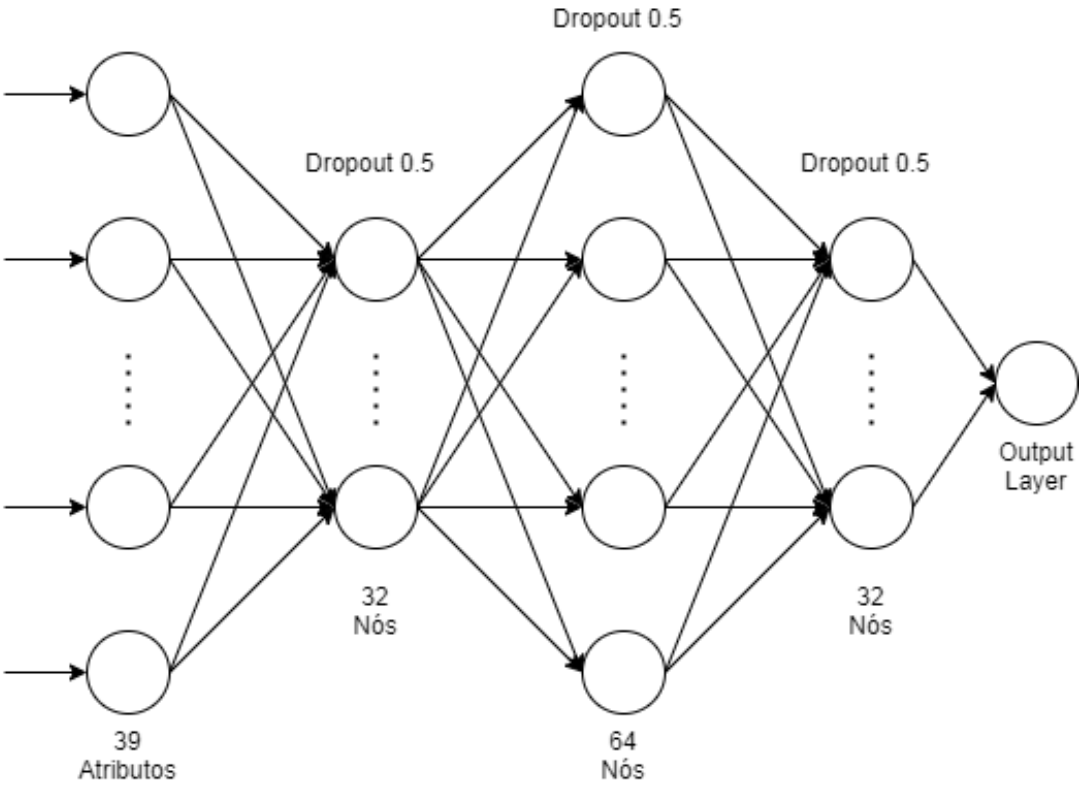

Figura 3.1: Visualização da rede criada, com 3 camadas ocultas e *dropout* em todas as camadas ocultas com taxa de $0,5$

Para previsões utilizando a marcação de alto custo para o mesmo ano, as métricas para o melhor modelo encontrado são detalhadas na Tabela 3.8. Para os modelos treinados para prever indivíduos de alto custo no ano seguinte, os melhores resultados foram detalhados na Tabela 3.9.

Tabela 3.8: Metricas obtidas utilizando os 5% maiores gastos como alvo no mesmo ano

	Valor	**Desvio Padrão**
Sensitivity	90%	0,018
Specificity	83%	0,02
Accuracy	86%	0,007
Precision	84%	0,01
F1 Score	87%	0,006

Todos os resultados são similares aos apresentados por Seyed [2], isso corrobora o conjunto mínimo de atributos definidos naquele trabalho, além de sugerir que resultados similares ou melhores podem ser alcançados com uma quantidade maior ou mais recente de dados.

Apesar de não ser um ponto de comparação direto, por não utilizar exatamente o mesmo dataset, temos como indicativo da qualidade do modelo gerado os resultados de trabalhos passados, como os apresentados por Meehan [3],

Tabela 3.9: Metricas obtidas utilizando os 5% maiores gastos como alvo no ano seguinte

	Valor	**Desvio Padrão**
Sensitivity	91%	0,02
Specificity	84%	0,03
Accuracy	88%	0,009
Precision	86%	0,02
F1 Score	88%	0,008

Tabela 3.10: Metricas apresentadas por Seyed [2] utilizando os 5% maiores gastos como alvo

	C 5.0	**CHAID**	**NN**
Sensitivity	56%	**90**%	87%
Specificity	**96**%	86%	86%
Accuracy	**96**%	86%	86%

exibidos na Tabela 3.11, que utilizou diversos algoritmos tradicionais de *machine learning* para gerar modelos para o mesmo problema.

Tabela 3.11: Resultados apresentados por Meehan [3]

	Logistic Regression	**Naive Bayes**	**J48**
Sensitivity	74%	**74,10**%	67,20%
Specificity	77,10%	76,90%	**82,20**%
Accuracy	**75,55**%	75,49%	74,73%
F-Measure	**75,50**%	**75,50**%	74,60%

Uma observação final pode ser feita ao observar as matrizes de confusão geradas tanto neste trabalho quanto aquelas obtidas por Seyed [2], que podem ser vistas nas Tabelas 3.12, 3.13 and 3.14

É possível notar, ao observar as matrizes que, enquanto as métricas apresentadas anteriormente tenham resultados similares, a proporção de casos onde o modelo devolveu uma previsão de alto custo para um paciente de baixo custo foi maior no modelo gerado por Seyed [2].

Para o modelo de rede neural isso significa que a *precision* fica em torno de 31%, o modelo CHAID apresenta uma *precision* próxima a 26%. Enquanto a média para os modelos gerados neste trabalho se mantém acima de 80%. Apesar de novamente não ser uma comparação direta, dada a diferença no tamanho dos conjuntos, isso também pode ser visto como um indicativo da qualidade do modelo gerado.

Tabela 3.12: Matriz de confusão apresentada por Seyed [2] utilizando os 5% maiores gastos como alvo (Rede Neural)

	Predicted HC	**Predicted LC**
Actual HC	405	56
Actual LC	891	6884

Tabela 3.13: Matriz de confusão apresentada por Seyed [2] utilizando os 5% maiores gastos como alvo (CHAID)

	Predicted HC	**Predicted LC**
Actual HC	460	52
Actual LC	1262	7790

Tabela 3.14: Matriz de confusão gerada por um de nossos testes, utilizando os 5% maiores gastos como alvo

	Predicted HC	**Predicted LC**
Actual HC	1160	229
Actual LC	100	1023

CAPÍTULO

4

Conclusões

Neste capítulo são apresentadas as conclusões deste trabalho. Na Seção 4.1 é realizado um paralelo entre os objetivos desta dissertação e os resultados obtidos. Na Seção 4.2 são discutidas algumas limitações das soluções propostas e na Seção 4.3 são apresentadas algumas direções de trabalhos futuros.

4.1 Resumo dos Objetivos e Principais Resultados

Os objetivos deste trabalho incluíram a implementação de modelos de redes neurais para o problema de previsão de pacientes de alto custo, a comparação desses resultados com aqueles encontrados em trabalhos anteriores, o desenvolvimento das ferramentas necessárias para a criação desses modelos, a fim de simplificar trabalhos futuros nessa área, e por fim, validar o conjunto de atributos definidos por Seyed [2] em um dataset maior e mais recente.

As principais conclusões derivadas deste trabalho incluem a validação do conjunto de atributos definido previamente usando dados da mesma fonte, em um período e quantidade maior, além dos resultados obtidos no problema de predição de pacientes de alto custo usando redes neurais. À partir das métricas e resultados encontrados neste trabalho, podemos concluir que as redes neurais são uma abordagem válida para o problema proposto, especialmente conforme mais dados se tornarem disponíveis futuramente, tanto por parte do dataset do MEPS quanto por outras fontes.

Nossas contribuições incluem a criação das ferramentas necessárias para o pré-processamento dos arquivos do MEPS em datasets utilizáveis, testados em 11 anos de dados, mais especificamente, o período entre 2006 e 2016. Essas

ferramentas permitem que o usuário crie modelos similares aos apresentados neste trabalho sem precisar necessáriamente lidar com a grande quantidade de dados inicialmente oferecidos pelos arquivos do MEPS, além de não precisar lidar com diferenças entre os arquivos no que diz respeito à mudanças de nomes de variáveis ou de seus valores, todas as ferramentas desenvolvidas estão disponíveis no github [48].

Os resultados obtidos à partir deste trabalho geraram também um artigo, ainda não publicado, mas aceito em uma conferência internacional (classificação A3, segundo o novo Qualis), *The 21st International Conference on Computational Science and its Applications* (ICCSA 2021), que será realizada entre os dias 13 e 16 de setembro de 2021, em colaboração com a Universidade de Cagliari, Itália [49].

4.2 Limitações

A principal limitação observada no desenvolvimento deste trabalho se dá na aplicação dos métodos utilizados neste trabalho no mundo real, em outras palavras, fora do conjunto de dados original, vendo que um potencial dataset deveria incluir informações similares àquelas observadas no dataset do MEPS. Isso exigiria um certo esforço no sentido de criar esse conjunto de dados para o cenário no qual se pretende obter resultados similares aos aqui encontrados ou adaptar um conjunto de dados já existente.

4.3 Trabalhos Futuros

Como trabalhos futuros é possível listar, primeiramente, a utilização de uma parcela maior do conjunto de dados do MEPS, quando assim for possível, ou a sua integração com datasets não utilizados neste trabalho. É posível também explorar soluções alternativas para o problema do desbalanceamento do conjunto de dados. Uma outra alternativa seria a utilização de três classes, ao invés de duas, alto, médio e baixo custo, por exemplo, transformando o problema em uma classificação multiclasse, o que aumentaria a quantidade de dados utilizados.

Para o problema de previsão de pacientes de alto custo, um trabalho futuro poderia investigar a utilização de técnicas de *clustering* utilizando o dataset descrito aqui, ou variações dele.

Por fim, a integração da solução proposta neste trabalho com possíveis fontes de dados alternativas, por exemplo do Datasus [40] pode ser um motivador para um projeto futuro, que vise a criação de uma estrutura para a coleta e tratamento desses dados.

Bibliografia

[1] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*, 2014. Citado nas páginas ix e 13.

[2] Seyed Abdolmotalleb Izad Shenas. Predicting high-cost patients in general population using data mining techniques, 2012. dx.doi.org/10.20381/ruor-6153. Citado nas páginas xi, 3, 4, 24, 26, 33, 34, 35, e 37.

[3] Jennica Meehan, Chun-An Chou, and Mohammad T. Khasawneh. Predictive modeling and analysis of high-cost patients. *Industrial and Systems Engineering Research Conference*, 2015. Citado nas páginas xi, 4, 33, e 34.

[4] Alberto Hideki Kanamura and Ana Luiza D'Ávila Viana. Gastos elevados em plano privado de saúde: com quem e em quê. *Documentos / Scielo*, 41:814–820, 2007. https://www.scielosp.org/scielo.php?pid=S0034-89102007000500016&script=sci_arttext. Citado na página 1.

[5] Babak Rashidi, Daniel M. Kobewka, David J. T. Campbell, Alan J. Forster, and Paul E. Ronksley. Clinical factors contributing to high cost hospitalizations in a canadian tertiary care centre. *BMC Health Services Research*, 1:27, 2017. bmchealthservres.biomedcentral.com/track/pdf/10.1186/s12913-017-2746-6. Citado na página 1.

[6] Alberto Hideki Kanamura and Ana Luiza D'Ávila Viana. The kaiser commission on medicaid and the uninsured. *Documentos / Kaiser Family Foundation*, 1:27, 2013. kaiserfamilyfoundation.files.wordpress.com/2010/06/7334-05.pdf. Citado na página 1.

[7] United States government. Medicaid.gov. www.medicaid.gov/about-us/organization/index.html. Citado na página 1.

[8] Yuriy Chechulin, Amir Nazerian, Saad Rais, and Kamil Malikov. Predicting patients with high risk of becoming high-cost healthcare users in ontario (canada). *Healthcare Policy*, 9:68–81, 2014. www.longwoods.com/content/23710. Citado na página 1.

[9] David Blumenthal, Gerard Anderson, Sheila Burke, Terry Fulmer, Ashish K. Jha, and Peter Long. Tailoring complex-care management, coordination, and integration for high-need, high-cost patients a vital direction for health and health care. *Documentos / National Academy of Medicine, Washington, DC*, 9:1–11, 2016. bit.ly/2UkH5FR. Citado nas páginas 1 e 2.

[10] Christy K. Boscardin, Ralph Gonzales, Kent L. Bradley, and Maria C. Raven. Predicting cost of care using self-reported health status data. *Insight Medical Publishing Group*, 2015. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4874657/pdf/nihms755930.pdf. Citado na página 4.

[11] Juan Lu, Erin Britton, Jacquelyn Ferrance, Emily Rice, Anton Kuzel, and Alan Dow. Identifying future high cost individuals within an intermediate cost population. *Documentos / Qual Prim Care*, 2016. Citado na página 5.

[12] Julajak Limsrivilai, Ryan W. Stidham, Shail M. Govani, Akbar K. Waljee, Wen Huang, and Peter D. R. Higgins. Factors that predict high health care utilization and costs for patients with inflammatory bowel diseases. *Clinical Gastroenterology and Hepatology Journal*, 2016. www.cghjournal.org/article/S1542-3565(16)30669-3/fulltext. Citado na página 5.

[13] David E. Kaplan, Michael K. Chapko, Rajni Mehta, Feng Dai, Melissa Skanderson, Ayse Aytaman, Michelle Baytarian, Kathryn D'Addeo, Rena Fox, Kristel Hunt, Christine Pocha, Adriana Valderrama, and Tamar H. Tadde. Healthcare costs related to treatment of hepatocellular carcinoma among veterans with cirrhosis in the united states. *Clinical Gastroenterology and Hepatology Journal*, 2018. www.cghjournal.org/article/S1542-3565(17)30861-3/fulltext. Citado na página 5.

[14] Wang Y, Iyengar V, Hu J, Kho D, Falconer E, Docherty JP, and Yuen GY. Predicting future high-cost schizophrenia patients using high dimensional administrative data. *Frontiers in Psychiatry*, 2017. doi.org/10.3389/fpsyt.2017.00114. Citado na página 5.

[15] Jialing Li Li Luo, Shuhao Lian, Xiaoxi Zeng, Lin Sun, Chunyang Li, Debin Huang, and Wei Zhang. Using machine learning approaches to predict high-cost chronic obstructive pulmonary disease patients in china. *Health Informatics Journal*, 2019. Citado na página 5.

[16] David Blumenthal, Bruce Chernof, Terry Fulmer, John Lumpkin, and Jeffrey Selberg. Caring for high-need, high-cost patients — an urgent priority. *The New England Journal of Medicine*, pages 909–911, 2016. www.nejm.org/doi/10.1056/NEJMp1608511. Citado na página 5.

[17] Douglas McCarthy, Jamie Ryan, and Sarah Klein. Models of care for high-need, high-cost patients: An evidence synthesis. *Documentos / The Commonwealth Fund*, 31, 2015. www.commonwealthfund.org/publications/issue-briefs/2015/oct/models-care-high-need-high-cost-patients-evidence-synthesis. Citado na página 5.

[18] Open Knowledge International. The open definition. opendefinition.org. Citado na página 7.

[19] Tribunal de Contas da União. 5 motivos para a abertura de dados na administração pública. *Documentos / Portal TCU*, 2015. portal.tcu.gov.br/biblioteca-digital/cinco-motivos-para-a-abertura-de-dados-na-administracao-publica.htm. Citado na página 8.

[20] Subchefia para Assuntos Jurídicos Presidência da República Casa Civil. Lei nº 12.527, de 18 de novembro de 2011. www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm. Citado na página 8.

[21] Tom M. Mitchell. *Machine Learning*. WCB McGraw-Hill, 1997. Citado na página 8.

[22] Google. Google translate. translate.google.com. Citado na página 11.

[23] Microsoft. Cortana. www.microsoft.com/pt-br/windows/cortana. Citado na página 11.

[24] Amit Gupta. Introduction to deep learning. www.aiche.org/resources/publications/cep/2018/june/introduction-deep-learning-part-1. Citado na página 11.

[25] Google. Machine learning crash course. developers.google.com/machine-learning. Citado nas páginas 11, 17, e 18.

[26] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Elsevier*, 2014. ac.els-cdn.com/S0893608014002135/1-s2.0-S0893608014002135-main.pdf?_tid=356ec394-af8e-4e83-bcf7-eeb104a3224d&acdnat=1532647754_fb220563f30cf02920cbc6b93645aded. Citado na página 11.

[27] Google. Google ai. ai.google. Citado na página 12.

[28] Jeff Dean and Campus Seoul. Google tech talk with jeff dean at campus seoul. youtu.be/QSaZGT4-6EY?t=9m20s. Citado na página 12.

[29] IBM Cloud Education. Overfitting. www.ibm.com/cloud/learn/overfitting. Citado na página 13.

[30] UCI Machine Learning Repository. Restaurant & consumer data data set. https://archive.ics.uci.edu/ml/datasets/Restaurant+%26+consumer+data. Citado na página 14.

[31] Sebastian Raschka. *Python Machine Learning*. Packt Publishing Ltd, Liverly Place, 35, Livery Street, Birmingham B3 2PB, UK, 2015. Citado na página 14.

[32] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009. Citado na página 18.

[33] David M W Powers. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *School of Informatics and Engineering, Flinders University of South Australia*, 2007. Citado na página 18.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. Citado na página 19.

[35] Python Software Foundation. What is python? executive summary. www.python.org/doc/essays/blurb/. Citado na página 19.

[36] Tensorflow. Página inicial. https://www.tensorflow.org. Citado na página 19.

[37] François Chollet et al. Keras. keras.io, 2015. Citado nas páginas 19 e 30.

[38] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore,

Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. Citado nas páginas 20 e 30.

[39] SAS Institute Inc. Statistical analysis system university edition. support.sas.com/software/products/university-edition. Citado na página 20.

[40] Ministério da Saúde. Datasus. datasus.saude.gov.br. Citado nas páginas 21 e 38.

[41] EU Open Data Portal. European union open data. data.europa.eu/euodp/data/. Citado na página 21.

[42] Agency for Healthcare Research and Quality. Medical expenditure panel survey. meps.ahrq.gov. Citado na página 21.

[43] Open Data Unit. ehealth ireland open data portal. data.ehealthireland.ie. Citado na página 21.

[44] Kaggle INC. Kaggle: Your home for data science. www.kaggle.com. Citado na página 21.

[45] Agency for Healthcare Research and Quality. Meps hc-181 2015 full year consolidated data file. https://meps.ahrq.gov/data_stats/download_data/pufs/h181/h181doc.pdf. Citado na página 22.

[46] Centers for Disease Control and Prevention. Assessing your weight. www.cdc.gov/healthyweight/assessing/index.html. Citado na página 26.

[47] François Chollet et al. Keras documentation. keras.io/getting-started/sequential-model-guide/. Citado na página 30.

[48] Repositório github. https://github.com/franklin-ll/Highcost-MEPS. Citado na página 38.

[49] ICCSA Society. The 21st international conference on computational science and its applications. https://iccsa.org/. Citado na página 38.