



**CENTRO UNIVERSITÁRIO
INSTITUTO DE EDUCAÇÃO SUPERIOR DE
BRASÍLIA - IESB
CIÊNCIA DA COMPUTAÇÃO**

LUIZ DOS SANTOS ARAUJO JUNIOR E PAULO HENRIQUE ALVES

**SISTEMA DE PREDIÇÃO DE
HIPERTIREOIDISMO UTILIZANDO
TÉCNICAS DE INTELIGENCIA ARTIFICIAL**

Brasília – DF

2020

Luiz dos Santos Araujo Junior e Paulo Henrique Alves

**SISTEMA DE PREDIÇÃO DE HIPERTIREOIDISMO
UTILIZANDO TÉCNICAS DE INTELIGENCIA
ARTIFICIAL**

Trabalho de Conclusão de Curso apresentado
como pré-requisito para obtenção do título
de Bacharel em Ciência da Computação pelo
Instituto de Educação Superior de Brasília.

Orientador: Prof^ª. Dra. Leticia Toledo
Maia Zoby

Brasília – DF

2020

SISTEMA DE PREDIÇÃO DE HIPERTIREOIDISMO UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL

Trabalho de Conclusão de Curso apresentado como pré-requisito para obtenção do título de Bacharel em Ciência da Computação pelo Instituto de Educação Superior de Brasília. Brasília, 05 de novembro de 2020.

Banca examinadora:

Prof^a. Dra. Leticia Toledo Maia Zoby

Prof. Cristiano Lehrer

Prof. José Márcio Araújo

AGRADECIMENTOS

A Deus por ter me dado forças para superar todos os obstáculos e dificuldades encontradas ao longo do desenvolvimento deste trabalho.

A minha Prof^a. Dra. Leticia Toledo Maia Zoby, por todo apoio, incentivo e todos os direcionamentos neste trabalho.

Aos nossos pais por todos os incentivos durante os anos de faculdade, por acreditarem em nós e nunca me deixarem perder a fé.

Aos meus grandes amigos da faculdade, pelas palavras de motivação, força, incentivo e otimismo ao longo desta jornada acadêmica.

RESUMO

O Hipertireoidismo, cujos principais sintomas são aumento de apetite, sem perda de peso recentemente, palpitações no coração, coração acelerado, protusão ocular e dentre outros, o hipertireoidismo é 5 a 10 vezes mais comum nas mulheres e com pico de incidência entre a 2^o e 4^o décadas de vida. Por esse motivo, há necessidade de utilizar técnicas computacionais com o intuito de auxiliar a detecção e encaminhamento imediato ao diagnóstico através de um exame sanguíneo para confirmação, diminuindo os impactos nas vidas social e garantindo um tratamento nas fases iniciais. Técnicas computacionais apresentam-se como uma importante aliada dos profissionais de saúde, principalmente no auxílio de diagnósticos, contribuindo para as mudanças dos processos tradicionais para processos cada vez mais precisos e automatizados. Este trabalho consiste na aplicação de técnicas de Inteligência Artificial (IA), com o objetivo de auxiliar na detecção de possíveis portadores de hipertireoidismo utilizando Redes Neurais Artificiais (RNA) do tipo Perceptron de Múltiplas Camadas (PMC) e Máquinas de Vetores de Suporte (MVS), foi construída uma base de dados (*dataset*) através de um questionário fornecido por um especialista em endocrinologia. Os dados obtidos passaram por análises e tratamentos para garantir a qualidade, e em seguida foi utilizado algoritmos, que é a entrada da arquitetura construída.

Palavras-chaves: Hipertireoidismo. Inteligência Artificial. Aprendizado de Máquina. RNA. PMC. MVS.

ABSTRACT

Hyperthyroidism, whose main symptoms are increased appetite, without weight loss recently, heart palpitations, fast heart, eye protrusion and among others, hyperthyroidism is 5 to 10 times more common in women and with a peak incidence between the 2nd and 4th decades of life. For this reason, it is necessary to use computational techniques in order to assist the detection and immediate referral to diagnosis through a blood test for confirmation, reducing the impacts on social lives and guaranteeing treatment in the early stages. Computational techniques are presented as an important ally of health professionals, mainly in the aid of diagnostics, contributing to the changes of traditional processes to increasingly accurate and automated processes. This work consists of the application of Artificial Intelligence (AI) techniques, with the objective of assisting in the detection of possible hyperthyroid patients using Artificial Neural Networks (RNA) of the Multilayer Perceptron type (PMC) and Support Vector Machines (MVS)), a database (textit dataset) was built using a questionnaire provided by an endocrinology specialist. The data obtained went through analyzes and treatments to guarantee quality, and then algorithms were used, which is the entry of the built architecture.

Key-words: Hyperthyroidism. Artificial Intelligence. Machine Learning. SVM. MLP.

LISTA DE ILUSTRAÇÕES

Figura 1 – O Bócio significa qualquer aumento do volume tireoidiano, podendo ser nodular (A e B) ou difuso (C e D).	13
Figura 2 – Fase de um processo de descoberta de conhecimento em base de dados.	14
Figura 3 – Representação de um perceptron.	18
Figura 4 – Exemplo de um problema linearmente separável.	19
Figura 5 – Rede Perceptron multicamadas	20
Figura 6 – Formação do hiperplano de separação através dos vetores de suporte.(SOARES, 2008)	22
Figura 7 – Padrões não-linearmente separáveis: separação linear no espaço de características. (SOARES, 2008)	23
Figura 8 – Arquitetura do trabalho.	25
Figura 9 – Tamanho da base de Dados.	27
Figura 10 – Escalonamento do conjunto.	28
Figura 11 – Implementação da IA.	29
Figura 12 – Acurácia do Modelo.	30
Figura 13 – Precisão de validação de cada <i>K-fold</i>	30
Figura 14 – Média de precisão em cada <i>K-fold</i>	31
Figura 15 – Matriz de confusão - PMC	32
Figura 16 – Matriz de confusão - MVS	32
Figura 17 – Relatório de classificação	33

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
IA	Inteligência Artificial
MVS	Máquinas de Vetores de Suporte
PMC	Perceptron de Múltiplas Camadas
RNA	Rede Neural Artificial

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Objetivos	10
1.2	Conteúdo e Organização	10
2	REVISÃO BIBLIOGRÁFICA	11
2.1	Hipertireoidismo	11
2.2	Inteligência Artificial	13
2.3	O processo de descoberta de conhecimento em base de dados (<i>KDD - Knowledge Discoverey in Databases</i>)	14
2.4	Aprendizado de Máquina	16
2.5	Redes Neurais Artificiais	17
2.6	Máquina de vetores de suporte	21
2.7	Trabalhos Correlatos	24
3	METODOLOGIA	25
3.1	Arquitetura	25
4	IMPLEMENTAÇÃO	27
4.1	Base de dados	27
4.2	Tratamento dos dados	27
4.3	Treinamento e Teste	28
4.4	Resultados	29
5	CONCLUSÃO	34
	REFERÊNCIAS	35
	APÊNDICE A – IMPLEMENTAÇÃO	38
	ANEXO A – QUESTIONÁRIO	43

1 INTRODUÇÃO

O hipertireoidismo é caracterizado pelo aumento da síntese e liberação dos hormônios tireoidianos pela glândula tireoide. Sendo assim, regula a função de órgãos importantes como o coração, o cérebro, o fígado e os rins. Ela produz os hormônios T3 (triiodotironina) e T4 (tiroxina). Dessa forma, garante o equilíbrio do organismo. A glândula possui forma de borboleta (com dois lobos) e se localiza na parte anterior do pescoço, logo abaixo do Pomo de Adão (Departamento de Tireoide, 2015).

A tireoide atua diretamente no crescimento e desenvolvimento de crianças e de adolescentes, na regulação dos ciclos menstruais, na fertilidade, no peso, na memória, na concentração, no humor e no controle emocional.

Quando a tireoide não funciona corretamente, pode liberar hormônios em quantidade insuficiente (hipotireoidismo) ou em excesso (hipertireoidismo). Nos dois casos, o volume da glândula aumenta, o que é conhecido como bócio. Esses problemas podem ocorrer em qualquer fase da vida e são simples de se diagnosticar (Departamento de Tireoide, 2015).

No Brasil, segundo dados do Ministério da Saúde, o hipertireoidismo é o agravo mais comum da tireoide tem prevalência de 2% na população geral e 8 vezes mais frequentes em mulheres do que em homens, tem como principal forma de apresentação (95% dos casos) decorrente de alterações na glândula tireoide (BRENTA, 2013).

Um software médico é um sistema que oferece ferramentas virtuais que têm o principal objetivo de facilitar o trabalho do profissional de saúde. Eles são desenvolvidos especialmente para a área de saúde e podem ser focados na otimização de diferentes processos.

Uma das principais características dos softwares é que eles automatizam muitas tarefas, isto é, os programas passam a realizar atividades que antes eram feitas manualmente pelos profissionais de saúde.

A medicina diagnóstica é uma das que mais se apoiam no uso de softwares inteligentes. É compreensível, uma vez que diagnósticos precisos são determinantes para o sucesso de um tratamento. A inteligência artificial pode ser usada para a obtenção de diagnósticos mais precisos e a recomendações de tratamentos.

Assim, são usadas diversas tecnologias que permitem melhor visualizações de tumores e outras doenças, processam dados de maneira mais rápida ou realizam exames rapidamente.

Uma grande vantagem de muitos softwares é que eles podem ser acessados de qualquer lugar ou dispositivo com acesso à internet. Isso favorece o acompanhamento à distância pelos médicos, o compartilhamento de diagnósticos entre médico e paciente e até

o conhecimento da opinião de outro especialista.

Diante exposto este trabalho, visa auxiliar os profissionais de endocrinologia no diagnóstico de possíveis pacientes com hipertireoidismo utilizando de técnicas de IA, como por exemplo as RNA's, que são modelos computacionais inspirados pelo sistema nervoso central de um animal que são capazes de realizar o aprendizado de máquina bem como o reconhecimento de padrões.

1.1 OBJETIVOS

O objetivo geral desse trabalho, é a implementação de um modelo de classificação e predição utilizando de técnicas de Inteligencia Artificial (IA), capazes de auxiliar na detecção de pessoas com hipertireoidismo com base em um questionário aplicado aos possíveis pacientes, afim de auxiliar o diagnóstico da doença junto a um especialista. Desta forma, os objetivos específicos propostos neste trabalho se constituem em:

- Aplicação de um questionário desenvolvido por um especialista em endocrinologia.
- Criação de uma base de dados ou *dataset* com os questionários preenchidos com casos de pacientes confirmados e não confirmados para que sirvam como treinamento da Inteligência Artificial (IA).
- Utilizar algoritmos de aprendizado de máquina (AM) e Redes neurais Artificiais (RNA) para o agrupamento ou clusterização de possíveis portadores e não portadores de hipertireoidismo.

1.2 CONTEÚDO E ORGANIZAÇÃO

Este trabalho foi dividido em 5 capítulos.

No Capítulo 2 é apresentado o referencial teórico e trabalhos correlatos que servem de suporte para o desenvolvimento do trabalho.

O Capítulo 3 descreve a arquitetura do trabalho e suas diversas etapas e cronograma de Trabalho.

No Capítulo 4 é apresentado como foi realizada a implementação do trabalho, materiais e métodos utilizados e a amostra de dados coletados.

No Capítulo 5 apresenta a conclusão do trabalho e trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Nesta seção será tratados sobre os tópicos iniciais sobre a inteligência artificial, hipertireoidismo, aprendizado de máquina e o processo de descoberta de conhecimento em base de dados (*KDD - Knowledge Discoverey in Databases*).

2.1 HIPERTIREOIDISMO

A tireoide é um dos maiores órgãos endócrinos do corpo, pesando cerca de 15 a 20 g. Localiza-se na região anterior do pescoço, sobre a cartilagem tireoide. Tem dois lobos unidos pelo istmo e, eventualmente, pode estar presente um terceiro lobo, o piramidal. É uma glândula extremamente vascularizada, que produz e armazena hormônios em grande quantidade, aos quais participam da regulação do metabolismo de todas as células do organismo. (SALES, 2016)

Tireotoxicose ou Hipertireoidismo é o estado clínico decorrente do excesso de hormônios tireoidianos circulantes sistemicamente, seja qual for a origem desses hormônios. (SALES, 2016):

- Ingestão de hormônio externo;
- Produção de hormônio por algum tumor;
- Produção excessiva de hormônio pela glândula tireoidiana;
- Liberação de hormônio pré-formado pela glândula tireoidiana que tenha sofrido algum dano celular – autoimune;
- infeccioso, químico ou mecânico com ruptura dos folículos e entre outras formas.

As causas mais comuns de hipertireoidismo, em ordem decrescente, são a doença de Graves (DG), o bócio multinodular tóxico (BMNT) e o adenoma tóxico (AT):

- A doença de graves é uma doença autoimune em que a tireoide é estimulada por auto-anticorpos contra o receptor do TSH.(VILLAR, 2017).
- O BMNT é surgimento de novos nódulos e/ou crescimento de nódulos preexistentes em glândulas tireoides após longos períodos em áreas carentes em iodo. Com o passar do tempo, geralmente esses nódulos progressivamente adquirem certa autonomia e começam a produzir hormônio tireoidiano em excesso. Acontece principalmente em indivíduos mais idosos, sendo a principal causa de hipertireoidismo nessa população de mais idade (SALES, 2016).

- O adenoma tóxico resulta, na maioria dos casos, de mutações no receptor do TSH. Nas duas situações, existe produção autônoma de T3 e T4, portanto elas se manifestam por níveis suprimidos de TSH e elevação do T3 e do T4. Em alguns casos, o T4 pode estar normal, caracterizando a T3-toxicose. Supressão do TSH, sem modificação dos HT, é típica do hipertireoidismo subclínico (HSC), situação encontrada em 0,3 a 1% da população geral e em 2% dos idosos. Entre os pacientes em uso de l-tiroxina para tratamento do hipotireoidismo, até 20% têm supressão do TSH, por uso de doses excessivas. Excepcionalmente, o hipertireoidismo pode resultar de tumor hipofisário secretor de TSH (tireotro-pinoma). Nesses casos, o TSH encontra-se elevado (em 77% dos pacientes) ou normal. (VILLAR, 2017)

Nas tireoidites subagudas (TSA), ocorre destruição da tireoide, com liberação de hormônios pré-formados pela glândula, o que pode levar à supressão do TSH e elevação das concentrações de T3 e T4. Nesses casos, a distinção com os estados de hiperfunção tireoidiana (p. ex., DG, BMNT e AT) é feita pela determinação da captação tireoidiana nas 24 h. Esta última se encontra muito baixa ou indetectável nas TSA e praticamente sempre elevada nas outras condições. Um quadro laboratorial similar é observado em pacientes que tomam doses excessivas de l-tiroxina (l-T4). Nesses casos, a diferenciação pode ser feita pela dosagem da tireoglobulina (Tg), que se mostrará suprimida pela ingestão da l-T4 e caracteristicamente elevada nas TSA, devido à liberação da Tg na circulação, decorrente da destruição dos folículos tireoidianos. Contudo, a ingestão de T3 (ainda muito utilizada por pacientes obesos que querem perder peso) pode resultarem supressão do TSH e T4, mas o T3 estará elevado. Tireotoxicose factícia é a terminologia empregada para definir a tireotoxicose decorrente da ingestão de l-T4 ou T3, sem o conhecimento do médico assistente. (VILLAR, 2017)

Bócio significa qualquer aumento do volume tireoidiano e pode ser difuso ou nodular (Figura 1). Sua presença nem sempre implica disfunção glandular. As terminologias bócio tóxico e bócio atóxico são utilizadas para denotar a presença ou ausência de hipertireoidismo, respectivamente. (VILLAR, 2017)

Figura 1 – O Bócio significa qualquer aumento do volume tireoidiano, podendo ser nodular (A e B) ou difuso (C e D).



Fonte: (VILLAR, 2017).

2.2 INTELIGÊNCIA ARTIFICIAL

Quando surgiu a IA, em meados dos anos 50 havia uma grande expectativa de que em pouco tempo, com o avanço da tecnologia dos computadores, as máquinas chegariam ao mesmo patamar de inteligência humana. Entretanto não foi o que houve mas propiciou-se a criação de muitos outros produtos que levaram outros campos da ciência da computação a evoluir, como banco de dados, engenharia de software e processamento compartilhado.(MEDEIROS, 2018)

Com a invenção da IA, a tecnologia passou a contar com a simulação de processos inteligentes que auxiliam no reconhecimento de padrões, na tomada de decisão ou na execução de tarefas.(MEDEIROS, 2018)

O conceito de inteligência artificial(IA) é algo bem amplo para nós da área de ciência da computação, a IA é uma área de pesquisa que tem como objetivo buscar que dispositivos computacionais possam ou que aumentem sua capacidade racional de modo que seja possível, "**pensar**".

Segundo Barr e Feigenbaum (1981), define IA como a parte da ciência da computação voltada para o desenvolvimento de sistemas de computadores inteligentes, isto é, sistemas que exibem características as quais associa-se com a inteligência no comportamento humano, por exemplo compreensão da linguagem, aprendizado, raciocínio, resolução de problemas, entre outros.

Segundo Charniak e McDermott (1985), IA é o estudo de faculdades mentais

através do uso de modelos computacionais.

Segundo Rich e Knight (1993), falam que a IA é o estudo de como fazer os computadores realizarem coisas que no momento, as pessoas fazem melhor.

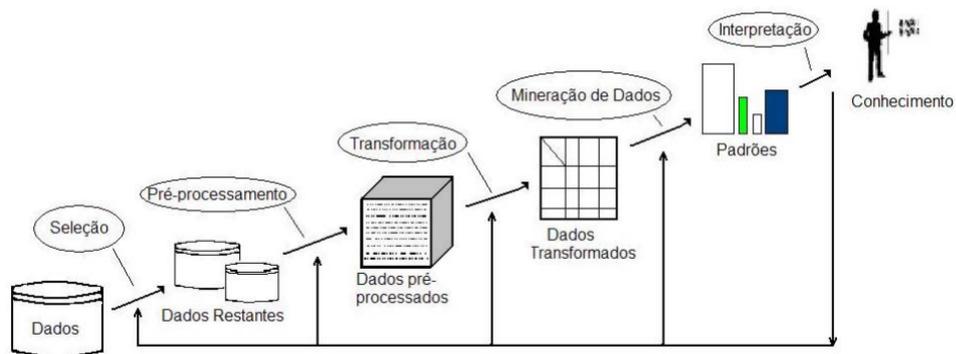
2.3 O PROCESSO DE DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS (*KDD - KNOWLEDGE DISCOVEREY IN DATABASES*)

Dado, informação e conhecimento embora sejam palavras que aparentemente se confundem como sinônimos, têm definições diferentes quando se trata do contexto de Mineração de Dados. O dado é um fato, um valor documentado ou um resultado de medição. Quando um sentido semântico ou um significado é atribuído aos dados, gera-se informação. Quando estes significados se tornam familiares, ou seja, quando um agente os aprende, este se torna consciente e capaz de tomar decisões a partir deles, e surge o conhecimento.(BOSCARIOLI, 2016)

A definição mais utilizada para o processo de KDD é a de Fayyad segundo ele, é o processo iterativo para identificar nos dados novos padrões que sejam válidos, novos, potencialmente úteis e interpretáveis.(FAYYAD; SMYTH, 1996)

Os passos conforme a Figura 2 (FAYYAD; SMYTH, 1996):

Figura 2 – Fase de um processo de descoberta de conhecimento em base de dados.



Fonte: (FAYYAD; SMYTH, 1996).

- A tarefa de Seleção dos dados é crítica porque os dados podem não estar disponíveis em formato apropriado para serem utilizados no processo de KDD. Ou, mesmo se disponíveis, os dados podem precisar ser rotulados com o auxílio de um especialista do domínio (ALMEIDA; DUMONTIER, 1996). Na seleção é escolhido o conjunto de dados pertencente a um domínio, contendo todas as possíveis variáveis e registros que farão parte da análise, também são definidos quais os dados são mais importantes. Os dados coletados podem possuir diferentes formatos como arquivo texto, banco de

dados, planilhas e etc. Cada formato de armazenamento será exigido uma ferramenta específica para a extração desses dados que pode se tornar complexa e se realizada de forma incorreta, podem significar o fracasso das etapas subsequentes.

- No Pré Processamento é a mais complexa, podendo tomar até 80% de todo tempo do processo e precisa ser feita com especialistas que conhecem bem o domínio de aplicação dos dados, já que suas atividades são por exemplo, a integração de dados heterogêneos e a eliminação de incompletude dos dados. Além disso, podem aparecer problemas que são específicos para cada aplicação e que dessa forma, precisam ser resolvidos com soluções específicas (MANNILA, 1996). Seguindo no pré processamento, serão realizadas tarefas que eliminem dados redundantes e inconsistentes, recuperem dados incompletos e avaliem possíveis dados discrepantes ao conjunto (*outliers*), dados ausentes (*missing values*), dados derivados e além de poder existir erros de digitação ou erros nos sistemas de captura dos dados por sensores. A Limpeza dos dados tem por objetivo assegurar a qualidade dos dados selecionados, pois vai determinar a eficiência dos algoritmos de mineração.
- Na Transformação após serem selecionados, pré-processados e limpos os dados necessitam ser armazenados e formatados adequadamente para que os algoritmos de aprendizado possam ser aplicados.
- A Mineração de dados é a etapa mais importante do processo de KDD. Segundo (POSSA, 1998), o cérebro humano, comprovadamente consegue fazer até 8 comparações ao mesmo tempo. A função da mineração de dados é justamente ampliar esta comparação para "**infinito**" e tornar visível aos olhos humano. Mineração de dados é o processo de proposição de várias consultas e extração de informações úteis, padrões e tendências frequentemente desconhecidas, a partir de grande quantidade de dados armazenada em banco de dados (THURASINGHAM, 1999). Mineração de dados tem sido considerada e classificada como uma mistura de pesquisas em estatística, inteligência artificial e banco de dados. Até recentemente não era reconhecido como um campo de interesse para os estatísticos, sendo mesmo considerado nesta área, como uma área de pesquisa 'pouco relevante'. Devido à sua importância prática, entretanto o campo tem emergido como uma área de crescimento acentuado e de elevada importância destacando-se pelo surgimento de diversos congressos científicos e produtos comerciais. (HAN; KAMBER, 2001)
- Na fase de Avaliação e Interpretação de resultados, o conhecimento adquirido através da técnica de *data mining* deve ser interpretado e avaliado para que o objetivo final seja alcançado. Os resultados do processo de descoberta de conhecimento podem ser mostrados de diversas formas. Esta fase envolve todos os participantes que avaliam de forma criteriosa os resultados proporcionando uma interpretação para o modelo.

2.4 APRENDIZADO DE MÁQUINA

Aprendizado de máquina é uma subárea da Inteligência Artificial que tem apresentado um crescimento enorme nas últimas décadas. Trata-se de algoritmos matemáticos, estatísticos e computacionais que são capazes de realizar um processo de inferência por meio de aprendizado baseado em exemplos. Há duas décadas, acreditava-se que os resultados alcançados por esses algoritmos seriam viáveis apenas nas mentes de autores de ficção científica. (FREITAS; JUNIOR, 2019)

Aprendizado de máquina é a área ideal para a automatização de processos, os quais podem ser "simples", como reconhecer padrões visuais, ou complexos, tais quais decisões de especialistas da área da saúde. Quando seres humanos lidam com problemas complexos, muitas vezes é impossível explicar o raciocínio que levou a tomar determinadas decisões. Por outro lado, é menos complexo realizarmos a coleta dos exemplos de decisões tomadas por seres humanos e usá-los como fonte para que o sistema aprenda a resolver o mesmo problema. (FREITAS; JUNIOR, 2019)

Utiliza técnicas para permitir que um dispositivo computacional adquira experiência, por meio do conhecimento que já aprendeu, para melhorar a sua performance. Ele tem como base as ideias do aprendizado indutivo, que como (KOTSIANTIS, 2007) diz, é o problema de induzir uma hipótese que cubra todos os exemplos positivos e não cubra nenhum dos exemplos negativos.

De acordo com KOTSIANTIS (2007), ao longo do tempo em que algoritmos de *Machine Learning* ou aprendizado de máquina eram criados, foi utilizado embasamento empírico, teórico ou até mesmo uma combinação de ambos. Esses sistemas têm sido desenvolvidos utilizando diferentes paradigmas de aprendizado, tais como estatístico, conexionista, baseado em instância, genético e sistemas de aprendizado simbólico entre outros.

Segundo Russell e Norvig (2003), para alguns sistemas de aprendizagem é necessário predizer se uma certa ação irá fornecer uma certa saída. Nesse sentido, é possível classificar os sistemas de AM nos seguintes modos:

- Aprendizagem supervisionada, no qual dado um conjunto de observações ou exemplos rotulados, isto é, conjunto de observações em que a classe, denominada também atributo meta, de cada exemplo é conhecida, o objetivo é encontrar uma hipótese capaz de classificar novas observações entre as classes já existentes.
- Aprendizagem não-supervisionado, no qual dado um conjunto de observações ou exemplos não rotulados, o objetivo é tentar estabelecer a existência de grupos ou similaridades nesses exemplos.

- Aprendizagem semi-supervisionado, no qual dado um pequeno conjunto de observações ou exemplos rotulados e um conjunto de observações ou exemplos não rotulados, o objetivo é utilizar ambos os conjuntos para encontrar uma hipótese capaz de classificar novas observações entre as classes já existentes. Aprendizagem semi-supervisionada é um meio termo entre aprendizagem supervisionada e não-supervisionada.
- Aprendizagem por reforço, no qual o agente aprendiz interage com o meio ambiente que o cerca e aprende uma política ótima de ação por experimentação direta com o meio. Dependendo de suas ações, o aprendiz é recompensado ou penalizado. O objetivo do aprendiz é desenvolver uma política ótima que maximize a quantidade de recompensa recebida ao longo da sua execução.

Segundo KOTSIANTIS (2007), um entendimento detalhado dos algoritmos de aprendizado de máquina pode levar também a um melhor entendimento da capacidade (e incapacidade) do aprendizado humano. A praticidade do aprendizado de máquina e sua interdisciplinaridade em áreas como estatística, inteligência artificial, teoria da informação, proporciona sua aplicação para diversas atividades. Visando uma maior experiência dos programas para que quando deparado com atividades possa tomar decisões práticas para contornar problemas presentes em áreas como inteligência artificial, probabilidade e estatística, complexidade computacional, teoria da informação, psicologia, neurociências, filosofia.

2.5 REDES NEURAIIS ARTIFICIAIS

Uma Redes Neurais Artificial (RNA) pode ser definida como sendo uma estrutura de processamento (rede), passível de implementação em dispositivos eletrônicos, composta por um número de unidades interconectadas (neurônios artificiais), sendo que cada unidade apresenta um comportamento específico de entrada/saída (computação local), determinado pela sua função de transferência, pelas interconexões com outras unidades, dentro de um raio de vizinhança, e possivelmente pelas entradas externas. (ZUBEN, 2013)

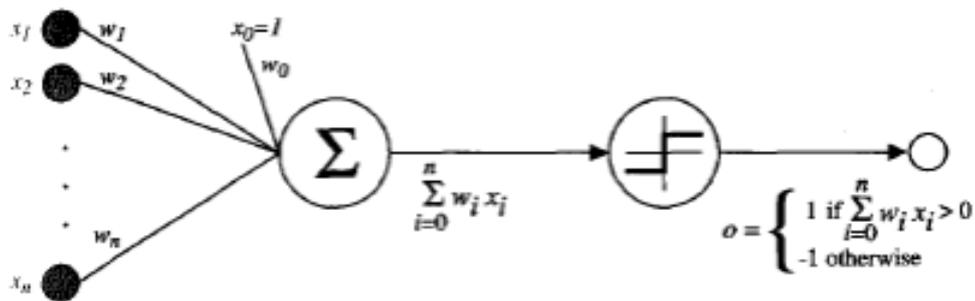
Uma rede neural artificial é um circuito composto por uma grande quantidade de unidades simples de processamento inspiradas no sistema neural. (NIGRIN, 1993)

Uma RNA é um sistema massivamente paralelo e distribuído, composto por unidades de processamento simples que possuem uma capacidade natural de armazenar e utilizar conhecimento. (HAYKIN, 1999)

As RNAs apresentam diversas características em comum com o sistema nervoso, (ZUBEN, 2013):

- O processamento básico de informação ocorre em diversas unidades simples denominadas de neurônios artificiais ou simplesmente neurônios (ou nós);
- Os neurônios estão interconectados gerando redes de neurônios, ou redes neurais;
- A informação (sinais) é transmitida entre neurônios através de conexões ou sinapses;
- A eficiência de uma sinapse, representada por um peso associado, corresponde à informação armazenada pelo neurônio e, portanto, pela rede neural; e
- O conhecimento é adquirido do ambiente através de um processo de aprendizagem que é, basicamente, responsável por adaptar os pesos das conexões aos estímulos recebidos do ambiente.

Figura 3 – Representação de um perceptron.

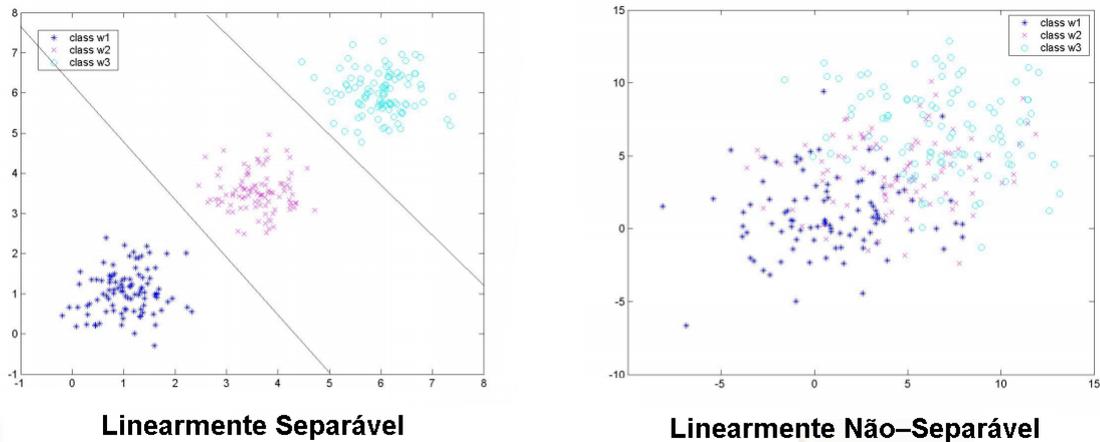


Fonte: (MITCHELL, 1997).

Como é representado na Figura 3, um perceptron é um tipo de aplicação de rede neural e o seu funcionamento é descrito por (MITCHELL, 1997), como uma função que a partir de um vetor de entradas de valor real calcula uma combinação linear delas. Caso o resultado for maior que o limiar, então gera o valor 1, se não gera o valor -1.

Conforme KOTSIANTIS (2007), diz que os perceptrons podem apenas classificar conjuntos de instâncias linearmente separáveis, como na Figura 4. Caso uma linha reta ou plano puder ser desenhado para separar as instâncias de entrada em suas categorias corretas, Então elas serão linearmente separáveis e o perceptrons encontrará a solução. Se as instâncias não forem linearmente separáveis, o aprendizado nunca chegará a um ponto, em que todas as instâncias sejam classificadas corretamente. Este princípio condicional foi enunciado como Teorema de Convergência do Perceptron (MINSKY; PAPERT, 1969).

Figura 4 – Exemplo de um problema linearmente separável.



Fonte: (KOERICH, 2014)

O Perceptron, foi construído em hardware e é a mais antiga rede neural ainda em uso hoje. Um *Perceptron* de camada única foi útil para classificar um conjunto de entradas de valor contínuo em uma de duas classes. Infelizmente, o Perceptron é limitado e foi comprovado como tal durante os “anos desiludidos” por *Marvin Minsky* e o livro de *Seymour Papert* de 1969, *Perceptrons* (Data Science Academy, 2019).

As redes Perceptron de Múltiplas Camadas (PMC), já são caracterizadas pela presença de pelo menos uma camada intermediária (escondida) de neurônios, situada entre a camada de entrada e a respectiva camada neural de saída. Conseqüentemente, as redes PMC possuem no mínimo duas camadas de neurônios, os quais estarão distribuídos entre as camadas intermediárias e a camada de saída (SILVA et al., 2016).

O resultado é que as redes de *Backpropagation* “aprendem” de forma mais lenta, pois necessitam, possivelmente, de milhares de iterações para aprender, mas geram um resultado muito preciso. (Data Science Academy, 2019)

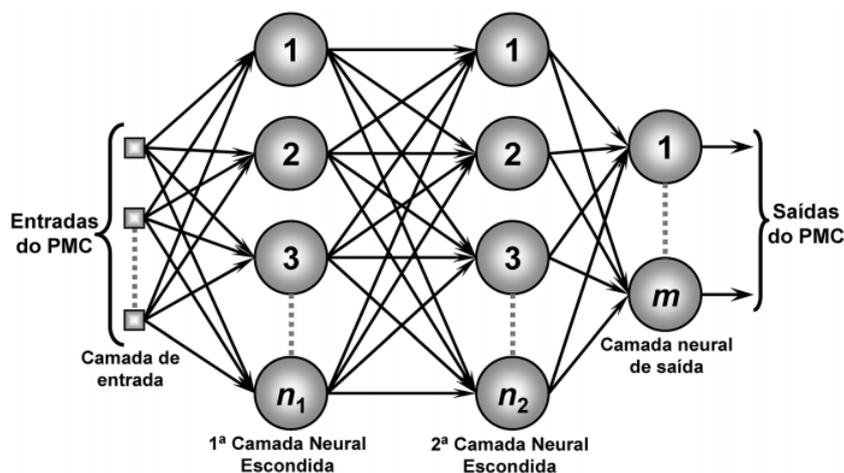
As redes PMC são ainda caracterizadas pelas elevadas possibilidades de aplicações em diversos tipos de problemas relacionados com as mais diferentes áreas do conhecimento, sendo também consideradas uma das arquiteturas mais versáteis quanto à aplicabilidade. Entre essas potenciais áreas, têm-se os seguintes destaques, (SILVA et al., 2016):

- Aproximação universal de funções;
- Reconhecimento de padrões;
- Identificação e controle de processos;
- Previsão de séries temporais;

- Otimização de sistemas.

A camada oculta possui esse nome porque não é possível prever a saída desejada nas camadas intermediárias. O vetor de valores de entrada x passa pela camada inicial, cujos valores de saída são ligados às entradas da camada seguinte, e assim por diante, até a RNA fornecer como resultado os valores de saída da última camada. Pode-se arranjar a RNA em várias camadas, tornando-a profunda e capaz de aprender relações cada vez mais complexas. A figura 5 mostra um exemplo de PMC. (FLAUZINO, 2010)

Figura 5 – Rede Perceptron multicamadas



Fonte: (MITCHELL, 1997)

Diferentemente do Perceptron, onde existe apenas um único neurônio de saída y , a PMC pode relacionar o conhecimento a vários neurônios de saída. O algoritmo de aprendizado da PMC é chamado backpropagation e composto de 4 passos: (FLAUZINO, 2010)

1º Passo: Inicialização

- Atribuir valores aleatórios para os pesos e limites;
- Escolha dos valores iniciais influencia o comportamento da rede;
- Na ausência de conhecimento prévio os pesos e limites devem ter valores iniciais aleatórios e pequenos uniformemente distribuídos.

2º Passo: Ativação

- Calcular os valores dos neurônios da camada oculta;
- Calcular os valores dos neurônios da camada de saída.

3º Passo: Treinar os Pesos

- Calcular os erros dos neurônios das camadas de saída e oculta;
- Calcular a correção dos pesos;
- Atualizar os pesos dos neurônios das camadas de saída e oculta.

4º Passo: Iteração

- Repetir o processo a partir do passo 2 até que satisfaça o critério de erro.

A grande vantagem destes algoritmos é a capacidade que têm de encontrar padrões na informação a eles disponibilizada e a adaptação aos problemas em questão. Contudo esta mesma capacidade de adaptação, também característica do cérebro humano, tem as suas desvantagens. As RNA têm tendência a ter problemas de *overfitting* que ocorre quando o modelo se adaptou muito bem aos dados com os quais está sendo treinado; porém, não generaliza bem para novos dados. Ou seja, o modelo “decorou” o conjunto de dados de treino, mas não aprendeu de fato o que diferencia aqueles dados para quando precisar enfrentar novos testes. Outro fator menos positivo destes métodos é o problema dos mínimos locais. Durante o processo de aprendizagem por vezes as correções levam a piores classificadores, quando isto acontece a RNA, como não melhorou o resultado da classificação, interrompe o processo de aprendizagem, julgando ter chegado à solução mais adequada. Contudo por vezes se o treino continuasse seriam encontradas soluções melhores até atingir o mínimo global. Apesar desta desvantagem, muitas vezes os mínimos locais encontrados têm desempenhos semelhantes ao mínimo global. (RUMELHART et al., 1986)

2.6 MÁQUINA DE VETORES DE SUPORTE

O Support Vector Machine (SVM), também conhecido como Máquina de vetores de suporte (MVS), foi elaborado com o estudo proposto por Boser, Guyon e Vapnik em 1992. Ele é um algoritmo de aprendizado supervisionado, cujo objetivo é classificar determinado conjunto de pontos de dados que são mapeados para um espaço de características multidimensional usando uma função kernel, abordagem utilizada para classificar problemas. Nela, o limite de decisão no espaço de entrada é representado por um hiperplano em dimensão superior no espaço. (BOSER et al., 1992)

As Máquinas de Vetores Suporte constituem uma técnica embasada na Teoria de Aprendizado Estatístico (VAPNIK, 1995) que vem recebendo grande atenção nos últimos anos. Os resultados da aplicação desta técnica são comparáveis aos obtidos por outros algoritmos de aprendizado, como as Redes Neurais Artificiais (RNAs).

Algumas das principais características das MVS que tornam seu uso atrativo são (SMOLA A. J.; SCHUURMANS, 1999):

- Boa capacidade de generalização: os classificadores gerados por uma MVS em geral alcançam bons resultados de generalização. A capacidade de generalização de um classificador é medida por sua eficiência na classificação de dados que não pertençam ao conjunto utilizado em seu treinamento. Na geração de preditores por MVS, portanto, é evitado o *overfitting*, situação na qual o preditor se torna muito especializado no conjunto de treinamento, obtendo baixo desempenho quando confrontado com novos padrões.
- Robustez em grandes dimensões: as MVS são robustas diante de objetos de grandes dimensões, como, por exemplo, imagens. Comumente há a ocorrência de *overfitting* nos classificadores gerados por outros métodos inteligentes sobre esses tipos de dados.
- Convexidade da função objetivo: a aplicação das MVS implica na otimização de uma função quadrática, que possui apenas um mínimo global. Esta é uma vantagem sobre, por exemplo, as Redes Neurais Artificiais, em que há a presença de mínimos locais na função objetivo a ser minimizada.
- Teoria bem definida: as MVS possuem uma base teórica bem estabelecida dentro da Matemática e Estatística.

As MVS foram definidas inicialmente para a classificação de padrões linearmente separáveis. A Figura 6 ilustra um exemplo de duas classes de padrões linearmente separáveis, dada por +1 e -1. (SOARES, 2008)

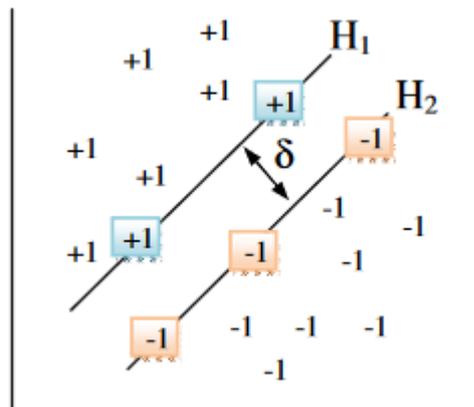


Figura 6 – Formação do hiperplano de separação através dos vetores de suporte.(SOARES, 2008)

No exemplo da Figura 6 é fácil observar uma linha de separação para as duas classes. Neste caso particular para duas classes linearmente separáveis, o SVM encontra um único hiperplano com máxima margem de separação denotada por δ . A solução

para o problema de classificação é representada pelos vetores de suporte, os quais são fundamentais na determinação do hiperplano de separação com margem máxima, onde esta é uma característica especial do SVM. A solução para o problema de classificação é representada pelos vetores de suporte, os quais são fundamentais na determinação do hiperplano de separação com margem máxima, onde esta é uma característica especial do SVM. (SOARES, 2008)

A Figura 7 (a) exemplifica a SVM de forma semelhante para separar classes que não podem ser separadas com um classificador linear, isto é, padrões não-linearmente separáveis. Na SVM para padrões não-linearmente separáveis, os padrões de entrada são transformados em um vetor de características de alta dimensionalidade, cujo objetivo é separar as características linearmente no espaço, como ilustrada na Figura 7 (b). (SOARES, 2008)

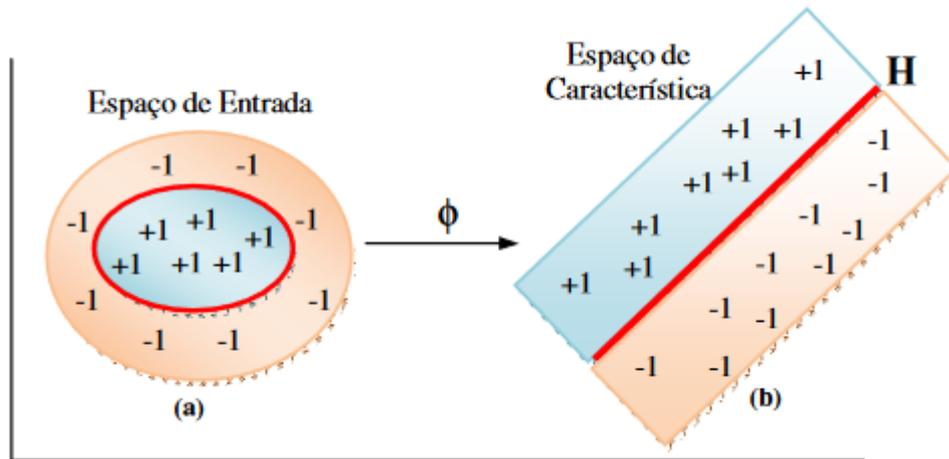


Figura 7 – Padrões não-linearmente separáveis: separação linear no espaço de características. (SOARES, 2008)

Como apresentado na Figura 7, O Kernel tem a vantagem de operar no espaço de entrada, onde a solução do problema de classificação é feito através da soma ponderada da função Kernel avaliada pelos vetores de suporte. Resumindo, o Kernel torna possível a construção de um hiperplano de separação ótimo no espaço de características sem ter que considerar o próprio espaço de características de forma explícita. (HAYKIN, 2001)

Uma vez que o espaço adequado de características é definido, a MVS seleciona o hiperplano particular, chamado de hiperplano de margem máxima (MMH - Maximal Margin Hyperplane), o qual corresponde à maior distância de seus padrões no conjunto de treinamento. Para isto a MVS faz uso de duas teorias: Dimensão VC (Vapnik – Chervonenkis) e a Minimização do Risco Estrutural. (SANTOS, 2004)

A dimensão VC de um conjunto de funções de classificação é o número máximo de exemplos de treinamento que podem ser aprendidos pela máquina sem erro, para todas as rotulações possíveis das funções de classificação. Nos problemas de aprendizagem

supervisionada, pretende-se obter o melhor desempenho de treinamento adequando-se a capacidade da máquina com a quantidade disponível de dados de treinamento para o problema em questão. método de minimização do risco estrutural fornece um procedimento indutivo, no qual utilizando a dimensão VC como uma variável de controle, pode-se obter este melhor desempenho. (SOARES, 2008)

2.7 TRABALHOS CORRELATOS

Ozyilmaz e Yildirim (2002), neste trabalho vários métodos de RNA foram utilizados para ajudar no diagnóstico de doenças da tireoide. As RNA's atuaram em cima do hormônios T3 e T4. A interpretação adequada dos dados funcionais da glândula é uma questão importante no diagnóstico da doença da tireoide.

Saiti et al. (2009), o autor focou em Máquinas de Vetores de Suporte (MVS) e redes neurais probabilísticas para a classificação do Hipotireoidismo, relacionado à produção insuficiente de hormônio tireoidiano, e hipertireoidismo, relacionado à produção excessiva de hormônio tireoidiano. A separação dessas duas doenças é muito importante para o diagnóstico da tireoide. Portanto esses métodos se baseiam principalmente em algoritmos de classificação para lidar com recursos redundantes e irrelevantes. A seleção de recursos é discutida como um problema importante via diagnóstico e demonstra que os prontuários médicos fornecem uma estrutura simples, geral e poderosa para a seleção de bons subconjuntos de recursos, levando a uma melhor taxa de diagnóstico.

Begum e Parkavi (2019), realizou um estudo sobre Máquinas de Vetores de Suporte (MVS) para classificação baseada em mineração de dados, A tarefa principal era detectar a doença nos estágios iniciais com maior precisão. As técnicas de mineração de dados desempenharam um papel importante no campo da saúde para a tomada de decisão, o diagnóstico de doenças e o melhor tratamento para os pacientes, a baixo custo.

O autor Shahid et al. (2019), realizou uma pesquisa sobre classificadores Random Forest (RF), MVS e K-Nearest Neighbors (K-NN) são comparados para avaliar a eficiência desses classificadores nos diagnósticos de doenças da tireoide usando o conjunto de dados de doenças da tireoide retirado de Repositório de aprendizado de máquina. A precisão geral da classificação de RF, MVS e K-NN é 98,50%, 97,02% e 95,81%, respectivamente. O resultado mostra que o desempenho do classificador de RF é melhor que MVS e K-NN para o diagnóstico de doença da tireoide usando o conjunto de dados.

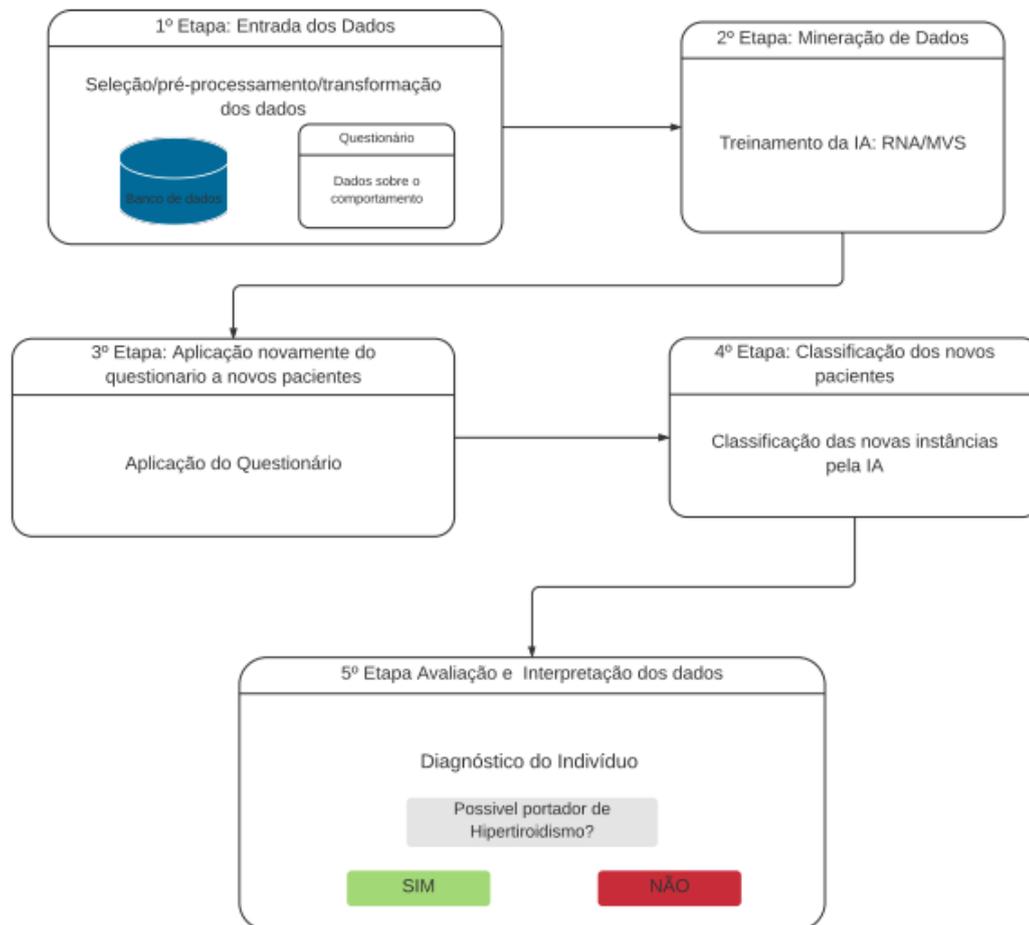
3 METODOLOGIA

Este capítulo, apresenta a arquitetura e técnicas utilizadas para o desenvolvimento do trabalho e o processo de implementação do mesmo.

3.1 ARQUITETURA

A arquitetura do trabalho consiste basicamente de 5 etapas, apresentadas na Figura 8 :

Figura 8 – Arquitetura do trabalho.



Fonte: Elaboração própria

- Etapa 1: Consiste na obtenção e seleção de dados. Estes dados são obtidos por meio dos questionários já preenchidos fornecidos por um especialista em endocrinologia, o questionário é composto por 22 questões sobre comportamento diário dos possíveis

portadores, vide anexo A. Nesta etapa também é realizada a preparação dos dados para serem utilizados pelos algoritmos de RNA e de AM. Algumas tarefas são necessárias:

- pré processamento: Importante pois a qualidade dos dados irão determinar a eficiência dos algoritmos, e eliminação dos dados redundantes e inconsistentes.
 - transformação: necessitam serem armazenados e formatados adequadamente (em um arquivo de excel) para que os algoritmos de aprendizado possam ser aplicados
- Etapa 2: Reside no treinamento supervisionado da Rede Neural Artificial (RNA) e da Máquina de vetores de suporte (MVS) em cima do *dataset* criado na etapa anterior. Nesta fase, o *dataset* é dividido em 10 partes sendo 9 de treinamento e 1 de teste.
 - Etapa 3: Consiste na aplicação do mesmo questionário a novos pacientes que não possuem diagnóstico. Esses dados serão a entrada da próxima etapa a fim de obter a classificação.
 - Etapa 4: Utilizando a rede já treinada (Etapa 2), as novas entradas serão classificadas como possível portador ou não de hipertireoidismo.
 - Etapa 5: Baseia-se na apresentação dos resultados obtidos após a fase de classificação, onde verifica-se qual foi o valor inserido para o indivíduo, se é ou não um possível portador do Hipertireoidismo, obtendo-se assim um primeiro diagnóstico e a orientação se deve procurar um endocrinologista, caso seja necessário.

4 IMPLEMENTAÇÃO

Este capítulo apresenta detalhadamente o processo de implementação do trabalho.

O trabalho foi implementado na linguagem de programação *Python*, vide o apêndice A por possuir um alto desempenho, produtividade e legibilidade, ideal para se trabalhar com problemas de análise de dados, visualização de dados e utilização de técnicas de AM e RNAs. Diversas bibliotecas do *Python* também foram utilizadas, como: *NumPy*, *Pandas*, *scikit-learn* e *Matplotlib* (BOBRIAKOV, 2018).

4.1 BASE DE DADOS

Foi criada uma base de dados para este trabalho, onde os dados foram obtidos através de um questionário preenchido sobre o comportamento diário dos possíveis portadores de hipertireoidismo. O questionário foi transcrito manualmente para um arquivo de excel formando assim a nossa base de dados. As respostas "sim", foram transcritas para o valor 1, e as resposta "não" foram transcritas para o valor 0, a base de dados criada, sem nenhum tratamento, é constituída de 506 respostas (Figura 8). O questionário desse trabalho foi fornecido por um profissional de saúde da área de endocrinologia, sendo que a mesma não quis se identificar.

Figura 9 – Tamanho da base de Dados.

In [2]:

```
# Carrega o dataset
base1 = pd.read_excel('Dataset.xlsx')
# Verifica o tamanho da base
len(base1)
```

Out[2]:

506

Fonte: Elaboração própria

4.2 TRATAMENTO DOS DADOS

Após a criação da base os dados passaram por um pré-processamento, utilizando a biblioteca *Pandas* do *Python*, essencial para tratamento e análise de dados.

Utilizando a biblioteca *scikit-learning* do *Python*, foi realizada o escalonamento no conjunto de dados, que basicamente deixa os dados na mesma escala. Os dados foram normalizados utilizando uma técnica *StandardScaler*. Sua principal característica ao normalizar os dados, é obter uma distribuição relativamente normal dos mesmos. Ideal para

que os algoritmos utilizados tenham um melhor desempenho e convirjam mais rápido (figura 10).

Figura 10 – Escalonamento do conjunto.

```
# Instancia o StandardScaler
scaler = StandardScaler()
# Deixa os atributos(previsores) na mesma escala - escalonar
previsores = scaler.fit_transform(previsores)
```

Fonte: Elaboração própria

4.3 TREINAMENTO E TESTE

Com o conjunto de dados preparado, esta etapa consiste na aplicação do modelo PMC e MVS, para classificação de possíveis portadores ou não de Hipertireoidismo. Foi utilizado uma das técnicas do *scikit-learning* mais importantes para a avaliação da capacidade de generalização de um modelo, o *Cross-Validation*, especificamente do tipo *K-Fold Cross-Validation*. A execução do *Cross-Validation* mantém de lado uma amostra dos dados a qual não é utilizada para treinar o modelo, e posteriormente é utilizada essa amostra para teste e validação.

O *dataset* é dividido em conjuntos de de 90% treino e 10% teste, usando o conjunto de treino para treinar o modelo e o conjunto de teste para avaliar o quanto o modelo generaliza para dados que ele ainda não viu. Devido à pequena quantidade de dados para treinar os modelos, remover uma parte deles para validação certamente apresentará um risco de perder os padrões, o que por sua vez aumenta o erro induzido. O ideal foi aplicar um método para o treinamento do modelo e também para deixar dados para validação. O *Cross-Validation* do tipo *K-Fold* faz exatamente isso.

O PMC e MVS foram configurado da seguinte maneira (figura 11):

- Uma camada oculta.
- 22 neurônios nas camadas Entrada
- 12 neurônios nas camadas ocultas, porque a classe de previsores mais a classe de resposta dividido por 2 tem por resultado um ponto flutuante sendo assim arredondado para 12.
- Função de ativação *reLU*: As entradas negativas são zeradas, ocorrendo assim a não ativação do neurônio.
- Função de ativação *Linear* para a MVS.

- Otimizador Adam, que faz basicamente uma descida de gradiente, mas aplica médias móveis com o passo anterior e com a derivada obtida via *Backpropagation*, realizando atualizações mais suaves; ele incorpora momento e taxa de aprendizado adaptativa com um *learning_rate=0.001*.
- Função de perda (função de pontuação de otimização): *binary_crossentropy* que é utilizada em classificação binária.
- A métrica *accuracy* que basicamente é uma função usada para avaliar o desempenho de um modelo.
- 10 *batch_size*. Número de amostras por atualização de gradiente.
- 100 *epochs* (época). 100 épocas para treinar o modelo, onde uma época é uma iteração sobre o todo x (previsores) e os y (classe) dados fornecidos.
- 1 neurônio na camada de saída (problema binário).
- Função de ativação sigmoid na camada de saída.

Figura 11 – Implementação da IA.

```
# Validação cruzada
kfold = StratifiedKFold(n_splits = 10, shuffle=True, random_state=4)
resultados_pmc = []
resultados_svm = []
for indice_treinamento, indice_teste in kfold.split(previsores,np.zeros(shape=(previsores.shape[0], 1))):

    # Classificador SVM
    svm = SVC(kernel='linear', random_state=1)
    svm.fit(previsores[indice_treinamento], classe[indice_treinamento])
    previsoes_svm = svm.predict(previsores[indice_teste])
    precisao_svm = accuracy_score(classe[indice_teste], previsoes_svm)
    matriz_svm = confusion_matrix(classe[indice_teste], previsoes_svm)
    resultados_svm.append(precisao_svm)

    # Classificador PMC
    pmc = Sequential()
    pmc.add(Dense(units=22, activation='relu', input_dim=22))
    pmc.add(Dense(units=12, activation='relu'))
    pmc.add(Dense(units=1, activation='sigmoid'))
    pmc.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    pmc.fit(previsores[indice_treinamento], classe[indice_treinamento], batch_size=10, epochs=100, verbose=0)
    previsoes_pmc = pmc.predict(previsores[indice_teste])
    previsoes_pmc = (previsoes_pmc > 0.5)
    precisao_pmc = accuracy_score(classe[indice_teste], previsoes_pmc)
    matriz_pmc = confusion_matrix(classe[indice_teste], previsoes_pmc)
    resultados_pmc.append(precisao_pmc)
```

Fonte: Elaboração própria

4.4 RESULTADOS

A acurácia é a proximidade de um resultado com o seu valor de referência real. Dessa forma, quanto maior a acurácia, mais próximo da referência ou valor real é o

resultado encontrado. A acurácia se refere a quão próximos da realidade são os resultados encontrados de forma automatizada ou com soluções de IA.

Neste trabalho o MVS teve acurácia de 80%, enquanto a RNA teve de 76%, conforme apresentado na figura 12.

Figura 12 – Acurácia do Modelo.

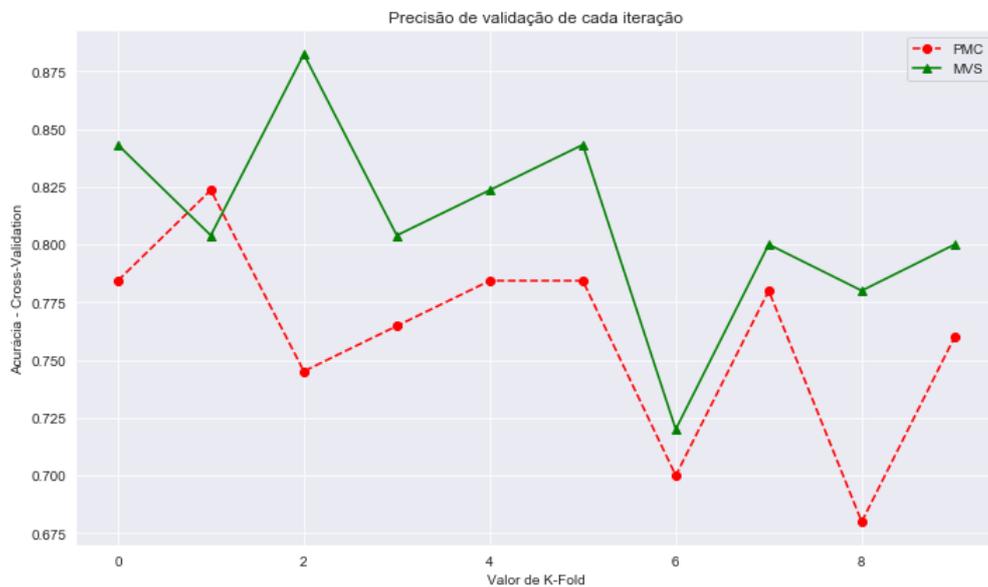
```
In [19]: # Apresenta a acurácia dos modelos
print('Acurácia do PMC =', precisao_pmc)
print('Acurácia do MVS =', precisao_svm)
```

Acurácia do PMC = 0.76
Acurácia do MVS = 0.8

Fonte: Elaboração própria

Já a figura 13 apresenta um gráfico com os pontos que representa a precisão de validação de cada *K-fold*, podendo ser observado que em algumas iterações a precisão foi diferente.

Figura 13 – Precisão de validação de cada *K-fold*



Fonte: Elaboração própria

De forma a complementar a figura 13. Cada *K-fold* teve como média de precisão vide a figura 14 e a média das precisões totais que corresponde a soma das médias de cada iteração dividido pelo total de iterações foi de 76% na RNA e no MVS foi de 80%.

Figura 14 – Média de precisão em cada *K-fold*

```

***** PMC *****
Precisão da validação de cada iteração:
[0.7843137254901961, 0.8235294117647058, 0.7450980392156863, 0.7647058823529411, 0.7843137254901961, 0.7843137254901961, 0.7,
0.78, 0.68, 0.76]

Média das precisões:
0.7606274509803921
***** MVS *****
Precisão da validação de cada iteração:
[0.8431372549019608, 0.803921568627451, 0.8823529411764706, 0.803921568627451, 0.8235294117647058, 0.8431372549019608, 0.72, 0.
8, 0.78, 0.8]

Média das precisões:
0.8099999999999999

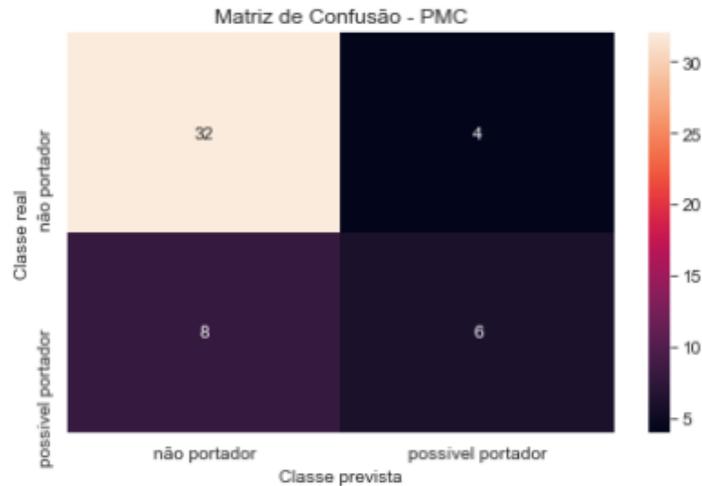
```

Fonte: Elaboração própria

A matriz de confusão é um tabela que mostra as frequências de classificação para cada classe do modelo (Figura 15) e (Figura 16).

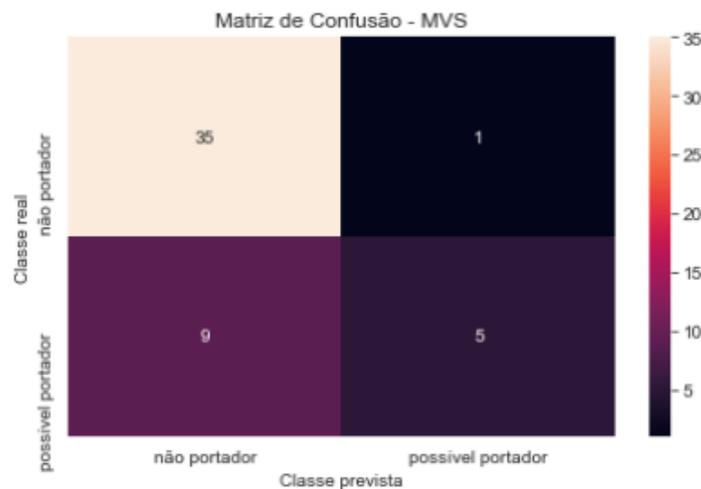
- Verdadeiro positivo: ocorre quando no conjunto real, a classe que estamos buscando foi prevista corretamente. Por exemplo, quando detetamos corretamente se o indivíduo é um possível portador. No modelo PMC foram classificados 32 de forma correta. Já no MVS foram classificados corretamente 35.
- Falso positivo: ocorre quando no conjunto real, a classe que estamos buscando prever foi prevista incorretamente, isso ocorre quando temos um não portador mas o modelo diz que ele é um possível portador, como falso positivo tivemos apenas 4 no modelo PMC e 1 no MVS.
- Falso verdadeiro: ocorre quando no conjunto real, a classe que não estamos buscando prever foi prevista corretamente, isso ocorre quando o modelo identifica um não portador de forma correta, como falso verdadeiro tivemos 8 no modelo PMC e 9 no MVS.
- Falso negativo: ocorre quando no conjunto real, a classe que não estamos buscando prever foi prevista incorretamente. Por exemplo, quando o indivíduo é um possível portador e o modelo previu incorretamente que ele não é, e nesse caso, tivemos 6 no modelo PMC e 5 no modelo MVS.

Figura 15 – Matriz de confusão - PMC



Fonte: Elaboração própria

Figura 16 – Matriz de confusão - MVS



Fonte: Elaboração própria

E a última técnica de avaliação utilizada foi o *classification_report* que verificou a qualidade da previsão do modelo. A Figura 17 apresenta o *classification_report*. O *classification_report* apresenta as seguintes métricas (The scikit-yb developers, 2016):

- *precision*: Precisão é a capacidade de um classificador de não rotular uma instância de positiva como realmente negativa. Para cada classe, é definida como a razão entre os verdadeiros positivos e a soma dos verdadeiros e falsos positivos.
- *recall*: Lembre-se é a capacidade de um classificador encontrar todas as instâncias positivas. Para cada classe, é definida como a razão entre os verdadeiros positivos e a soma dos verdadeiros positivos e falsos negativos.

- *f1 score*: A pontuação F1 é uma média harmônica ponderada de precisão e recordação, de modo que a melhor pontuação seja 1,0 e a pior seja 0,0. De um modo geral, as pontuações F1 são menores que as medidas de precisão, à medida que incorporam precisão e recuperação em seu cálculo. Como regra geral, a média ponderada de F1 deve ser usada para comparar modelos de classificadores.
- *support*: Suporte é o número de ocorrências reais da classe no conjunto de dados especificado. O suporte desequilibrado nos dados de treinamento pode indicar fragilidades estruturais nas pontuações relatadas do classificador e pode indicar a necessidade de amostragem estratificada ou reequilíbrio.

Figura 17 – Relatório de classificação

```

***** Relatório de classificação do PMC *****
      precision    recall  f1-score   support

     0         0.80      0.89      0.84         36
     1         0.60      0.43      0.50         14

 accuracy                   0.76         50
 macro avg         0.70      0.66      0.67         50
 weighted avg      0.74      0.76      0.75         50

***** Relatório de classificação do MVS *****
      precision    recall  f1-score   support

     0         0.80      0.97      0.88         36
     1         0.83      0.36      0.50         14

 accuracy                   0.80         50
 macro avg         0.81      0.66      0.69         50
 weighted avg      0.81      0.80      0.77         50

```

Fonte: Elaboração própria

5 CONCLUSÃO

O Hipertireoidismo, cujos principais sintomas são aumento de apetite, sem perda de peso recentemente, palpitações no coração, coração acelerado, protusão ocular e dentre outros. O hipertireoidismo é 5 a 10 vezes mais comum nas mulheres e com pico de incidência entre a 2^o e 4^o décadas de vida. Por esse motivo, há necessidade de utilizar técnicas computacionais com o intuito de auxiliar a detecção e encaminhamento imediato ao diagnóstico através de um exame sanguíneo para confirmação, diminuindo os impactos nas vidas social e garantindo um tratamento nas fases iniciais.

Utilizando-se especificamente algoritmos de MVS e PMC, foi possível indicar indivíduos como possíveis portadores do hipertireoidismo, auxiliando em um diagnóstico.

Como resultados, o percentual de acerto em classificar portadores e não portadores ficou entre 76% e 80%. Foi criada uma base de dados com 506 instâncias, que serviu de entrada para os modelos utilizados.

Os diferenciais do trabalho em relação aos trabalhos correlatos, foi um algoritmo agindo no pré exame. podendo ajudar um especialista em endocrinologia a indicar uma confirmação via exame de sangue.

Uma abordagem importante para trabalhos futuros, consiste na construção de uma base de dados maior, para um aperfeiçoamento maior do PMC e do MVS, a criação de uma interface mais amigável para facilitar a inserção das informações pelo usuário e o desenvolvimento para casos de hipotireoidismo.

REFERÊNCIAS

- ALMEIDA; DUMONTIER. Uso de redes neurais em avaliação de riscos de inadimplencia. *Revista de Administração FEA/USP*, 1996. Citado na página 14.
- BARR, A.; FEIGENBAUM, E. A. *The Handbook of Artificial Intelligence*. [S.l.]: Butterworth-Heinemann, 1981. ISBN 978-0-86576-089-9. Citado na página 13.
- BEGUM, A.; PARKAVI, A. *Prediction of thyroid Disease Using Data Mining Techniques*. 2019. Disponível em: <<https://ieeexplore.ieee.org/document/8728320>>. Acesso em: 27 abr. 2020. Citado na página 24.
- BOBRIAKOV, I. *PYTHON LIBRARIES FOR DATA SCIENCE*. 2018. Disponível em: <<http://datascienceacademy.com.br/blog/top-20-bibliotecas-python-para-data-science/>>. Acesso em: 01 out. 2019. Citado na página 27.
- BOSCARIOLI, L. A. da Silva e Sarajane Marques Peres e C. *Introdução à Mineração de Dados com Aplicações em R*. [S.l.], 2016. 17-18 p. Citado na página 14.
- BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. *A Training Algorithm for Optimal Margin Classifiers. In: ANNUAL WORKSHOP ON COMPUTACIONAL LEARNING*. [S.l.: s.n.], 1992. 144-152 p. Citado na página 21.
- BRENTA, G. Força tarefa em hipotireoidismo da sociedade latino-americana de tiroide (lats). 2013. Citado na página 9.
- CHARNIAK, E.; MCDERMOTT, D. *Introduction to artificial intelligence*. [S.l.: s.n.], 1985. Citado na página 13.
- Data Science Academy. *Deep Learning Book*. 2019. Disponível em: <<http://www.deeplearningbook.com.br/>>. Acesso em: 09 jun. 2020. Citado na página 19.
- Departamento de Tireoide. *O que É a Tireoide?* 2015. Disponível em: <<http://www.tireoide.org.br/o-que-e-a-tireoide/>>. Acesso em: 01 jun. 2020. Citado na página 9.
- FAYYAD, G. P.-S. U.; SMYTH, P. From data mining to knowledge discovery in databases. 1996. Citado na página 14.
- FLAUZINO, I. da S. . D. H. S. . R. A. Redes neurais artificiais para engenharia e ciências aplicadas. 2010. Citado na página 20.
- FREITAS, A. L.; JUNIOR, O. V. S. *MACHINE LEARNING Desafios para um Brasil competitivo*. [S.l.]: Revista da Sociedade Brasileira de Computação, 2019. 16 p. Citado na página 16.
- HAN, J.; KAMBER, M. Data minning - concepts and techniques. 2001. Citado na página 15.
- HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. [S.l.: s.n.], 1999. 35 p. Citado na página 17.

- HAYKIN, S. *Redes Neurais - Princípios de Práticas (2ª ed.)*. [S.l.]: Editora Bookman, 2001. Citado na página 23.
- KOERICH, A. L. *Redes Neurais Artificiais*. 2014. Disponível em: <<http://www.ppgia.pucpr.br/~alekoe/AM/2014/7-RNA-AM-2014.pdf>>. Acesso em: 09 jun. 2020. Citado na página 19.
- KOTSIANTIS, S. B. *Machine Learning*. [S.l.]: McGraw-Hill Science/Engineering/Math. [S.l.: s.n.], 2007. 3-24 p. Citado 3 vezes nas páginas 16, 17 e 18.
- MANNILA, H. *Data Mining: Machine Learning, Statistics and databases*. In *Proceedings of the eight IEEE International Conference*, p. 2–9, 1996. Citado na página 15.
- MEDEIROS, L. F. de. *Inteligência Artificial Aplicada: Uma abordagem introdutória*. Curitiba, 2018. 17-18 p. Citado na página 13.
- MINSKY, M.; PAPERT, S. A. Perceptrons an introduction to computational geometry. 1969. Citado na página 18.
- MITCHELL, T. *Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering*. [S.l.: s.n.], 1997. 432 p. ISBN ISBN 0070428077. Citado 2 vezes nas páginas 18 e 20.
- NIGRIN, A. *Neural Networks For Pattern Recognition*. [S.l.: s.n.], 1993. 26 p. Citado na página 17.
- OZYILMAZ, L.; YILDIRIM, T. *Diagnosis of thyroid disease using artificial neural network methods*. 2002. Disponível em: <<https://ieeexplore.ieee.org/document/1199031>>. Acesso em: 25 mar. 2020. Citado na página 24.
- POSSA. *Data Mining técnicas para exploração de dados*. Universidade Federal de Minas Gerais, 1998. Citado na página 15.
- RICH, E.; KNIGHT, K. *Artificial Intelligence*. [S.l.]: Makron Books, 1993. ISBN 9788534601221. Citado na página 14.
- RUMELHART, E. et al. *Learning representations by back-propagating errors*. *Nature*. [S.l.: s.n.], 1986. 323, 533–536 p. Citado na página 21.
- RUSSELL, S.; NORVIG, P. *Artificial Intelligence – A Modern Approach*. [S.l.]: Prentice Hall, 2 edition, 2003. Citado na página 16.
- SAITI, F. et al. *Thyroid Disease Diagnosis Based on Genetic Algorithms Using PNN and SVM*. 2009. Disponível em: <<https://ieeexplore.ieee.org/document/5163689>>. Acesso em: 15 abr. 2020. Citado na página 24.
- SALES, P. *O essencial em Endocrinologia*. Rio de Janeiro: EDITORA GUANABARA KOOGAN LTDA., 2016. 830 p. Citado na página 11.
- SANTOS, C. R. *Análise Automática de Assinaturas Manuscritas Baseada nos Princípios da Grafoscopia*. *Dissertação de Mestrado, Pontifícia Universidade Católica do Paraná*. [S.l.: s.n.], 2004. Citado na página 23.

SHAHID, A. H. et al. *A Study on Label TSH, T3, T4U, TT4, FTI in Hyperthyroidism and Hypothyroidism using Machine Learning Techniques*. 2019. Disponível em: <<https://ieeexplore.ieee.org/document/9002284>>. Acesso em: 25 abr. 2020. Citado na página 24.

SILVA, I. N. da; SPATTI, D. H.; FLAUZINO, R. A. *Redes Neurais Artificiais para engenharia e ciências aplicadas*. [S.l.]: Artliber Editora, 2016. 35 p. (2º Edição). Citado na página 19.

SMOLA A. J., B. P. S. B.; SCHUURMANS, D. *Introduction to Large Margin Classifiers*. [S.l.: s.n.], 1999. 1–28 p. Citado na página 22.

SOARES, H. B. Análise e classificação de imagens de lesões da pele por atributos de cor, forma e textura utilizando máquina de vetor de suporte. Universidade do Rio Grande do Norte, 2008. Disponível em: <<https://repositorio.ufrn.br/jspui/handle/123456789/15118>>. Acesso em: 30 set. 2020. Citado 4 vezes nas páginas 6, 22, 23 e 24.

The scikit-yb developers. *Classification Report*. 2016. Disponível em: <https://www.scikit-yb.org/en/latest/api/classifier/classification_report.html>. Acesso em: 07 jun. 2020. Citado na página 32.

THURASINGHAM, B. *Data minning*. 1999. Citado na página 15.

VAPNIK, V. N. *The Nature of Statistical Learning Theory*. [S.l.: s.n.], 1995. Citado na página 21.

VILLAR, L. *Endocrinologia Clínica*. Rio de Janeiro: EDITORA GUANABARA KOOGAN LTDA., 2017. 252 p. (Quinta edição). Citado 3 vezes nas páginas 11, 12 e 13.

ZUBEN, P. L. N. de C. . F. J. V. *Redes neurais artificiais*. 2013. Citado na página 17.

APÊNDICE A – IMPLEMENTAÇÃO

In []:

```
# Importação das bibliotecas
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import keras
import warnings

from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from keras.models import Sequential
from keras.layers import Dense
from sklearn.model_selection import learning_curve
from sklearn.svm import SVC

%matplotlib inline
sns.set_style('darkgrid')

# Ocultar Warnings
warnings.filterwarnings('ignore')
```

In []:

```
# Carrega o dataset
base1 = pd.read_excel('Dataset.xlsx')
# Verifica o tamanho da base
len(base1)
```

In []:

```
# Verifica o cabeçalho do DataFrame
base1.head()
```

In []:

```
# Define as colunas dos atributos previsores
previsores = base1.iloc[:, 0:22].values
# Define a coluna da classe
classe = base1.iloc[:, 22].values
```

In []:

```
previsores[1,0:22]
```

In []:

```
classe
```

In []:

```
# Instancia o StandardScaler
scaler = StandardScaler()
# Deixa os atributos(previsores) na mesma escala - escalonar
previsores = scaler.fit_transform(previsores)
```

In []:

```
previsores
```

In []:

```
# Validação cruzada
kfold = StratifiedKFold(n_splits = 10, shuffle=True, random_state=4)
resultados_pmc = []
resultados_svm = []
for indice_treinamento, indice_teste in kfold.split(previsores,np.zeros(shape=(previsores.shape[0], 1))):

    # Classificador SVM
    svm = SVC(kernel='linear', random_state=1)
    svm.fit(previsores[indice_treinamento], classe[indice_treinamento])
    previsoes_svm = svm.predict(previsores[indice_teste])
    precisao_svm = accuracy_score(classe[indice_teste], previsoes_svm)
    matriz_svm = confusion_matrix(classe[indice_teste], previsoes_svm)
    resultados_svm.append(precisao_svm)

    # Classificador PMC
    pmc = Sequential()
    pmc.add(Dense(units=22, activation='relu', input_dim=22))
    pmc.add(Dense(units=12, activation='relu'))
    pmc.add(Dense(units=1, activation='sigmoid'))
    pmc.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    pmc.fit(previsores[indice_treinamento], classe[indice_treinamento], batch_size=10,
    epochs=100, verbose=0)
    previsoes_pmc = pmc.predict(previsores[indice_teste])
    previsoes_pmc = (previsoes_pmc > 0.5)
    precisao_pmc = accuracy_score(classe[indice_teste], previsoes_pmc)
    matriz_pmc = confusion_matrix(classe[indice_teste], previsoes_pmc)
    resultados_pmc.append(precisao_pmc)
```

In []:

```
# Apresenta a acurácia dos modelos
print('Acurácia do PMC =', precisao_pmc)
print('Acurácia do MVS =', precisao_svm)
```

In []:

```
# Apresenta a classe real das instâncias
classe_teste = classe[indice_teste]
print(classe_teste)
```

In []:

```
print(previsoes_svm)
```

In []:

```
# Apresenta as previsões do MLP
print(previsoes_pmc)
```

In []:

```
# Apresenta as precisões das iterações do K-Fold Cross-Validation e a média das precisões em relação aos modelos
print('***** PMC *****')
print('Precisão da validação de cada iteração:')
print(resultados_pmc)
print('\n')
print('Média das precisões:')
print(np.asarray(resultados_pmc).mean())

# Apresenta as precisões das iterações do K-Fold Cross-Validation e a média das precisões em relação aos modelos
print('***** MVS *****')
print('Precisão da validação de cada iteração:')
print(resultados_svm)
print('\n')
print('Média das precisões:')
print(np.asarray(resultados_svm).mean())
```

In []:

```
# Visualização da precisão de cada iteração do K-Fold Cross-Validation em relação aos m
odelos
plt.figure(figsize=(10, 6))
plt.plot(resultados_pmc, 'o--', color='red', label='PMC')
plt.plot(resultados_svm, '^-', color='green', label='MVS')
plt.legend()
plt.title('Precisão de validação de cada iteração')
plt.xlabel('Valor de K-Fold')
plt.ylabel('Acurácia - Cross-Validation')
plt.tight_layout()
```

In []:

```
# Apresenta a matriz de confusão do PMC
matrizPMC = plt.subplot()
sns.heatmap(matriz_pmc, annot=True, ax = matrizPMC)

matrizPMC.set_xlabel('Classe prevista')
matrizPMC.set_ylabel('Classe real')
matrizPMC.set_title('Matriz de Confusão - PMC')
matrizPMC.xaxis.set_ticklabels(['não portador', 'possível portador'])
matrizPMC.yaxis.set_ticklabels(['não portador', 'possível portador'])
plt.tight_layout()
```

In []:

```
# Apresenta a matriz de confusão do MVS
matrizMVS = plt.subplot()
sns.heatmap(matriz_svm, annot=True, ax = matrizMVS)

matrizMVS.set_xlabel('Classe prevista')
matrizMVS.set_ylabel('Classe real')
matrizMVS.set_title('Matriz de Confusão - MVS')
matrizMVS.xaxis.set_ticklabels(['não portador', 'possível portador'])
matrizMVS.yaxis.set_ticklabels(['não portador', 'possível portador'])
plt.tight_layout()
```

In []:

```
#Apresenta o relatório de classificação dos modelos
print('\t***** Relatório de classificação do PMC *****')
print(classification_report(classe[indice_teste], previsoes_pmc))

#Apresenta o relatório de classificação dos modelos
print('\t***** Relatório de classificação do MVS *****')
print(classification_report(classe[indice_teste], previsoes_svm))
```

In []:

```
#Novo paciente  
user = np.array([1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1])
```

In []:

```
#Predição do novo paciente  
  
predSVM = svm.predict([user])  
predPMC = pmc.predict([user.reshape(1,-1)])  
predPMC = (predPMC>0.5)  
print(predSVM)  
print(predPMC)
```

ANEXO A – QUESTIONÁRIO

1) Você tem apresentado intolerância ao calor?	SIM	NÃO
2) Qual seu Sexo?	SIM	NÃO
3) Sente dores no peito?	SIM	NÃO
4) Você tem suado mais que o habitual?	SIM	NÃO
5) Seu apetite aumentou?	SIM	NÃO
6) Falta de ar, dificuldade em respirar?	SIM	NÃO
7) Diminuição da capacidade de concentração?	SIM	NÃO
8) Apesar do aumento do apetite, você perdeu peso recentemente?	SIM	NÃO
9) Sente irritação ocular?	SIM	NÃO
10) Dor nos olhos ao olhar para a luz brilhante?	SIM	NÃO
11) Massa do pescoço aumentada?	SIM	NÃO
12) Tem sentido palpitações ou o coração acelerado?	SIM	NÃO
13) Sente náusea constantemente?	SIM	NÃO
14) Tem apresentado fadiga ou fraqueza muscular (dificuldade para subir escadas ou levantar da poltrona/cadeira)?	SIM	NÃO
15) Tem apresentado tremores das mãos?	SIM	NÃO
16) Tem vomitado constantemente?	SIM	NÃO
17) Sente-se mais irritado?	SIM	NÃO
18) Tem tido dificuldade para dormir?	SIM	NÃO
19) Aumento da sede?	SIM	NÃO
20) Aumento da micção?	SIM	NÃO
21) Notou protusão dos olhos ("olhos para fora") ou aparência de "olhar assustado"?	SIM	NÃO
22) Houve alguma alteração do hábito intestinal, como aumento da frequência de evacuações ou fezes de consistência mais amolecida?	SIM	NÃO