

Henrique Levandoski Richa

Renan Belem Biavati

Vitória Izabel Mendes Pinto

INVESTIGAÇÃO DE ANOMALIAS CARDÍACAS EM DATASETS DE ELETROCARDIOGRAMAS - PROPOSTA DE PESQUISA CIENTÍFICA -

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Cientista da Computação.

Área de concentração: Ciência da Computação

Orientador: Andrey Cabral Meira

Curitiba

2024

Henrique Levandoski Richa

Renan Belem Biavati

Vitória Izabel Mendes Pinto

INVESTIGAÇÃO DE ANOMALIAS CARDÍACAS EM DATASETS DE ELETROCARDIOGRAMAS - PROPOSTA DE PESQUISA CIENTÍFICA -

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Cientista da Computação.

Área de concentração: Ciência da Computação

Orientador: Andrey Cabral Meira

Curitiba

2024

RESUMO

A medicina cardiológica, assim como todos os campos científicos de utilidade e necessidade na vida social, tem evoluído constantemente, buscando métodos mais eficazes para o diagnóstico de doenças cardíacas. O eletrocardiograma (ECG) é uma ferramenta diagnóstica essencial que fornece uma representação gráfica da atividade elétrica do coração, permitindo a identificação de uma variedade de condições cardíacas. A interpretação dos padrões de ECG é crucial para a detecção precoce de eventos cardíacos adversos, como infartos do miocárdio e arritmias. No entanto, a complexidade e a sutileza dos dados de ECG exigem uma análise que fez levantar a hipótese, se existe a necessidade de suporte computacional para melhorar a eficácia de análise humana sobre os dados. Neste sentido, este estudo se propõe a realizar este processo, fazendo o pré-processamento dos dados de ECG, e depois, fazendo a aplicação e avaliação de técnicas como Máquina de Vetores de Suporte (SVM), Perceptron Multicamadas (MLP), entre outras técnicas de aprendizado de máquina supervisionado com experimentação dos hiperparâmetros, realizando assim a supervisão humana e ajuste dos parâmetros que são relativos às especificidades do aprendizado da máquina sobre o conjunto de dados. O conjunto de dados utilizados é o MIT-BIH Arrhythmia Dataset, que é público e proveniente dos laboratórios no Beth Israel Hospital de Boston (hoje Beth Israel Deaconess Medical Center) no Instituto de Tecnologia de Massachusetts (MIT). A proposta deste trabalho é investigar a eficácia dessas técnicas computacionais para um diagnóstico de anomalias em ECGs. O estudo adota uma metodologia estruturada em sprints seguindo a metodologia Scrum, e seu objetivo é contribuir para o avanço no entendimento do uso de técnicas de aprendizado de máquina na identificação de patologias cardíacas, e com isso, fornecer recomendações para a implementação prática dessas técnicas no ambiente clínico e oferecendo insights valiosos para profissionais de saúde e pesquisadores na área.

Palavras-chaves: eletrocardiograma; aprendizado; .

ABSTRACT

Cardiological medicine, like all scientific fields of utility and necessity in social life, has constantly evolved, seeking more effective methods for diagnosing heart diseases. The electrocardiogram (ECG) is an essential diagnostic tool that provides a graphic representation of the electrical activity of the heart, allowing the identification of a variety of heart conditions. Interpretation of ECG patterns is crucial for early detection of adverse cardiac events such as myocardial infarctions and arrhythmias. However, the complexity and subtlety of ECG data require analysis that has led to the hypothesis whether there is a need for computational support to enhance the effectiveness of human analysis on the data. In this sense, this study proposes to carry out this process, pre-processing the ECG data, and then applying and evaluating techniques such as Support Vector Machine (SVM), Multilayer Perceptron (MLP), among others. supervised machine learning techniques with hyperparameter experimentation, thus performing human supervision and adjustment of parameters that are relative to the specifics of machine learning on the data set. The dataset used is the MIT-BIH Arrhythmia Dataset, which is public and comes from laboratories at Boston's Beth Israel Hospital (now Beth Israel Deaconess Medical Center) at the Massachusetts Institute of Technology (MIT). The purpose of this work is to investigate the effectiveness of these computational techniques for diagnosing anomalies in ECGs. The study adopts a methodology structured in sprints following the Scrum methodology, and its objective is to contribute to advancing the understanding of the use of machine learning techniques in identifying cardiac pathologies, and with this, providing recommendations for implementation practice of these techniques in the clinical environment and offering valuable insights for healthcare professionals and researchers in the area.

Keywords: Electrocardiogram; Machine Learning.

SUMÁRIO

RESUMO	3
ABSTRACT	4
LISTA DE FIGURAS	7
LISTA DE TABELAS	8
LISTA DE QUADROS	9
LISTA DE ABREVIATURAS E SIGLAS	10
CAPÍTULO 1 - INTRODUÇÃO	11
1.1 SOBRE O CONTEÚDO DA INTRODUÇÃO	11
1.2 OBJETIVOS	12
1.3 ESTRUTURA DO DOCUMENTO	12
CAPÍTULO 2 - REVISÃO DA LITERATURA	14
2.1 Introdução ao Eletrocardiograma	14
2.2 APRENDIZADO DE MÁQUINA	17
2.2.1 APRENDIZADO SUPERVISIONADO	20
2.2.2 APRENDIZADO NÃO SUPERVISIONADO	24
2.2.3 APRENDIZADO POR REFORÇO	25
2.3 OTIMIZAÇÃO DE HIPERPARÂMETROS	26
2.4 PRÉ-PROCESSAMENTO	26
2.5 APRENDIZADO DE MÁQUINA NA SAÚDE	28
CAPÍTULO 3 - MÉTODO DE PESQUISA	31
3.1 CARACTERIZAÇÃO DA PESQUISA	31
3.2 MÉTODO DE PESQUISA	31
3.2.1 SELEÇÃO DE DADOS	31
3.2.1.1 MIT-BIH Arrhythmia Database	31
3.2.1.2 PTB DIAGNOSTIC ECG DATABASE	32
3.2.2 PRÉ-PROCESSAMENTO E EXECUÇÃO	33

3.2.2.1 Exclusão do PTB Diagnostic ECG Database

33

CAPÍTULO 4 - VALIDAÇÃO DA PESQUISA CIENTÍFICA**34**

4.1	AMBIENTE	34
4.1.1	HARDWARE	34
4.1.1	SOFTWARE	35
4.2	CONDIÇÕES PARA INICIAR A VALIDAÇÃO	36
4.2.1	Instalação de SOFTWARES	37
4.2.2	Configurações do Ambiente de Desenvolvimento	37
4.2.3	PREPARAÇÃO DOS DADOS	37
4.3	PROCEDIMENTOS PARA REALIZAR A VALIDAÇÃO	38
4.4	ANÁLISE DOS RESULTADOS	39
4.4.1	Análise do Modelo SVC	39
4.4.2	Análise do Modelo KNN	42
4.4.3	Análise do Modelo Random Forest	45
4.5	DISCUSSÃO DE RESULTADOS	50
4.5.1	Resultados Gerais dos Modelos	50
4.6	CONSIDERAÇÕES FINAIS	52

CAPÍTULO 5 - INFORMAÇÕES DA PESQUISA**53**

5.1	CRONOGRAMA	53
5.2	RISCOS	54

REFERÊNCIAS BIBLIOGRÁFICAS**57**

LISTA DE FIGURAS

Figura 1 - Morfologia de um ECG saudável.....	15
Figura 2 - Exemplos de Eletrocardiogramas.....	16
Figura 3 - Classificador KNN.....	21
Figura 4 - Árvore Binária.....	22
Figura 5 - SVM.....	22
Figura 6 - MLP.....	24
Figura 7 - K-Means Clustering.....	25
Figura 8 - Representação do Aprendizado por Reforço.....	26
Figura 9 - Conjunto de Treino MIT-BIH Arrhythmia Database.....	27
Figura 10 - Conjunto de Treino PTB Diagnostic ECG Database.....	28
Figura 11 - Matriz de Confusão do SVM.....	42
Figura 12 - Aprendizado no Número de Vizinhos.....	43
Figura 13 - Matriz de Confusão para KNN.....	45
Figura 14 - Matriz de Confusão do Modelo Padrão para RandomForest.....	47
Figura 15 - Gráfico de Desempenho para RandomForest.....	48
Figura 16 - Gráfico de Comparação de Pesos para RandomForest.....	49
Figura 17 - Matriz de Confusão do Modelo Otimizado para RandomForest.....	50

LISTA DE TABELAS

Tabela 1 - Softwares.....	36
Tabela 2 - Bibliotecas Python.....	36
Tabela 3 - Média Ponderada com Diferentes Kernel.....	39/40
Tabela 4 - Média Ponderada com Diferentes Gamma.....	40
Tabela 5 - Resultados Padrão para SVC.....	41
Tabela 6 - Resultados Otimizados para SVC.....	41
Tabela 7 - Parâmetros Padrões para KNN.....	42/43
Tabela 8 - Resultados Padrões para KNN.....	44
Tabela 9 - Resultados Otimizados para KNN.....	44
Tabela 10 - Parâmetros Padrões para RandomForest.....	45/46
Tabela 11 - Resultados Padrões para RandomForest.....	46
Tabela 12 - Resultados Otimizados para Random Forest.....	49
Tabela 13 - Resultados Padrões com Média Ponderada.....	51
Tabela 14 - Resultados Otimizados com Média Ponderada.....	52
Tabela 15 - Planejamento do Trello.....	53/54
Tabela 16 - Riscos do Projeto.....	55

LISTA DE QUADROS

Quadro 1 - Quadro descritivo das classes de dados.....	32
--	----

LISTA DE ABREVIATURAS E SIGLAS

ECG	Eletrocardiograma
SVM	Support Vector Machine
MLP	Multilayer Perceptron
ML	Machine Learning
SVC	Support Vector Classification
SVM	Support Vector Machine
KNN	K Nearest Neighbors

CAPÍTULO 1 - INTRODUÇÃO

1.1 Sobre o conteúdo da Introdução

O eletrocardiograma é uma ferramenta essencial na medicina, fornecendo *insights* valiosos sobre a saúde cardíaca de um paciente. Tradicionalmente, a interpretação do ECG, depende da experiência e do julgamento clínico do médico. No entanto, com o volume crescente de dados e a complexidade das condições cardíacas, torna-se interessante buscar abordagens mais quantitativas, objetivas e precisas. É então, o porque a tecnologia de aprendizado de máquina é aplicável, oferecendo novas perspectivas e desafios (SCHWARZ, 2009), (WASIMUDDIN et al. 2020).

Tendo em vista que segundo Breen, Kelly, Kernohan (2022), o ECG se tornou o procedimento laboratorial cardiovascular mais comumente usado para pacientes apresentando queixas cardíacas, além disso, também foi relatado que os cardiologistas identificam corretamente entre 53% e 96% das anormalidades em um ECG, e se um erro na interpretação pode acarretar a morte do paciente, a taxa ideal seria 100% em todos os casos.

Portanto, este trabalho visa explorar a eficiência e eficácia de métodos de aprendizado de máquina na busca por anomalias em ECG, através de revisões de literatura sobre o procedimento e os dados extraídos de ECG, e os métodos de aprendizado de máquina, identificando os métodos mais eficazes e as melhores práticas na intersecção da medicina e da tecnologia. Além disso, este trabalho aspira contribuir para o avanço dos campos e fornecer uma base sólida para pesquisas futuras.

1.2 Objetivos

O objetivo geral do trabalho é aplicar e avaliar técnicas de aprendizado de máquina sobre dados de ECG, para maior eficácia nos diagnósticos de possíveis doenças cardíacas.

Para alcançar este objetivo, o estudo tem os seguintes objetivos específicos:

- a) Realizar uma revisão da literatura para entender o campo da análise de ECG usando aprendizado de máquina;
- b) Utilizar o pré-processamento de dados para configurar e refinar a aplicabilidade das técnicas de aprendizado de máquina.
- c) Comparar e contrastar diferentes técnicas de aprendizado de máquina em termos de precisão, eficácia e aplicabilidade na detecção de anomalias em ECGs, investigando possíveis melhorias nos resultados obtidos com as técnicas de forma a garantir a correta análise das anomalias;
- d) A partir das técnicas de aprendizagem de máquina empregadas, identificar melhoria ou piora nos índices de diagnóstico de doenças cardíacas;
- e) Com os resultados obtidos, propor melhorias no processo de análise ECG e objetivamente sinalizar onde as técnicas são úteis pro campo científico da saúde cardíaca.

1.3 Estrutura do documento

- O Capítulo 1 apresenta a parte introdutória da pesquisa, explicita o contexto, os campos de estudo e exploração, e também, fornece os elementos e objetivos respectivos.
- O Capítulo 2 são desenvolvidos conceitos que determinam o contexto da pesquisa, explica a morfologia gráfica do eletrocardiograma e as técnicas de aprendizado de máquina.
- O Capítulo 3 apresenta como será a coleta de dados da pesquisa, seus métodos, etapas necessárias e análises dos bancos de dados cogitados.

- O Capítulo 4 apresenta o desenvolvimento dos métodos de pesquisa para garantia de resultados, registra a aplicabilidade e ações utilizadas para validar a pesquisa e compara resultados dos métodos para análise.
- O Capítulo 5 apresenta o cronograma e os riscos da pesquisa.

CAPÍTULO 2 - REVISÃO DA LITERATURA

No campo da medicina, especificamente na análise de eletrocardiogramas (ECG), a evolução tecnológica e o advento do aprendizado de máquina trouxeram novas perspectivas e desafios. Esta seção busca traçar um panorama do que já foi explorado, identificando as contribuições significativas e as lacunas ainda presentes na literatura.

2.1 Introdução ao Eletrocardiograma

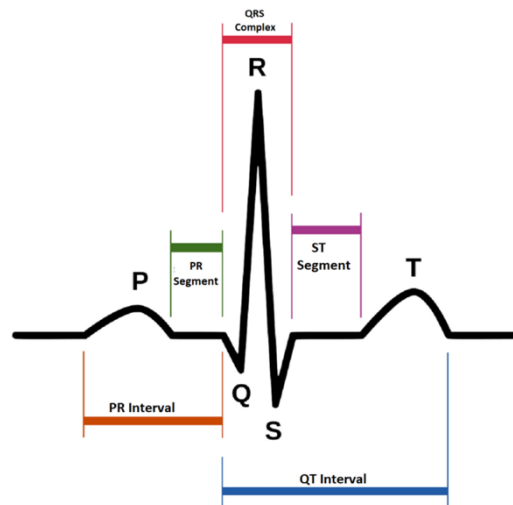
O eletrocardiograma é uma representação gráfica da atividade elétrica do coração. Desenvolvido no início do século XX, e melhorado desde então. O ECG rapidamente se tornou uma ferramenta indispensável na medicina, permitindo aos médicos identificar e diagnosticar uma variedade de condições cardíacas. Ao longo dos anos, sua importância na detecção precoce de doenças cardíacas, como infarto no miocárdio e arritmias, tem sido inestimável. Os achados em um ECG, como os citados, fornecem pistas valiosas sobre a saúde do coração de um paciente. O exame é realizado por meio de eletrodos fixados na pele, que captam a atividade elétrica do coração. Esta atividade é caracterizada pela variação na quantidade de íons de sódio dentro e fora das células musculares cardíacas. O resultado é um gráfico, com ondas e complexos específicos (como as ondas P¹, QRS² e T³), que representam diferentes fases da atividade cardíaca e podem ser comparados com padrões normais para avaliar a saúde cardíaca do paciente. Abaixo está situado um exemplo de um eletrocardiograma (JAMBUKIA, DABHI, PRAJAPATI, 2015).

¹ Ativação dos átrios

² Ativação dos ventrículos

³ Movimento de repolarização ventricular

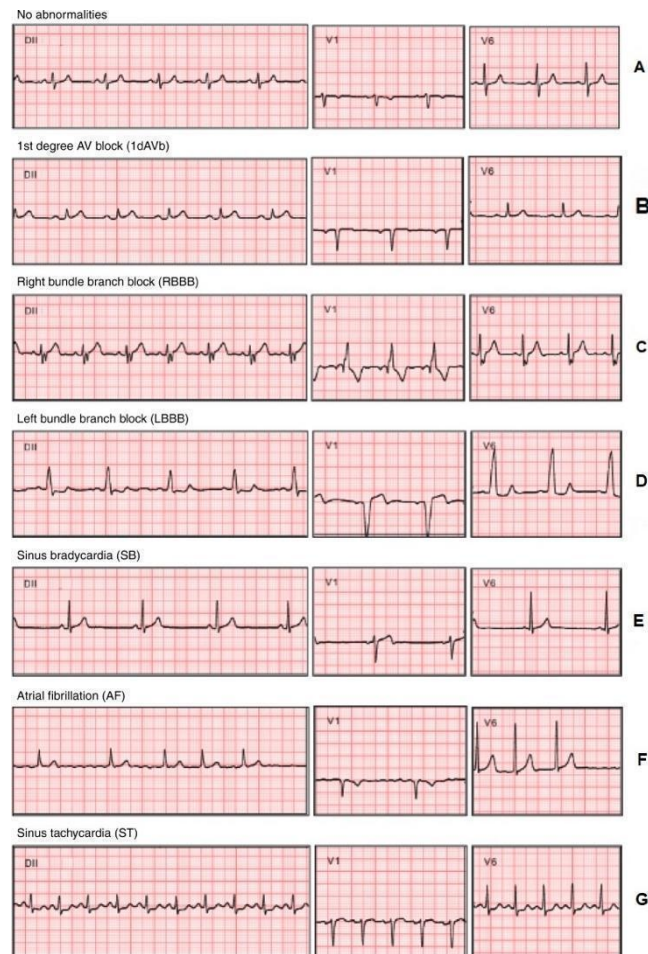
Figura 1 - Morfologia de um ECG saudável



Fonte: AZIZ, AHMED, ALOUINI (2021)

Por meio dos resultados obtidos em um eletrocardiograma, é possível identificar uma variedade de condições cardíacas. Algumas das patologias que podem ser detectadas incluem o bloqueio atrioventricular de 1º grau, bloqueio de ramo direito, bloqueio de ramo esquerdo, bradicardia sinusal, fibrilação atrial e taquicardia sinusal.

A Figura 2 apresenta uma compilação de eletrocardiogramas, mostrando uma variedade de padrões cardíacos, variando de ritmos normais a várias anormalidades. Cada painel, rotulado de A à G, representa um tipo específico de leitura de ECG. A leitura desses padrões é crucial para o diagnóstico clínico de condições cardíacas e demonstram a complexidade de algumas das variações que podem ser detectadas e analisadas usando técnicas de aprendizado de máquina, como investigado no contexto da pesquisa.

Figura 2 - Exemplos de Eletrocardiogramas

Fonte: RIBEIRO et al. (2020)

Onde:

- A. Sem anormalidades, representa um eletrocardiograma saudável.
- B. Bloqueio de atrioventricular de 1º grau é um atraso na condução dos impulsos elétricos entre as câmaras superiores (átrios) e as câmaras inferiores (ventrículos) do coração (PAIXÃO, 2022).
- C. Bloqueio de Ramo Direito é um bloqueio no ramo direito do sistema de condução elétrica do coração (IKEDA, 2021).
- D. Bloqueio de Ramo Esquerdo ocorre quando a condução elétrica normal do ventrículo esquerdo é interrompida ou atrasada, fazendo com que o ventrículo direito seja ativado antes do ventrículo esquerdo. Isso resulta em mudanças na mecânica, perfusão e carga de trabalho do ventrículo esquerdo (PÉREZ-RIERA, 2018).

- E. Bradicardia sinusal refere-se a uma frequência cardíaca mais lenta do que o normal, originada no nó sinusal, que é o marcapasso natural do coração. Especificamente, a bradicardia sinusal é definida por uma frequência cardíaca inferior a 60 batimentos por minuto (bpm) em adultos, quando medida em repouso.
- F. Fibrilação atrial é a arritmia supraventricular caracterizada por atividade elétrica atrial desorganizada, secundária a múltiplos focos de despolarização atrial (NETO, MOREIRA, MIRANDA, 2016).
- G. Taquicardia sinusal é um ritmo cardíaco regular no qual o coração bate mais rápido que o normal e resulta em um aumento no débito cardíaco (HENNING, KRAWIEC, 2023).

2.2 Aprendizado de Máquina

O termo “aprendizado de máquina” foi criado em 1959 por Arthur Samuel. A inteligência artificial (IA) e o aprendizado de máquina podem ser confundidos, e realmente podem ser parecidos na instância de que ambos realizam tarefas que necessitam inteligência humana para serem cumpridos, porém há diferenças entre os dois métodos. A IA refere-se ao conceito geral de máquinas que tomam decisões automaticamente, já o aprendizado de máquina (ML) é a capacidade de computadores aprenderem sem serem explicitamente programados. A essência do ML é usar algoritmos que analisam dados de entrada para prever valores de saída, identificar padrões e aprender com experiências anteriores (Handelman, et al. 2018).

Segundo Netto e Neto (2021), o aprendizado de máquina consiste em um compilado de métodos e algoritmos de detecção de padrão, classificação ou direções de dados a fim da obtenção de insights valiosos a partir de dados de maneira rápida e completamente autônoma. Em essência, são estratégias matemáticas destinadas a instruir algoritmos a reconhecer e interpretar padrões específicos.

No entanto, o processo de modelagem em ML não está isento de desafios, sendo dois dos principais o underfitting e o overfitting. Underfitting ocorre quando um modelo é demasiadamente simplista, falhando em captar a complexidade dos dados e, conseqüentemente, apresentando um desempenho insatisfatório tanto em dados

de treino quanto de teste. *Overfitting*, por outro lado, acontece quando um modelo é excessivamente complexo, capturando ruídos e detalhes específicos do conjunto de treino ao ponto de prejudicar sua capacidade de generalização para novos dados. Para mitigar o overfitting, técnicas de regularização como Lasso e Ridge são empregadas, ajustando os coeficientes do modelo para promover uma melhor generalização e robustez (KOLLURI, et al. 2020).

Logo, é necessária uma análise profunda dos resultados para identificar se os modelos estão de fato capturando os padrões subjacentes nos dados ou se estão simplesmente memorizando as respostas. As métricas de avaliação desempenham um papel muito importante nesta análise, fornecendo uma visão quantitativa da performance do modelo. Cada métrica oferece uma perspectiva única e, quando usadas em conjunto, podem oferecer uma visão abrangente da eficácia do modelo.

- Matriz de Confusão: Uma tabela que aponta os verdadeiros positivos e negativos, e os falsos positivos e negativos.
- Precisão: Mede a proporção de previsões positivas corretas em relação ao total de previsões positivas feitas pelo modelo.

$$Precisão = \frac{VP}{VP+FP} \quad (1)$$

- Acurácia: Proporção de previsões corretas (positivas e negativas) em relação ao total de previsões.

$$Acurácia = \frac{VP+VN}{VP+FP+FN+VN} \quad (2)$$

- F1-Score: Combina precisão e recall em uma única métrica que busca um equilíbrio entre as duas, útil quando as classes são desbalanceadas.

$$F1 - Score = 2 \times \frac{Precisão \times Recall}{Precisão + Recall} \quad (3)$$

- Recall: Avalia a proporção de casos positivos reais que foram corretamente identificados pelo modelo.

$$Recall = \frac{VP}{VP+FN} \quad (4)$$

- Média Macro e Ponderada (Weighted AVG): Calcula a média das métricas para cada classe, ponderada pelo tamanho da classe.

$$Weighted\ AVG = \sum_{C \in \mathcal{C}} (Peso\ da\ Classe \times Métrica\ para\ Classe) \quad (5)$$

- Especificidade: Avalia a capacidade do modelo de detectar resultados negativos.

$$Especificidade = \frac{VN}{VN+FP} \quad (6)$$

Com o uso dessas métricas, é possível não apenas ajustar os modelos para melhorar sua performance, mas também para entender as compensações entre diferentes tipos de erros. Por exemplo, em um contexto médico, pode-se preferir um modelo com um recall mais alto para garantir que todos os casos positivos sejam identificados, mesmo à custa de aceitar mais falsos positivos. Porém, mesmo tendo um modelo com um bom recall, as outras métricas não são descartáveis, pois por mais que a pesquisa busque por anomalias, não há ganho algum em um modelo que prevê todos os casos como anomalias em um cenário médico, pois o médico perderia um grande tempo para rever os eletrocardiogramas, e não apenas os casos necessários.

Enfim, o aprendizado de máquina pode ser categorizado em três abordagens principais: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço (BONACCORSO, 2018), (WU, NAGAHASHI, 2014).

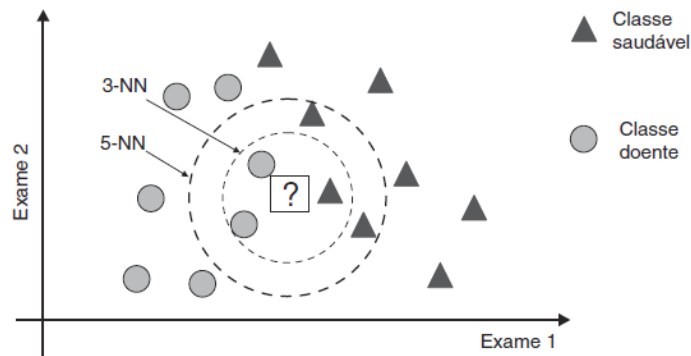
○ 2.2.1 Aprendizado supervisionado

Envolve treinar um algoritmo que é treinado com dados já rotulados, ou seja, as entradas e saídas correspondentes são as mesmas. O objetivo principal é que o algoritmo saiba prever a saída correta para novos dados de entrada. Esse tipo de aprendizado é comumente usado quando se quer fazer estimativas que possam classificar objetos ou prever resultados contínuos, tendo duas principais funcionalidades: classificação, para rótulos discretos e regressão, para rótulos contínuos (NASTESKI, 2017). O aprendizado supervisionado envolve os seguintes métodos:

- **XGBoost:** é um algoritmo avançado de aprendizado de máquina que otimiza árvores de decisão usando uma técnica de expansão de Taylor é um termo de regularização para evitar o overfitting. Desenvolvido pelo Dr. Chen Tianqi, é conhecido por sua rapidez e eficácia na classificação. Utiliza a árvore de regressão CART como base e combina a função de perda com regularização para manter um equilíbrio entre adaptação e generalização (LI, YIN, YANG, 2020).
- **AdaBoost:** proposto por Yoav Freund e Robert Saphire, o AdaBoost busca por uma melhora dos "aprendizes" fracos, ele ajusta iterativamente um conjunto de pesos sobre os dados de treinamento. Os pesos dos aprendizes se iniciam com pesos iguais, porém em cada iteração, os pesos dos exemplos classificados erroneamente são aumentados. O que força o aprendiz fraco a melhorar os exemplos mais difíceis do conjunto de treinamento (WANG, 2012).
- **RandomForest:** é um método de ML que utiliza múltiplas árvores de decisão para realizar tarefas de classificação ou regressão, oferecendo resultados precisos ao combinar as previsões de cada árvore. Ele é eficiente em capturar interações complexas entre variáveis automaticamente, o que o torna valioso em aplicações onde as relações entre os dados não são facilmente modeladas por técnicas simples (RIGATTI, 2017).

- K-Nearest Neighbors (KNN):** é um algoritmo simples de ML, em que pode ser usado tanto para classificação quanto para regressão, porém ele apresenta um problema significativo, ele fica lento conforme o tamanho dos dados em uso aumenta (MAHESH, 2018). O KNN classifica um ponto de um plano que ainda não foi rotulado usando como base seus vizinhos. Ou seja, o número de vizinhos é pré-determinado e conhecido como K, então o algoritmo calcula a distância do ponto não determinado com a quantidade K de vizinhos para descobrir quais são os mais próximos. O cálculo da distância pode ser alterado, mas por padrão, a métrica utilizada para calcular a distância dos vizinhos mais próximos é a euclidiana (FACELI, et al. 2021).

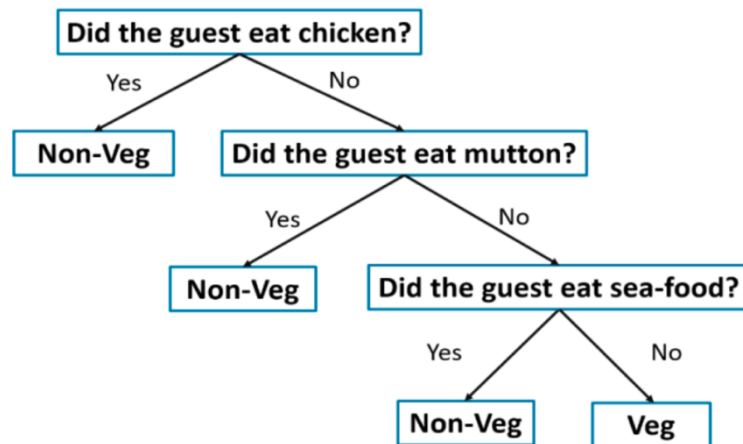
Figura 3 - Classificador KNN



Fonte: FACELI, et al. (2021)

- Árvore de Decisão:** é uma estrutura de grafo que representa escolhas e seus resultados na forma de uma árvore. Os nós representam decisões ou pontos de divisão, e os ramos representam as regras ou condições de decisão. Cada nó interno indica um atributo de um grupo a ser classificado, e cada ramo indica um possível valor para esse atributo (NASTESKI, 2017).

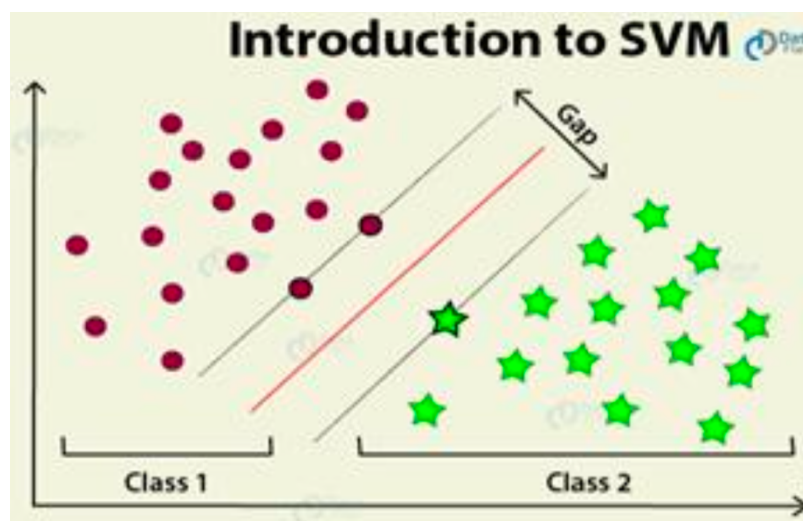
Figura 4 - Árvore Binária



Fonte: (MAHESH, 2018)

- Support Vector Machine (SVM):** é uma técnica avançada de aprendizado de máquina, pode analisar dados para uma classificação ou regressão, porém, o algoritmo possui uma desvantagem, em datasets muito grandes, a complexidade de tempo e espaço aumenta linearmente. Com base no “kernel trick”, o modelo pode realizar uma classificação linear e não linear, com o objetivo de mapear implicitamente entradas em espaços de recursos de alta dimensão. O SVM desenha margens entre as classes de forma a maximizar a distância entre a margem e as classes, minimizando assim o erro de classificação (ABDULLAH, ABDULAZEEZ, 2021).

Figura 5 - SVM

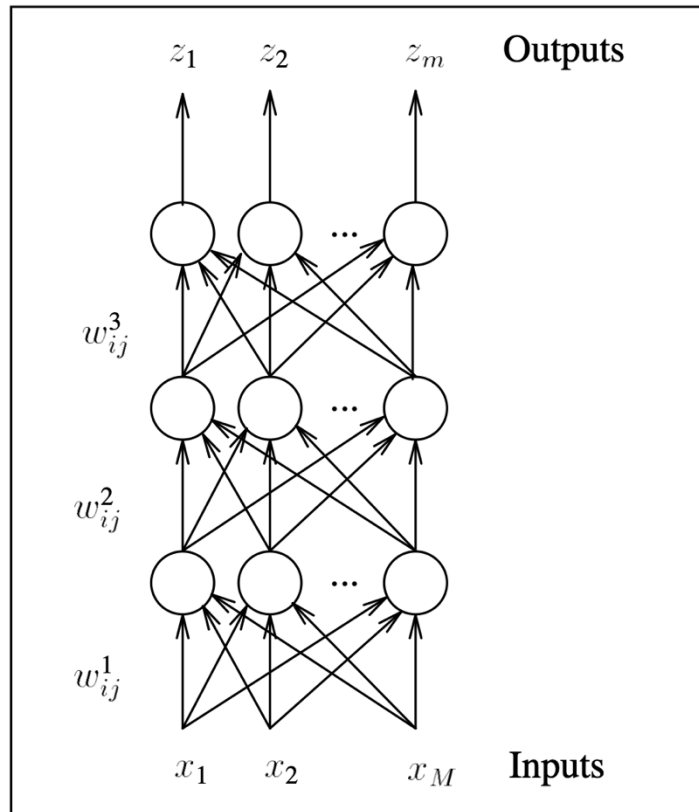


Fonte: (MAHESH, 2018)

Neste cenário, o SVM trabalha para encontrar a linha de decisão ideal, ajustando a posição até que a lacuna entre as classes seja maximizada. Uma vez determinada, essa linha de separação pode ser usada para classificar novos pontos, dependendo de em qual lado da linha o novo dado irá se alocar.

- **GradientBoosting:** faz parte da mesma família do XGBoost, ele é um tipo de algoritmo mais limitado em questão de tempo de execução pelo fato de ele não fazer o uso de multithread. Em sua iteração ele cria uma árvore de decisão de cada vez, o que no método XGBoost acontece de forma paralela (ALI, et al, 2023).
- **Multilayer Perceptron (MLP):** é conhecido pela sua arquitetura de rede neural, ele é composto por múltiplas camadas de neurônios, sendo a camada de entrada, as camadas ocultas e a camada de saída. Cada neurônio da rede neural está conectado a todos os outros neurônios da camada seguinte, com seus pesos agregados aos neurônios. Durante seu treinamento, a rede realiza o backpropagation, que minimiza o erro de predição (POPESCU, 2009).

Figura 6 - MLP



Fonte: (RUCK, ROGERS, KABRISKY, 1990)

○ 2.2.2 Aprendizado Não Supervisionado

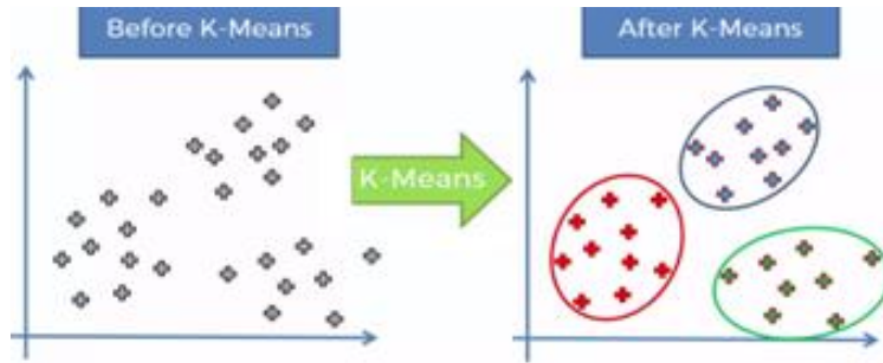
Se baseia no conjunto de dados sem rótulos. O objetivo é encontrar estruturas ou padrões nos dados. O uso de clusterização está presente neste método, que significa que o algoritmo na tentativa de encontrar os padrões, agrupa os dados em subconjuntos (ou clusters) com base em suas similaridades. Assim, os dados dentro de um mesmo cluster são mais semelhantes entre si do que com os de outros clusters, revelando relações e estruturas subjacentes no conjunto de dados.

Alguns dos métodos não supervisionados de aprendizagem de máquina são K-Means Clustering e Análise de Componentes Principais (MAHESH, 2018).

- **K-Means Clustering:** é um dos algoritmos mais simples quando se fala de aprendizado não supervisionado, o método visa classificar um conjunto de dados em um número específico de clusters, definindo k centros para esse cluster. A localização desses centros é crucial, pois diferentes posições levam

a resultados distintos. Idealmente, os centros devem ser posicionados o mais distante possível um dos outros.

Figura 7 - K-Means Clustering

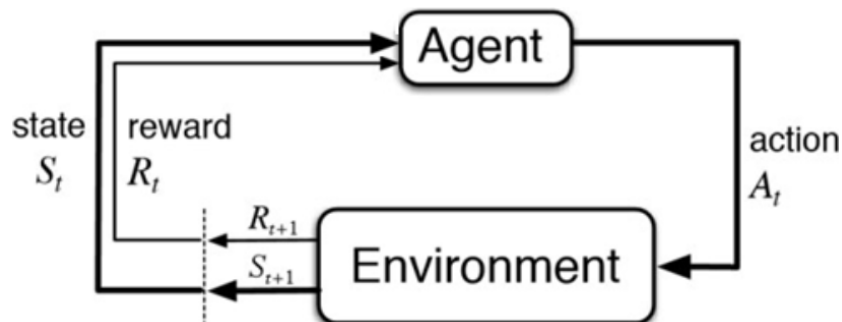


Fonte: (MAHESH, 2018)

- **Análise de Componentes Principais (PCA):** é uma técnica estatística que transforma variáveis correlacionadas em componentes principais linearmente não correlacionados. Ela reduz a dimensionalidade dos dados para otimizar cálculos. Após a transformação, os pontos são agrupados com base no centro mais próximo, e os centroides são recalculados conforme os clusters formados.
- **2.2.3 Aprendizado por Reforço**

É baseado em feedback, ou seja, o algoritmo toma suas decisões em um ambiente a fim de maximizar sua recompensa cumulativa. Ao longo do tempo, o agente aprende quais ações levam às melhores recompensas através de tentativa e erro, ajustando suas estratégias com base nas consequências de suas ações anteriores para alcançar o melhor desempenho possível.

Figura 8 - Representação do Aprendizado por Reforço



Fonte: (MAHESH, 2018)

Como pode-se visualizar na Figura 5, o agente realiza uma ação no ambiente. Em resposta a essa ação, o ambiente fornece ao agente uma recompensa e o próximo estado. Com base nesse feedback, o agente então decide qual será a próxima ação a tomar (MAHESH, 2018).

○ **2.3 Otimização de Hiperparâmetros**

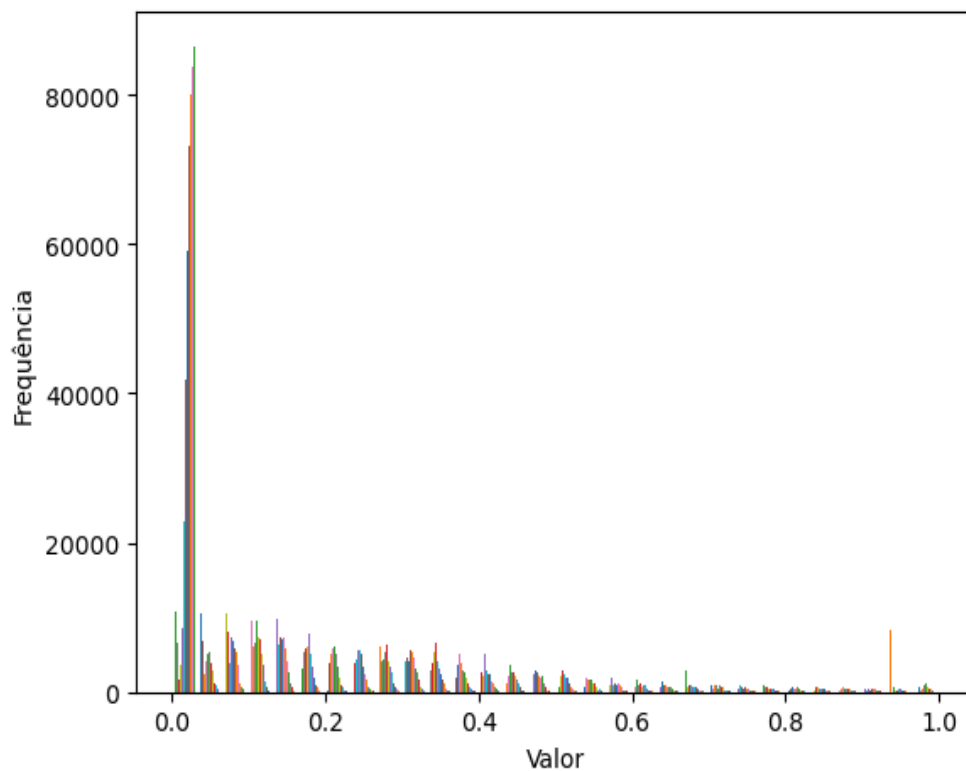
A otimização de hiperparâmetros é muito importante no desenvolvimento de modelos de aprendizado de máquina, visando maximizar o desempenho em variadas aplicações. Diferentes algoritmos de ML requerem a escolha cuidadosa de hiperparâmetros, que são definidos antes da execução dos algoritmos e alguns casos distintos podem ser modificados durante o uso deles, como é o caso das redes neurais. Porém, não há uma regra certa na escolha dos hiperparâmetros, sua escolha pode variar de aplicação a aplicação. A escolha ideal destes pode influenciar significativamente na arquitetura e eficácia do modelo. Comumente, a escolha do hiperparâmetros é feita manualmente, testando seu desempenho geral no aprendizado dos dados (YANG, SHAMI, 2020).

○ **2.4 Pré-processamento**

O pré-processamento é um dos primeiros e mais importantes passos para iniciar a classificação de sinais e imagens, pois com seu uso é possível a remoção de ruídos, balanceamento de classes, normalização de features, remoção de outliers, extração de características, entre outras funções. No campo do MIT-BIH Arrhythmia

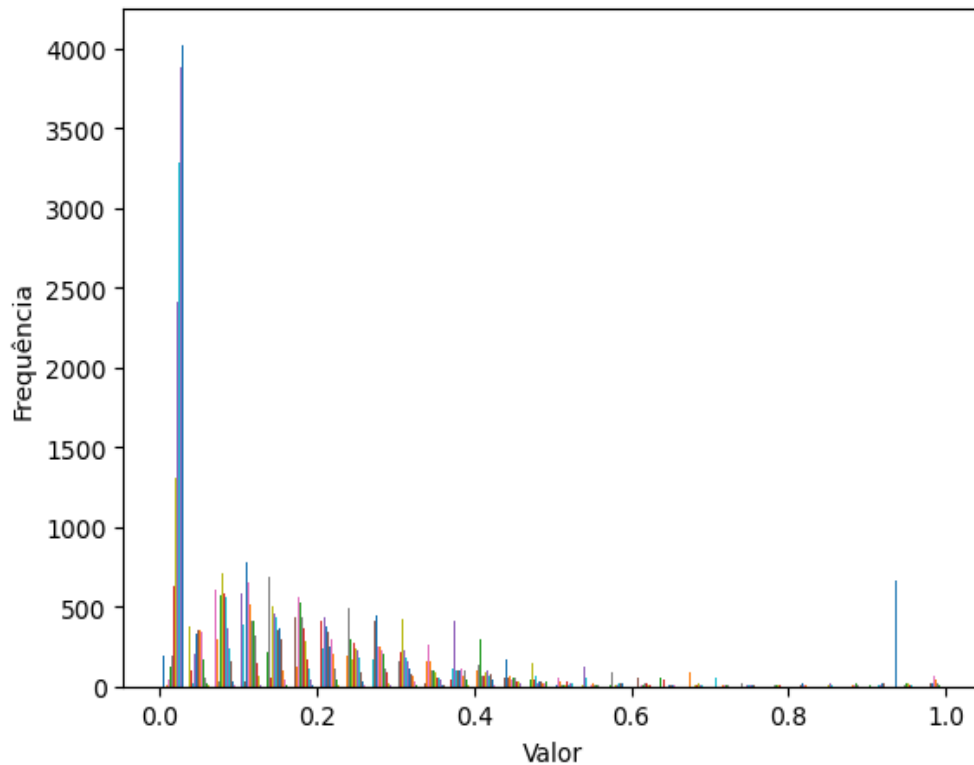
Database não é diferente, o pré-processamento é fundamental no uso dele, pois suas classes são extremamente desbalanceadas, como explicado no Capítulo 3. Esse desbalanceamento de classes pode causar um overfitting na classificação das classes com mais dados e reduzir a acurácia geral do modelo (SHOUGHI, MOHAMMAD, 2021). As Figuras 8 e 9 mostradas abaixo representam os conjuntos de treino dos datasets MIT-BIH Arrhythmia Database e PTB Diagnostic ECG Database.

Figura 9 - Conjunto de Treino MIT-BIH Arrhythmia Database



Fonte: Autoria Própria (2024)

Figura 10 - Conjunto de Treino PTB Diagnostic ECG Database



Fonte: Autoria Própria (2024)

Como é possível identificar nas Figuras 8 e 9, ambos os datasets utilizados no entorno da pesquisa estão normalizados, portanto não há necessidade de utilizar normalização dos dados na etapa de pré-processamento.

○ **2.5 Aprendizado de Máquina na Saúde**

O aprendizado de máquina tem a aspiração de transformar a medicina. Progressos tecnológicos, como o aumento significativo na capacidade de processamento e a vasta disponibilidade de dados clínicos por meio de prontuários eletrônicos, bem como o desenvolvimento do aprendizado de máquina, têm aproximado essa visão da realidade. O aprendizado de máquina tem o potencial de aprimorar a precisão de diagnósticos e oferecer tratamentos mais personalizados aos pacientes (Handelman, et al. 2018), assim podendo melhorar não só a velocidade dos atendimentos médicos, mas também sua qualidade e disponibilidade (MOREIRA, SALERNO, TSUNODA, 2020, p. 3).

À medida que a tecnologia evolui, o campo da detecção de anomalias em ECG tem se beneficiado significativamente do aprendizado de máquina. Diversas pesquisas têm explorado essa intersecção entre saúde e tecnologia. A seguir, estão os destaques de alguns estudos pertinentes sobre o assunto.

(AHMED, et al, 2020) utilizaram os métodos KNN, Árvore de Decisão, ANN com pesos ponderados, SVM, LSTM com pesos ponderados e uma abordagem com Ensemble. Seus melhores resultados foram utilizando Ensemble com uma acurácia de 97.664%, precisão de 96.904% e recall de 97.062%.

(TSAI, MORSHED, 2022) utilizaram os métodos SVM, KNN, Ensemble Tree, Rede Neural, Árvore de Decisão, Naive Bayes e Regressão Logística. Porém, os pesquisadores utilizaram o dataset BIDMC Congestive Heart Failure (CHF). Seus melhores resultados foram com a Rede Neural, com acurácia de 98.64%, precisão de 99.3% e f1-score de 99.42%.

(LATIF, ANEZI, ZIKRIA, 2020) utilizaram cinco métodos de aprendizado de máquina em sua pesquisa: Multilayer Perceptron, Radial Basis Function Network, Logistic Model, Naive Bayes e por último Random Forest. Seus melhores resultados foram obtidos a partir do método Random Forest, com uma acurácia de 97.45.

(MI, et al, 2023) utilizaram os métodos Back Propagation Neural Network, SVM e KNN. Seus melhores resultados vieram da BP Neural Network, com uma acurácia próxima de 99.1%.

(JAMBUKIA, DABHI, PRAJAPATI, 2015) realizaram uma survey de técnicas e algoritmos de classificação de batimentos cardíacos e encontraram pesquisas que utilizaram Redes Neurais, Máquinas de Vetores de Suporte (SVM), Transformada Discreta de Wavelet, entre outras, porém concluíram que redes neurais, especialmente MLP mostraram-se mais eficazes na classificação de ECG em termos de precisão.

(APANDI, IKEURA, HAYAKAWA, 2018) fizeram um levantamento de vários métodos como SVM, redes neurais, etc. focados em realizar a detecção de arritmias, especialmente usando o banco de dados MIT-BIH.

(SHOUGHI, DOWLATSHAHI, 2021) implementaram uma classificação automática de doenças cardiovasculares com pré-processamento e um classificador que consiste em rede neural convolucional e rede BLSTM. Os resultados das avaliações sobre um conjunto de dados desequilibrados do MIT-BIH mostram que esta abordagem com treinamento de 50 épocas alcançou cerca de 99% de precisão para todas as cinco categorias, variando de 98,85% a 99,87%.

CAPÍTULO 3 - MÉTODO DE PESQUISA

3.1 Caracterização da pesquisa

Esse trabalho será desenvolvido com abordagem descritiva e caráter experimental. Focado em analisar e identificar anomalias em ECG utilizando métodos de aprendizagem de máquina, com o propósito de auxiliar no diagnóstico prático e preciso de possíveis doenças cardíacas, comparando a eficácia de diferentes métodos de aprendizagem sobre o mesmo conjunto de dados.

Os dados a serem analisados provêm do conjunto de dados utilizados são o MIT-BIH Arrhythmia Dataset e o PTB Diagnostic ECG Database, que são públicos e provenientes dos laboratórios no Beth Israel Hospital de Boston (hoje Beth Israel Deaconess Medical Center) e no Instituto de Tecnologia de Massachusetts (MIT)

3.2 Método de Pesquisa

○ 3.2.1 Seleção de Dados

Como objeto para a experimentação foi selecionado um conjunto de dados disponibilizado pelo Kaggle, sendo formado por MIT-BIH Arrhythmia Dataset e o PTB Diagnostic ECG Database, que são dados públicos que podem ser encontradas individualmente no PhysioNet.

○ 3.2.1.1 MIT-BIH Arrhythmia Dataset

Os dados foram recolhidos pela MIT-BIH na parte clínica do Beth Israel Hospital, entre os anos 1975 e 1979. O conjunto de dados em questão contém 48 registros anotados de 47 indivíduos. Além disso, é dividido em séries de '100' e '200', escolhidos aleatoriamente entre um conjunto de 4000 fitas Holter de 24 horas com o objetivo de incluir uma variedade de fenômenos raros, mas clinicamente importantes. Os registros possuem em média 30 minutos de duração, com dois sinais amostrados a 360 Hz, totalizando aproximadamente 109.000 batimentos.

Quadro 1 - Descrição das classes de dados

AAMI Classes	Heartbeat Types
Normal beats (N)	Normal beats (N), Left bundle branch block (L), Right bundle branch block (R), Atrial escape beat (e), Nodal (junctional) escape beat (j)
Supraventricular ectopic Beats (S)	Atrial premature contraction (A), Aberrated atrial premature beat (a), Nodal (junctional) premature beat (J), Supraventricular premature beat (S)
Ventricular ectopic beats (V)	Ventricular premature contraction (V), Ventricular escape beat (E)
Fusion beats (F)	The fusion of ventricular and normal beat (F)
Unclassifiable beats (Q)	Paced beat (/), Fusion of paced and normal beat (f), Unclassified beat (Q)

Fonte: (GOLDBERGER, 2000)

○ 3.2.1.2 PTB Diagnostic ECG Database

O conjunto de dados em questão apresenta 14.552 amostras com uma taxa de amostragem de 125 Hz. As amostras pertencem à 294 pacientes, com idades de 17 a 87 anos, englobando 209 homens e 81 mulheres. Entre os indivíduos em questão, parte deles demonstra uma análise saudável no eletrocardiograma e parte demonstra doenças cardíacas. A coleta foi uma iniciativa da Physikalisch-Technisch Bundesanstalt (PTB), o Instituto Nacional de Metrologia da Alemanha. Os eletrocardiogramas foram coletados sob a supervisão do Professor Michael Oeff, M.D., no departamento de cardiologia da Universidade Clínica Benjamin Franklin.

○ 3.2.2 Pré-processamento e execução

O experimento tem sua base fundamentada nas hipóteses:

Hipótese nula: Nenhum método analisado apresenta melhoria possível a ser aplicada quanto aos resultados obtidos.

Hipótese alternativa: Encontradas mudanças que promovem resultados melhores dos métodos utilizados.

Para comprovação de tais hipóteses se utiliza o conjunto de dados mencionados, que são amplamente reconhecidos na área de pesquisa médica. Para busca de resultados limpos e precisos será feito um pré-processamento, filtrando os elementos com limpeza de dados, visando remover registros com muitos elementos nulos, e redução de ruídos, para remoção de elementos que não apresentem sentido ao aspecto da pesquisa.

Com o pré-processamento realizado, se parte para a aplicação dos métodos selecionados KNN, XGBoost, RandomForest, MLPClassifier, GradientBoosting, SVC e AdaBoost. Os métodos serão comparados com seus resultados baseados nos atributos de Precisão, Recall, F1-Score e Acurácia e Especificidade.

Em visão de melhoria dos atributos e de resultados será feita uma experimentação de hiperparâmetros, com alguns conjuntos específicos como por exemplo learning rate (Taxa de aprendizado), learning rate decay (Declínio da taxa de aprendizado) e momentum (Dinâmica)

○ **3.2.2.1 Exclusão do PTB Diagnostic ECG Database**

Após a análise dos datasets MIT-BIH Arrhythmia Database e PTB Diagnostic ECG Database, concluímos que seria muito difícil a tentativa de junção dos dois datasets pela diferença das classes, o PTB Diagnostic ECG Database possuindo apenas “normal” e “anormal” como elementos categóricos não suportaria a especificidade das 5 classes do MIT-BIH, e seria necessário um reducionismo ou contorcionismo de dados que levaria tempo e poderia prejudicar o andamento do trabalho, sendo assim, excluimos o uso do PTB Diagnostic ECG Database.

CAPÍTULO 4 - VALIDAÇÃO DA PESQUISA CIENTÍFICA

A essência de qualquer pesquisa científica reside na sua capacidade de suportar a análise rigorosa da validação. Este capítulo é dedicado a dissecar o processo de validação adotado neste estudo, que é fundamental para assegurar a integridade da aplicabilidade dos resultados. Através de uma abordagem sistemática que abrange desde a coleta de dados até a modelagem preditiva, este capítulo fornece uma visão transparente das metodologias empregadas para garantir que os achados sejam não apenas replicáveis, mas também robustos e confiáveis no campo de classificação de batimentos cardíacos.

4.1 Ambiente

O ambiente é fundamental para garantir que os experimentos sejam realizados de maneira controlada e que os resultados sejam confiáveis e replicáveis. Esta seção descreve os componentes de hardware e software necessários, bem como as qualificações das pessoas envolvidas no processo de validação.

4.1.1 Hardware

Para replicar o estudo contido nesta pesquisa não é necessário algum hardware específico, porém quanto mais potente é o hardware, mais rápida será a execução dos experimentos e a análise dos dados, tendo em vista que foi utilizado alguns métodos de aprendizado de máquina. A validação dos modelos de aprendizado de máquina para a classificação dos batimentos cardíacos foi realizada em um computador com as seguintes especificações:

- CPU: Apple M1 Pro (8 núcleos).
- Memória: 16 GB.
- Armazenamento: 494,38 GB.
- Sistema Operacional: macOS Sonoma (Versão 14.0)

É importante destacar que a paralelização foi utilizada para otimizar o processo de treinamento e validação dos modelos, aproveitando a arquitetura multi-core do processador M1 Pro. Isso permitiu uma execução mais eficiente dos algoritmos, reduzindo o tempo necessário para a conclusão dos experimentos. Mesmo assim, o estudo pode ser replicado em hardware com especificações mais modestas, embora com tempos de processamento potencialmente mais longos.

4.1.2 Software

Para a realização deste estudo, foi utilizada uma combinação de linguagem de programação, ambiente de desenvolvimento e bibliotecas específicas. A seguir, são apresentadas as ferramentas de software empregadas, bem como suas respectivas versões e propósitos dentro do contexto da pesquisa.

Para a realização deste estudo, foi necessária a utilização de uma série de ferramentas de software que incluem um ambiente de desenvolvimento integrado (IDE), uma linguagem de programação poderosa e diversas bibliotecas especializadas. Estas ferramentas foram escolhidas por sua robustez, ampla adoção na comunidade científica e capacidade de lidar eficientemente com operações de processamento de dados e aprendizado de máquina. As versões específicas das ferramentas e bibliotecas são especificadas para garantir sua replicabilidade com os resultados obtidos. A seguir, são detalhadas as ferramentas de software utilizadas no estudo, com suas respectivas versões e descrições breves, fornecendo uma visão geral do ambiente de software necessário para a execução e validação dos experimentos realizados.

A Tabela 1 apresenta o ambiente de desenvolvimento e a linguagem de programação utilizados, enquanto a Tabela 2 lista as bibliotecas Python específicas empregadas, destacando suas funções principais e sua fonte.

Tabela 1 - Softwares

Ferramenta	Versão	Descrição	Fonte
------------	--------	-----------	-------

Visual Studio Code	1.83.1	Editor de código fonte	https://code.visualstudio.com/
Python	3.10.9	Linguagem de programação	https://www.python.org/downloads/
Jupyter Notebook	6.5.2	Interface de programação Interativa	https://jupyter.org/install

Fonte: Autoria Própria (2024)

Tabela 2 - Bibliotecas Python

Biblioteca	Versão	Descrição	Fonte
pandas	1.5.3	Análise e manipulação de dados	https://pandas.pydata.org/
scikit-learn	1.2.1	Algoritmos de aprendizado de máquina	https://scikit-learn.org/stable/
matplotlib	3.8.4	Criação de tabelas	https://matplotlib.org/
joblib	1.1.1	Serialização de modelos de ML	https://joblib.readthedocs.io/
numpy	1.26.4	Processamento de lista de números, textos e objetos.	https://numpy.org
tabulate	0.9.0	Organização de prints com tabelas	https://pypi.org/project/tabulate/
seaborn	0.13.2	Visualização de dados	https://seaborn.pydata.org/

4.2 Condições para Iniciar a Validação

Para que a validação seja realizada com sucesso, algumas condições devem ser estabelecidas e cumpridas. Este capítulo descreve as etapas essenciais para preparar o ambiente de validação.

4.2.1 Instalação de Softwares

Antes de iniciar a validação dos modelos de aprendizado de máquina desenvolvidos neste estudo, é necessário preparar o ambiente de desenvolvimento e garantir que todos os envolvidos estejam devidamente capacitados para realizar os testes. Este capítulo visa descrever as etapas necessárias para configurar o ambiente de validação, incluindo a instalação de softwares.

Antes de proceder com a validação, é necessário instalar o Python na versão 3.10.9, bem como as ferramentas de desenvolvimento Visual Studio Code e Jupyter Notebook. Além disso, todas as bibliotecas Python utilizadas no projeto devem ser instaladas nas versões especificadas na Tabela 2.

4.2.2 Configurações do Ambiente de Desenvolvimento

Recomenda-se a criação de um ambiente virtual Python para evitar conflitos entre as dependências do projeto e outras bibliotecas instaladas no sistema. O ambiente virtual pode ser criado e gerenciado utilizando ferramentas como [venv](#) ou [conda](#). Uma vez que ativado, todas as instalações de bibliotecas devem ser realizadas dentro deste ambiente isolado.

4.2.3 Preparação dos Dados

Os dados utilizados neste estudo, conforme citado no [Capítulo 3.2](#), são provenientes do MIT-BIH Arrhythmia Database. O conjunto de dados é de referência na área de pesquisa cardíaca e estão disponíveis publicamente para fins de pesquisa:

- MIT-BIH Arrhythmia Database: [PhysioNet](#)

Após a obtenção do conjunto de dados, os dados foram carregados utilizando a biblioteca pandas, uma escolha popular para a manipulação dos dados em Python devido à sua eficiência e facilidade de uso. Os dados foram então divididos em características (X), que representam as medições dos batimentos cardíacos, e rótulos (y), que correspondem às classes de arritmia.

Antes de prosseguir para a validação dos modelos, foi realizada uma averiguação cuidadosa dos dados para assegurar que não houvesse inconsistências ou desequilíbrios significativos que pudessem afetar o treinamento dos modelos. Isso incluiu a análise de distribuição das classes e a verificação da necessidade de técnicas de balanceamento ou normalização, que foram aplicadas conforme necessário.

Com os dados devidamente preparados e verificados, o estudo está pronto para avançar para a etapa de validação dos modelos de aprendizado de máquina descrita no Capítulo 4.3.

4.3 Procedimentos para Realizar a Validação

A validação dos modelos de aprendizado de máquina é conduzida através de um processo descrito no Capítulo 4.2.3. Com os dados prontos e verificados, passamos para a etapa de treinamento e validação dos modelos selecionados.

Os algoritmos escolhidos, incluindo SVC, KNN e Random Forest, são escolhidos alguns parâmetros para tentar otimizar ao máximo o desempenho do aprendizado dos modelos. A validação cruzada é utilizada para avaliar a eficácia de cada modelo, fornecendo uma estimativa robusta de seu desempenho em dados não vistos anteriormente.

Os resultados são analisados, permitindo uma comparação entre os modelos e a seleção daqueles com melhor desempenho. Estes modelos são então salvos para o uso futuro, garantindo a replicabilidade do estudo e a aplicabilidade prática dos modelos em situações reais.

4.4 Análise dos Resultados

Nesta seção, serão discutidos os resultados de cada modelo, com foco em métricas-chave como acurácia, precisão, recall, f1-score, especificidade e a matriz de confusão. Os cadernos criados no Jupyter Notebook para cada modelo estão disponíveis no repositório do GitHub ⁴.

4.4.1 Análise do Modelo SVC⁵

O modelo SVC, como qualquer outro método das bibliotecas scikit-learn quando treinado sem parâmetros, executa com seus parâmetros padrões, que podem ser visualizados na documentação do Support Vector Classifier (SVC) do Scikit Learn.

Sabendo que cada método possui seus parâmetros próprios, foi escolhido o parâmetro “kernel” para realizar o primeiro experimento, pois segundo HAN, QUBO, MENG (2012) kernel é um dos parâmetros mais importantes e influentes no modelo SVC, é com seus diferentes kernels que o modelo separa os dados. Suas variações são *linear*, *poly*, *rbf* e *sigmoid*. Em seus resultados o que obteve os melhores resultados foi o rbf, como mostra a Tabela 3.

Tabela 3 - Média Ponderada com Diferentes Kernel

Kernel	Precisão	Recall	F1-Score	Especificidade
linear	0.92	0.92	0.91	0.916
poly	0.96	0.96	0.95	0.955
rbf	0.97	0.97	0.97	0.9679
sigmoid	0.75	0.79	0.76	0.8238

Fonte: Autoria Própria (2024)

Como podemos ver, todas as métricas com o kernel em rbf ficaram melhores, portanto o próximo experimento foi feito utilizando múltiplos valores para o parâmetro

⁴ <https://github.com/Viihizaa/Projeto-Transformador-I-Prototipo>

⁵ <https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC>

gamma e foi mantido o kernel *rbf*. Suas possibilidades variaram entre *scale* e *auto*, e seus resultados podem ser analisados na Tabela 5.

Tabela 4 - Média Ponderada com Diferentes Gamma

Gamma	Precisão	Recall	F1-Score	Especificidade
scale	0.97	0.97	0.97	0.9679
auto	0.97	0.97	0.97	0.9679

Fonte: Autoria Própria (2024)

Sabendo que o modelo se adaptou melhor com o kernel definido em *rbf*, o próximo parâmetro experimentado foi o *gamma*, que para o treinamento do modelo não foi impactante, portanto, o próximo experimento foi utilizado o mesmo kernel, com ambos os diferentes possíveis valores para *gamma* e pelo fato de *C* ser uma variável que aceita números racionais foram testados valores com um espaçamento razoável entre eles, e seus melhores resultados originaram novos testes com espaçamentos menores. Os primeiros testes do parâmetro *C* foram com os valores 0.1, 1, 10 e 100. Como o melhor caso foi com *C* em 100, os próximos testes foram feitos utilizando um intervalo de 25, os valores experimentados foram 50, 75, 100, 125 e 150 com o melhor resultado de *C* em 75, com resultados de precisão, recall e f1-score com média ponderada de 0,98, 0.98 e 0,98, respectivamente.

Seguindo essa mesma lógica para os parâmetros *max_iter* e *shrinking*, portanto a Tabela 6 representa os resultados para o modelo sem a otimização de hiperparâmetros.

Tabela 5 - Resultados Padrões para SVC

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.969	0.97	0.97	0.97	0.9679
Aritmética	0.969	0.94	0.77	0.84	0.9679

Fonte: Autoria Própria (2024)

Podemos ver com as Tabelas 5 e Tabela 6, que a experimentação trouxe melhorias na predição, abaixo estão tanto o melhor conjunto encontrado, quanto seus resultados.

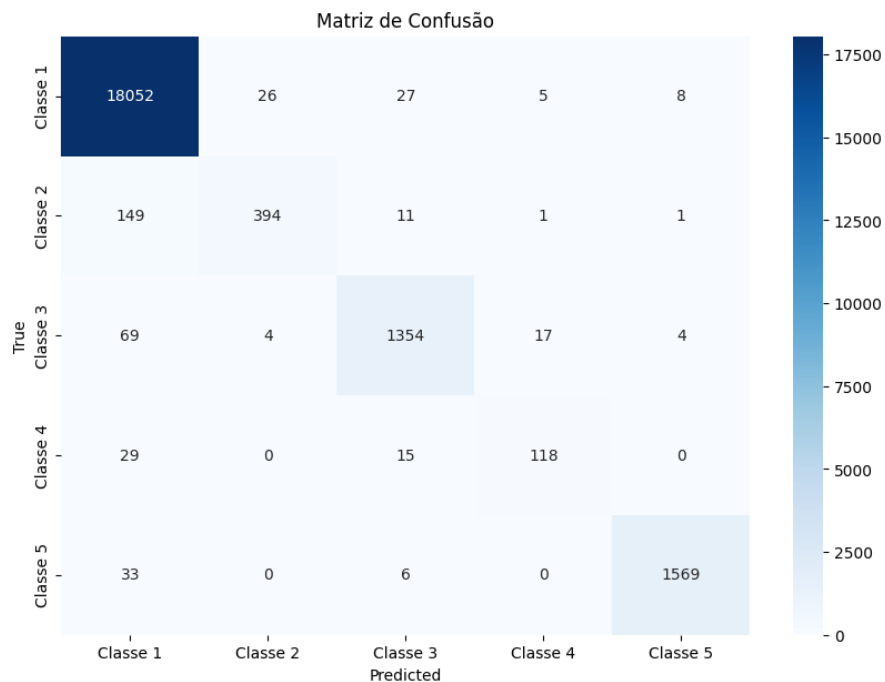
- C: 75
- gamma: scale
- kernel: rbf
- max_iter: 12500
- shrinking: True

Tabela 6 - Resultados Otimizados para SVC

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.98	0.98	0.98	0.98	0.98
Aritmética	0.98	0.94	0.87	0.90	0.98

Fonte: Autoria Própria (2024)

Portanto, é possível ver uma clara melhora em todas as métricas tanto com a média ponderada das classes, quanto a média aritmética, o que mostra que a experimentação do modelo com diferentes hiperparâmetros está sendo benéfico para os dados. Abaixo, podemos verificar também a matriz de confusão do modelo com este melhor conjunto treinado, na Figura 1.

Figura 11 - Matriz de Confusão do SVM

Fonte: Autoria Própria (2024)

4.4.2 Análise do Modelo KNN⁶

A experimentação com o modelo de aprendizado de máquina KNN, bem como o SVM, possui seus próprios parâmetros padrões quando executados, neste capítulo será mostrado como foi feita esta experimentação com o KNN. Sabendo disso, a Tabela 7 mostra os valores padrões do modelo.

Tabela 7 - Parâmetros Padrões para KNN

Parâmetro	Padrão
n_neighbors	5
weights	uniform
algorithm	Auto
leaf_size	30
p	2
metric	minkowski

⁶ <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

metric_params	None
n_jobs	None

Fonte: Autoria Própria (2024)

Sabendo das métricas padrões usadas para treinar este modelo, os primeiros testes vieram no número de vizinhos (`n_neighbors`), e este foi treinado usando o número de 1 até 1000 vizinhos, para testar se o aumento deste número faria com que o modelo tivesse um melhor desempenho. Como apresentado na Figura 2.

Figura 12 - Aprendizado no Número de Vizinhos



Fonte: Autoria Própria (2024)

Como é possível verificar na Figura 2, quanto maior o número de vizinhos, pior foi a habilidade do modelo na classificação dos batimentos. Assim, foi descoberto que o melhor número de vizinhos é de 1.

Após essa descoberta, os próximos parâmetros testados foram o *weights* e o *algorithm*, e assim que testado, trouxeram os mesmos resultados entre suas possibilidades, tanto com valor *uniform* quanto com *distance* para *weights*, e os mesmos resultados para o parâmetro usado para calcular os vizinhos mais próximos (*algorithm*), testados com valores de *auto*, *ball_tree*, *kd_tree* e *brute*. Obtendo uma acurácia média de 0,97.

Para verificar todas conjuntos separados de parâmetros, foi utilizado um GridSearch, assim, chegando no conjunto abaixo com os melhores resultados:

- n_neighbors: 1
- weights: uniform
- algorithm: auto
- p: 1
- leaf_size: 20

Tendo em vista que o parâmetro leaf_size só afeta a velocidade de execução, e não o seu desempenho, o GridSearch indicou que o valor 20 para este parâmetro por ser o primeiro a ser passado, assim como o weights e algorithm, que quando testados de maneira separada, apresentaram os mesmos resultados. E em termos de resultado, podemos ver na Tabela 8 os resultados padrões do modelo KNN com o dataset MIT-BIH Arrhythmia Dataset.

Tabela 8 - Resultados Padrões para KNN

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.974	0.972	0.973	0.972	0.983
Aritmética	0.974	0.922	0.846	0.879	0.983

Fonte: Autoria Própria (2024)

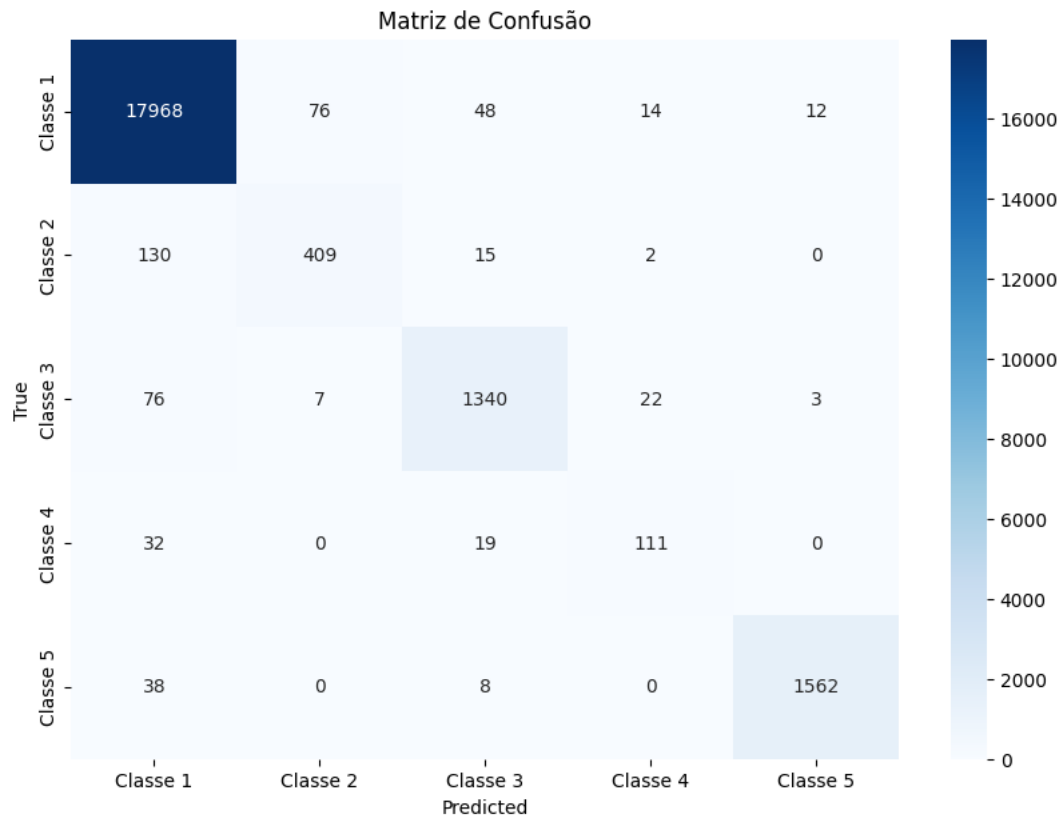
E na Tabela 9, estão os resultados otimizados para o modelo KNN, com a experimentação realizada.

Tabela 9 - Resultados Otimizados para KNN

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.977	0.976	0.977	0.976	0.976
Aritmética	0.977	0.908	0.878	0.892	0.976

Fonte: Autoria Própria (2024)

Na Figura 3 pode-se ver a matriz de confusão do KNN, onde as classes 1 e 5 trazem melhores resultados, pela quantidade de instâncias dentro de cada uma das classes.

Figura 13 - Matriz de Confusão do KNN

Fonte: Autoria Própria (2024)

4.4.3 Análise do Modelo Random Forest⁷

Para a fase de prototipação do estudo foi feita uma execução do método RandomForest utilizando a biblioteca disponível pelo SkLearn.

Inicialmente o método foi executado em sua versão default tendo os seguintes valores de atributos:

Tabela 10 - Parâmetros Padrões para RandomForest

Parâmetro	Padrão
n_estimators	100
criterion	gini
max_depth	None

⁷

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier>

min_samples_split	2
min_samples_leaf	1
min_weight_fraction_leaf	0.0
max_features	sqrt
max_leaf_nodes	None
min_impurity_decrease	0.0
bootstrap	True
oob_score	False
n_jobs	None
random_state	None
verbose	0
warm_start	Fase
class_weight	None
ccp_alpha	0.0
max_samples	None
monotonic_cst	None

Fonte: Autoria Própria (2024)

Nessa versão os resultados são listados na tabela:

Tabela 11 - Resultados Padrões para RandomForest

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.967	0.967	0.967	0.965	0.964
Aritmética	0.967	0.963	0.788	0.852	0.964

Fonte: Autoria Própria (2024)

Para critério de avaliação foi construída uma matriz de confusão da execução:

Figura 14 - Matriz de Confusão do Modelo Padrão para RandomForest



Fonte: Autoria Própria (2024)

Iniciando a experimentação dentro dos próprios atributos foi realizada uma validação cruzada da acurácia da execução do método alterando o atributo “n_estimators”, que é relativo ao número de árvores, entre os valores 1 e 100, para visualização dos resultados foi gerado um gráfico com o MSE (Mean Squared Error):

Figura 15 - Gráfico de Desempenho do RandomForest

Fonte: Autoria Própria (2024)

Ao comparar os resultados, foi estipulado que o melhor resultado veio da execução com 73 árvores. Seguindo para outro atributo utilizando o melhor resultado para “n_estimators” foi realizado o mesmo processo de validação cruzada porém alterando “class_weight” entre os valores “balanced” e “balanced_subsample”, gerando o histograma:

Figura 16 - Gráfico de Comparação de Pesos para RandomForest



Fonte: Autoria Própria (2024)

Apesar dos resultados possuírem valores aproximados, se notou que o valor 'balanced' teve uma acurácia maior em relação ao outro valor.

Utilizando os dois valores encontrados como melhores em resultado foi construída a seguinte tabela:

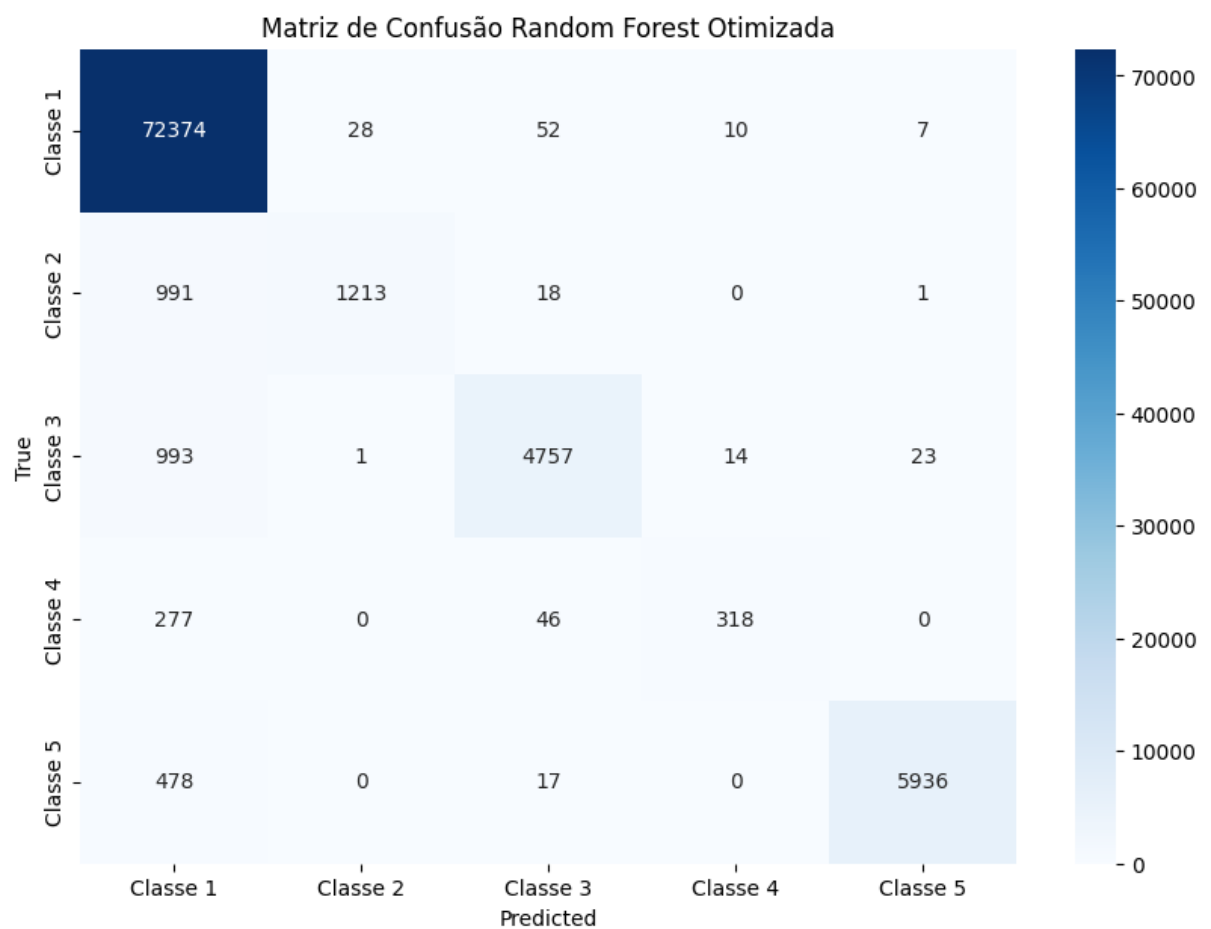
Tabela 12 - Resultados Otimizados para RandomForest

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.966	0.966	0.966	0.964	0.963
Aritmética	0.966	0.967	0.788	0.854	0.963

Fonte: Autoria Própria(2024)

Foi construída também uma matriz de confusão para esse teste otimizado:

Figura 17 - Matriz de Confusão do Modelo Otimizado para RandomForest



Fonte: Autoria Própria (2024)

4.5 Discussão de Resultados

Os modelos SVM, KNN e Random Forest foram analisados e otimizados individualmente para melhorar seu desempenho em relação ao conjunto de dados. Cada modelo foi inicialmente treinado com seus parâmetros padrão e, em seguida, submetido a uma série de experimentações com diferentes hiperparâmetros.

4.5.1 Resultados Gerais dos Modelos

Os resultados mostraram que a otimização dos hiperparâmetros teve um impacto significativo no desempenho dos modelos, em geral. No caso do SVC, o parâmetro "kernel" foi identificado como o mais influente, com kernel rbf apresentando o melhor desempenho em todas as métricas avaliadas, como mostra a Tabela 3. Subsequentemente, a variação do parâmetro gamma entre "scale" e "auto" não trouxe

diferenças significativas, já os parâmetros *C*, *shrinking* e *max_iter*, quando adicionados, obtiveram uma pequena melhora em suas métricas com média ponderada das classes e uma grande melhora em todas as métricas com média aritmética, como mostrado na Tabela 5 e 6.

Para o modelo KNN, a variação do número de vizinhos (*n_neighbors*) de 1 a 1000 revelou que 1 vizinho proporcionou os melhores resultados. Outros parâmetros *weights* e *algorithm* também foram ajustados, mas não apresentaram mudanças em seu desempenho. A combinação otimizada de parâmetros indicada pelo GridSearch foi: *algorithm=auto*, *leaf_size=20*, *n_neighbors=1*, *p=1* e *weights=uniform*, em testes separados dos parâmetros, foi descoberto que para os dados contidos no conjunto de dados utilizado, as únicas melhores surgiram do número de vizinhos (*n_neighbors*) e do parâmetro "*p*" com valor 1.

Já no caso do Random Forest, a experimentação com o número de estimadores (*n_estimators*) de 1 a 100 indicou que 73 árvores proporcionam o melhor desempenho. Depois, foram testados diferentes valores para o peso das classes (*classes_weight*) entre "*balanced*" e "*balanced_subsample*", onde as médias aritméticas melhoraram levemente e as médias ponderadas não apresentaram mudanças significativas, como apresentado nas Tabelas 11 e 12.

Tabela 13 - Resultados Padrões com Média Ponderada

Modelo	Acurácia	Precisão	Recall	F1-Score	Especificidade
KNN	0.974	0.972	0.973	0.972	0.983
SVC	0.969	0.97	0.97	0.97	0.9679
Random Forest	0.96	0.97	0.97	0.97	0.96

Fonte: Autoria Própria (2024)

Tabela 14 - Resultados Otimizados com Média Ponderada

Modelo	Acurácia	Precisão	Recall	F1-Score	Especificidade
KNN	0.977	0.976	0.977	0.976	0.976
SVC	0.98	0.98	0.98	0.98	0.98
Random Forest	0.96	0.97	0.97	0.96	0.96

Fonte: Autoria Própria (2024)

4.6 Considerações Finais

Sendo a principal hipótese do estudo a possibilidade de melhora em resultados de métodos de classificação com essa comparação, fica demonstrado como essa trilha está se concretizando, e o objetivo vem sendo alcançado em alguns métodos.

Em conclusão, com as otimizações atuais, os métodos apresentaram, em sua maioria, melhoras. Porém para promover um aumento nos valores de seus relatórios, serão feitas, em próximos passos da prototipação do estudo, mais alterações em parâmetros dos métodos, utilizando ferramentas como o Grid Search ou mudanças individuais nos parâmetros para determinar valores e parâmetros a serem utilizados na execução em busca de resultados elevados.

CAPÍTULO 5 - INFORMAÇÕES DA PESQUISA

5.1 Cronograma

Para organizar as tarefas e objetivos por etapa do projeto, utilizamos a metodologia Scrum, e as atividades foram organizadas em Sprints. A sequência das Sprints apresentadas reflete a ordem planejada de desenvolvimento das atividades e tarefas do projeto. O registro detalhado de cada Sprint, bem como das tarefas associadas, foi feito usando a plataforma Trello. A seguir, a tabela apresenta o planejamento das Sprints e os respectivos backlogs definidos para cada uma delas.

Tabela 15 - Planejamento do Trello

Sprint	Backlog	Data Limite
00	- Definir o escopo do projeto - Entender a estrutura inicial do documento - TDE 1 - Submissão de Proposta	4 de março
01	TDE 2 – Planejamento e Vídeo	18 de março
02	- Realizar Pesquisa - Capítulo 2 – Revisão de Literatura	24 de março
03	Capítulo 3 – Método de Pesquisa	29 de março
04	Capítulo 1 - Introdução	2 de abril
05	Capítulo 4 – Informações da Pesquisa	7 de abril
06	Resumo e Abstract	9 de abril
07	- Revisão das Listas - Revisão das Referências Bibliográficas - Revisão do Documento	14 de abril
08	Entrega do Documento	15 de abril

09	Revisão pós resultado da avaliação e adição do novo capítulo 4	30 de abril
10	Desenvolvimento do Protótipo	25 de maio
11	Capítulo 4 – Validação da Pesquisa Científica	2 de junho
12	Revisão e Entrega da Avaliação 2	9 de junho

Fonte: Autoria Própria (2024)

5.2 Riscos

A identificação e gestão de riscos são fundamentais para garantir a execução bem-sucedida do projeto. É essencial reconhecer que riscos são inerentes em qualquer projeto e, muitas vezes, são imprevisíveis. No entanto, uma gestão proativa e sistemática de riscos permite não apenas identificar e avaliar essas ameaças, mas também desenvolver estratégias para mitigá-las ou contorná-las.

A análise de riscos envolve não apenas a identificação, mas também a priorização. Alguns riscos podem ter um impacto significativo no projeto, mas têm uma probabilidade baixa de ocorrência, enquanto outros podem ser mais prováveis, mas com um impacto menor. Avaliar a severidade e a probabilidade de cada risco permite alocar recursos e esforços de forma mais eficaz.

A gestão de riscos é um processo contínuo. À medida que o projeto avança, novos riscos podem surgir, enquanto outros podem se tornar obsoletos. Portanto é de verdadeira importância revisar e atualizar regularmente a análise de riscos, adaptando as estratégias conforme necessário. Além disso, a gestão de riscos é um processo contínuo.

A gestão de riscos começa com o desenvolvimento dos requisitos de entrada do projeto. À medida que o design evolui, novos riscos podem ser identificados. Para sistematicamente identificar, e quando necessário, reduzir esses riscos, o processo de gerenciamento de riscos é integrado ao processo de design (BARTOO, 2003, p. 166).

Com base na metodologia proativa descrita, a Tabela 2 a seguir cataloga os riscos identificados no projeto, proporcionando uma visão clara de sua probabilidade,

impacto e severidade. Esta tabela é um instrumento vital no planejamento estratégico, auxiliando na preparação e resposta adequada a cada potencial desafio identificado.

Tabela 16 - Riscos do Projeto

Id	Descrição	P	I	S	Ação de Prevenção	Ação de Contingência
01	Dificuldades na escrita ou formatação do documento	Média	Média	Média	Participar de workshops ou cursos de redação	Solicitar revisão de colegas ou usar ferramentas externas
02	Falta de artigos relevantes sobre o tema	Baixa	Alta	Alta	Realizar uma revisão bibliográfica preliminar	Ajustar o foco da pesquisa ou buscar fontes internacionais
03	Problemas de saúde ou pessoais	Baixa	Alta	Alta	Manter um equilíbrio entre estudo e descanso	Solicitar suporte adicional do orientador
04	Dificuldades em realizar experimentos ou testes	Média	Média	Média	Planejar experimentos com antecedência	Revisar a metodologia ou buscar alternativas
05	Tempo de Execução dos Métodos	Média	Média	Média	Planejar momentos e prever possíveis tempos de execução	Utilizar máquinas disponibilizadas pelo orientador de uso exclusivo para a execução dos métodos

Legenda P: probabilidade (alta, média, baixa)

I: impacto (alto, médio, baixo)

S: severidade ($S=P \times I$)

A Tabela de Riscos detalha a avaliação e o planejamento para cada risco identificado, incluindo estratégias de prevenção e ações de contingência para cada cenário. A avaliação de cada risco é baseada em sua probabilidade (P), impacto (I), e severidade calculada (S), permitindo a priorização apropriada. Sendo as estratégias de prevenção usadas para mitigar o risco antes que ele afete o desenvolvimento do projeto, enquanto planos de contingência são estabelecidos para responder efetivamente caso o risco se concretize.

REFERÊNCIAS BIBLIOGRÁFICAS

ABDULLAH, D. M.; ABDULAZEEZ, A M.; Machine learning applications based on SVM classification a review. Qubahan Academic Journal, v. 1, n. 2, p. 81-90, 2021.

ALI, Z. A.; et al. "Exploring the power of eXtreme gradient boosting algorithm in machine learning: A review." Academic Journal of Nawroz University 12.2 (2023): 320-334.

APANDI, Ziti Fariha Mohd; IKEURA, Ryojun; HAYAKAWA, Soichiro. Arrhythmia detection using MIT-BIH dataset: A review. In: 2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA). IEEE, 2018. p. 1-5.

GNOATTO, R; FRANZEN, E. ANÁLISE DO DESEMPENHO DE HIPERPARÂMETROS DE APRENDIZAGEM DE MÁQUINA APLICADOS NA PREVISÃO DA TAXA DE ROTATIVIDADE DE CLIENTES. Revista Destaques Acadêmicos, v. 15, n. 4, 2024.

GOLDBERGER, A.; AMARAL, L.; GLASS, L.; HAUSDORFF, J.; IVANOV P. C.; Mark R; MIETUS, J. E.; MOODY, G. B.; PENG, C. K.; STANLEY, H. E.; PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220.

HAN, S; QUBO, C; MENG, H; "Parameter selection in SVM with RBF kernel function," World Automation Congress 2012, Puerto Vallarta, Mexico, 2012, pp. 1-4.

JAMBUKIA, S. H.; DABHI, V. K.; PRAJAPATI, H. B. Classification of ECG signals using machine learning techniques: A survey. In: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTER ENGINEERING AND APPLICATIONS, 2015, Ghaziabad, Índia. Anais... Ghaziabad: [s.n.], 2015. p. 714-721. doi: 10.1109/ICACEA.2015.7164783.

LATIF, G.; ANEZI, F. Y. Al; ZIKRIA, M.; ALGHAZO, J.; "EEG-ECG Signals Classification for Arrhythmia Detection using Decision Trees," 2020 Fourth International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 2020, pp. 192-196, doi: 10.1109/ICISC47916.2020.9171084.

MI, F.; LI, B.; CHENG, X.; ZHAO, Y.; LI, M.; JING, J.; "Classification and Processing of MIT-BIH Arrhythmia-Based on BP Algorithm," 2023 International Conference on Intelligent Supercomputing and BioPharma (ISBP), Zhuhai, China, 2023, pp. 72-76, doi: 10.1109/ISBP57705.2023.10061303.

NASTESKI, V. Uma visão geral dos métodos de aprendizado de máquina supervisionados. Horizons, v. 4, n. 51-62, p. 56, 2017.

POPESCU, M. C.; BALAS, V. E.; PERESCU-POPESCU, L.; MASTORAKIS, N.; Multilayer perceptron and neural networks. WSEAS Transactions on Circuits and Systems, 2009.

RUCK, D. W.; ROGERS, S. K.; KABRISKY, M.; Feature selection using a multilayer perceptron. Journal of neural network computing, 1990.

SHOUGHI, A.; MOHAMMAD, B.D. "A practical system based on CNN-BLSTM network for accurate classification of ECG heartbeats of MIT-BIH imbalanced dataset." 2021 26th international computer conference, Computer Society of Iran (CSICC). IEEE, 2021.

STALIN, S; RAJKUMAR P; SUBBUTHAI, P. "Feature Extraction and Classification for ECG Signal Processing based on Artificial Neural Network and Machine Learning Approach." International Conference on Inter Disciplinary Research in Engineering and Technology, 2015.

TSAI, I. H.; MORSHED, B. I.; "Beat-by-beat Classification of ECG Signals with Machine Learning Algorithm for Cardiac Episodes," 2022 IEEE International Conference on Electro Information Technology (eIT), Mankato, MN, USA, 2022, pp. 311-314, doi: 10.1109/eIT53891.2022.9813902.

WANG, R. (2012). AdaBoost for feature selection, classification and its relation with SVM, a review. Physics Procedia, 25, 800-807.

YANG, L; SHAMI, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, v. 415, p. 295-316, 2020.