



Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Departamento de Engenharia Elétrica

Thiago José Ribeiro Guimarães

## **Trabalho de Conclusão de Curso**

# **Identificação de Doenças Cardíacas a partir de Eletrocardiogramas Utilizando Machine Learning**

Campina Grande - PB

Julho de 2019

Thiago José Ribeiro Guimarães

## **Trabalho de Conclusão de Curso**

# **Identificação de Doenças Cardíacas a partir de Eletrocardiogramas Utilizando Machine Learning**

*Trabalho de Conclusão de Curso submetido à Coordenação de Curso de Graduação de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.*

Área de Concentração: Machine Learning.

Orientador: Gutemberg Gonçalves dos Santos Júnior

Campina Grande - PB

Julho de 2019

Thiago José Ribeiro Guimarães

## **Trabalho de Conclusão de Curso**

# **Identificação de Doenças Cardíacas a partir de Eletrocardiogramas Utilizando Machine Learning**

*Trabalho de Conclusão de Curso submetido à Coordenação de Curso de Graduação de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.*

Trabalho aprovado em: Campina Grande - PB,        /        /

---

**Gutemberg Gonçalves dos Santos Júnior, UFCG**  
Professor Orientador

---

**Danilo Freire de Souza Santos, UFCG**  
Professor Avaliador

Campina Grande - PB  
Julho de 2019

Dedico este trabalho a Deus, Nossa Senhora, meus pais e irmãs por todo apoio e força que foram de suma importância para conclusão dessa etapa.

# Agradecimentos

Agradeço a Deus e Nossa Senhora, por toda interseção e força nos momentos de precisão, sem as suas presenças não seria possível concluir essa etapa.

Minha Mãe e meu Pai, tenho uma eterna gratidão, pelo esforço realizado para que eu tivesse uma educação de qualidade, mesmo diante das diversas dificuldades. Bem como, o apoio diante dos obstáculos encontrados e exemplos de pessoas e casal a serem seguidos. Saibam que durante toda essa jornada foi possível ver o amor de vocês nos mais simples gestos.

As minhas irmãs, Cynthia e Priscylla, por todos os momentos de descontração que tornaram as dificuldades encontradas mais leves e pela companhia diária sempre com alegria e amor.

Não poderia deixar de citar o professor Gutemberg Gonçalves dos Santos Júnior pela orientação nesse trabalho, contribuições e conhecimentos compartilhados. Em especial, por toda orientação acadêmica, companheirismo e amizade desde o terceiro período da minha graduação.

Deixo também meus agradecimentos aos amigos - Vinicius, Breno, Zé Wesley, Telmo, Leo, Yan, Baia, Indio e tantos outros - adquiridos ao longo do curso. Graças a todas as reuniões de estudos e momentos de descontração que adentraram a madrugada, e algumas até viraram o dia, que cheguei ao final desse ciclo. Bem como, a minha família de EJC - Iluminados em Cristo, de forma especial a Jordanna e Rayane.

Ainda a todos que contribuíram de forma direta ou indireta que não foram citados, explicitamente, mas sem os quais não seria possível concluir essa etapa final.

“Do. Or do not. There is no try.”

Master Yoda

# Resumo

As doenças cardíacas são a principal causa de morte em todo o mundo, atingindo países desenvolvidos e subdesenvolvidos, e vem apresentando um crescimento contínuo e alarmante. Um diagnóstico precoce deste tipo de anomalia é fundamental para diminuição da mortalidade. A principal, e mais comum, forma de diagnóstico de doenças cardíacas é o Eletrocardiograma (ECG), sendo realizado em clínicas e laboratórios sob supervisão de um profissional da área. Consequentemente é fácil observar a necessidade de um diagnóstico preventivo com intuito de minimizar os altos índices de mortalidade desses tipos de doenças. Portanto, esse trabalho tem por objetivo o desenvolvimento de um programa inteligente, com fundamentos em *Machine Learning*, para facilitar o processo de identificação de doenças cardíacas a partir de sinais de eletrocardiogramas.

**Palavras-chave:** Doenças Cardíacas, Eletrocardiograma, Machine Learning.

# Abstract

Heart disease is the leading cause of death worldwide, reaching developed and underdeveloped countries, and it has been showing continued and alarming growth. An early diagnosis of this type of anomaly is essential to reduce mortality. The main, and most common, way of diagnosing heart disease is the Electrocardiogram (ECG), being performed in clinics and laboratories under the supervision of a professional in the area. Consequently, it is easy to observe the need for a preventive diagnosis in order to minimize the high mortality rates of these types of diseases. Therefore, the objective of this work is the development of an intelligent program, based on Machine Learning, to facilitate the identification of cardiac diseases from electrocardiograms.

**Keywords:** Heart Disease, Electrocardiogram, Machine Learning.



# Lista de ilustrações

Figura 1 – Mortes por ano devido a doenças cardiovasculares. . . . .	4
Figura 2 – Registro de ECG obtido por Einthoven. . . . .	6
Figura 3 – Ondas registradas em um ECG. . . . .	6
Figura 4 – Inteligência Artificial e seus subconjuntos. . . . .	8
Figura 5 – Tipos de <i>Machine Learning</i> . . . . .	10
Figura 6 – Exemplos de <i>Supervised Learning</i> . . . . .	10
Figura 7 – Fluxo da <i>Reinforcement Learning</i> . . . . .	11
Figura 8 – Exemplo de <i>Clustering</i> . . . . .	12
Figura 9 – Dois casos de aplicação do SVM. . . . .	13
Figura 10 – Classificação pelo método <i>Decision Tree Classifier</i> . . . . .	14
Figura 11 – Exemplo do método para $k = 3$ e $k = 6$ . . . . .	15
Figura 12 – Diagrama de blocos do <i>dataset</i> . . . . .	18
Figura 13 – Diagrama de blocos dos métodos . . . . .	19
Figura 14 – Quantidade de amostras pertencentes as classes. . . . .	20
Figura 15 – Gráfico de Correlação entre as Variáveis das amostras. . . . .	20
Figura 16 – Eficácia obtida pelo Método <i>K Nearest Neighbors Classifier</i> . . . . .	21
Figura 17 – Eficácia obtida pelo Método SVM. . . . .	21
Figura 18 – Eficácia obtida pelo Método <i>Decision Tree Classifier</i> . . . . .	22
Figura 19 – Eficácia obtida pelo Método <i>Random Forest Classifier</i> . . . . .	22
Figura 20 – Melhores resultados dos Métodos. . . . .	23
Figura 21 – Influência do Banco de Treino na Eficácia dos Métodos. . . . .	24

# Lista de tabelas

Tabela 1 – Características do <i>Dataset</i> . . . . .	18
Tabela 2 – Melhores Desempenhos Alcançados. . . . .	23
Tabela 3 – Dados de Eficácia ao variar o Tamanho do <i>Dataset</i> de Treino . . . . .	24

# Lista de abreviaturas e siglas

ECG	Eletrocardiograma
IA	Inteligência Artificial
IBM	International Business Machines Corporation
SBC	Sociedade Brasileira de Cardiologia
SOBRAC	Sociedade Brasileira de Arritmias Cardíacas
SVM	Support Vector Machine
RNA	Rede Neural Artificial

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	Objetivos	2
1.2	Organização do Trabalho	2
1.3	Metodologia	3
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>4</b>
2.1	Doenças Cardíacas	4
2.2	Eletrocardiogramas	5
2.3	<i>Machine Learning</i>	7
2.3.1	Contexto geral	7
2.3.2	Tipos de <i>Machine Learning</i>	10
2.3.3	Métodos Aplicados	12
2.3.3.1	<i>Support Vector Machine</i>	12
2.3.3.2	<i>Decision Tree Classifier</i>	13
2.3.3.3	<i>K Nearest Neighbors Classifier</i>	14
2.3.3.4	<i>Random Forest Classifier</i>	15
2.3.4	Outros Métodos	16
<b>3</b>	<b>DESENVOLVIMENTO DO SOFTWARE E RESULTADOS</b>	<b>17</b>
3.1	Implementação do <i>Software</i>	17
3.2	Resultados e Discussões	19
<b>4</b>	<b>CONCLUSÕES</b>	<b>25</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>26</b>

# 1 INTRODUÇÃO

As doenças cardiovasculares são a principal causa de morte no mundo [1]. De acordo com a Organização Mundial de Saúde, por ano mais de 300 mil pessoas no Brasil sofrem um infarto e até 2040 as mortes por doenças cardiovasculares devem aumentar em 250%. Fundamentado nos dados supracitados fica claro que o cenário atual, que diz respeito a mortes por doenças cardiovasculares, é no mínimo alarmante.

Diante deste cenário é posto um universo que diz respeito à arritmia cardíaca. A Sociedade Brasileira de Arritmias Cardíacas (SOBRAC) define essa anomalia como: “A arritmia cardíaca é uma alteração que ocorre na geração ou na condução do estímulo elétrico do coração e pode provocar modificações no ritmo cardíaco. Podendo ser benigna ou maligna”. Quando não identificada e tratada esse tipo de distúrbio pode causar paradas cardíacas, doenças no coração e a morte súbita. A sociedade brasileira apresenta uma alta incidência de arritmias cardíacas, principalmente na população idosa. Esse tipo de anomalia acomete mais de 20 milhões de pessoas e é culpado por mais de 320 mil mortes súbitas anualmente no Brasil [2].

O diagnóstico deste tipo de doença é feito, comumente, a partir de Eletrocardiogramas (ECG), que é a reprodução da atividade elétrica cardíaca a partir de eletrodos. E a realização desse exame ocorre em laboratórios ou clínicas sob supervisão de um profissional da área. Logo, a identificação precoce de uma doença cardiovascular, em especial de arritmias cardíacas, é de suma importância para prevenção de maiores danos a saúde da população brasileira. Porém, comumente, a sociedade desconhece ou até nunca realizou algum tipo de exame ou tratamento de prevenção a respeito destas. Esse fato pode ser justificado por diversos fatores, como: conhecimento por parte da população; falta de agilização do processo no Sistema Público de Saúde (SUS) e atuação em regiões longe dos centros urbanos.

Diversos algoritmos para identificação de doenças cardíacas a partir de ECG com *Machine Learning* já foram propostos, e realizadas comparações entre os diversos métodos e resultados obtidos [3]. Ainda a utilização desse tipo de programa para classificação e identificação de doenças cardíacas, com auxílio de grandes bancos de dados de ECG já é uma realidade [4]. Outros estudos abordam a temática com intuito de identificar esse tipo de anomalia em tempo real, implementando algoritmos com *Machine Learning*, para classificação dos sinais [5][6]. É válido ressaltar que, diante do cenário exposto, a linguagem de programação *Python* se mostra bastante promissora para realização desses projetos, tendo em vista a grande quantidade de bibliotecas e materiais, como livros [7], existentes na área.

É nesse contexto que o presente projeto se enquadra, buscando implementar um programa para identificação de doenças cardíacas, utilizando Machine Learning, com intuito de agilizar e facilitar o diagnóstico.

## 1.1 Objetivos

O objetivo geral do presente Trabalho de Conclusão de Curso é a implementação de um *software* para identificação de doenças cardíacas a partir de ECG, utilizando *Machine Learning*. É pretendido que o *software* seja capaz de identificar a existência ou não de anomalias em sinais de eletrocardiogramas de um banco de dados real.

De forma mais específica, tem-se por finalidade:

- Estudar sobre doenças cardíacas e Machine Learning, com ênfase na sua utilização para resolução de problemas de identificação e classificação;
- Obtenção de uma base de dados de eletrocardiogramas para treinamento e validação do programa concebido;
- Implementar um *software* capaz de identificar a presença de doenças cardíacas a partir de sinais de ECG;
- Verificar a eficácia do algoritmo implementado, avaliando os erros e acertos no processo identificação.

## 1.2 Organização do Trabalho

Em um primeiro momento, é feita uma breve introdução do trabalho proposto, bem como expostos os objetivos almejados e metodologia aplicada. Doravante, o relatório é organizado da seguinte forma:

- Capítulo 2: Fundamentação Teórica. Neste capítulo são apresentados os conceitos relevantes para o entendimento do trabalho;
- Capítulo 3: Desenvolvimento. Nele são expostos a metodologia aplicada, o projeto do *software* proposto e os resultados alcançados;
- Capítulo 4: Conclusões. São apresentadas as conclusões obtidas e sugestões para trabalhos futuros na área.

Por fim, são postas as referências do trabalho em questão.

## 1.3 Metodologia

A metodologia utilizada para realização dos objetivos deste projeto de TCC consistiu, em um primeiro momento, em uma revisão bibliográfica - por meio de artigos científicos, dissertações, teses e livros - para identificação de doenças cardíacas a partir de eletrocardiogramas e no processo de implementação de programas com Machine Learning. Com intuito de identificar e determinar a técnica mais adequada para execução do projeto.

Em seguida, foi realizada uma busca por uma base de dados, para que seja possível a validação do projeto, (*dataset*) que atenda as especificações desejadas, sendo: Possuir amostras de sinais de eletrocardiogramas normais e com anomalias. Dessa forma, foi identificado e utilizado o *The PTB Diagnostic ECG Database* <sup>1</sup>.

De posse da base de dados, os esforços foram voltados na implementação do programa, feito com a linguagem de programação *Python* na plataforma de desenvolvimento *PyCharm*. Em que, o projeto utiliza quatro métodos - *K Nearest Neighbors Classifier*, *Support Vector Machine*, *Decision Tree Classifier* e *Random Forest Classifier* - de aprendizagem de máquina para identificação de anomalias cardíacas a partir de um eletrocardiograma.

Por fim, os resultados obtidos para cada método foram analisados e discutidos, bem como a influência do tamanho do banco de dados de treino dos métodos na eficácia do projeto.

---

<sup>1</sup> Disponível em: <https://physionet.org/physiobank/database/ptbdb/>

## 2 FUNDAMENTAÇÃO TEÓRICA

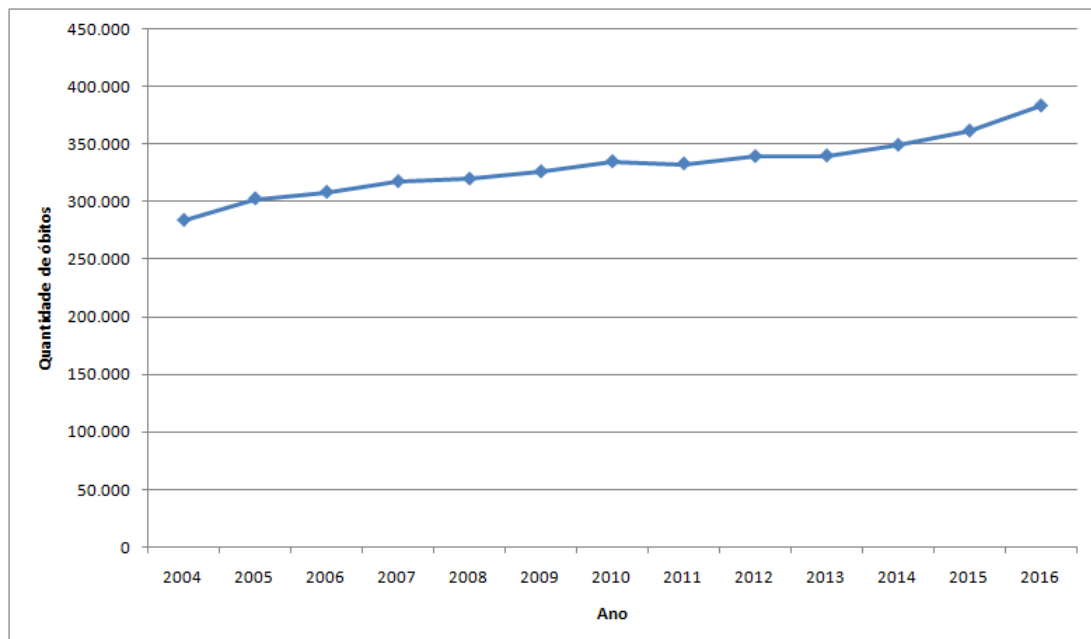
Neste Capítulo serão descritos os principais fundamentos necessários para o bom entendimento do trabalho em questão. Dessa forma, são descritos os conceitos e contexto de doenças cardíacas, eletrocardiogramas e *Machine Learning*.

### 2.1 Doenças Cardíacas

As doenças cardíacas dizem respeito a qualquer tipo de anomalia, podendo ser crônica, que afeta alguma parte do sistema cardíaco humano. Conforme supracitado, os índices de mortalidade por doenças cardíacas vem atingindo valores alarmantes e esse cenário vem se propagando a décadas.

Como posto na Fig. 1, o número de mortes apresenta uma tendência consolidada de crescimento na última década, de acordo com a Sociedade Brasileira de Cardiologia (SBC) [8]. Esse cenário pode ser justificado por diversos fatores de risco como: Hipertensão, Colesterol, Tabagismo, Estresse, Sedentarismo e Diabetes.

Figura 1 – Mortes por ano devido a doenças cardiovasculares.



Fonte: SBC, 2019.

Ainda, a SBC, dispõe de um cardiômetro que contabiliza as mortes por doenças vasculares no Brasil em tempo real, sendo do início de 2019 até o dia 26/06/2019 cerca de 191 mil mortes. De acordo com a SBC [8]:



"São mais de 1 mil mortes por dia, cerca de 43 por hora, 1 morte a cada 1,5 minutos (90 segundos). As doenças cardiovasculares causam o dobro de mortes que aquelas devidas a todos os tipos de câncer juntos, 2,3 vezes mais que as todas as causas externas (acidentes e violência), 3 vezes mais que as doenças respiratórias e 6,5 vezes mais que todas as infecções incluindo a AIDS. A Sociedade Brasileira de Cardiologia estima que, ao final deste ano, quase 400 mil cidadãos brasileiros morrerão por doenças do coração e da circulação. Muitas dessas mortes poderiam ser evitadas ou postergadas com cuidados preventivos e medidas terapêuticas."

Portanto a identificação precoce desse tipo de doença é de suma importância, visto que um tratamento preventivo diminui consideravelmente os riscos de vida. Dentre os diversos tipos de exames que avaliam as condições do paciente, no que diz respeito ao sistema cardíaco, como: eletrocardiograma, ecocardiograma, teste ergométrico, cateterismo cardíaco, dentre outros. Em que, o mais usual é o primeiro citado que será detalhado na próxima subseção.

## 2.2 Eletrocardiogramas

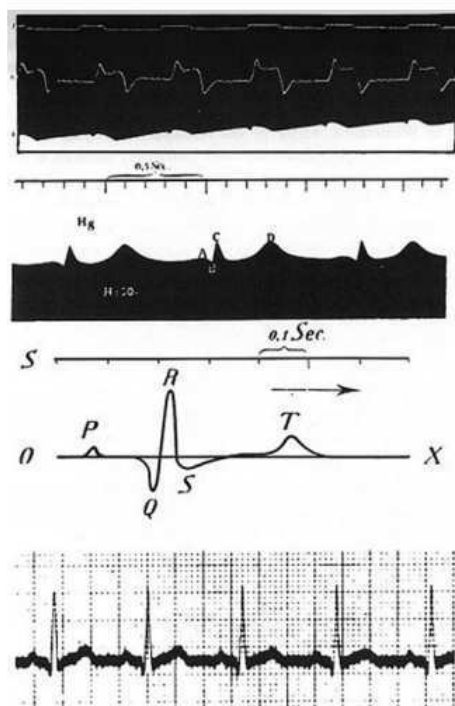
O Eletrocardiograma (ECG) consiste em um exame que reproduz a atividade elétrica do coração durante os movimentos cardíacos, em que estes são consequência da variação na concentração citosólica de íons de cálcio controlada pelo miocárdio. Dessa forma, é possível identificar padrões referentes a atividade elétrica cardíaca no corpo humano, referentes as contrações e relaxamentos do coração causadas pelo miocárdio.

O primeiro registro de um eletrocardiograma, visto na Fig. 2, data no ano de 1902, pelo holandês Willem Einthoven [9], em que seu equipamento pesava aproximadamente 270 kg e necessitava de cinco pessoas para ser operado. Vale ressaltar, que a devido as dimensões do equipamento e distância considerável do hospital mais próximo, Einthoven conectou seu dispositivo a linha telefônica, transmitindo os sinais dos pacientes no hospital para seu laboratório - surgindo a telemedicina.

Diante deste cenário, diversas doenças cardíacas como as arritmias foram compreendidas graças a nova abordagem desenvolvida. Posteriormente, Thomas Lewis visitou o laboratório de Einthoven no ano de 1909 e continuou a pesquisa na área buscando aprimorar os estudos até então existentes, chegando a ser considerado o sucessor de Einthoven.

Ainda, em 1931 Frank N. Wilson, provou matematicamente que é possível o registro da atividade elétrica do coração em qualquer parte do corpo e introduziu as derivações unipolares que permitiu a padronização do sistema com 12 derivações, que é empregado desde então.

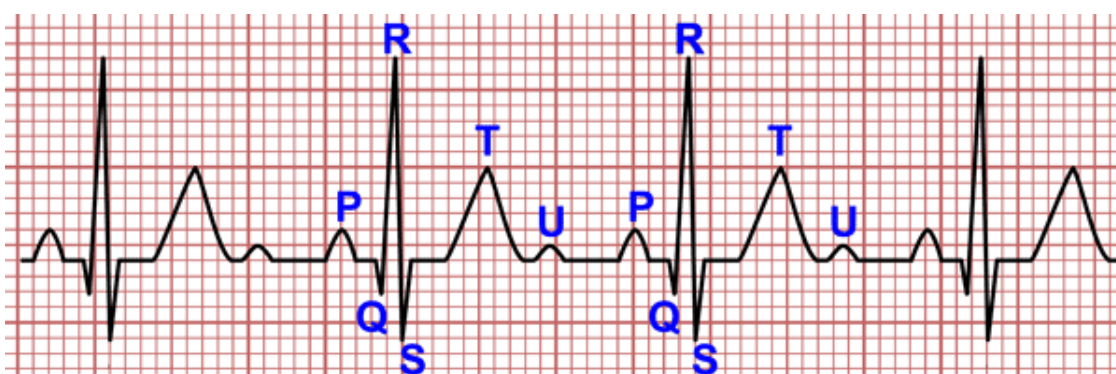
Figura 2 – Registro de ECG obtido por Einthoven.



Fonte: Fye WB, 1994.

Com os avanços tecnológicos os instrumentos para aferição da atividade cardíaca tiveram suas dimensões reduzidas e se tornaram mais leves. Na Fig. 3 é visualizada a representação atual de um ECG e os principais pontos a serem observados no exame.

Figura 3 – Ondas registradas em um ECG.



Fonte: <https://pt.my-ekg.com/generalidades-ecg/ondas-eletrocardiograma.html>, 2019.

Abaixo são descritas as ondas do ECG [10].

- Onda P: É a primeira onda do ciclo e corresponde a despolarização dos átrios, a parte inicial é referente ao átrio direito e a final ao esquerdo. Tem duração inferior a 0,10 s e apresenta uma tensão máxima de 0,25 mV;

- Complexo QRS: Formado pelas ondas Q, R e S, e representa a despolarização dos ventrículos. Tem duração entre 0,06 s e 0,10 s.
  - Onda Q: É a primeira onda negativa do complexo
  - Onda R: É a primeira onda positiva do complexo QRS, podendo ou não ser precedida por uma onda Q;
  - Onda S: É a onda negativa subsequente da onda R.
- Onda T: Retrata a repolarização dos ventrículos, sendo normalmente positiva e assimétrica. Possui duração inferior a 0,60 s e uma tensão máxima de 0.5 mV;
- Onda U: É uma onda positiva de baixa tensão, com relação as demais. Ocorre após a onda T e é comumente desconsiderada nos exames de eletrocardiograma, visto que sua origem ainda é objeto de estudo.

Logo, qualquer registro de atividades cardíacas que venham a divergir do comportamento supracitado irá corresponder a alguma anomalia consequente de doenças cardíacas. Como por exemplo:

- Uma onda P com amplitude e/ou duração majoradas é consequência de uma sobrecarga nos átrios;
- Um complexo QRS é totalmente negativo, pode se diagnosticar uma necrose no miocárdio;
- Alterações na onda T podem significar uma cardiopatia isquêmica;
- Ondas U com altas amplitudes podem ser causadas por uma hipocalemia moderada ou grave.

Dessa forma, de posse de todas as informações postas, fica clara a fundamental importância do ECG no diagnóstico de doenças cardíacas, sendo um exame comumente utilizado por médicos cardiologistas. A facilidade que esse instrumento trouxe no diagnóstico de doenças cardíacas é facilmente observada.

## 2.3 *Machine Learning*

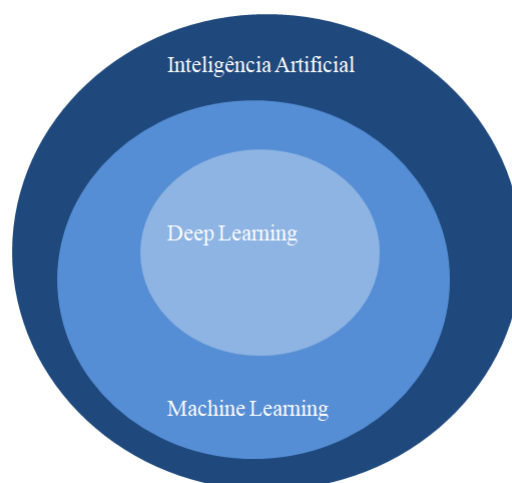
### 2.3.1 Contexto geral

*Machine Learning*, ou aprendizagem de máquina, é um subcampo do desenvolvimento de Inteligência Artificial (IA). Essa vertente surgiu a partir do desenvolvimento da ciência da computação no que diz respeito a reconhecimento de padrões e da própria IA.

Um aspecto importante a ser destacado é a diferenciação entre IA, *machine learning* e *deep learning*. Em que, todas essas definições são vinculadas, porém são conceitos distintos.

Na Fig. 4, fica claro a diferenciação entre as três definições supracitadas. A IA é a capacidade de uma máquina simular uma inteligência humana. Já *machine learning* é um subconjunto da IA que permite os computadores realizarem um processo de aprendizagem sem serem programados claramente para este objetivo, podendo se adaptar a novos cenários e dados. E *deep learning* é um subconjunto do *machine learning* que diz respeito a redes neurais complexas.

Figura 4 – Inteligência Artificial e seus subconjuntos.



Fonte: Própria do Autor, 2019.

O nascimento do *machine learning* ocorre no ano de 1950, por Alan Turing [11]. Ele criou o conhecido "Teste de Turing" que consiste em um diálogo de texto entre duas pessoas e uma máquina, em que uma pessoa se comporta como "juiz" e deve identificar qual dos dois outros participantes é a máquina. Caso a máquina não seja identificada se diz que passou no teste de Turing.

Outros marcos de suma importância na história da evolução do *Machine Learning* são postos abaixo [12].

**1952:** Arthur Samuel implementa um programa com capacidade de aprendizagem. O *software* jogava damas e a cada partida melhorava seu desempenho;

**1958:** Frank Rosenblatt projeta a primeira rede neural artificial - intitulada Perceptron;

**1967:** É implementado o algoritmo de "Nearest Neighbor", sendo considerado um marco para o reconhecimento de padrões por computadores;

- 1997:** O computador "*Deep Blue*", da *International Business Machines Corporation* (IBM), derrota o campeão mundial de xadrez Gary Kaspárov;
- 2006:** Geoffrey Hinton utiliza o termo *Deep Learning* para explicar a respeito de redes neurais profundas;
- 2011:** O computador "Watson" da IBM vence a competição Jeopardy, que consiste em um jogo de perguntas orais;
- 2014:** O *Facebook* desenvolve um algoritmo de reconhecimento facial, intitulado *Deep-Face*;
- 2015:** A Amazon lança sua própria plataforma de *machine learning*.

Diante do histórico posto fica claro que a utilização de *machine learning* vêm atraindo grandes atenções para seu desenvolvimento e aplicações comerciais [13], por exemplo:

- Serviços financeiros: identificando insights e prevenindo fraudes;
- Governo: análise de grandes quantidades de dados;
- Operadoras de saúde: ajudando profissionais da área a identificar anomalias, aperfeiçoando o diagnóstico;
- Marketing e vendas: recomendando produtos e serviços com base no histórico de compras e buscas;
- Transportes: analisando os dados de tráfego com intuito de gerar rotas mais eficientes.

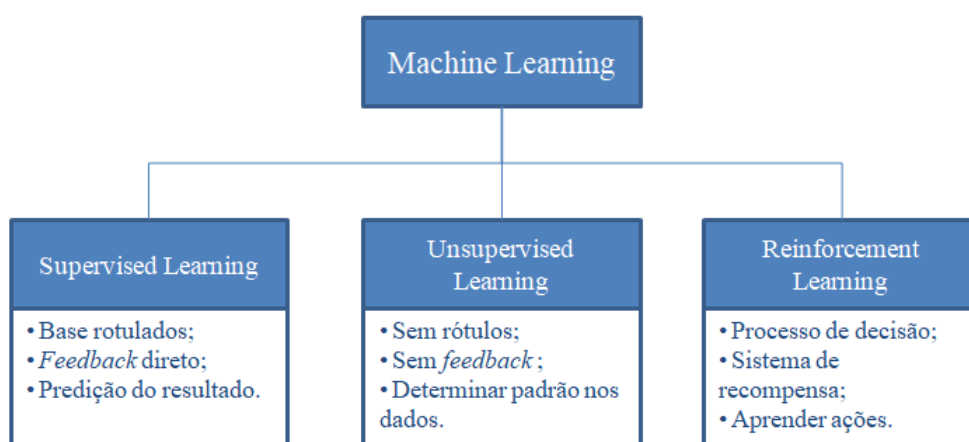
Exemplos de empresas conhecidas que utilizam *machine learning* são: Youtube, Netflix, Google, Amazon, Facebook e Instagram. De acordo com Raschka e Mirjalili [7]:

"The application and science of algorithms that make sense of data, is the most exciting field of all the computer sciences! We are living in an age where data comes in abundance; using self-learning algorithms from the field of machine learning, we can turn this data into knowledge. Thanks to the many powerful open source libraries that have been developed in recent years, there has probably never been a better time to break into the machine learning field and learn how to utilize powerful algorithms to spot patterns in data and make predictions about future events."

### 2.3.2 Tipos de *Machine Learning*

Ao tratar de *Machine Learning*, é de suma importância salientar os três tipos existentes, visualizados na Fig 5.

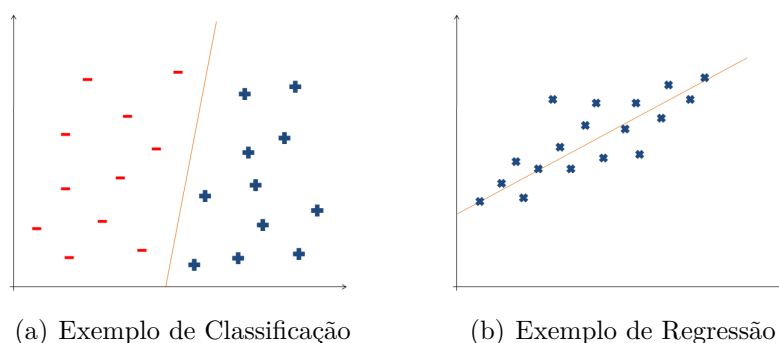
Figura 5 – Tipos de *Machine Learning*.



Fonte: Raschka e Mirjalili, 2017.

**Supervised Learning:** O principal objetivo da aprendizagem supervisionada é conceber um modelo, a partir de uma base de dados rotulada, que permita realizar previsões a respeito de dados não vistos ou futuros. Logo, o termo supervisionado compreende as amostras em que é conhecida a saída. As duas principais subcategorias de *supervised learning*, são: Classificação - quando a base de dados é discreta e o intuito é classificar amostras futuras, Fig. 7(a) - e Regressão - para valores contínuos, Fig. 7(b). Um exemplo para aplicação desse tipo de aprendizagem é um sistema de spam para e-mail, em que o programa pode classificar novos e-mails como spam ou não, a partir de uma série de classificações anteriores feitas pelo usuário.

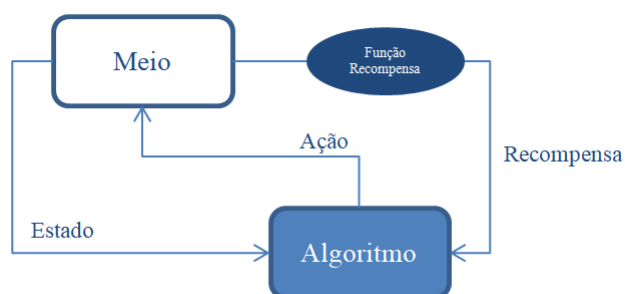
Figura 6 – Exemplos de *Supervised Learning*.



Fonte: Raschka e Mirjalili, 2017.

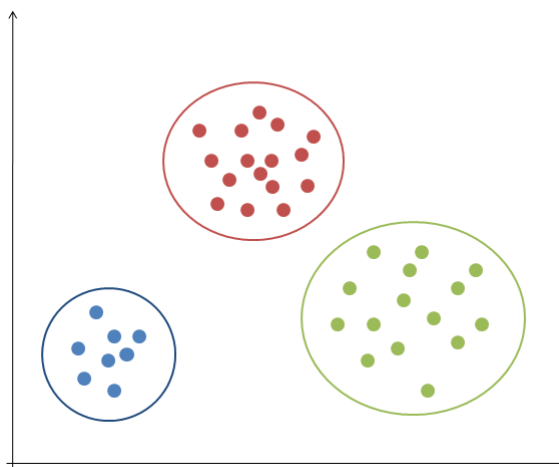
**Reinforcement Learning:** Nesse tipo de aprendizagem o algoritmo irá melhorar seu desempenho com base em suas interações com o meio. Para isso um sistema de recompensas, ou um sinal de prêmio, irá sinalizar quando uma ação do programa é a correta ou não. Vale ressaltar que o *feedback* fornecido pela recompensa pode determinar como certa uma ação que talvez não fosse a ótima para o caso em questão. Na Fig. 7 é visualizado o conceito desse tipo de aprendizagem. Logo, uma abordagem em que uma função de recompensa seja capaz de majorar o prêmio, com base nas ações, é bastante promissor. Um exemplo clássico para esse tipo de aprendizagem é um modelo para jogar xadrez, em que a partir da ampla gama de movimentos possíveis no tabuleiro a recompensa é definida pela vitória ou derrota no fim do jogo.

Figura 7 – Fluxo da *Reinforcement Learning*.



Fonte: Raschka e Mirjalili, 2017.

**Unsupervised Learning:** Na aprendizagem não supervisionada o programa irá lidar com uma base de dados desconhecida, sem rótulos ou estrutura conhecida. Dessa forma, esse tipo é capaz de extrair informações a partir da estrutura da base de dados, sem a orientação de rótulos ou uma função recompensa. A principal técnica para a aprendizagem não supervisionada é o *Clustering*, que consiste em uma técnica de divisão do banco de dados em subgrupos, ou "*clusters*", a partir da similaridade dos dados.

Figura 8 – Exemplo de *Clustering*.

Fonte: Raschka e Mirjalili, 2017.

### 2.3.3 Métodos Aplicados

Nas próximas seções são expostos, de forma sucinta, os conceitos referentes aos métodos de *machine learning* utilizados no presente trabalho.

#### 2.3.3.1 *Support Vector Machine*

O método *Support Vector Machine* (SVM) é uma técnica de *Supervised Learning* [14]. Seu fundamento é a criação de um hiperplano de separação para classificação das amostras, em que esse hiperplano é obtido a partir do *dataset* de treino. Portanto, o hiperplano irá dividir o universo em duas partes pertencentes a cada classe.

Na Fig. 9, são observados dois casos de divisão para um *dataset*. No caso 1 é traçada uma reta, de comportamento simples, porém apresenta tolerância a erros na classificação. Já no caso 2, um polinômio mais complexo é utilizado, a divisão é realizada sem nenhum erro.

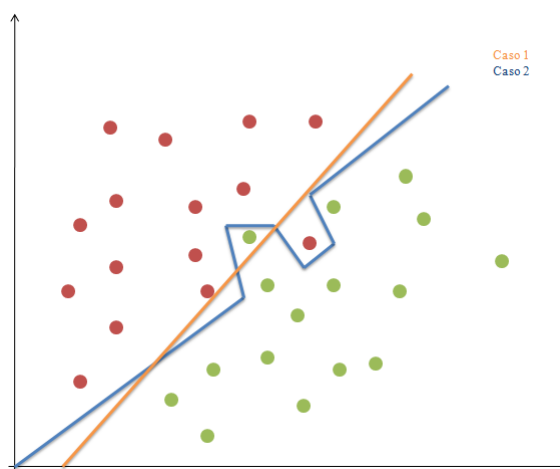
Logo, o principal ponto a ser analisado nesse método é o *trade off* a ser considerado entre taxa de acerto e tempo de execução, visto que uma um ótimo desempenho requer curvas mais complexas e conseqüentemente um esforço computacional superior.

No que diz respeito a implementação desse método, os seguintes parâmetros devem ser definidos:

- Kernel: Diz respeito a característica do polinômio a ser utilizado na concepção do hiperplano;



Figura 9 – Dois casos de aplicação do SVM.



Fonte: Própria do Autor, 2019.

- Regularization: É referente a otimização do algoritmo quanto a taxa de acertos, um alto valor desse parâmetro garante uma menor tolerância a erros;
- Gamma: Esse parâmetro define a influência dos pontos no polinômio de divisão, um baixo *Gamma* considera os pontos distantes da margem de divisão para gerar o polinômio e um alto considera apenas os pontos mais próximos;
- Margin: Configura a distância do hiperplano de separação para os pontos mais próximos a fronteira, uma boa *Margim* garante uma distância equidistante entre os dados pertencentes as duas classes.

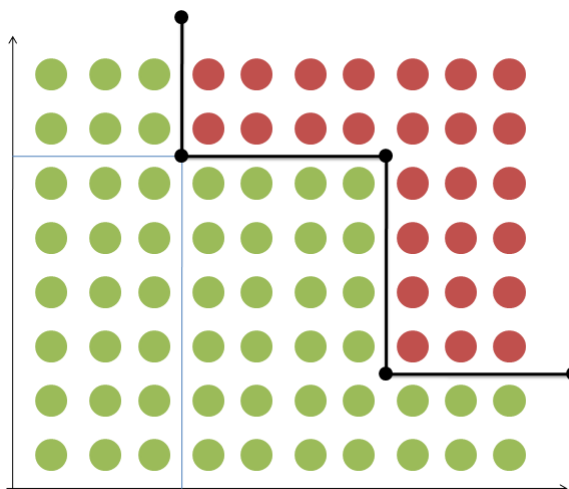
### 2.3.3.2 Decision Tree Classifier

O método *Decision Tree Classifier* é uma técnica de *Supervised Learning* [15]. Esse método se baseia na divisão das amostras em regiões, criadas por diversas linhas de separação.

Na Fig. 10, fica claro que para realizar a divisão entre as classes é necessário mais de uma linha. Logo são traçadas  $n$  linhas, com intuito de criar uma fronteira entre as amostras.

Assim como no SVM, nesse método também irá existir um *trade off* para sua implementação, em que o algoritmo pode criar diversas linhas de fronteira conseguindo uma divisão pura entre as classes, porém com um elevado esforço computacional. Bem como, possui os seguintes parâmetros:

- Impureza: Diz respeito a tolerância a erros de classificação, uma alta impureza proporciona um desempenho rápido, porém uma baixa precisão;

Figura 10 – Classificação pelo método *Decision Tree Classifier*.

Fonte: Própria do Autor, 2019.

- Entropia: É o nível de aleatoriedade dos elementos;
- Ganho de informação: É referente ao ganho obtido ao aplicar um recurso para divisão do conjunto, permitindo comparar diferentes abordagens;
- Número máximo de amostras: Define a quantidade máxima de amostras por fronteira;

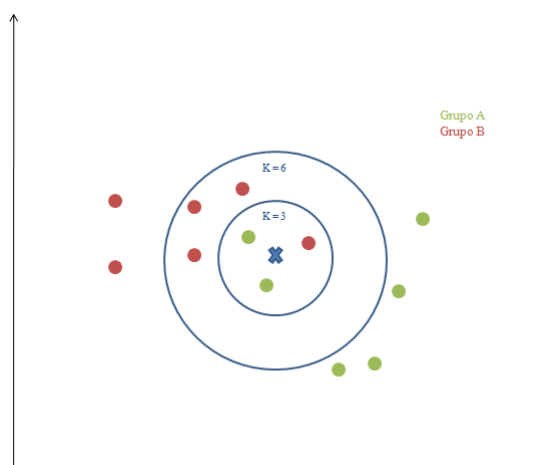
### 2.3.3.3 *K Nearest Neighbors Classifier*

O método *K Nearest Neighbors Classifier* é uma técnica de *Supervised Learning* [16]. Sua lógica de classificação se baseia na proximidade da amostra com relação aos seus vizinhos, em que o objeto é atribuído ao grupo mais comum entre seus  $k$  vizinhos mais próximos. Na Fig. 11 por exemplo, é visualizado que para  $k = 3$  a amostra irá pertencer ao grupo A, enquanto para  $k = 6$  será classificado no grupo B.

De forma sucinta, as amostras serão classificadas de acordo com o grupo pertencente a maioria de seus vizinhos, em que a quantidade de vizinhos utilizada para classificação é definida pelo parâmetro  $k$ .

A seleção dos vizinhos a serem considerados pode ser feita pelos seguintes métodos:

**Força Bruta:** O programa irá calcular a distância, de forma vetorial, da amostra para todos os pontos do conjunto e determinar o mais próximo. A complexidade do tempo de execução irá então ser calculada como  $D * N^2$ , em que  $N$  é o número de amostras em  $D$  dimensões. Portanto, quanto maior o *dataset* menor o desempenho dessa abordagem;

Figura 11 – Exemplo do método para  $k = 3$  e  $k = 6$ .

Fonte: Própria do Autor, 2019.

**K-D Tree:** A ideia básica desse método pode ser exemplificada da seguinte forma: se o ponto A estiver muito distante do ponto B e o ponto B estiver muito próximo do ponto C, sabemos que os pontos A e C estão muito distantes, sem precisar calcular explicitamente sua distância. Logo o método não irá calcular a distância para todos os pontos. A complexidade do tempo de execução irá então ser calculada como  $D * N * \log(N)$ , em que  $N$  é o número de amostras em  $D$  dimensões. Portanto, essa abordagem claramente apresenta um rendimento maior, quando comparado a força bruta, porém a influência da dimensão do *dataset* é mais significativa;

**Ball Tree:** Essa abordagem cria, no espaço dimensional do *dataset*, hipersferas alinhadas para determinação dos vizinhos. Dessa forma, a complexidade do tempo de execução é definido como  $D * \log(N)$ , em que  $N$  é o número de amostras em  $D$  dimensões.

Logo, no que diz respeito a implementação desse método, existem dois principais parâmetros a serem definidos. O primeiro é o número de vizinhos ( $k$ ) a ser considerado e seguido pelo método de abordagem de seleção dos vizinhos.

#### 2.3.3.4 *Random Forest Classifier*

O método *Random Forest Classifier* é uma técnica que combina mais de um algoritmo para classificação [17]. Em um primeiro momento, ele cria um conjunto de árvores de decisão, proveniente de um banco de treino aleatório. Posteriormente, analisa as respostas obtidas para diferentes árvores de decisão para então classificar o objeto.

Uma forma clara para entendimento pode ser feita considerando um banco de dados de treinamento  $[X1 X2 X3 X4]$ , com seus respectivos *targets*  $[T1 T2 T3 T4]$ . O

algoritmo irá criar três árvores de decisão, considerando subconjuntos do conjunto de treinamento, p.e.:  $[X1 X2 X3]$ ,  $[X1 X2 X4]$  e  $[X2 X3 X4]$ . Então o programa irá selecionar o grupo da amostra, com base na resposta mais recorrente dentre as árvores de decisão.

Uma otimização desse método é adicionar um sistema de relevância para as respostas obtidas pelas árvores, em que árvores com maior taxa de acerto terão um peso maior na decisão final, obtendo assim resultados mais precisos. Portanto, os principais parâmetros para concepção desse método são: O número de árvores utilizadas pelo algoritmo, os critérios do método original de árvore de decisão e o número mínimo do conjunto a ser utilizado.

### 2.3.4 Outros Métodos

Além dos quatro métodos supracitados, atualmente existem diversos algoritmos de *Machine Learning* já consolidados, como:

- Métodos de *Supervised Learning*:
  - *Naive Bayes Classification*: Utiliza conceitos probabilísticos simples com base no teorema com forte independência entre características;
  - Regressão Linear: Consiste na formulação de uma equação linear, que garanta o menor erro possível entre as amostras;
  - Regressão Logística: Funciona de forma semelhante a regressão linear, porém utiliza um modelo binomial para concepção do modelo;
- Métodos de *Unsupervised Learning*:
  - *Clustering*: Agrupa as amostras de acordo com as suas semelhanças, podendo ter o algoritmo fundamentado na densidade ou disposição, por exemplo.

Ainda, outra abordagem bastante eficaz e complexa são as Redes Neurais Artificiais (RNA). Estas são modelos matemáticos inspirados na estrutura neural de organismos inteligentes, dessa forma, esses algoritmos são capazes de aprender com base em experiências e se tornarem mais eficientes.

Uma RNA é composta em diversas unidades de processamento que tem seu funcionamento simplificado e são conectadas por canais de comunicação que determinam sua influência (ou peso). Logo, cada unidade trabalha em uma parte local do modelo e juntas definem o comportamento e interações de toda rede neural artificial.

## 3 DESENVOLVIMENTO DO SOFTWARE E RESULTADOS

As seções a seguir abordam a implementação do *software*, os resultados alcançados e uma discussão a respeito dos mesmos.

### 3.1 Implementação do *Software*

A revisão bibliográfica, posta na fundamentação teórica, teve por principal intuito identificar métodos de classificação para emprego no projeto, e os seguintes se destacaram: *K Nearest Neighbors Classifier*, *Support Vector Machine*, *Decision Tree Classifier* e *Random Forest Classifier*. Esse destaque é justificado pela presença de materiais *open source*, como bibliotecas e funções já implementadas, referentes a esses métodos e um baixo esforço computacional, quando comparados a outras técnicas como redes neurais.

Logo, com os métodos selecionados, os esforços foram voltados na implementação do programa. Nessa etapa, o primeiro ponto a ser definido para implementação do projeto foi a definição da linguagem de programação a ser utilizada e, diante da alta complexidade dos métodos a linguagem *Python* se mostrou altamente propícia. Essa escolha é justificada em dois fatos, sendo: A facilidade proporcionada pela linguagem e o grande número de bibliotecas *open source* já implementadas para *machine learning*.

Ainda o projeto foi concebido na plataforma de desenvolvimento *PyCharm*<sup>1</sup> que se destaca, quando comparada as demais, pelo assistente *Get Smart* que proporciona diversas facilidades na implementação do código. Como o reconhecimento de bibliotecas e inclusão destas ao projeto partir da chamada de funções pertencentes a estas.

Com todo ambiente de desenvolvimento do projeto definido, a busca por um *dataset*, que atenda as especificações do projeto, foi iniciada. Logo, o *The PTB Diagnostic ECG Database* [18] foi identificado e na Tabela 1 são postas as principais características dessa base de dados.

Vale ressaltar que o *dataset* utilizado possui diversas classes de doenças, entretanto devido o baixo número de amostras para as classes era inviabilizado o treinamento dos métodos de forma confiável. Ainda, o *input* do *dataset* em cada método será uma onda completa de ECG, desde a onda P até a U, amostrada discretamente na frequência de 125 Hz, posta na Tabela 1.

Para implementação do código foram utilizadas as seguintes bibliotecas *open source*:

---

<sup>1</sup> Disponível em: <https://www.jetbrains.com/pycharm/>

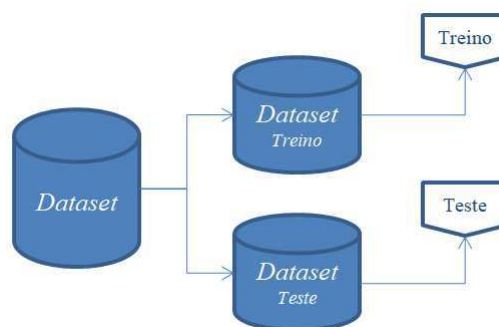
Tabela 1 – Características do *Dataset*.

Conteúdo do Banco de Dados	
Número de pacientes	294
Número de amostras	14552
Número de categorias	2
Frequência de amostragem	125 Hz

Fonte: Physionet's PTB Diagnostic Database, 2016

- Numpy: Trabalhar com *arrays*;
- Matplotlib: Elaboração e *plot* dos gráficos;
- Dataset: Manipulação e obtenção de informações sobre o banco de dados;
- *Warnings*: Ignorar os sinais de aviso gerados pelo código;
- Pandas: Tratar os arquivos no formato *.csv*;
- Scikit-learn: Importar os métodos de *machine learning*.

Após realizar as chamadas das bibliotecas supracitadas é feito o particionamento do *dataset* em dois, para que seja possível realizar o treinamento e doravante feito o teste para validação do método com dados distintos dos que foram utilizados no treino. Na Fig. 12 é visualizado o particionamento do *dataset*

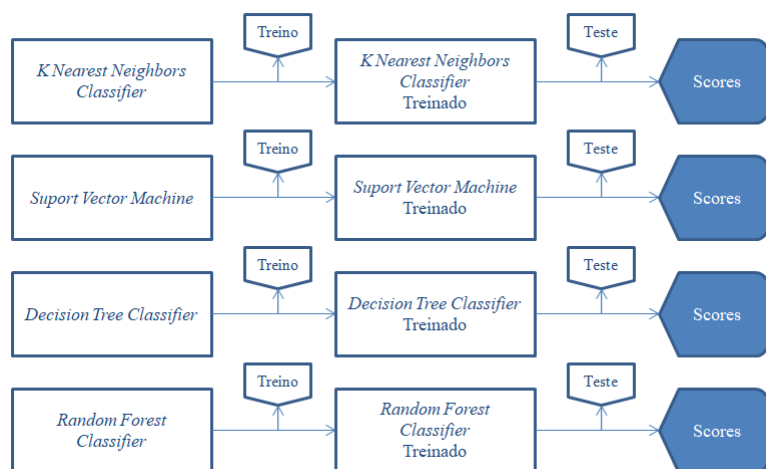
Figura 12 – Diagrama de blocos do *dataset*.

Fonte: Própria do Autor, 2019.

Posteriormente, os métodos são chamados de forma separada para treinamento e feito o teste com os demais dados. A Fig. 13 exemplifica esse processo.

No método *K Nearest Neighbors Classifier* foi configurado o modo *default* para abordagem de seleção dos vizinhos. Já o número de vizinhos *k* é definido por um *array*

Figura 13 – Diagrama de blocos dos métodos



Fonte: Própria do Autor, 2019.

que varia de 1 a 20, com intuito de comparar a eficácia do método para diferentes valores de  $k$ .

No SVM o parâmetro definido como variável foi o *kernel*, sendo: linear, poly, rbf e sigmoid. Enquanto os demais parâmetros configurados no modo *default* do método. Para a técnica *Decision Tree Classifier*, foram comparados os *scores* obtidos para diferentes valores do número máximo de amostras. Já no *Random Forest Classifier*, o número de estimadores foi definido por um *array* com os valores de: [10, 100, 200, 500, 1000].

Ainda, foram selecionados para cada método, os que obtiveram maior *score* ao variar os parâmetros de configuração, e comparados seus rendimentos ao modificar o tamanho do banco de dados de treinamento. E então, analisar a influência do *dataset* de treino no desempenho dos métodos.

Vale ressaltar, que devido a normalização natural do *dataset* não foi utilizado *feature scaling* nas amostras. Visto que, sua aplicação não traria ganhos nos *scores* dos métodos e iria majorar o custo computacional do projeto.

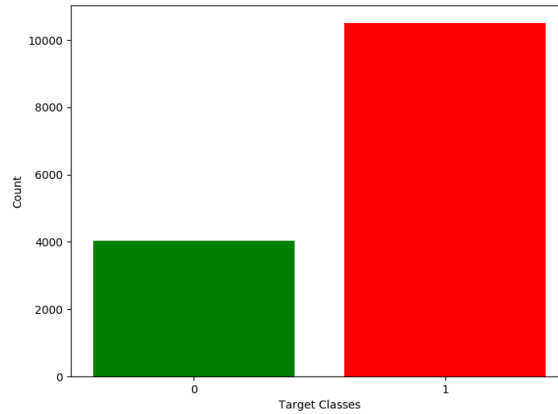
O código implementado é disponibilizado no Github, pelo link: [https://github.com/Thiago83/TCC\\_ECG](https://github.com/Thiago83/TCC_ECG).

## 3.2 Resultados e Discussões

No Gráfico da Fig. 14 é visualizado a quantidade de amostras pertencentes as duas classes, sendo: *target* = 1, referente a amostras com anomalias, e *target* = 0, amostras normais. Logo, fica claro que o *dataset* possui uma quantidade superior de amostrar com anomalias. Ainda o grande número de dados é um fator positivo para treinamento dos

métodos.

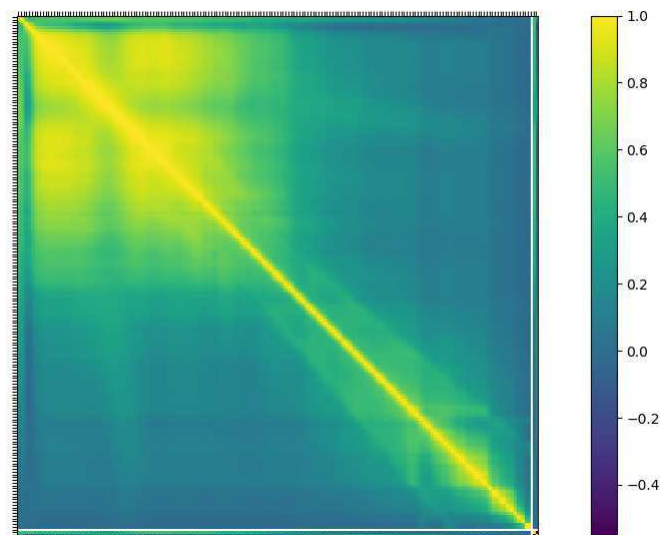
Figura 14 – Quantidade de amostras pertencentes as classes.



Fonte: Própria do Autor, 2019.

Outra análise do *dataset* utilizado é a respeito da correlação existente entre as variáveis de cada amostra. Logo, no Gráfico da Fig. 15 é observado que existe uma alta correlação entre as ondas iniciais do ECG, nele são postas as correlações ponto a ponto de todas as amostras do *dataset*. Esse fator tem uma influência positiva na eficácia dos métodos, pois evidencia a existência de um padrão no início das amostras.

Figura 15 – Gráfico de Correlação entre as Varáveis das amostras.



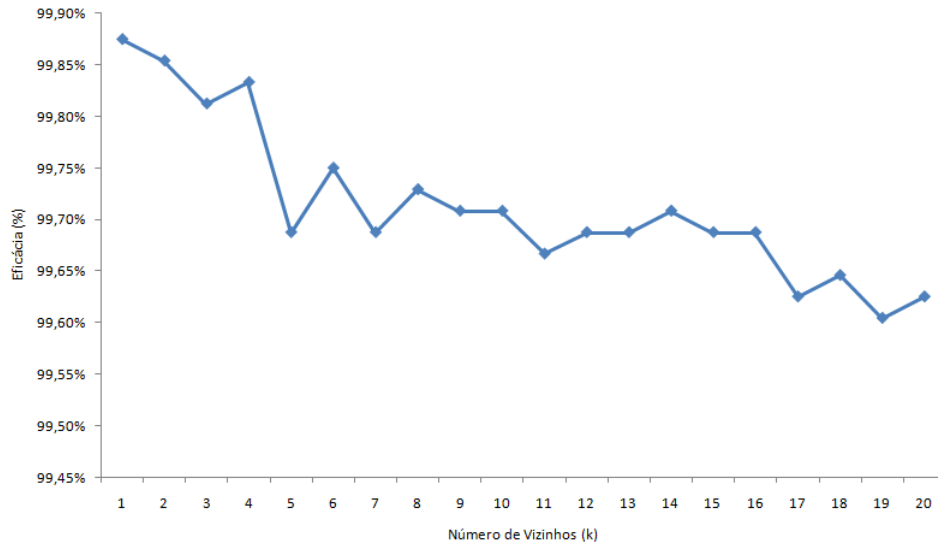
Fonte: Própria do Autor, 2019.

Logo, de posse do *dataset* cada método foi implementado variando um de seus parâmetros com intuito de obter uma eficácia ótima para cada método, como supracitado. Nos Gráficos da Fig. 16 e da Fig. 17 são postos os *scores obtidos* para os métodos *K Nearest*



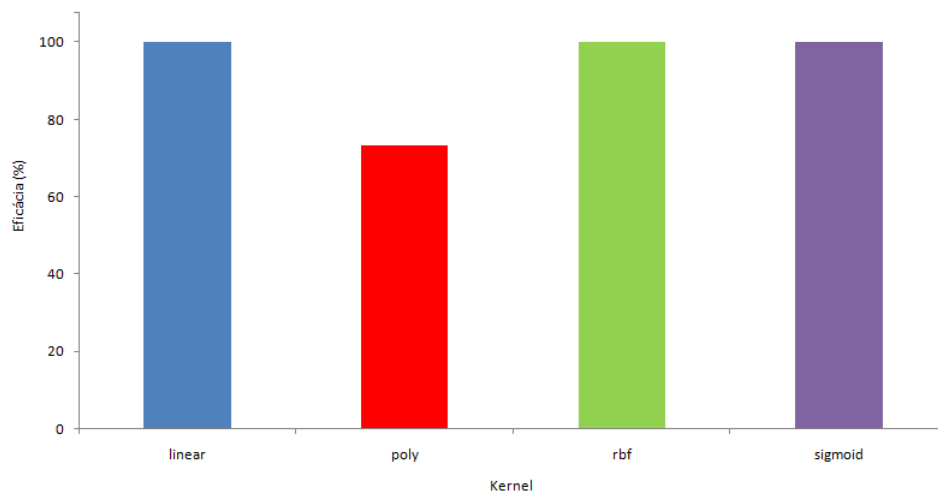
*Neighbors Classifier*, com  $k = [1 \dots 20]$ , e *Support Vector Machine*, com o *kernel* = [linear, poly, rbf, sigmoid], respectivamente.

Figura 16 – Eficácia obtida pelo Método *K Nearest Neighbors Classifier*.



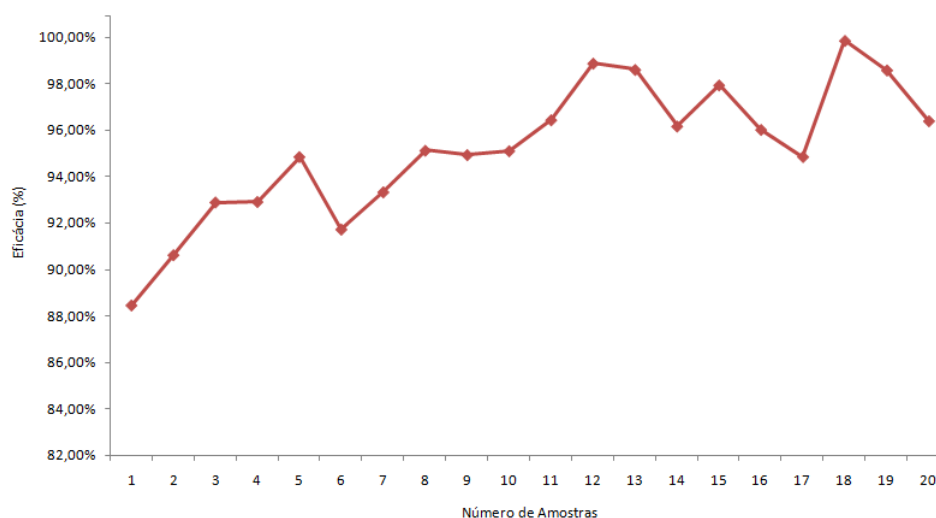
Fonte: Própria do Autor, 2019.

Figura 17 – Eficácia obtida pelo Método SVM.

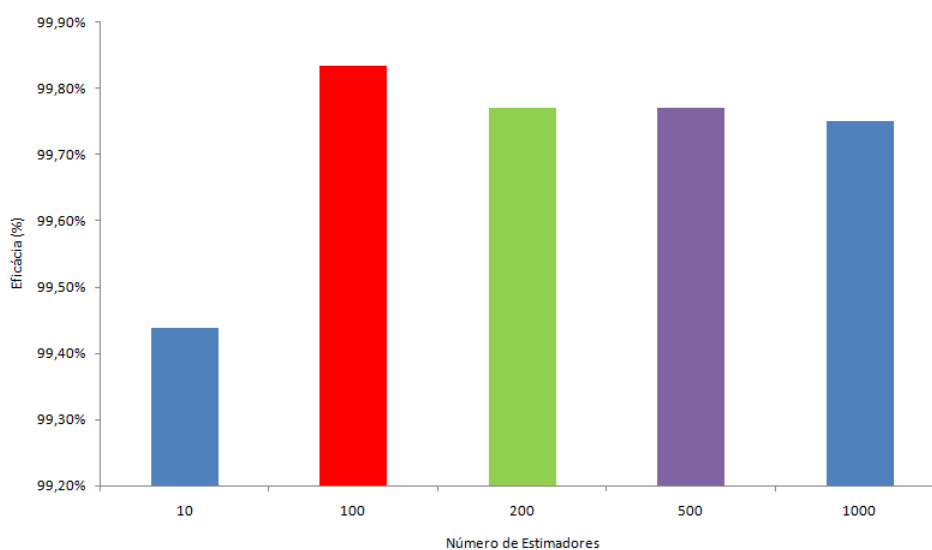


Fonte: Própria do Autor, 2019.

E nos Gráficos da Fig. 18 e da Fig. 19, são visualizados os resultados para os métodos *Decision Tree Classifier*, com número máximo de amostras =  $[1 \dots 20]$ , e *Random Forest Classifier*, com o número de estimadores =  $[10, 100, 200, 500, 1000]$ .

Figura 18 – Eficácia obtida pelo Método *Decision Tree Classifier*.

Fonte: Própria do Autor, 2019.

Figura 19 – Eficácia obtida pelo Método *Random Forest Classifier*.

Fonte: Própria do Autor, 2019.

Portanto, fica claro, que todos os métodos obtiveram ótimos índices de acertos ao identificar sinais de ECG no *dataset* em questão. Em que, as melhores eficácias para cada método, são postos na Tabela 2 e visualizados no Gráfico da Fig. 20.

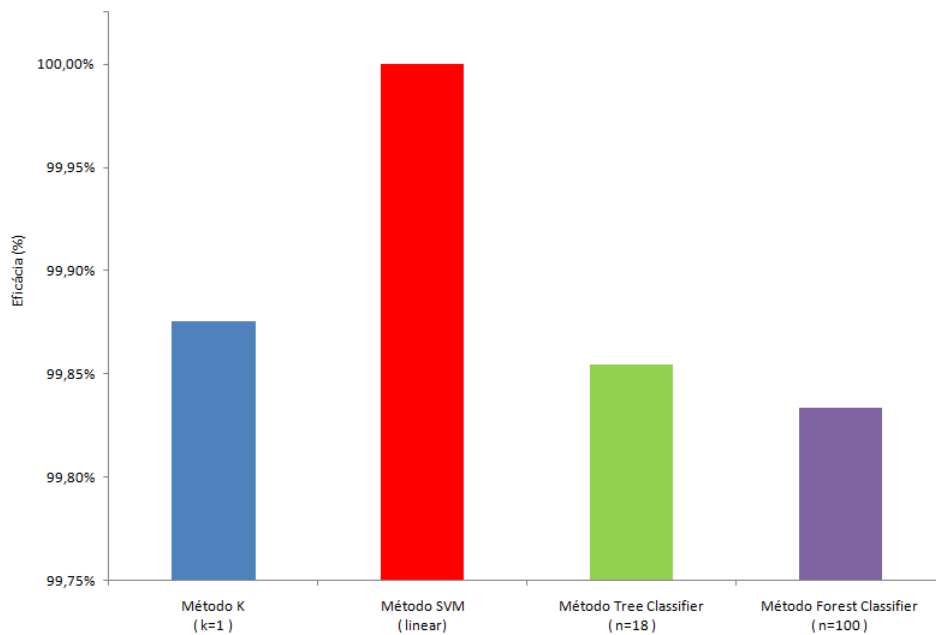
Diante dos ótimos resultados supracitados, é possível validar a aplicação de todos os quatro métodos utilizados para identificação de anomalias em sinais de ECG, com ênfase no Método SVM que obteve uma taxa de 100 % de acertos. Vale ressaltar, que os altos índices de acertos podem ser justificados fundamentalmente em dois fatores, sendo:

Tabela 2 – Melhores Desempenhos Alcançados.

Método	Parâmetro	Eficácia
K Nearest Neighbors Classifier	$k = 1$	98,87 %
Support Vector Machine	kernel = linear	100 %
Decision Tree Classifier	$n = 18$	99,85 %
Random Forest Classifier	$n = 100$	99,83 %

Fonte: Própria do Autor, 2019.

Figura 20 – Melhores resultados dos Métodos.



Fonte: Própria do Autor, 2019.

- Correlação existente no *dataset*;
- Alto número de amostras para treinamento dos métodos.

Ainda, na Tabela 3 são postos os resultados de eficácia dos métodos ao variar o tamanho do *dataset* de treino, e visualizados no Gráfico da Fig. 21.

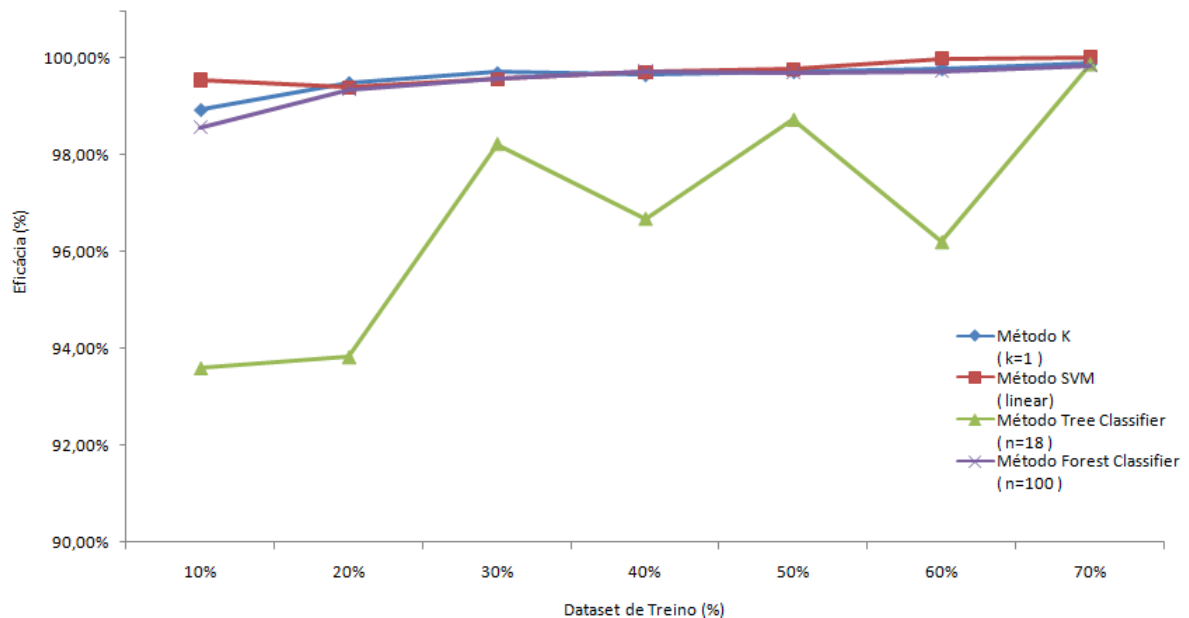
Tabela 3 – Dados de Eficácia ao variar o Tamanho do *Dataset* de Treino

<i>Dataset</i> (%)	K Neighbors ( k = 1 )	SVM (linear)	Decision Tree (n=18)	Random Forest (n=100)
10	98,92%	99,54%	93,59%	98,56%
20	99,48%	99,39%	93,82%	99,35%
30	99,69%	99,55%	98,21%	99,58%
40	99,65%	99,69%	96,67%	99,73%
50	99,71%	99,75%	98,71%	99,69%
60	99,77%	99,99%	96,19%	99,73%
70	99,88%	100,00%	99,85%	99,83%

Fonte: Própria do Autor, 2019.

Dessa forma, fica comprovada a influência positiva obtida ao majorar o número de amostras utilizadas no treinamento dos métodos. Visto que, ao aumentar a quantidade de dados utilizados para treinamento, os métodos se tornam mais "inteligentes" no processo de identificação. Bem como, é constatado que o método *Tree Classifier* se mostrou mais sensível a variação no tamanho do *dataset* de treino que os demais.

Figura 21 – Influência do Banco de Treino na Eficácia dos Métodos.



Fonte: Própria do Autor, 2019.

## 4 CONCLUSÕES

Este trabalho, expôs a implementação de um *software*, utilizando os conceitos de *Machine Learning*, com intuito de identificar anomalias em sinais de eletrocardiograma e, conseqüentemente classificar pacientes com doenças cardíacas. Dessa forma, o principal objetivo deste trabalho foi alcançado de forma essencial.

O *software* proposto obteve resultados ótimos para os quatro métodos propostos, sendo: *K Nearest Neighbors Classifier*, *Support Vector Machine*, *Decision Tree Classifier* e *Random Forest Classifier*. Bem como, foi comprovada a influência positiva obtida na eficácia dos métodos ao aumentar o tamanho da base de dados utilizada para treino. Vale ressaltar, que a existência de bibliotecas *open source* já implementadas favorece o desenvolvimento de programas e pesquisa na área.

Diante disto, e dos recentes avanços na área, destacando as redes neurais, é constatado que aplicações em *Machine Learning* são capazes de atuar em uma ampla gama de aplicações com resultados confiáveis. Portanto, a utilização de *softwares* inteligentes para facilitar o diagnóstico de doenças cardíacas se apresenta como um avanço promissor, visto o crescimento alarmante dessas enfermidades publicado pela Organização Mundial de Saúde.

Outro fato a ser destacado é a importância de encontrar *datasets* que se apliquem ao projeto em estudo. Este é um ponto primordial ao se trabalhar com *Machine Learning*, considerando que são necessários grandes quantidades de amostras para treinar o programa, com intuito de garantir uma alta eficácia, e posteriormente testar com novos dados, para obter resultados de acertos do projeto.

Este trabalho se deteve ao desenvolvimento de um *software* capaz de identificar a existência de anomalias em um sinal de eletrocardiograma. Logo, como sugestão de trabalhos futuros na área, a evolução do *software* para identificação do tipo de anomalia é bastante promissora. Ainda, é válido destacar a possibilidade de embarcar este projeto em um equipamento de fácil utilização e transporte, de forma a facilitar a identificação de doenças cardíacas em regiões de difícil acesso ou carência de profissionais de saúde.

# REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Governo do Brasil. Cerca de 300 mil brasileiros morrem de doenças vasculares por ano. Disponível em: <<http://www.brasil.gov.br/noticias/saude/2017/09/cerca-de-300-mil-brasileiros-morrem-de-doencas-cardiovasculares-por-ano>>. Acesso em: 17 abril 2019.
- [2] Sociedade Brasileira de Arritmias Cardíacas (SOBRAC). A Campanha. Disponível em: <<http://www.sobrac.org/campanha/a-campanha/>>. Acesso em: 17 abril 2019.
- [3] ROOPA, C.K.; HARISH, B. S. A Survey on various Machine Learning Approaches for ECG Analysis. International Journal of Computer Applications. Vol 163 – No 9. Abril 2017.
- [4] TISON, G.H.; ZHANG, J.; DELLING, F.N.; DEO, R.C. Automated and Interpretable Patient ECG Profiles for Disease Detection, Tracking and Discovery. California, USA, Julho 2018.
- [5] JIN, Z.; SUN, Y.; CHENG, A.C. Predicting Cardiovascular Disease from Real-Time Electrocardiographic Monitoring: Na Adaptive Machine Learning Approach on a Cell Phone. 31th Annual International Conference of the IEEE EMBS. Minnesota, USA, September 2009.
- [6] NASHIF, S.; RAIHAN, R.; ISLAM, R.; IMAM, M.H. Heart Disease Detection by Using Machine Learning Algorithms and a Real-Time Cardiovascular Health Monitoring System. World Journal of Engineering and Technology. November, 2018.
- [7] RASCHKA, S.; MIRJALILI, V. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow. 2<sup>a</sup> ed. Birmingham: Packt, 2017.
- [8] SBC. Sociedade Brasileira de Cardiologia. Disponível em: <<https://www.cardiol.br/>>. Acesso em: 20 de abril 2019.
- [9] Fye WB. A history of the origin, evolution and impact of electrocardiography. Am J Cardiol. 1994; 73:937-49.
- [10] My EKG. Ondas do Eletrocardiograma. Disponível em: <<https://pt.my-ekg.com/generalidades-ecg/ondas-eletrocardiograma.html>>. Acesso em: 17 junho 2019.
- [11] A. M. TURING, I. Computing Machinery And Intelligence, Mind, Volume LIX, Issue 236, October 1950, Pages 433–460.
- [12] PACHECO, V. Una Breve Historia del Machine Learning. Disponível em:

<<https://empresas.blogthinkbig.com/una-breve-historia-del-machine-learning/>>. Acesso em: 19 junho 2019.

[13] SAS. Machine Learning O que é e qual sua importância?. Disponível em: <[https://www.sas.com/pt\\_br/insights/analytics/machine-learning.html](https://www.sas.com/pt_br/insights/analytics/machine-learning.html) > .Acesso em : 19junho2019.

[14] PATEL, S. Chapter 2 : SVM (Support Vector Machine). Disponível em: <<https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72>>. Acesso em: 07 maio 2019.

[15] PATEL, S. Chapter 3 : Decision Tree Classifier. Disponível em: <<https://medium.com/machine-learning-101/chapter-3-decision-trees-theory-e7398adac567>>. Acesso em: 07 maio 2019.

[16] PATEL, S. Chapter 4: K Nearest Neighbors Classifier. Disponível em: <<https://medium.com/machine-learning-101/k-nearest-neighbors-classifier-1c1ff404d265>>. Acesso em: 07 maio 2019.

[17] PATEL, S. Chapter 5: Random Forest Classifier. Disponível em: <<https://medium.com/machine-learning-101/chapter-5-random-forest-classifier-56dc7425c3e1>>. Acesso em: 07 maio 2019.

[18] Bousseljot R, Kreiseler D, Schnabel, A. Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet. Biomedizinische Technik, Band 40, Ergänzungsband 1 (1995) S 317